CICS Transaction Server for z/OS

CICS セキュリティー・ガイド

バージョン *5* リリース *5*

CICS Transaction Server for z/OS

CICS セキュリティー・ガイド

バージョン *5* リリース *5*

注記

本書および本書で紹介する製品をご使用になる前に、 559 ページの『特記事項』に記載されている情報をお読みください。

本書は、IBM CICS Transaction Server for z/OS バージョン 5 リリース 5 (製品番号 5655-Y04) および新しい版で明記されていない限り、以降のすべてのリリースおよびモディフィケーションに適用されます。

お客様の環境によっては、資料中の円記号がバックスラッシュと表示されたり、バックスラッシュが円記号と表示されたりする場合があります。

原典： CICS Transaction Server for z/OS
CICS Security Guide
Version 5 Release 5

発行： 日本アイ・ビー・エム株式会社

担当： トランスレーション・サービス・センター

© Copyright IBM Corporation 1989, 2018.

目次

この PDF について ix

第 1 章 RACF による CICS セキュリティーの概要 1

CICS のセキュリティー機能 1
CICS セキュリティーの保護対象 1
CICS ユーザー 2
CICS 領域ユーザー ID 4
CICS デフォルト・ユーザー ID 6
端末ユーザー・セキュリティー 7
非端末セキュリティー 8
QUERY SECURITY コマンド. 8
相互通信セキュリティー 8
サインオン・セキュリティー用のパスチケット . . 9
API および SPI 制限の DFHAPIR parmlib メンバー 10
RACF ファシリティー. 12
RACF 管理 12
RACF 管理責任の委任. 13
RACF プロファイル 14
RACF ユーザー・プロファイル. 15
RACF グループ・プロファイル. 23
RACF データ・セット・プロファイル 24
RACF 一般リソース・プロファイル 25
RACF classes for CICS リソース 27
システム・リソースを保護するための RACF クラス. 28
DB2ENTRY リソースのリソース・クラス . . . 32
RACF コマンドの要約. 32
データおよびユーザーのセキュリティー分類 . . 36
独自のリソース・クラスの定義. 36
RACF プロファイル内のフィールドへのアクセスの制御 38

第 2 章 単一 CICS 領域での RACF 保護の実装 39

CICS システム・リソースのセキュリティー . . . 39
RACF の CICS インストール要件. 39
CICS 領域のユーザー ID の指定 41
RACF での CICS プロシージャーの実行を許可する 42
CICS 領域ユーザー ID のユーザー・プロファイルの定義 43
デフォルトの CICS ユーザー ID を RACF に定義する 46
MVS ログ・ストリームへのアクセスを許可する 48
CICS データ・セットへのアクセスの許可 . . . 49
一時記憶域プールへのアクセスを許可する . . . 53
一時記憶域サーバーへのアクセスを許可する . . 54
名前付きカウンター・プールおよびサーバーへのアクセスの許可 55
SMSVSAM サーバーへのアクセスを許可する . . 57
CICS 領域へのアクセスの許可 58
CICS 領域の z/OS Communications Server ACB オープンの制御 60
ユーザー ID 伝搬の制御 60
CICS 環境での代理ジョブ実行依頼 61
CICS 領域ユーザー ID を代理ユーザーとして許可する 62
CICS 環境での JES スプールの保護 62
EXEC PGM=DFHSIP ステートメントの **PARM** パラメーターまたは SYSIN での **HPO** 使用の許可. . . 63
セキュリティー関連システム初期設定パラメーター 63
CICS ユーザーの検証 72
CICS 端末ユーザーの識別 72
セキュア・サインオンのためのパスチケットの実装 72
FEPI アプリケーションでのパスチケットの使用 74
サインオン・プロセス. 75
サインオフ・プロセス. 77
特定のエントリー・ポートから CICS へのアクセスの制御 79
エントリー・ポート・プロファイルの定義 . . . 79
サインオンおよびサインオフ・アクティビティーの監査 83
Preset terminal security 83
MVS システム・コンソールを CICS 端末として使用する 87
ユーザーの CICS 関連データの取得 90
大/小文字混合パスワードのサポート. 93
各国語および非端末トランザクション 93
トランザクション・セキュリティー 94
トランザクション接続セキュリティーを制御する CICS パラメーター. 94
RACF へのトランザクション・プロファイルの定義 97
許可障害とエラーメッセージ 98
非端末トランザクションの保護. 99
リソース・セキュリティー 100
リソース・セキュリティーの概要. 100
XRES リソース・セキュリティー・パラメーターを使用したセキュリティー. 101
CICS および RACF によるリソース・セキュリティー検査 105
一時データのセキュリティー 110
ファイルのセキュリティー. 112
ジャーナルとログ・ストリームのセキュリティー 114
開始されたトランザクションのセキュリティー 115

© Copyright IBM Corp. 1989, 2018

EXEC CICS RUN TRANSID によって開始されたトランザクションのセキュリティー 119
XPCT 検査トランザクションのセキュリティー 120
アプリケーション・プログラムのセキュリティー 121
一時記憶域のセキュリティー 122
z/OS UNIX ファイルのセキュリティー . . . 123
プログラム仕様ブロックのセキュリティー. . . 128
CEDF、CEDG、CEDX、または CEDY の下で実行されるトランザクションのセキュリティー検査 129
リソースの総称プロファイルの定義 130
XRES リソース・セキュリティー・パラメーターを使用したセキュリティー. 131
リソース・セキュリティー 134
CICS および RACF リソース検査の要約 134
リソースの総称プロファイルの定義 136
リソースおよびコマンドの検査の相互参照. . . . 137
代理ユーザー・セキュリティー 154
代理ユーザー検査が適用される状態 154
代理ユーザー検査の RACF 定義 159
代理ユーザー検査の RACF 定義の例 161
CICS コマンド・セキュリティー 162
コマンド・セキュリティーの概要. 162
コマンド・セキュリティー検査の対象となる CICS リソース 164
コマンド・セキュリティーを指定するためのパラメーター 171
CEDF の下で実行されるトランザクションのセキュリティー検査 172
CEMT の考慮事項. 173
アプリケーション・プログラムの許可障害. . . 173
QUERY SECURITY コマンドを使用したセキュリティー検査. 174
QUERY SECURITY の動作 174
RESTYPE オプション 176
RESCLASS オプション 181
ユーザー定義リソースを RACF に指定する . . 182
ユーザーの代理権限の照会 184
QUERY SECURITY のロギング 185
例: リソース・セキュリティー検査のための **QUERY SECURITY** コマンドの使用 185
CICS トランザクションのセキュリティー. . . . 186
| 内部読み取りプログラムに JCL ジョブを実行依頼
| する場合のセキュリティー 193
CICS が CICS 領域のユーザー ID を特定する状況と方法. 196

第 3 章 ID 伝搬および分散セキュリティー 199
ID 伝搬のサポートおよび要件. 200
ID 伝搬を使用するためのネットワーク・トポロジーの例. 202
ID 伝搬の構成 204
ID 伝搬のための RACF の構成 205

第 4 章 外部セキュリティー・マネージャーの呼び出し 207
MVS ルーターの概要. 207
ESM 出口プログラムによる CICS 関連情報へのアクセス方法 208
RACF ユーザー出口パラメーター・リスト . . 208
インストール・データ・パラメーター・リスト 209

第 5 章 CICSPlex SM のセキュリティー 211
CICSPlex SM セキュリティーの実装 211
だれが CICSPlex SM リソースへのアクセスを必要とするかの判別 212
CICSPlex SM セキュリティーに関する一般要件 216
CICSPlex SM データ・セットのプロファイルの作成 216
MAS エージェント・ユーザー ID の判別. . . 217
CICSPlex SM 開始タスクの定義 218
CMAS における CICSPlex SM トランザクションの定義 218
管理対象 CICS 領域でのトランザクションの定義 220
CICSPlex SM Web ユーザー・インターフェース・セキュリティーのセットアップ 221
プロファイルでの CICSPlex SM リソース名の指定 227
プラットフォームおよびアプリケーションのセキュリティー 246
CICS シミュレーション・セキュリティーのアクティブ化 250
CICS 代理セキュリティー検査に関する考慮事項 252
CICSPlex SM のセキュリティーのアクティブ化 253
CICSPlex SM の RACF プロファイルのリフレッシュ. 254
CICSPlex SM セキュリティー検査シーケンス 255
ユーザー提供の外部セキュリティー・マネージャーの呼び出し 260
CICSPlex SM ESM インターフェースの概要 260
MVS ルーターの概要. 260
CICSPlex SM セキュリティーの制御点. . . . 261
サンプル・タスク: セキュリティー 262
例: すべての CICSPlex SM リソースの保護 . . 262
例: CICSPlex SM オペレーターへの適切な権限の付与. 263
例: ユーザーへの MVS システム A 上のすべてのトランザクションに対する読み取り権限の付与 264

第 6 章 プラットフォームおよびアプリケーションのセキュリティー 265

第 7 章 相互通信のセキュリティー. . . 271
相互通信セキュリティーの概要 271
相互通信セキュリティーの概要 271
相互通信セキュリティーの計画 272
相互通信バインド時のセキュリティー 272

相互通信リンク・セキュリティー. 272
相互通信のユーザー・セキュリティー 273
相互通信のためのトランザクション、リソース、コマンド、および代理ユーザーのセキュリティー 274
相互通信セキュリティー・レベルの要約 . . . 274
LU6.2 セキュリティーの実装 275
LU6.2 でのバインド時のセキュリティー . . . 275
Link security with LU6.2 281
LU6.2 接続のリンク・セキュリティーの指定 282
LU6.2 でのユーザー・セキュリティー 282
SNA プロファイルおよび接続時セキュリティー 288
LU6.2 でのトランザクション・セキュリティー、リソース・セキュリティー、およびコマンド・セキュリティー 289
LU6.2 でのトランザクション・ルーティング・セキュリティー. 291
LU6.2 での機能シップ・セキュリティー . . . 293
LU6.2 での分散プログラム・リンク・セキュリティー. 294
LU6.2 での AOR で実行されるセキュリティー検査 296
LU6.1 セキュリティーの実装 298
LU6.1 でのリンク・セキュリティー. 298
LU6.1 接続のリンク・セキュリティーの指定 298
LU6.1 での ATTACHSEC の指定 299
LU6.1 でのトランザクション・セキュリティー、リソース・セキュリティー、およびコマンド・セキュリティー 299
LU6.1 での機能シップ・セキュリティー . . . 300
LU6.1 での AOR で実行されるセキュリティー検査 302
APPC パスワード有効期限管理 303
APPC パスワード有効期限管理の概要 303
APPC PEM を使用するために必要なこと. . . 304
外部セキュリティー・インターフェース . . . 305
PEM クライアントと CICS PEM サーバーの役割 306
APPC PEM 処理の概要. 308
PEM クライアントのセットアップ 315
PEM クライアントの入出力データ 317
IPIC セキュリティーの実装 326
IPIC バインド時のセキュリティー 326
IPIC リンク・セキュリティー. 328
IPIC ユーザー・セキュリティー 330
IPIC のトランザクション・セキュリティー、リソース・セキュリティー、およびコマンド・セキュリティー 332
CICS ルーティング・トランザクション CRTE 333
IPIC を使用した AOR で実行されるセキュリティー検査. 334
MRO セキュリティーの実装 335
MRO アクセス方式選択時のセキュリティーへの影響 335
MRO でのバインド時のセキュリティー . . . 336
MRO を使用したログオン・セキュリティー検査 336
MRO とのリンク・セキュリティー 339
MRO 接続のリンク・セキュリティーの指定 . . 341
MRO でのユーザー・セキュリティー 341
MRO でのトランザクション・セキュリティー、リソース・セキュリティー、およびコマンド・セキュリティー 343
MRO でのトランザクション・ルーティング・セキュリティー 345
MRO での機能シップ・セキュリティー . . . 346
MRO での分散プログラム・リンク・セキュリティー 348
MRO での AOR で実行されるセキュリティー検査 349
データ・テーブルのセキュリティー. 350
CICS 共用データ・テーブルのセキュリティー 350
カップリング・ファシリティー・データ・テーブルのセキュリティー 354

第 8 章 TCP/IP クライアントのセキュリティー. 359

TCP/IP クライアントのセキュリティーについて 359
メッセージ保護. 359
識別と認証 362
セキュリティー・プロトコルのサポート . . . 368
SSL を使用するための CICS の構成 375
RACF でのプロファイルのセットアップ . . . 376
認証局からの証明書の要求. 377
手動での鍵リングの作成. 378
DFH$RING を使用した証明書付きの鍵リングの作成 379
新規 RACF 証明書の作成 381
RACF ユーザー ID と証明書との関連付け . . 382
CICS 領域ユーザー ID によって所有されていない既存の証明書の使用 383
CICS TS で使用するための RACF サイト証明書の構成. 384
証明書を Untrusted としてマーク付けする . . 385
SSL のシステム初期設定パラメーター 385
SSL の TCPIPSERVICE 属性 386
SSL 暗号スイート仕様ファイルの作成 388
暗号化ネゴシエーションのカスタマイズ . . . 389
CICS TS システムを NIST SP800-131A 準拠にする 390
CICS を使用するための LDAP の構成. . . . 392
証明書失効リスト (CRL) の使用 394

第 9 章 データ・ソースのセキュリティー 399

Db2 のセキュリティー 399
CICS での Db2 関連リソースへのアクセスの制御 400
CICS 領域用および CICS トランザクション用に、許可 ID を Db2 に提供する. 410
ユーザーに対する Db2 内のリソースへのアクセス許可. 419
Db2 マルチレベル・セキュリティーおよび行レベル・セキュリティー 423

DBCTL のセキュリティー 424
CICS による PSB 許可検査 424
データ・セットのセキュリティー: 暗号化 425

第 10 章 外部インターフェースのセキュリティー 429
EXCI のセキュリティー 429
MRO ログオン・セキュリティーおよびバインド時のセキュリティーの使用 429
リンク・セキュリティー 430
ユーザー・セキュリティー 431
代理ユーザー検査 432
ONC RPC のセキュリティー 433
ONC RPC のセキュリティー 433
CICS のセキュリティーとそれが CICS ONC RPC の稼働に及ぼす影響 433
リソース・チェッカーの作成 436

第 11 章 Java アプリケーションのセキュリティー 439
OSGi アプリケーションに関するセキュリティーの構成 439
Liberty JVM サーバーに関するセキュリティーの構成 440
Liberty のエンジェル・プロセス 443
Liberty JVM サーバーでのユーザー認証 . . . 446
ユーザーに Liberty JVM サーバー内のアプリケーションを実行する権限を与える 449
OAuth 2.0 を使用したアプリケーションの許可 451
SAF ロール・マッピングを使用した許可 . . . 455
LDAP レジストリーを使用した Liberty JVM サーバーのセキュリティーの構成 457
Java 鍵ストアを使用した Liberty JVM サーバー用の SSL (TLS) の構成 461
RACF を使用した Liberty JVM サーバー用の SSL (TLS) の構成 462
Liberty JVM サーバーでの SSL (TLS) クライアント証明書認証のセットアップ 463
Java セキュリティー・マネージャーの有効化 . . . 465

第 12 章 CICS Web サポートのセキュリティー 469
HTTP サーバーとしての CICS: 認証および識別 469
HTTP クライアントとしての CICS: 認証および識別 471
HTTP 基本認証のパスワード有効期限管理 . . . 472
CICS システムと CICS Web サポートのリソース・セキュリティー 475
インバウンド・ポートのセキュリティー . . . 476
CICS システム・コンポーネントのセキュリティー 477
アプリケーションが生成する応答のリソース・セキュリティーとトランザクション・セキュリティー 477
文書テンプレートを使用する静的応答のリソース・レベル・セキュリティー 481
CICS Web サポートでの SSL 482
Application Transparent Transport Layer Security (AT-TLS) の概要 483
Atom フィードのセキュリティー 495

第 13 章 Web サービスを保護するためのサポート 497
Web Services Security を実装するための前提条件 498
SOAP Web サービスの保護の計画 499
SOAP メッセージを保護するためのオプション . . 500
Security Token Service を使用した認証 502
Trust クライアント・インターフェース . . . 504
SOAP メッセージへの署名 505
署名アルゴリズム 505
署名された SOAP メッセージの例 505
暗号化された SOAP メッセージの CICS サポート 506
暗号化アルゴリズム 507
暗号化された SOAP メッセージの例 508
Web Services Security に合わせた RACF の構成 508
ID 伝搬のためのプロバイダー・モードの Web サービスの構成 511
Web Services Security に合わせたパイプラインの構成 513
カスタムのセキュリティー・ハンドラーの作成 . . 517
メッセージ・ハンドラーからの Trust クライアントの起動 519
z/OS Connect のセキュリティー 520
z/OS Connect サービスおよび API の許可の構成 521

第 14 章 BTS のセキュリティー . . . 523
BTS のリソース・セキュリティー 523
プロセスおよびアクティビティーのユーザー ID 523
プロセスおよびアクティビティーの接続時セキュリティー 524
BTS のコマンド・セキュリティー 525

第 15 章 CICS-MQ アダプターのセキュリティー 527
CICS-MQ アダプター・トランザクションのセキュリティーの実装 527
MQCONN コマンドおよび MQMONITOR コマンドのセキュリティー 528
CICS-MQ アダプター・ユーザー ID 529
CICS-MQ アダプターの接続セキュリティー . . . 529

第 16 章 CICS-MQ ブリッジのセキュリティー 531

第 17 章 Kerberos のサポート 537
Kerberos 用の RACF の構成 537
Kerberos 用の CICS Web サービスの構成 . . . 538
Kerberos アプリケーションの開発 539

3270 エミュレーター・サインオンでの Kerberos セキュリティー・トークンの使用. 540

第 18 章 RACF を使用した Multi-Factor Authentication のサポート 543

第 19 章 SAML 用の CICS の構成 . . 545
SAML 用の CICS の構成の検証 546
SAML トークンを使用するためのプロバイダー・パイプラインの構成 547
SAML トークンを使用するためのリクエスター・パイプラインの構成 549
CICS セキュリティー・トークン・サービスの構成 550
STS 構成ファイル. 552
SAML 対応プログラムを開発するパターン . . . 553

第 20 章 SAML アプリケーションの開発 555
SAML 対応の初期プログラムの開発. 555
SAML 対応の初期プログラム開発のパターン 556
アウトバウンド要求で検証済み SAML トークンを使用するプログラムの開発 556
SAML トークンを作成したり拡張したりするプログラムの開発 557

特記事項. 559

索引 565

この PDF について

この PDF では、CICS® システム全体のセキュリティーを計画して実装する方法が説明されています。この情報は、CICS で使用されるリソースへのアクセスを制御する責任を担う、セキュリティー管理者を対象としています。それらのリソースは、CICS 領域における CICS 端末、ユーザー、またはトランザクションによって使用され、またそれらの領域で稼働する CICS アプリケーション・プログラムによって使用されます。この PDF は、インストールに関する要件をセキュリティー管理者に伝える必要がある、CICS システム・プログラマーも対象としています。

この PDF は、「*RACF Security Guide*」(SC34-7423-00) に取って代わるものです。

本書で使用される用語と表記について詳しくは、IBM® Knowledge Center の CICS 資料で使用されている表記規則および用語 を参照してください。

この PDF の日付

この PDF は 2018 年 12 月 14 日 に作成されました。

© Copyright IBM Corp. 1989, 2018

第 1 章 RACF による CICS セキュリティーの概要

このパートは、CICS セキュリティーの概要です。RACF® によって提供される機能を使用して、CICS システムおよびそれらのシステム内のリソースを無許可アクセスから保護する方法を大まかに説明します。

CICS のセキュリティー機能

オンライン・トランザクション処理システム (何万というユーザーをサポートすることもよくある) として、CICS は、アクセスを管理するリソースを確実に保護して無許可アクセスからセキュアにするために、セキュリティー・システムの保護を必要とします。CICS には、無許可アクセスからリソースを保護するいくつかの機能が用意されています。

必要なセキュリティーを CICS 領域に提供するため、CICS は MVS™ System Authorization Facility (SAF) を使用して、CICS トランザクション処理内の適切な時点で、RACF などの外部セキュリティー・マネージャー (ESM) に許可要求を転送します。

Business Transaction Services の具体的なセキュリティー情報については、BTS のセキュリティーを参照してください。

CICS セキュリティーの保護対象

CICS はアプリケーション・プログラム、アプリケーション・データ、およびアプリケーション出力を管理します。これらの資産の漏えい、破壊、または破損を防ぐには、まず CICS システム・コンポーネント自体を保護する必要があります。

CICS システムへの漏えいが生じる恐れのある領域は 2 つあります。一つは、CICS の外部にあるソースからの漏えいです。TSO ユーザーまたはバッチ・ジョブから CICS 管理資産への無許可アクセスを防止する主な手段として、RACF データ・セット保護を使用できます。

考えられるもう一つの領域は、CICS ユーザーから生じる漏えいです。CICS では次のように、CICS ユーザーのアクティビティーを、特定の個別ユーザーが使用を許可される機能のみに制限できる、さまざまなセキュリティーおよび制御メカニズムが提供されます。

トランザクション・セキュリティー
: トランザクションを実行しようとするユーザーにその資格があるかどうかを確認します。

リソース・セキュリティー
: CICS リソースを使用するユーザーにその資格があるかどうかを確認します。

コマンド・セキュリティー
: CICS システム・プログラミング・コマンドを使用するユーザーにその資格があるかどうかを確認します。

© Copyright IBM Corp. 1989, 2018

CICS 自体には、固有の資産を外部アクセスから保護する機能はありません。プログラム・ライブラリーへのアクセスを、CICS 領域に制限し、さらには、承認されたアプリケーションやシステム変更の組み込みを担当するユーザーに制限する必要があります。同様に、CICS や CICS アプリケーションによって使用されるデータ・セットおよびデータベースは、承認されたバッチ処理および操作プロシージャーのみがアクセス可能でなければなりません。

CICS は、資料による裏付けのないインターフェース、またはサポートされていないインターフェースを使用して CICS セキュリティーをバイパスするアプリケーション・プログラムから、システムを保護しません。そのようなプログラムがシステムにインストールされないようにするのは、お客様側で行ってください。

CICS は、アプリケーション・ソース・ライブラリーを保護しません。無許可または未テストのアプリケーション・プログラムが実動アプリケーション・ベースに導入されるのを防ぐ手順を確立し、その手順に従うようにしてください。さらに、システムに承認されているライブラリーおよびそれらのライブラリーへの変更を制御することにより、システムの完全性を守る必要があります。

CICS ユーザー

CICS セキュリティーがアクティブである場合、トランザクション接続要求、およびトランザクションによるリソースへのアクセス要求は、ユーザー に関連付けられます。

ユーザーが要求を行うと、CICS はユーザーに要求を行う権限があるかどうかを判別するために外部セキュリティー・マネージャーを呼び出します。ユーザーに正しい権限がない場合、CICS は要求を拒否します。

多くの場合、ユーザーは人間のオペレーターであり、端末またはワークステーションを介して CICS と対話します。ただし、必ずそうであるわけではなく、ユーザーはクライアント・システムで実行するプログラムである場合もあります。一般に、CICS ユーザーは、ユーザー *ID* (または *userid*) によって識別されるエンティティーです。

すべての CICS ユーザーは、セキュリティー・マネージャーに対して定義されている必要があります。セキュリティー・マネージャーが RACF である場合、各ユーザーに関する情報はユーザー・プロファイルに保管されます。

以下に、CICS トランザクションまたは CICS リソースのユーザーを識別するいくつかの方法を示します。

- 人間のオペレーターは、端末セッションの開始時にサインオンします (そのためにユーザー ID を入力します)。ユーザー ID は、端末オペレーターがサインオフするまで端末に関連付けられたままになります。端末から実行されるトランザクションと、それらのトランザクションによって行われる要求は、このユーザー ID に関連付けられます。

  詳しくは、CICS 端末ユーザーの識別を参照してください。
- ユーザー ID は端末に永続的に関連付けられています。端末から実行されるトランザクションと、それらのトランザクションによって行われる要求は、事前設定ユーザー ID に関連付けられます。

詳しくは、Preset terminal securityを参照してください。

- Secure Sockets Layer (SSL) を使用して CICS と通信するクライアント・プログラムは、自身を識別するためのクライアント証明書を提供します。セキュリティー・マネージャーは、証明書をユーザー ID にマップします。クライアントの要求を処理するトランザクションと、そのトランザクションによってさらに行われる要求は、このユーザー ID に関連付けられます。

  詳しくは、TCP/IP クライアントのセキュリティーについてを参照してください。
- CICS アプリケーション・プログラムは、USERID オプションを指定した START コマンドを発行します。開始済みトランザクションと、そのトランザクションによって行われる要求は、指定されたユーザー ID に関連付けられます。

  詳しくは、開始されたトランザクションのセキュリティーを参照してください。
- トランザクションは、区画内一時データ・キューのトリガー・レベルに達すると開始されます。 USERID 属性がキューの TDQUEUE 定義に指定されている場合、開始済みトランザクションと、そのトランザクションによって行われる要求は、指定されたユーザー ID に関連付けられます。

  詳しくは、Protecting non-terminal transactionsを参照してください。
- 別のシステムで実行中のリモート・プログラムは、CICS に接続要求を送信するときにユーザー ID を提供します。接続済みトランザクションと、そのトランザクションによって行われる要求は、指定されたユーザー ID に関連付けられます。

  詳しくは、相互通信のユーザー・セキュリティーを参照してください。
- リモート・システムが CICS に接続し、リモート・システムへの接続に対してリンク・セキュリティーが指定されています。リモート・システムから呼び出されるトランザクションと、そのトランザクションによって行われる要求は、リンクのユーザー ID に関連付けられます。詳しくは、相互通信リンク・セキュリティーを参照してください。
- CICS Business Transaction Services (BTS) プロセスは、RUN コマンドによってアクティブ化され、DEFINE PROCESS コマンドは USERID オプションを指定しました。プロセスが実行されるトランザクションと、そのトランザクションによって行われる要求は、指定されたユーザー ID に関連付けられます。

  詳細については、BTS の概要を参照してください。
- 第 2 フェーズの PLT プログラムは、CICS の初期設定中に実行されます。PLTPISEC システム初期設定パラメーターの値に応じて、プログラムによって行われた要求は、PLTPIUSR システム初期設定パラメーターで指定されたユーザー ID に関連付けられます。

  詳しくは、PLT プログラムを参照してください。

個々のユーザーを識別するユーザー ID に加えて、CICS が使用するユーザー ID が 2 つあります。それらは、以下のとおりです。

領域ユーザー **ID**
: CICS システム (システムの個々のユーザーではない) がリソースへのアクセスを要求するときの許可検査に使用されます。

  詳細については、CICS 領域ユーザー IDを参照してください。

デフォルト・ユーザー **ID**
: 他の、より具体的なユーザー識別が存在しない場合、CICS リソースを保護するするために使用されるセキュリティー属性を持つユーザーを識別します。

  詳細については、CICS デフォルト・ユーザー IDを参照してください。

ユーザー ID はそれだけでは無許可アクセスからシステムを保護しません。多くの場合、ユーザー ID は、本人であるユーザー以外の人にも知られています。偽名の使用を避けるために、ユーザーを認証するには、別の情報 (個々のユーザーのみが知っている) を提供する必要があります。以下に例を示します。

- 端末ユーザーの場合、ユーザーがサインオン時に提供するパスワードによってそのユーザーが認証されます。
- SSL を使用するクライアントの場合、クライアント証明書によってそのクライアントが認証されます。

CICS 領域ユーザー ID

CICS 領域ユーザー ID は、CICS システム (システムの個々のユーザーではない) がリソースへのアクセスを要求するときの許可検査に使用されます。

CICS はこの領域ユーザー ID を、以下のリソースの許可を検査するときに使用します。

MVS システム・ログ・ストリーム
: 詳しくは、 48 ページの『MVS ログ・ストリームへのアクセスを許可する』を参照してください。

CICS システム・データ・セット
: 詳しくは、 49 ページの『CICS データ・セットへのアクセスの許可』を参照してください。

CICS ユーザー・データ・セット
: 詳しくは、 52 ページの『ユーザー・データ・セットへのアクセスを許可する』を参照してください。

一時記憶域データ共用サーバー
: 詳しくは、 54 ページの『一時記憶域サーバーへのアクセスを許可する』を参照してください。

SMSVSAM サーバー
: 詳しくは、 57 ページの『SMSVSAM サーバーへのアクセスを許可する』を参照してください。

名前付きカウンター・サーバー
: 詳しくは、 55 ページの『名前付きカウンター・プールおよびサーバーへのアクセスの許可』を参照してください。

z/OS® Communications Server ACB
詳しくは、 60 ページの『CICS 領域の z/OS Communications Server ACB オープンの制御』を参照してください。

JES スプール・データ・セット
詳細については、 62 ページの『CICS 環境での JES スプールの保護』を参照してください。

CICS 領域間プログラム
詳しくは、 336 ページの『MRO を使用したログオン・セキュリティー検査』を参照してください。

カップリング・ファシリティー・データ・テーブル
詳しくは、 354 ページの『カップリング・ファシリティー・データ・テーブルのセキュリティー』を参照してください。

RACF 鍵リング
詳しくは、 378 ページの『手動での鍵リングの作成』を参照してください。

CICS は、以下の場合にも領域ユーザー ID を使用することがあります。

JES 内部読み取りプログラムにジョブを実行依頼する場合
詳しくは、 60 ページの『ユーザー ID 伝搬の制御』を参照してください。

CICS 実行時に使用される他のユーザー **ID** の代理として
詳しくは、 154 ページの『代理ユーザー・セキュリティー』を参照してください。

RACF に渡されるリソース名の接頭部として
詳細については、 63 ページの『セキュリティー関連システム初期設定パラメーター』で **SECPRFX** システム初期設定パラメーターの説明を参照してください。

CICS 提供の非端末トランザクションを実行する場合
詳しくは、Security for CICS-supplied transactionsを参照してください。

LU6.2 を使用する **CICS** システム間連絡
詳しくは、 275 ページの『LU6.2 セキュリティーの実装』を参照してください。

LU6.1 を使用する **CICS** システム間連絡
詳しくは、 298 ページの『LU6.1 セキュリティーの実装』を参照してください。

MRO を使用する **CICS** システム間連絡
詳しくは、 335 ページの『MRO セキュリティーの実装』を参照してください。

エンジェル・プロセスを使用する **Liberty JVM** サーバーで
詳細については、Liberty サーバーのエンジェル・プロセスを参照してください。

CICS 領域ユーザー ID は、初期設定時に CICS 領域に割り当てられます。これはジョブまたは開始タスクに関連付けられるユーザー ID です。詳しくは、 41 ページの『CICS 領域のユーザー ID の指定』を参照してください。

CICS デフォルト・ユーザー ID

CICS デフォルト・ユーザー ID は、他の、より具体的なユーザー識別が存在しない場合に、CICS リソースを保護するするために使用されるセキュリティー属性を持つユーザーを識別します。

デフォルト・ユーザー ID は、**DFLTUSER** システム初期設定パラメーターに指定されます。パラメーターを指定しない場合、デフォルト・ユーザー ID は CICSUSER です。

- これは、ユーザーがサインオンする前、およびユーザーがサインオフした後に、端末またはコンソールに割り当てられます。ただし端末またはコンソールに、指定された事前設定セキュリティーがある場合を除きます。詳しくは、 72 ページの『CICS ユーザーの検証』を参照してください。
- これは、端末に関連付けられていない一時データのトリガー・レベル・トランザクションに対して、ユーザー ID が一時データ・キューの定義に指定されていない場合に割り当てられます。詳細については、 157 ページの『一時データのトリガー・レベル・トランザクション』を参照してください。
- これは、SECURITYNAME 属性が CONNECTION 定義に指定されていない場合に、LU6.1 接続および LU6.2 接続のリンク・ユーザー ID として使用されます。詳細については、 298 ページの『LU6.1 でのリンク・セキュリティー』および 281 ページの『Link security with LU6.2』を参照してください。デフォルト・ユーザー ID は、IPIC のリンク・ユーザー ID として使用することもできます。詳細については、IPIC による AOR で行われるセキュリティー検査を参照してください。
- これは、接続要求にセキュリティー・パラメーターが含まれておらず、CONNECTION 定義が USEDFLTUSER(YES) を指定している場合に、LU6.2 セッションおよび MRO セッションによって接続されるトランザクションに割り当てられます。詳細については、 288 ページの『SNA プロファイルおよび接続時セキュリティー』および 341 ページの『MRO でのユーザー・セキュリティー』を参照してください。デフォルト・ユーザー ID は、IPIC セッションによって接続されたトランザクションに割り当てることもできます。詳細については、IPIC ユーザー・セキュリティーを参照してください。
- CICS ルーティング・トランザクション (CRTE) を使用して開始されたアプリケーション所有領域 (AOR) でのトランザクションの場合、CRTE の使用時に端末ユーザーが AOR にサインオンしない場合、デフォルト・ユーザー ID が使用されます。詳細については、CICS ルーティング・トランザクション、CRTE を参照してください。
- これは、アプリケーション所有領域 (AOR) が共用データ・テーブルとして定義されているリモート・ファイルに対して要求を発行し、AOR がファイル所有領域 (FOR) にサインオンできない場合に使用されます。詳しくは、 350 ページの『CICS 共用データ・テーブルのセキュリティー』を参照してください。
- より明示的な ID がない場合、これは、CICS に接続する TCP/IP クライアントを識別するために使用されます。詳しくは、 362 ページの『識別』を参照してください。
- USERID が明示的に指定されていない場合 (例えば、動的にインストールされた MQMONITOR DFHMQINI)、これは MQMONITOR の USERID 属性に割り当てられます。

- デフォルトで z/OS Connect、Liberty、および WEBSERVICE に使用される初期ユーザー ID。
- デフォルトで非認証 z/OS Connect、Liberty、および WEBSERVICE 要求に使用されるユーザー ID。

  Liberty JVM サーバーのセキュリティーを構成する方法の詳細については、Liberty JVM サーバーのセキュリティーの構成 (Configuring security for a Liberty JVM server)を参照してください。 z/OS Connect に対する権限を構成する方法の詳細については、z/OS Connect サービスおよび API の許可の構成を参照してください。

端末ユーザー・セキュリティー

無許可アクセスからリソースを保護するため、CICS はシステムの個々のユーザーを一意に識別する何らかの手段が必要です。

この目的のために、まずユーザー・プロファイル と呼ばれる RACF データベース内の項目を作成して、ユーザーを RACF に定義します。ユーザーは、CICS に対して身元を証明するために、CICS 提供のサインオン・トランザクション CESN で RACF ユーザー識別 (ユーザー ID) と関連パスワード、またはオペレーター ID カード (OIDCARD) を指定してサインオンします。別の方法として、ユーザーは、独自のインストール済み環境で開発された同等のトランザクションを使用して、この目的のために提供された EXEC CICS SIGNON コマンドを発行することもできます。

ユーザーが CESN トランザクションに入ると、CICS は RACF の呼び出しによってユーザー ID およびパスワードを検査します。端末ユーザーのサインオンが有効な場合、CICS ユーザー・ドメインはサインオン・ユーザーの状況を常に把握します。それ以降、CICS は RACF を呼び出して許可検査を行うときは、そのユーザーに関する情報を使用します。**GMTRAN** システム初期設定パラメーターを使用すると、ユーザーがサインオンを完了できなかった場合の処理を制御できます。例えば、その後のすべてのトランザクションで CICS デフォルト・ユーザー ID を使用したり、端末セッションを切断したりできます。

RACF によって提供される端末セキュリティー機能について詳しくは、 79 ページの『端末プロファイル』を参照してください。CICS での端末ユーザー・セキュリティーの使用については、 72 ページの『CICS ユーザーの検証』を参照してください。

一部の端末の場合、および CICS 端末として使用される MVS コンソールの場合は、事前設定端末セキュリティー を使用するのが妥当かもしれません。事前設定端末セキュリティーを使用すると、CICS に定義された端末にユーザー ID を永続的に関連付けることができます。これは、端末のインストール時に CICS が端末に暗黙的に「サインオン」する (端末はその後にサインオンされるのではない) ことを意味します。事前設定セキュリティーは、多くの場合、キーボードがなくユーザーがサインオンできない装置 (プリンターなど) に対して定義されます。

端末ユーザー・セキュリティーに代わる手段として、通常のディスプレイ端末でこのセキュリティー形式を使用することもできます。これにより、事前設定セキュリティーを備えた端末に物理的にアクセスできるユーザーは誰でも、CICS にサインオ

ンする必要なく、その端末に対して許可されているトランザクションに入ることができます。端末は設置されている限りサインオンの状態を維持し、それに対して明示的なサインオフを実行することはできません。事前設定セキュリティーを備えたディスプレイ端末に関連付けられたユーザー ID が機密トランザクションの使用を許可される場合は、その端末が、アクセス制限された安全な場所にあるようにしてください。例えば、CICS ネットワーク・コントロール・センター内に物理的に配置された端末は、事前設定セキュリティーに適している可能性があります。

非端末セキュリティー

端末に関連付けられていないトランザクションのセキュリティーを指定できます。

端末に関連付けられていないトランザクションとしては、以下のものがあります。

- 開始された非端末トランザクション
- 一時データのトリガー・レベル・トランザクション
- CICS 初期設定時に実行されるプログラム・リスト・テーブル (PLT) プログラム

非端末セキュリティーについて詳しくは、 99 ページの『非端末トランザクションの保護』を参照してください。

QUERY SECURITY コマンド

CICS 制御対象リソースに対して CICS セキュリティー検査を使用することに加え (またはその代替として)、**QUERY SECURITY** コマンドを使用して、CICS アプリケーション内のセキュリティー・アクセスを制御できます。この方式により、CICS リソース・プロファイル以外の、リソースのセキュリティー・プロファイルを RACF に対して定義できます。さらに、標準リソース・クラスから使用できるセキュリティー検査よりも、さらに詳細なレベルのセキュリティー検査を可能にします。

詳しくは、以下を参照してください。

- QUERY SECURITYを参照してください。
- トランザクション内のリソース・セキュリティー検査のために RACF がサポートするリソース・クラスについては、 25 ページの『RACF 一般リソース・プロファイル』を参照してください。
- リソース・セキュリティー検査の詳細については、 100 ページの『リソース・セキュリティー』を参照してください。

相互通信セキュリティー

多数の CICS 領域を相互に接続するには、相互通信、例えば、ACF/Communications Server などの SNA アクセス方式を使用する SNA を介したシステム間通信 (SNA を介した ISC) を使用して、必要な通信プロトコルを提供します。相互接続されたシステムには基本的なセキュリティー原則が適用されますが、リソース定義がより複雑になり、セキュリティー要件も増えます。

APPC (LU6.2) セッション・セキュリティー
: CICS が使用する SNA を介した ISC プロトコルの 1 つが拡張プログラム間通信 (APPC) であり、これは SNA アーキテクチャーの LU6.2 部分の CICS 実装です。CICS は APPC セッション、接続、およびパートナーをリソースとして扱い、それらすべてにセキュリティー要件があります。CICS は、APPC 環境に以下のセキュリティー・メカニズムを提供します。

- バインド時の (またはセッション・) セキュリティーは、無許可のリモート・システムが CICS に接続するのを防ぎます。
- リンク・セキュリティーは、リモート・システムが接続を介したアクセスを許可されている CICS トランザクションおよびリソースの完全なセットを定義します。
- ユーザー・セキュリティーは、ユーザーが CICS トランザクションに接続すること、およびトランザクションが使用するようプログラムされているリソースと SPI コマンドすべてにアクセスすることを許可されているかどうかを検査します。

詳細については、『 275 ページの『LU6.2 セキュリティーの実装』』を参照してください。

複数領域操作 **(MRO)**
: 相互通信を使用するもう一つの方法は、複数領域操作 (MRO) です。これは、システム・ネットワーク体系 (SNA) アクセス方式に関係なく、単一システプレックス内の CICS 領域間のリンクに使用できます。MRO セキュリティーについて詳しくは、335 ページの『MRO セキュリティーの実装』を参照してください。

IP 相互接続 **(IPIC)** セキュリティー
: IPIC 接続のセキュリティー・メカニズムは APPC (LU6.2) 接続のセキュリティー・メカニズムと似ていますが、実装は以下のように異なります。

- バインド時のセキュリティーによって、許可が与えられていないリモート・システムが CICS に接続しないようにします。IPCONN の場合、バインド・セキュリティーは Secure Sockets Layer (SSL) クライアント証明書の交換により施行されます。
- リンク・セキュリティーは、リモート・システムが IPCONN を介したアクセスを許可されている CICS トランザクションおよびリソースの完全なセットを定義します。
- ユーザー・セキュリティーは、ユーザーが CICS トランザクションに接続すること、およびトランザクションが使用するようプログラムされているリソースと SPI コマンドすべてにアクセスすることを許可されているかどうかを検査します。ユーザー・セキュリティーはリンク・セキュリティーのサブセットです。つまりユーザーは、自分のユーザー ID でアクセス可能と定義されたセット内にリソースが含まれていても、リンク・ユーザー ID でアクセス可能なリソースのセット内にそのリソースが含まれていない場合、そのリソースにアクセスすることはできません。

IPIC 接続について詳しくは、システム間の通信を参照してください。

サインオン・セキュリティー用のパスチケット

パスチケットとは、別の CICS 領域などの特定のシステムの特定のアプリケーションにプログラムがサインオンするために使用できる、安全な方法で表されたパスワードのことです。特定のパスチケットは 1 回の認証にだけ使用可能で、生成されてから 10 分以内に使用する必要があります。パスチケットは、パスワードを使用できる場所であればどこでも使用できます。

PassTicket を使用する理由

パスワードの代わりにパスチケットを使用すれば、宛先システムにサインオンするためにアプリケーションがパスワードを保管する (またはユーザーにパスワードの再入力を求める) 必要はなく、パスワードはネットワーク上を送信されません。

仕組み

発信システム上のクライアントは、RACF パスチケット生成プログラムのアルゴリズムを使用して、宛先システムのためにパスチケットを生成する必要があります。

発信システムが CICS である場合は、サインオン・ユーザーのパスチケットを作成するために、アプリケーションは **EXEC CICS REQUEST PASSTICKET** または **FEPI REQUEST PASSTICKET** コマンドを発行して、RACF (または、パスチケットをサポートする、機能的に同等の外部セキュリティー・マネージャー) に、パスチケットの生成を要求します。

宛先システムは、外部セキュリティー・マネージャーを使用して、ユーザー ID とパスチケットを認証します。既存のコマンドおよびプロシージャーを認証に使用することができます。

CICS は、ユーザー ID およびパスチケットを認証する際に、RACF などの外部セキュリティー・マネージャーを呼び出して、提供されたパスチケットがその領域用の指定されたユーザー ID のものであるかどうかを検査します。指定されたユーザー ID が、指定されたグループ ID に結び付けられていることを確認するために、オプションの検査を実行することもできます。

詳細情報

『z/OS Security Server RACF Security Administrator's Guide』の『Using the Secured Signon Function』では、パスチケットについて詳しく説明されています。

「z/OS Security Server RACF マクロおよびインターフェース」の『パスチケット生成プログラム・アルゴリズムのプログラムへの組み込み』では、RACF パスチケット生成プログラムに使用されるアルゴリズムについて詳しく説明されています。

72 ページの『セキュア・サインオンのためのパスチケットの実装』では、パスチケットを CICS 環境に実装する方法が示されています。

API および SPI 制限の DFHAPIR parmlib メンバー

DFHAPIR parmlib メンバーは、制限付き CICS API コマンド および SPI コマンドを識別する規則を格納します。 CICS 変換プログラムは、この parmlib メンバーを使用して、変換中に、指定されたコマンド規則に対してプログラム・ソースを検査します。デフォルトでは、コマンド規則は、DFHAPIR が配置されている LPAR 上のすべてのユーザーに適用されます。ただし、RACF のプロファイルを使用して、一部のユーザーまたは LPAR を免除できます。

DFHAPIR parmlib メンバーの RACF プロファイルを定義する必要がある場合

FACILITY クラスの RACF プロファイル DFHAPIR.lpar によって、ユーザーがコマンド規則ファイルの対象であるかどうかが定義されます。ユーザーがこのプロファイルに対するアクセス権限を持っていない場合、またはプロファイルが定義されていない場合は、ユーザーはコマンド規則ファイルの対象になります。ユーザーがこのプロファイルに対する READ アクセス権限を持っている場合、ユーザーはコマンド規則ファイルの対象になりません。

以下に、2 つの共通シナリオの例を示します。

- コマンド規則はほとんどのユーザーに適用されますが、一部のユーザーは免除されます。
- コマンド規則は一部の LPAR に適用され、すべてに適用されるわけではありません。

例: DFHAPIR parmlib メンバーの RACF プロファイルの定義

RACF FACILITY クラスは、DFHAPIR parmlib メンバーを保護するために使用されます。以下に、2 つの共通シナリオの RACF プロファイルとアクセス・リストの定義方法の例を示します。

例 **1:** すべての **LPAR** に適用される一般プロファイルの定義

この例では、すべての LPAR に適用されるプロファイルを作成します。UACC(NONE) は、コマンド規則がデフォルトですべてのユーザーに適用されることを意味します。

```
RDEFINE FACILITY DFHAPIR.** UACC(NONE)
```

一部のユーザーがコマンド規則をバイパスできるようにするには、PERMIT コマンドを使用してアクセス・リストをセットアップすることによって、プロファイルに対する READ アクセス権限をユーザーに付与します。

```
PERMIT DFHAPIR.** CLASS(FACILITY) ID(user) ACCESS(READ)
```

例 **2:** 特定の **LPAR** に適用されるプロファイルの定義

コマンド規則を特定の LPAR でのみ実行するには、DFHAPIR.*lpar* プロファイルをセットアップします。ここで、*lpar* は 4 文字の LPAR 名です。これらは、以下のように、ユーザーがコマンド規則ファイルの対象となっている LPAR に対してセットアップする必要があります。

```
RDEFINE FACILITY DFHAPIR.** UACC(READ)
RDEFINE FACILITY DFHAPIR.lpar UACC(NONE)
```

より一般的なプロファイルの場合、UACC(NONE) は、コマンド規則が LPAR 上のすべてのユーザーに適用されることを意味します。必要に応じて、アクセス・リストを使用して、プロファイルに対する READ アクセス権限を免除ユーザーに付与します。

```
PERMIT DFHAPIR.lpar CLASS(FACILITY) ID(user) ACCESS(READ)
```

RACF ファシリティー

CICS は、リソースを保護するために多数の RACF 機能を使用します。

RACF には、以下の機能があります。

- システム・リソースの個々のユーザーを識別する情報や、保護を必要とするリソースを識別する情報を記録するために必要な機能。ユーザーおよびリソースに関して RACF に定義する情報は、ユーザーおよびリソースのプロファイルに保管されます。
- どのユーザーまたはユーザー・グループが、プロファイルが定義されているリソースへのアクセスを許可されるか、またはリソースへのアクセスから除外されるかを定義する機能。特定のリソースへのアクセスを許可されたユーザーまたはユーザー・グループを記録する情報は、リソースを保護するプロファイル内のアクセス・リストに保持されます。
- MVS システムで稼働しているサブシステムまたはジョブによって発行された要求を処理し、RACF に定義されたユーザーの ID を認証し、リソースに対するそれらのユーザーのアクセス許可を検査する手段。
- ユーザーのサインオンとサインオフ、RACF コマンドの発行、保護リソースへのアクセス試行などのセキュリティー関連イベントをログに記録する機能。成功した保護リソースへのアクセス試行は、MVS システム管理機能 (SMF) によって記録される可能性があります。成功したかどうかにかかわらず、保護リソースへのすべてのアクセス試行を記録する場合は、RACF 監査を使用します (z/OS Security Server RACF 監査担当者のガイドを参照してください)。 RACF 監査員は、RACF 報告書作成プログラムを実行して、SMF レコードに基づくレポートを生成することができます。SMF への RACF 監査メッセージのロギングについて詳しくは、 109 ページの『SMF への RACF 監査メッセージのロギング』を参照してください。

  RACF を使用した監査機能 (RACF コマンドでの監査オペランドの指定、および RACF 報告書作成プログラムを使用した監査対象セキュリティー関連アクティビティーのレポート生成) の実行について詳しくは、 z/OS Security Server RACF 監査担当者のガイドを参照してください。

RACF 管理

1 つ以上の CICS 領域、および CICS アプリケーションのユーザーに関わるセキュリティー管理者としてのお客様の仕事は、インストール済み環境のデータが適切に保護されるようにすることです。

RACF を使用して、すべてのシステム・リソース (このマニュアルでは特に CICS リソース) を保護する責任があります。

RACF の重要な特長の 1 つに、その階層的管理構造があります。 RACF セキュリティー管理者はその階層の頂点に位置する者として定義され、システム全体のセキュリティーを管理する権限を持ちます。お客様が RACF セキュリティー管理者でない場合は、RACF プロファイルやシステム全体の設定を操作できる十分な権限をお客様に委任するよう、RACF セキュリティー管理者に依頼してください。さらに、

RACF 監査員とも協力する必要があります。RACF 監査員は、RACF によって生成されたレコードの監査に基づいて、セキュリティー関連アクティビティーのレポートを作成できます。

RACF セキュリティー管理者は、システム SPECIAL 属性、グループ SPECIAL 属性、または他の権限の組み合わせを持っています。

- システム SPECIAL 属性を持っている場合は、任意の RACF コマンドを発行でき、任意の RACF プロファイルを変更できます (一部の監査関連オペランドを除く)。
- グループ SPECIAL 属性を持っている場合、権限は、SPECIAL 属性が設定されている RACF グループの範囲に制限されます。
- その他の権限には、以下のものがあります。
  - CLAUTH (クラス権限) 属性。これにより、特定の RACF クラスに RACF プロファイルを定義できます。
  - 既存の RACF プロファイルの OWNER であることに付随する権限。これにより、プロファイルのリスト作成、アクセスの変更、プロファイルの削除が可能になります。
  - RACF グループの CONNECT や JOIN グループなどのグループ権限を持っていること。

RACF コマンドの発行に必要な権限の詳細と、権限の委任および RACF グループの範囲については、 z/OS Security Server RACF 監査担当者のガイドを参照してください。

RACF コマンドを発行するための RACF 要件については、z/OS Security Server RACF コマンド言語解説書 のコマンドの説明を参照してください。

TSO セッションから LISTUSER コマンドを発行することにより、システム SPECIAL 属性を持っているかグループ SPECIAL 属性を持っているかを確認できます。システム SPECIAL 属性を持っている場合、SPECIAL は出力の最初の部分の USER ATTRIBUTES 句の後に出現します。グループ SPECIAL 属性を持っている場合、SPECIAL は、RACF グループへの接続を記述したオフセット・セクション内の USER ATTRIBUTES 句の後に出現します。LISTUSER 出力の例を含む詳細な説明については、z/OS Security Server RACF ユーザーズ・ガイドを参照してください。

RACF 管理責任の委任

CICS セキュリティー管理者として、以下のタスクを実行します (システム SPECIAL 属性を持っていない場合は、必要な権限を取得してください)。

- **CICS** 関連の一般リソース・クラス内でプロファイルを定義および保守します。一般に、この権限を付与するには、指定されたクラスの CLAUTH (クラス権限) 属性をユーザーに割り当てます。例えば、RACF セキュリティー管理者は次のコマンドを発行できます。

  ```
  ALTUSER your_userid CLAUTH(TCICSTRN)
  ```

このコマンドにより、同じ POSIT 番号のすべてのクラスにアクセスできるようになります。POSIT 番号は、クラス記述子テーブル (CDT) の ICHERCDE マクロのオペランドです。詳細については、 26 ページの『CICS クラスのアクティブ化』を参照してください。

- その他のリソース・クラス内でプロファイルを定義および保守します。本書で言及される一般リソース・クラスの多く (例えば APPL、APPCLU、FACILITY、OPERCMDS、SURROGAT、TERMINAL、VTAMAPPL など) は、CICS 以外の製品の操作に影響を及ぼします。お客様が RACF セキュリティー管理者でない場合は、RACF セキュリティー管理者にプロファイルの定義を依頼する必要があるかもしれません。
- **RACF** ユーザー・プロファイルをシステムに追加します。一般に、この権限を付与するには、ユーザー・プロファイルで「USER」に CLAUTH (クラス権限) 属性を割り当てます。例えば、RACF セキュリティー管理者は次のコマンドを発行できます。

  ```
  ALTUSER your_userid CLAUTH(USER)
  ```

  システムにユーザーを追加するときは、そのユーザーにデフォルトの接続グループを割り当ててください。これにより、(ユーザーをグループのメンバーとして追加することで) グループのメンバーシップが変更されます。したがって、グループ内の JOIN グループ権限またはグループ内のグループ SPECIAL 属性を持っている場合、あるいはグループの OWNER である場合は、CLAUTH(USER) によりユーザーをシステムに追加し、それらのユーザーをグループのスコープ内にあるグループに接続することができます。
- **RACF** システム全体の設定をリストし、**CICS** に関連するすべてのプロファイルを処理します。この操作の権限を付与するには、RACF グループをセットアップし、特定の CICS 関連 RACF プロファイルがそのグループのスコープに含まれるようにし、グループ SPECIAL 属性を持つグループにユーザーを接続します。例えば、RACF セキュリティー管理者は次のコマンドを発行できます。

  ```
  CONNECT your_userid GROUP(applicable-RACF_groupid) SPECIAL
  ```

SETROPTS GENERICOWNER コマンドが有効であり、接頭部付けがアクティブである場合は、管理者を割り当てることができます。この操作を行うには、接頭部を高位修飾子として使用して、各クラスに総称プロファイルを作成します。例えば、以下のような項目が含まれています。

```
RDEFINE TCICSTRN cics_region_id.** UACC(NONE)
        OWNER(cics_region_administrator_userid)
```

SETROPTS GENERIC コマンドは、総称プロファイルを定義する前に使用する必要があります (32 ページの『RACF コマンドの要約』を参照)。

RACF 管理の委任について詳しくは、z/OS Security Server RACF セキュリティー管理者のガイドを参照してください。

RACF プロファイル

RACF では、プロファイル はユーザー、ユーザーのグループ、または 1 つ以上のコンピューター・リソースのセキュリティー特性を記述します。

ユーザー・プロファイル
: ユーザー・プロファイル とは、RACF で定義されたユーザーを記述したも

のです。このプロファイルの情報には、ユーザー ID、ユーザー名、ユーザーのパスワード、プロファイル所有者、ユーザー属性、その他のデータが含まれます。また、ユーザー・プロファイルには CICS を含めたサブシステムについてのユーザー関連の情報も含まれます。

グループ・プロファイル
: グループ・プロファイル とは、ユーザーのグループを定義するものです。プロファイルの情報には、グループ名、プロファイル所有者、およびグループ内のユーザーが含まれます。

データ・セット・プロファイル
: データ・セット・プロファイル は、1 つ以上のデータ・セットに対して RACF 保護を提供します。プロファイルの情報には、データ・セット・プロファイル名、プロファイル所有者、汎用アクセス権限、アクセス・リスト、およびその他のデータが含まれています。

  データ・セット・プロファイルは、総称にすることも、個別にすることもできます。

  - 総称プロファイル は、類似する名前および同一のセキュリティー要件を持つ複数のリソースを保護します。
  - 個別プロファイル は、単一のリソースを保護します。

一般リソース・プロファイル

: 一般リソース・プロファイル は、データ・セット以外のコンピューター・リソースに対して RACF 保護を提供します。プロファイルの情報には、一般リソース・プロファイル名、プロファイル所有者、汎用アクセス権限、アクセス・リスト、およびその他のデータが含まれています。類似した特性を持つ一般リソースは、同じクラス に属します。

  総称プロファイルと同様に、リソース・グループ・プロファイル は、同一のセキュリティー要件を持つ複数のリソースを保護します。ただし、リソースの名前が類似している必要はありません。類似した特性を持つリソース・グループ・プロファイルは、同じリソース・グループ・クラス に属します。

  リソース・プロファイルは、総称にすることも、個別にすることもできます。

  - 総称プロファイル は、類似する名前および同一のセキュリティー要件を持つ複数のリソースを保護します。
  - 個別プロファイル は、単一のリソースを保護します。

RACF ユーザー・プロファイル

ユーザー・プロファイルとは、RACF で定義されたユーザーを記述したものです。このプロファイルの情報には、ユーザー ID、ユーザー名、ユーザーのパスワード、プロファイル所有者、ユーザー属性、その他のデータが含まれます。また、ユーザー・プロファイルには CICS を含めたサブシステムについてのユーザー関連の情報も含まれます。

ユーザー・プロファイルは、1 つ以上のセグメントから構成されます。つまり、RACF セグメントが 1 つと、その他のオプションのセグメントです。CICS ユーザーにとって重要なセグメントは以下のとおりです。

- RACF セグメント。RACF ユーザー・プロファイルの基本情報を保持します。『RACF セグメント』を参照してください。
- CICS セグメント。CICS ユーザーごとにデータを保持します。 17 ページの『CICS セグメント』を参照してください。
- LANGUAGE セグメント。ユーザーの各国語設定を指定します。 22 ページの『LANGUAGE セグメント』を参照してください。

RACF セグメント

RACF ユーザーは、RACF がそのユーザーの RACF プロファイルに関連付ける、英数字のユーザー ID によって識別されます。

RACF に定義する「ユーザー」は、CICS 端末ユーザーなどの人間でなくても構いません。例えば CICS 環境では、RACF ユーザー ID は、CICS を開始タスクとして開始するために使用するプロシージャーに関連付けることができます。また、ユーザー ID は (事前設定セキュリティーのために) CICS 端末に関連付けることができます。以下のリストは、RACF がユーザーについて保持する基本セグメント情報の一部を示しています。

キーワード
: 説明

USERID
: ユーザーのユーザー ID

NAME
: ユーザーの名前

OWNER
: ユーザーのプロファイルの所有者 (RACF 管理者または管理者が権限を与えた他のユーザー、あるいは RACF グループ)

DFLTGRP
: ユーザーが属しているデフォルト・グループ

AUTHORITY
: デフォルト・グループにおけるユーザーの権限

PASSWORD
: ユーザーのパスワード

ユーザー・プロファイルの RACF セグメントは、ADDUSER コマンド、または RACF ISPF パネルを使用して定義します。CICS ユーザーのユーザー・プロファイルの RACF セグメントを計画する場合は、それが含まれるグループを特定します。まず、ユーザーの RACF 管理単位を特定します。例えば、同じマネージャーの下にいるすべてのユーザーや、受注を行うすべてのユーザーを、管理単位として考慮できます。 RACF はこれらの単位を、CICS システム・リソースにアクセスするための類似の要件を持つ個々のユーザーのグループとして処理します。

ユーザーをシステムに追加するために必要な手順の概要については、z/OS Security Server RACF セキュリティー管理者のガイドを参照してください。

CICS セグメント

RACF ユーザー・プロファイルの CICS セグメントには、CICS ユーザーのデータが含まれます。

CICS がオペレーター情報を検索する順序について詳しくは、 90 ページの『ユーザーの CICS 関連データの取得』を参照してください。

CICS セグメントで指定できる情報は、以下のとおりです。

OPCLASS({1|*number*})
CICS は、CICS トランザクション内で開始される基本マッピング・サポート (BMS) メッセージをルーティングするときに、オペレーター・クラスを使用しますオペレーター・クラスは、1 から 24 までの数値です。

EXEC CICS ROUTE コマンドを (オプションの) OPCLASS パラメーター付きで発行する CICS トランザクションを使用するユーザーに対して、オペレーター・クラスを指定します。自動ルーティングを行うには、ユーザー・プロファイルの CICS セグメントで、対応する値をオペレーター・クラスとして指定します。

メッセージ・ルーティングについて詳しくは、メッセージ・ルーティングを参照してください。

OPIDENT({blank|*name*})
それぞれのオペレーターに割り当てる 1 文字から 3 文字のオペレーター ID コード。

オペレーターがサインオンすると、CICS は、CICS 端末管理テーブル (TCTTE) 内のオペレーターの端末項目にコードを保管します。このオペレーター ID は、特定の CICS メッセージに表示されます。また、BMS メッセージをルーティングするために EXEC CICS ROUTE コマンドで使用することもできます。CEDA LOCK コマンドでも使用されます。

メッセージ・ルーティングについて詳しくは、メッセージ・ルーティングを参照してください。

OPPRTY({0|*number*})
オペレーター優先順位値は、CICS 端末でオペレーターが呼び出す CICS トランザクションのタスク優先順位を決定するときに CICS が使用する 10 進数です。優先順位の値は 0 から 255 の範囲で指定できます。255 が最高の優先順位です。

CICS は、オペレーター優先順位、端末優先順位、およびトランザクション優先順位の合計を使用して、トランザクションのディスパッチング優先順位を決定します。

TIMEOUT({0000|*hhmm*})
ユーザーが最後に端末を使用してから、CICS が端末を「タイムアウトにする」までに経過するべき時間。

この時間は、0 から 9959 までの 10 進整数でなければなりません (最後の 2 桁は分数を表しており、00 から 59 までの値を指定します。左側の桁は時間を表します)。

1 時間 8 分を指定するには、ここで値 0108 を指定します。例えば、以下のような項目が含まれています。

```
ALTUSER userid CICS(TIMEOUT(0108))
```

値 0 (デフォルト) は、端末がタイムアウトしないことを意味します。

XRFSOFF({NOFORCE|FORCE})

CICS 持続セッションの再始動、および拡張リカバリー機能 (XRF) のサインオフ・オプションです。持続セッションの再始動または XRF テークオーバーの後に CICS がオペレーターをサインオフするかどうかを指示するには、この情報を指定します。

FORCE

持続セッションの再始動または XRF テークオーバーが発生した場合に CICS にオペレーターを自動的にサインオフさせるには、FORCE を指定します。

NOFORCE

持続セッションの再始動または XRF テークオーバーが発生した場合に CICS にオペレーターのサインオン状態を維持させるには、NOFORCE を指定します。

類似する端末のグループ・レベルや CICS システム・レベルで XRFSOFF 機能を指定できます。

- 類似する端末のグループに対して XRFSOFF 機能を指定するには、TYPETERM リソース定義の RSTSIGNOFF 属性を使用します。
- この機能をシステム・レベルで指定するには、**RSTSIGNOFF** システム初期設定パラメーターを使用します。

どちらの場合も、デフォルト値は NOFORCE です。システム初期設定テーブルまたは TYPETERM で FORCE オプションを指定すると、CICS セグメントに指定された NOFORCE の値はオーバーライドされます。

一時記憶域サーバーへのアクセスの許可 は、システム初期設定パラメーター、TYPETERM 定義、および CICS セグメントでの FORCE または NOFORCE 指定の組み合わせによって、持続セッションの再始動または XRF テークオーバーの後に端末のサインオン状態を維持するかどうかが決まることを示します。持続セッションの再始動または XRF テークオーバーの後も端末のサインオン状態を維持するには、**3** つの場所すべてで NOFORCE が指定されている必要があります。

表 *1. FORCE* オプションと *NOFORCE* オプションの効果

TYPETERM 定義	**CICS** セグメント	システム初期設定パラメーター	結果として生じる端末状況
FORCE	FORCE	FORCE	サインオフ
FORCE	FORCE	NOFORCE	サインオフ
FORCE	NOFORCE	FORCE	サインオフ
FORCE	NOFORCE	NOFORCE	サインオフ
NOFORCE	FORCE	FORCE	サインオフ
NOFORCE	FORCE	NOFORCE	サインオフ

表 *1. FORCE* オプションと *NOFORCE* オプションの効果 *(*続き*)*

TYPETERM 定義	**CICS** セグメント	システム初期設定パラメーター	結果として生じる端末状況
NOFORCE	NOFORCE	FORCE	サインオフ
NOFORCE	NOFORCE	NOFORCE	サインオン

注**:** テークオーバーが **RSTSIGNTIME** システム初期設定パラメーターで指定された時間を超えた場合、ゼロ以外の TIMEOUT 値を持つ端末のユーザーは、テークオーバー後にサインオン状態は維持されません。例えば、XRFSOFF=NOFORCE が設定されたシステムで、次のように指定されているとします。

```
ALTUSER USER1 CICS(XRFSOFF(NOFORCE) TIMEOUT(10))
ALTUSER USER2 CICS(XRFSOFF(NOFORCE) TIMEOUT(1))
```

システム初期設定パラメーターに XRFSTME=5 が指定されているシステムで持続セッションの再始動または XRF テークオーバーが発生し、再始動またはテークオーバーにかかった時間が 5 分を超える場合、USER1 はサインオン状態が維持されませんが、USER2 はサインオンしたままになります。

CICS セグメントでのデフォルト値の指定**:**

リストされるデフォルトは、そのユーザー ID に対して CICS セグメントが定義されている場合のみ有効です。

CICS セグメントをデフォルトにするには、以下のように定義します。

```
ADDUSER userid DFLTGRP(group_name) NAME(user_name)
        OWNER(group_id|userid)
        PASSWORD(password)
        CICS
```

例えば、デフォルト・ユーザー属性ではなくシステム・デフォルトを適用したい場合、またはテスト・システムをセットアップしていて、使用したい値をまだ決めていない場合は、この方法で CICS セグメントを定義することができます。

CICS セグメントを完全に省略した場合は、 90 ページの『ユーザーの CICS 関連データの取得』で説明されているとおりにデフォルトが取得されます。

CICS セグメントの一部のオプションを指定し、その他を省略した場合は、省略されたオプションに対してこの文書で説明するデフォルトが適用されます。

以下のようにして、CICS セグメントを除去できます。

```
ALTUSER userid NOCICS
```

CICS ユーザーのセグメント・データの作成または更新**:**

CICS ユーザーの CICS セグメント・データを作成または更新するには、新しいユーザーに対して RACF ADDUSER コマンドで、または既存のユーザーに対して ALTUSER コマンドで CICS オプションを指定します。

例えば以下のコマンドは、新規 CICS ユーザーを、関連 CICS オペレーター・データと共に RACF データベースに追加します。

```
ADDUSER userid  DFLTGRP(group_name) NAME(user_name) OWNER(group_id)
        PASSWORD(password)
        CICS(OPCLASS(1,2,...,n) OPIDENT(identifier) OPPRTY(priority)
             TIMEOUT(timeout_value) XRFSOFF(NOFORCE))
        LANGUAGE(PRIMARY(primary_language))
```

以下の ALTUSER コマンドの例は、RACF データベース内の既存のユーザーに CICS オペレーター・データを追加します。

```
ALTUSER userid
        CICS(OPCLASS(1,2,...,n) OPIDENT(identifier) OPPRTY(priori`ty)
             TIMEOUT(timeout_value) XRFSOFF(NOFORCE))
        LANGUAGE(PRIMARY(primary_language))
```

これらのコマンドを発行して CICS オペレーター・データを定義する前に、CICS 提供の RACF 動的解析検証ルーチンがリンク・リスト内の APF 許可ライブラリーにインストールされていることを確認してください。これらの出口について詳しくは、39 ページの『CICS 提供の RACF 動的解析検証ルーチン』を参照してください。

システム SPECIAL 属性を持っていない場合は、RACF セキュリティー管理者に、ユーザー・プロファイル内の CICS セグメントおよび LANGUAGE セグメントをリストまたは更新する権限を要求してください。これらのセグメントのリストまたは更新は、RACF FIELD クラスにプロファイルを作成することによって行われます。詳しくは、38 ページの『RACF プロファイル内のフィールドへのアクセスの制御』を参照してください。

opclass を変更したいけれどもリストを再指定したくない場合は、ADDOPCLASS および DELOPCLASS オペランドを使用できます。例:

```
   ALTUSER userid
CICS(ADDOPCLASS (1,2)
     DELOPCLASS (6,7))
```

リモート・ユーザーの ***RACF*** プロファイルの変更*:*

サインオン・リモート・ユーザー、または物理端末やコンソールを直接使用しないサインオン・ユーザーの RACF プロファイルでの特定の変更は、タイプ 71 の RACF イベント通知 (ENF) によって CICS に通知されます。

CICS は、特定の RACF コマンドがユーザーのグループ許可に影響を与えるときに送信される、RACF タイプ 71 ENF をモニターします。ユーザー ID に対する変更の通知によって、**USRDELAY** システム初期設定パラメーターに指定された設定が指定変更されます。そのため、**USRDELAY** 設定を確認してください。

- z/OS 1.11 以降の場合、これらの RACF コマンドは、REVOKE オプション付きの **ALTUSER**、**CONNECT**、および **REMOVE** です。
- APAR OA39486 用の PTF を適用済みの z/OS 1.13 以降の場合、これらの RACF コマンドは、REVOKE オプション付きの **ALTUSER**、**CONNECT**、**REMOVE**、**DELGROUP**、および **DELUSER** です。

例えば、ユーザーが **ALTUSER** コマンドで日付を指定せずに REVOKE オプションを使用して、ユーザー ID を即効で取り消すと、CICS に通知されます。ただし、ユ

ーザー ID の有効期限が切れている場合、またはローカル領域にサインオンしているユーザー ID (例えば、サインオンに CESN トランザクションを使用する TOR) の場合は、CICS に通知されません。

RACF プロファイルの変更が行われ、CICS がユーザー ID の新規接続要求を受け取ると、CICS はそのユーザー ID の暗黙的サインオンを実行し、新しい RACF プロファイル情報が使用されます。そのユーザーの既存のタスクは、タスクが接続されたときに有効であった RACF プロファイルを引き続き使用します。

CICS が RACF に加えられた変更を迅速に検出できるように **USRDELAY** システム初期設定パラメーターに低い値を指定した場合、CICS は RACF プロファイルの変更が生じると即時に通知を受け取るようになったため、この値を引き上げることをお勧めします。 **USRDELAY** 値を大きくすることの主な影響は、 RACF 制御ブロックに使用されるストレージの量が増えることです。

ユーザーを取り消すなどしてサインオン・リモート・ユーザーの RACF プロファイルを変更した場合、以下のいずれかの状態となるまで、CICS は最初の接続要求時に設定された許可を引き続き使用します。

- トランザクションが同期点を実行する。
- 接続要求が終了する。
- RACF が CICS にユーザー・プロファイルの変更を通知し、そのサインオン・ユーザー ID に関連付けられている接続要求が、LOCAL を除く ATTACHSEC のすべてのオペランドに対して完了したためにサインオフとなる。
- RACF が CICS にユーザー・プロファイルの変更を通知し、新規接続要求が行われ、**USRDELAY** システム初期設定パラメーターの値が有効期限切れになっていないためにサインオフとなる。このサインオフの後には、サインオンが続きます。

CICS デフォルト・ユーザー**:**

CICS を外部セキュリティーと共に使用している場合、CICS は、サインオンしないすべての CICS 端末ユーザーに対して、CICS デフォルト・ユーザー のセキュリティー属性を割り当てます。

さらに CICS は、独自の CICS セグメント・データを持たないサインオン・ユーザーに対して、デフォルト・ユーザーの CICS セグメントからのオペレーター・データを割り当てます。CICS がデフォルトのセキュリティー属性およびオペレーター・データを割り当てるようにするには、CICS デフォルト・ユーザー ID を RACF に定義します。次に、**DFLTUSER** システム初期設定パラメーターを指定して、使用するデフォルト・ユーザーを CICS に知らせます。**DFLTUSER** パラメーターでデフォルト・ユーザー ID を指定しない場合、CICS は「CICSUSER」という名前を使用します。

インストール済み環境で定義されたオペレーター・データを **DFLTUSER** パラメーターで使用するか、あるいはデフォルトを使用するかにかかわらず、ユーザー ID が RACF に定義されていること、および領域ユーザー ID がデフォルト・ユーザーを使用するための代理セキュリティーをインストールしていることが不可欠です (154 ページの『代理ユーザー・セキュリティー』を参照)。

CICS は、システム初期設定時にデフォルト・ユーザーを「サインオン」します。システム初期設定パラメーターとして **SEC=YES** が指定されていて、**CICS** がデフォルト・ユーザー **ID** を「サインオン」できない場合、**CICS** の初期設定は失敗します。

CICS は、明示的にサインオンしない端末ユーザーに対し、デフォルト・ユーザー ID のセキュリティー属性を使用してすべてのセキュリティー検査を実行します。これらのセキュリティー検査には、トランザクション接続セキュリティー検査だけでなく、リソースおよびコマンドのセキュリティー検査が含まれます。

注**:** デフォルト・ユーザーの RACF プロファイルによってゼロ以外の TIMEOUT 値が指定される場合、その値はサインオンしない端末には適用されません。

LANGUAGE セグメント

言語セグメントは、ユーザーが CICS 発行メッセージを受け取るときの各国語に関する情報を保持します。

2 つの言語を指定できますが、CICS は各ユーザーに言語を 1 つだけ割り当てます。1 次言語が指定されていて、**NATLANG** システム初期設定パラメーターに指定された単一文字コードに対応する場合、CICS は 1 次言語を割り当てます。そうでない場合、CICS は、同じ基準に従って 2 次言語を割り当てます。 1 次言語も 2 次言語も **NATLANG** 値に対応していない場合は、他の場所から言語を提供する必要があります。 90 ページの『サインオン時の CICS 関連データの取得』を参照してください。

優先する各国語を RACF ユーザー・プロファイルの LANGUAGE セグメントに指定するには、その言語が CICS システムに定義されていることを確認し、ADDUSER または ALTUSER コマンドで **LANGUAGE** パラメーターを使用します。次の例は、ALTUSER コマンドでの言語要求を示します。

```
ALTUSER userid LANGUAGE(PRIMARY(language_code) SECONDARY(language_code))
```

指定できる情報は、以下のとおりです。

LANGUAGE
CICS ユーザーの 1 次言語と 2 次言語を指定します。 CICS は、セグメントに定義された言語を受け入れて使用しますが、RACF システム全体のデフォルトを無視します。これは、CICS には各国語に関して CICS システム初期設定パラメーター **NATLANG** で指定する、独自のシステム・デフォルトがあるためです。

PRIMARY(*language_code*)
システム・デフォルトをオーバーライドする、ユーザーの 1 次言語を指定します。 インストールした各国語機能に応じて、各国語コードの 3 文字コードの 1 つをこのパラメーターに指定できます。

SECONDARY(*language_code*)
システム・デフォルトをオーバーライドする、ユーザーの 2 次言語を指定します。インストールした各国語機能に応じて、各国語コードの 3 文字コードの 1 つをこのパラメーターに指定できます。

注**:**

1. CICS メッセージは、英国英語、中国語 (簡体字)、および日本語でのみサポートされます。これら 3 つの言語以外の言語を指定した場合は、デフォルトで英語が使用されます。
2. CICS は、以下の SETROPTS コマンドによって定義された RACF のデフォルト言語を無視します。

   ```
   SETROPTS LANGUAGE(PRIMARY(...) SECONDARY(...))
   ```

各国語について詳しくは、 93 ページの『各国語および非端末トランザクション』を参照してください。

RACF グループ・プロファイル

RACF で個々のユーザー・プロファイルを定義するだけでなく、グループ・プロファイルも定義することができます。グループ・プロファイルとは、ユーザーのグループを定義するものです。

グループ・プロファイルには、グループに関する情報 (例えば、所有者、サブグループ、接続しているユーザーのリスト) を含めることができます。グループ・プロファイルの定義および使用方法について詳しくは、「z/OS Security Server RACF セキュリティー管理者のガイド」を参照してください。グループ・プロファイルはリソース・グループ・プロファイルとは異なります。リソース・グループ・プロファイルはリソースのグループを定義するものであり、 25 ページの『RACF 一般リソース・プロファイル』で説明します。

グループのメンバーであるユーザーは、保護リソースへの共通アクセス権限を共有することができます。例えば、以下のようなグループをセットアップするとします。

- 同じ部門で働くユーザー
- RACF で保護することにしたトランザクション、ファイル、端末装置、その他のリソースの同一セットを操作するユーザー
- 同じ領域にサインオンするユーザー (複数の CICS 領域がある場合)

CICS 環境では、グループ・プロファイルにより次のようなさまざまなメリットがもたらされます。

- リソースへのアクセスの制御が簡単になる
- グループの SPECIAL 属性や CONNECT グループ権限を使用して権限を割り当てることが可能になる
- ストレージ中のプロファイルのリフレッシュが少なくて済む

制御のポイントを、リソース・プロファイルのアクセス・リストではなく、グループ内でのユーザー ID の有無に置くことを目指してください。あるメンバーが部門を離れた場合、その部門のユーザー・グループからユーザー ID を除去すると、すべての特権が取り消されます。プロファイルのその他の管理は必要ありません。このようにして RACF 管理を最小限に抑えて、リソース・プロファイル数が過剰にならないようにします。

RACF はリソース・プロファイルのコピーをストレージ内に保持しているため、これらのプロファイルに対する変更は、ストレージ内のプロファイルがリフレッシュされるまではシステムに反映されません。

リソースにアクセスする権限は、リソース・プロファイルの一部であるアクセス・リストに保持されています。この権限は、ユーザーまたはグループに対して認可されます。アクセス・リストでユーザーを追加または削除するには、主記憶域でプロファイルをリフレッシュします。詳しくは、 26 ページの『主記憶域内のリソース・プロファイルのリフレッシュ』を参照してください。

既にアクセス・リストに入っているグループにユーザーを結び付けたり、ユーザーをそこから削除したりすると、そのユーザーはプロファイルをリフレッシュしなくともグループの権限を獲得または失います。そのグループ内で CONNECT グループ権限を持つユーザーは誰でも、CONNECT および REMOVE コマンドを使用して、そのグループのメンバーシップを変更できます。このように、影響を受けるプロファイルのアクセス・リストを (PERMIT コマンドを使用して) 変更する必要はありません。 CICS 一般リソース・プロファイルを変更しない場合は、そのストレージ内コピーをリフレッシュする必要はありません。ただし、ユーザーのグループ・メンバーシップが変更された場合、ユーザーはもう一度サインオンしなければならない可能性があります。

グループの作成によって得られるその他の利点については、「z/OS Security Server RACF セキュリティー管理者のガイド」を参照してください。

次のコマンド・シーケンスは、ユーザーの新規グループを作成し、ユーザーを既存のグループから新規グループへ移動します。

```
ADDGROUP group_name2
REMOVE user1 GROUP(group_name1)
CONNECT user1 GROUP(group_name2)
```

CICS に通知されます。

RACF データ・セット・プロファイル

RACF 機能を使用して、保護するデータ・セットのプロファイルを定義することにより、直接アクセス・ストレージ・デバイス (DASD) およびテープのデータ・セットを保護できます。

データ・セット・プロファイルを RACF に定義するための規則については、z/OS Security Server RACF セキュリティー管理者のガイドおよびz/OS Security Server RACF コマンド言語解説書で説明しています。例については、z/OS Security Server RACF ユーザーズ・ガイドを参照してください。

次の 2 つの RACF カテゴリーのデータ・セットを保護するプロファイルを定義します

1. ユーザー・データ・セット用のプロファイル。この場合、高位修飾子は RACF ユーザー ID です。すべての RACF 定義ユーザーは、自分のデータ・セットを保護できます。
2. グループ・データ・セット用のプロファイル。この場合、高位修飾子は RACF グループ名です (RACF グループについては、 23 ページの『RACF グループ・プロファイル』を参照)。RACF 定義ユーザーは、必要な権限または属性を持っていれば、グループ・データ・セットを RACF 保護することができます。(詳しくは、z/OS Security Server RACF セキュリティー管理者のガイドを参照してください。)

注: データ・セット・プロファイルは、CICS 端末ユーザーには適用されず、CICS 領域ユーザー ID にのみ適用されます

総称データ・セット・プロファイル

総称プロファイルを使用することで、データ・セットを保護するために必要なプロファイルの数を削減でき、RACF データベースの必須サイズも削減できます。さらに、総称プロファイルはボリューム固有ではありません (つまり、総称プロファイルによって保護されるデータ・セットは任意のボリューム上に置くことができます)。

通常、総称データ・セット・プロファイル名は、総称文字を指定して (例えばプロファイル名にパーセント (%) またはアスタリスク (*) を使用して) 指定します。データ・セット・プロファイルの場合、RACF の拡張総称命名が有効であれば、RACF はアスタリスク (*) と二重アスタリスク (**) を区別します。 RACF DATASET クラスの総称プロファイル名を制御する規則については、z/OS Security Server RACF コマンド言語解説書を参照してください。

例えば、CICSTS55.CICS という名前のグループがある場合、「CICSTS55.CICS.**」という総称プロファイルを定義でき、このプロファイルのアクセス・リストにあるすべてのユーザーは、許可レベルで、高位修飾子 CICSTS55.CICS を持つデータ・セットにアクセスできます。例えば、以下のような項目が含まれています。

```
ADDSD 'CICSTS55.CICS.**' UACC(NONE) NOTIFY(admin_userid)
```

総称プロファイルを定義する前に、 32 ページの『RACF コマンドの要約』で説明されているように、SETROPTS GENERIC コマンドを使用します。

注: 本書の例では二重アスタリスク (**) を示していますが、これには拡張総称命名が有効であることが必要です。拡張総称命名が有効でない場合は、二重アスタリスクの代わりに単一のアスタリスク (*) を使用します。 (拡張総称命名は、RACF SETROPTS EGN コマンドを発行して有効にします。 SETROPTS EGN は、データ・セット名のみに影響を与えることに注意してください。拡張総称命名は、TCICSTRN などの一般リソース・プロファイルでは常に有効になっています。)

RACF 一般リソース・プロファイル

RACF は、CICS がリソースに使用する多数のリソース・クラスを提供します。

一般リソース・プロファイル は、データ・セット以外のコンピューター・リソースに対して RACF 保護を提供します。プロファイルの情報には、一般リソース・プロファイル名、プロファイル所有者、汎用アクセス権限、アクセス・リスト、およびその他のデータが含まれています。類似した特性を持つ一般リソースは、同じクラス に属します。

総称プロファイルと同様に、リソース・グループ・プロファイル は、同一のセキュリティー要件を持つ複数のリソースを保護します。ただし、リソースの名前が類似している必要はありません。類似した特性を持つリソース・グループ・プロファイルは、同じリソース・グループ・クラス に属します。

これらについては、 27 ページの『RACF classes for CICS リソース』で説明しています。 その他の RACF リソース・クラスには、CICS およびその他のサブシス

テムで使用されるリソース用のプロファイルが含まれます。これらのグループについては、 28 ページの『システム・リソースを保護するための RACF クラス』で説明しています。

リソースの保護

リソースを保護するには、以下の手順を実行する必要があります。

1. 適切なリソース・クラス内でリソースのプロファイルを定義します。
2. 以下を指定するアクセス・リスト を定義します。
   - リソースへのアクセスが許可されるユーザー
   - 各ユーザーに許可されるアクセスのレベル

CICS クラスのアクティブ化

CICS 領域によるセキュリティー検査で使用するために CICS リソース・クラスをアクティブ化するには、RACF **SETROPTS** コマンドを使用します。

CICS リソース・クラスがアクティブ RACF クラス記述子テーブルに定義されると、管理者はそのクラスに一般リソース・プロファイルを定義できます。詳しくは、 25 ページの『RACF 一般リソース・プロファイル』の RDEFINE および PERMIT の説明を参照してください。管理者が定義したプロファイルを CICS システムが使用する前に、クラスをアクティブ化する必要があります。

SETROPTS コマンドの形式は、SETROPTS CLASSACT(*classname*) です。例えば、以下のような項目が含まれています。

```
SETROPTS CLASSACT(TCICSTRN)
```

CDT 定義内で同じ POSIT 番号を持つ RACF 一般リソース・クラスのすべてのセットは、まとめてアクティブ化および非アクティブ化されます。したがって、IBM 提供の CICS クラスを 1 つ指定するだけで、IBM 提供のすべての CICS 関連クラスをアクティブ化できます。IBM 提供のクラスと同じ POSIT 番号を使用して独自のインストール定義クラスを定義する場合、それらは IBM 提供のクラスと共にアクティブ化および非アクティブ化されます。インストール定義クラスのセットに対して別の制御を行うには、別の POSIT 番号でそれらを定義します。(POSIT 番号について詳しくは、z/OS Security Server RACF マクロおよびインターフェースを参照。)

主記憶域内のリソース・プロファイルのリフレッシュ

以下の TSO コマンドを使用して、RACLIST に定義されているクラスをリフレッシュします。

```
SETROPTS RACLIST(xxxxxxxx) REFRESH
```

ここで、*xxxxxxxx* はリフレッシュされる RACF クラスです。**CEMT PERFORM SECURITY REBUILD** コマンドは NOT REQUIRED と応答します。

メンバー・クラスまたはリソース・グループ・クラスのプロファイルを追加または更新した後、次のコマンドを発行する必要があります。

```
SETROPTS RACLIST (TCICSTRN) REFRESH
```

GCICSTRN リソース・グループ・クラスにプロファイルを追加した場合でも、TCICSTRN メンバー・クラスのみに対してこのコマンドを発行する必要があります。このコマンドを使用すると、RACF で定義が更新され、CICS は変更されたプロファイルを使用します。

RACF classes for CICS リソース

CICS リソースを保護するには、適切なクラスまたは適切なリソース・グループ・クラスで、そのリソース用に一般リソース・プロファイルを作成する必要があります。RACF は CICS リソース用にいくつかのクラスを提供しています。 また、独自のリソース・クラスを定義することもできます。

注:

1. 独自のクラスを定義する場合は、クラス名の最初の文字が重要です。例えば、CICS プログラム用に独自のクラスを定義する場合、選択する名前は、メンバー・クラスの場合は M、リソース・グループ・クラスの場合は N で始まる必要があります。
2. DB2ENTRY リソースにはデフォルトのリソース・クラス名はありません。これらのリソースには独自のリソース・クラスを定義する必要があります。詳細については、DB2ENTRY リソースのリソース・クラスを参照してください。
3. RCICSRES クラスのプロファイル名には、それらが適用される CICS リソース・タイプを示す接頭部が含まれていなければなりません。例えば、ATOMSERVICE 定義は ATOMSERVICE.*name*、バンドルは BUNDLE.*name*、CICS 文書テンプレートは DOCTEMPLATE.*name*、EP アダプターは EPADAPTER.*name*、EP アダプター・セットは EPADAPTERSET.*epadapterset_resource_name*、イベント・バインディングは EVENTBINDING.*name*、XML 変換は XMLTRANSFORM.*name* です。

表 2. *CICS* リソースの *RACF* 提供リソース・クラス

メンバー・クラス	リソース・グループ・クラス	説明
TCICSTRN	GCICSTRN	CICS トランザクション、通常の接続セキュリティー
PCICSPSB	QCICSPSB	CICS PSB
ACICSPCT	BCICSPCT	CICS 開始トランザクションおよび以下の EXEC CICS コマンド。 COLLECT STATISTICS TRANSACTION DISCARD TRANSACTION INQUIRE TRANSACTION SET TRANSACTION
DCICSDCT	ECICSDCT	CICS 一時データ・キュー
FCICSFCT	HCICSFCT	CICS ファイル
JCICSJCT	KCICSJCT	CICS ジャーナル
MCICSPPT	NCICSPPT	CICS プログラム
SCICSTST	UCICSTST	CICS 一時記憶域キュー
CCICSCMD	VCICSCMD	EXEC CICS SYSTEM コマンドおよび EXEC CICS FEPI システム・コマンド

表 2. *CICS* リソースの *RACF* 提供リソース・クラス (続き)

メンバー・クラス	リソース・グループ・クラス	説明
RCICSRES	WCICSRES	文書テンプレート、バンドル、EP アダプター、EP アダプター・セット、イベント・バインディング、ATOMSERVICE 定義、および XML 変換

CICSPlex SM リソースの RACF クラス

CICSPlex SM リソースの保護は、以下の一般リソース・クラスによって提供されます。

CPSMOBJ
CICSPlex SM リソースへのアクセスを制御します。対応するリソース・グループ・クラスは GCPSMOBJ です。詳しくは、 211 ページの『CICSPlex SM セキュリティーの実装』を参照してください。

CPSMXMP
CICSPlex SM でシミュレートされた CICS セキュリティー検査からの免除を制御します。詳細については、 251 ページの『セキュリティー検査からのユーザーおよびリソースの免除』を参照してください

RACF リソース・クラス名

リソース・グループ・プロファイルを使用することで、リソース・クラス内で維持する必要があるプロファイルの数を削減できます。

また、重複メンバー名の定義を避けていれば、このメソッドを使用することで、CICS が初期設定時に作成する RACF ストレージ内のプロファイルのためのストレージ要件を削減できます。

RACF はストレージ内検査サービスを提供しており、そのサービスがないと RACF で必要になってしまう入出力操作を回避できます (これは RACROUTE REQUEST=FASTAUTH マクロを利用して実行されます)。この目的のために、CICS の初期設定時に、CICS は RACF にそのリソース・プロファイルを主記憶域にロードするように要求します。

管理を容易にするには、プロファイルを重複して定義しないようにします。 RACF がプロファイルをストレージにロードするときに重複が検出されると、RACF はそれらのプロファイルを ICHRLX02 選択出口に従ってマージします。選択出口がインストールされていない場合、RACF は、RLX2P データ域で指示されているデフォルトのマージ規則に従います。これについて詳しくは、「z/OS Security Server RACF セキュリティー管理者のガイド」の『グループ・プロファイル間の矛盾の解決』を参照してください。

システム・リソースを保護するための RACF クラス

CICS は多数のシステム・リソースを使用しており、これらは無許可アクセスから保護する必要があります。この保護は、いくつかの一般リソース・クラス内のプロファイルによって提供されます。

APPCLU

z/OS Communications Server セッションの確立中に、APPC パートナー論理装置 (LU タイプ 6.2) の ID を検査します。詳しくは、 277 ページの『APPCLU 一般リソース・クラスでのプロファイルの定義』を参照してください。

APPL 端末ユーザーによる z/OS Communications Server アプリケーション (CICS を含む) へのアクセスを制御します。詳しくは、 58 ページの『CICS 領域へのアクセスの許可』を参照してください。

CONSOLE

コンソールへのユーザー・アクセスを制御します。詳しくは、 82 ページの『コンソール・プロファイル』を参照してください。

DIGTCERT

デジタル証明書および関連情報が含まれます。詳しくは、 381 ページの『新規 RACF 証明書の作成』を参照してください。

FACILITY

FACILTY 一般リソース・クラスは、異なる複数のシステム・リソースを保護するために使用されます。これらのリソースについては、 30 ページの『FACILITY 一般リソース・クラスによって保護されるリソース』で説明します。

FIELD

RACF プロファイルのフィールドへのアクセスを制御します。詳細については、 38 ページの『RACF プロファイル内のフィールドへのアクセスの制御』を参照してください。

IDIDMAP

IDIDMAP リソース・プロファイルには、分散 ID フィルターが含まれます。RACF では、分散 *ID* フィルター という語を使用して、RACF ユーザー ID と 1 つ以上の分散 ID の間のマッピングの関連付けを記述します。詳しくは、 205 ページの『ID 伝搬のための RACF の構成』を参照してください。

JESSPOOL

JES スプール・データ・セットを保護します。詳細については、 62 ページの『CICS 環境での JES スプールの保護』を参照してください。

LOGSTRM

CICS がシステム・ログおよび一般ログに使用する MVS ログ・ストリームへのアクセスを制御します。詳しくは、 48 ページの『MVS ログ・ストリームへのアクセスを許可する』を参照してください。

OPERCMDS

- 特定の CICS 領域に向けて MODIFY コマンドを発行できるコンソール・ユーザーを制御します。詳しくは、 87 ページの『MVS システム・コンソールを CICS 端末として使用する』を参照してください。
- CICS が発行できるオペレーター・コマンドを制御します。例えば、コマンド・リスト・テーブル (CLT) 内のコマンドや、MODIFY ネットワーク・コマンドなどです。

PROGRAM
CICS を開始できるユーザーを制御します。詳しくは、 40 ページの『CICS ロード・ライブラリーの保護』を参照してください。

PROPCNTL
CICS から JES 内部読み取りプログラムに実行依頼されるジョブで、USER オペランドが指定されていないものに対しては、CICS 領域ユーザー ID が伝搬されないようにします。詳しくは、 60 ページの『ユーザー ID 伝搬の制御』を参照してください。

PTKTDATA
パスチケットの生成および検証に使用された暗号鍵が含まれます。詳しくは、 9 ページの『サインオン・セキュリティー用のパスチケット』を参照してください。

STARTED
MVS 開始ジョブのユーザー ID を提供するプロファイルが含まれます。詳しくは、 42 ページの『開始ジョブの STARTED プロファイルの使用』を参照してください。

SUBSYSNM
サブシステム (CICS のインスタンスなど) が VSAM ACB をオープンし、VSAM レコード・レベル共用 (RLS) 機能を使用することを許可します。詳細については、 57 ページの『SMSVSAM サーバーへのアクセスを許可する』を参照してください。

SURROGAT
他のユーザー ID の代理となることができるユーザー ID を指定します。詳しくは、 154 ページの『代理ユーザー・セキュリティー』を参照してください。

TERMINAL
個々の端末でユーザーがサインオンできるかどうかを制御します。対応するリソース・グループ・クラスは GTERMINL です。詳しくは、 83 ページの『Preset terminal security』を参照してください。

VTAMAPPL
ユーザーが SNA ACB をオープンできるかどうかを制御します。詳しくは、 60 ページの『CICS 領域の z/OS Communications Server ACB オープンの制御』を参照してください。

注**:** VTAM® は、現在は z/OS Communications Server (for SNA または IP) です。

FACILITY 一般リソース・クラスによって保護されるリソース

FACILTY 一般リソース・クラスは、異なる複数のシステム・リソースを保護するために使用されます。

それらは、以下のとおりです。

ライブラリー・ルックアサイド **(LLA)** ライブラリー
FACILITY クラスは、プログラムが LLACOPY マクロを使用する機能を制御します。詳しくは、 52 ページの『MVS ライブラリー・ルックアサイド (LLA) 機能によるアクセスの許可』を参照してください。

CICS 領域間通信プログラム、**DFHIRP**

FACILITY クラスは、領域が CICS 領域間通信プログラム DFHIRP にログオンする機能を制御します。詳しくは、 335 ページの『MRO セキュリティーの実装』を参照してください。

共用データ・テーブル

FACILITY クラスは、ファイル所有領域が共用データ・テーブル・サーバーとしての役割を果たす機能を制御します。詳しくは、 350 ページの『CICS 共用データ・テーブルのセキュリティー』を参照してください。

カップリング・ファシリティー・データ・テーブル

FACILITY クラスは以下を制御します。

- CICS 領域がカップリング・ファシリティー・データ・テーブル (CFDT) に接続する機能
- CFDT サーバーが CFDT プールのサーバーとしての役割を果たす機能
- CFDT サーバー領域がその CFDT プールのカップリング・ファシリティー・リスト構造に接続する機能

詳しくは、 354 ページの『カップリング・ファシリティー・データ・テーブルのセキュリティー』を参照してください。

名前付きカウンター・サーバー

FACILITY クラスは以下を制御します。

- CICS 領域が名前付きカウンター・サーバーに接続する機能
- 名前付きカウンター・サーバーが名前付きカウンター・プールのサーバーとしての役割を果たす機能
- 名前付きカウンター・サーバー領域がその名前付きカウンター・プールのカップリング・ファシリティー・リスト構造に接続する機能

詳しくは、 55 ページの『名前付きカウンター・プールおよびサーバーへのアクセスの許可』を参照してください。

共用一時記憶域

FACILITY クラスは以下を制御します。

- CICS 領域が共用一時記憶域 (TS) サーバーに接続する機能
- 共用 TS サーバーが共用 TS プールのサーバーとしての役割を果たす機能
- 共用 TS サーバー領域がその共用 TS プールのカップリング・ファシリティー・リスト構造に接続する機能

詳しくは、 53 ページの『一時記憶域プールへのアクセスを許可する』を参照してください。

ログ・ストリーム

FACILITY クラスは、CICS 領域が MVS ログ・ストリームに使用されるカップリング・ファシリティー・リスト構造に接続する機能を制御します。詳しくは、 48 ページの『MVS ログ・ストリームへのアクセスを許可する』を参照してください。

Db2® 定義内の **AUTHTYPE** ユーザー **ID** および **COMAUTHTYPE** ユーザー **ID**

FACILITY クラスは以下を制御します。

- ユーザーが AUTHID 属性または COMMAUTHID 属性を指定する DB2CONN 定義および DB2ENTRY 定義をインストールする機能

- ユーザーが AUTHID 属性または COMMAUTHID 属性を指定する CREATE DB2CONN コマンドまたは CREATE DB2ENTRY コマンドを使用する機能
- ユーザーがインストール済みの DB2CONN または DB2ENTRY の AUTHID 属性または COMMAUTHID 属性を変更する機能

詳しくは、CICS 内の DB2 関連リソースへのアクセス制御を参照してください。

CICSPlex® SM リソース
FACILITY クラスは多くの CICSPlex SM リソースへのアクセスを制御します。詳しくは、 211 ページの『CICSPlex SM セキュリティーの実装』を参照してください。

API および **SPI** 制限の **DFHAPIR parmlib** メンバー
FACILITY クラスは、制限付き CICS API および SPI コマンドを識別する規則を格納する DFHAPIR parmlib メンバーにユーザーがアクセスする機能を制御します。詳しくは、 10 ページの『API および SPI 制限の DFHAPIR parmlib メンバー』を参照してください。

HPO システム初期設定パラメーターのオーバーライド
FACILITY クラスは、EXEC PGM=DFHSIP ステートメントまたは SYSIN データ・セットの **PARM** パラメーターで **HPO** を指定する機能を制御します。

DB2ENTRY リソースのリソース・クラス

CICS は、IBM 提供の RACF リソース・クラスがない、CICS 定義の DB2ENTRY リソースに対するリソース・セキュリティー検査をサポートします。

DB2ENTRY リソースについて、ユーザー定義クラス名でセキュリティー・プロファイルを定義し、 **XDB2** システム初期設定パラメーターを使用して、CICS に対してクラス名を指定します。 **XDB2** システム初期設定パラメーターの構文は XDB2=NO|*name* ですが、これは他のセキュリティー・システム初期設定パラメーターのようにデフォルトのクラス名をサポートすることはしません。CICS で使用する Db2 リソース・クラス名の定義方法の例としては、DFH$RACF サンプル・ジョブを使用してください。

デフォルトの CICS リソース・クラスには DB2ENTRY プロファイルを定義しないでください。CICS は RACLIST を使用して、指定した X*name* システム初期設定セキュリティー・パラメーターに従って、デフォルトのリソース・クラス内にあるプロファイルをアクティブ化します。**XDB2** は、特に DB2ENTRY リソース用に定義されたユーザー定義クラス名を指定することになります。

RACF コマンドの要約

保護された CICS リソースを扱う RACF アクティビティーの多くは、一般リソース・プロファイルの作成、変更、および削除を伴います。

注:

1. ここで説明するコマンド、および例で使用されるオペランドは完全なものではありません。

2. ここで示すコマンド・シーケンスは、ある特定のタスクを実行する一つの方法を示します。使用できるコマンド・シーケンスは他にもある可能性があります。

RACF コマンドの全詳細については、z/OS Security Server RACF コマンド言語解説書を参照してください。

一般リソース・プロファイルの作成
: 一般リソース・プロファイルを作成するには、RDEFINE コマンドを使用します。通常、プロファイルを作成したときは、続いて PERMIT コマンドを用いてそのプロファイルに対するアクセス・リストを作成します。

  この例では、3 つの RDEFINE コマンドは、CEMT、CEDA、および CEDB という 3 つのプロファイルを TCICSTRN リソース・クラスに定義します。3 つの PERMIT コマンドは、2 つのユーザー・グループが各トランザクションにアクセスできるようにします。

```
RDEFINE TCICSTRN CEMT UACC(NONE)
       NOTIFY(sys_admin_userid)
RDEFINE TCICSTRN CEDA UACC(NONE)
       NOTIFY(sys_admin_userid)
RDEFINE TCICSTRN CEDB UACC(NONE)
       NOTIFY(sys_admin_userid)
PERMIT  CEMT CLASS(TCICSTRN) ID(group1, group2) ACCESS(READ)
PERMIT  CEDA CLASS(TCICSTRN) ID(group1, group2) ACCESS(READ)
PERMIT  CEDB CLASS(TCICSTRN) ID(group1, group2) ACCESS(READ)
```

リソース・グループ・プロファイルの作成
: リソース・グループ・クラスでプロファイルを定義するには、RDEFINE コマンドを使用し、ADDMEM オペランドを指定して、リソースをグループのメンバーとして追加します。通常、プロファイルを作成したときは、続いて PERMIT コマンドを用いてそのプロファイルに対するアクセス・リストを作成します。

  この例では、RDEFINE コマンドは、CICSTRANS というリソース・グループ・プロファイルを GCICSTRN リソース・グループ・クラスに定義します。PERMIT コマンドは、2 つのユーザー・グループがプロファイル内のすべてのトランザクションにアクセスできるようにします。

```
RDEFINE GCICSTRN CICSTRANS UACC(NONE)
       ADDMEM(CEMT, CEDA, CEDB)
       NOTIFY(sys_admin_userid)
PERMIT  CICSTRANS CLASS(GCICSTRN) ID(group1, group2) ACCESS(READ)
```

一般リソース・プロファイルの作成
: RDEFINE コマンドを使用して、一般リソース・クラスにプロファイルを作成します。

```
RDEFINE class profile UACC(NONE)
```

  ここで、
  class は一般リソース・クラスの名前です。
  profile は新規プロファイルの名前です。

  プロファイルへのデフォルト・アクセスが存在しないようにするには、UACC(NONE) を指定します。

一般リソースへのアクセスの許可
: 一般リソースへのアクセスを許可するには、PERMIT コマンドを使用して、一般リソース・プロファイルのアクセス・リストを作成します。

```
PERMIT profile CLASS(class)
       ID(user) ACCESS(authority)
```

ここで、

profile は新規プロファイルの名前です。

class は一般リソース・クラスの名前です。

user は、リソースへのアクセス権限が付与されるユーザー (またはユーザー・グループ)です。

authority は、ユーザーに付与される権限のレベルです。

アクセス・リストからの項目の除去

アクセス・リストからユーザーまたはグループの項目を除去するには、PERMIT コマンドで、ACCESS オペランドでなく DELETE オペランドを使用します。

```
PERMIT profile_name CLASS(class_name)
       ID(user|group) DELETE
```

プロファイルの変更

プロファイルを変更するには (たとえば、UACC を NONE から READ に変更するには)、RALTER コマンドを使用します。

```
RALTER class_name profile_name UACC(READ)
```

プロファイルの削除

プロファイルを削除するには、RDELETE コマンドを使用します。例えば、以下のような項目が含まれています。

```
RDELETE class_name profile_name
```

プロファイルからのコピー

プロファイル間でアクセス・リストをコピーできます。これを行うには、PERMIT コマンドで FROM オペランドを指定します。

```
PERMIT profile_name CLASS(class_name)
       FROM(existing_profile_name) FCLASS(class_name)
```

プロファイル間で情報もコピーできます。これを行うには、RDEFINE または RALTER コマンドで FROM オペランドを指定します。

```
RDEFINE class_name profile_name
        FROM(existing-profile_name) FCLASS(class_name)
```

注: リソース・グループ・プロファイルを使用する場合は、この操作を計画しないでください。プロファイルをコピーするときに、RACF は (ADDMEM オペランドで指定された) メンバーをコピーしません。また、新規プロファイルが既存のプロファイルの正確なコピーでないかもしれない点は、他にもあります。例えば、RACF はリソース・プロファイル所有者のユーザー ID をアクセス・リストに入れ、その際に ALTER アクセス権限を指定します。詳細については、z/OS Security Server RACF コマンド言語解説書で、該当するコマンドの FROM オペランドの説明を参照してください。

クラス内のプロファイルのリスト

特定のクラス内のプロファイルの名前をリストするには、SEARCH コマンドを使用します。以下のコマンドは、TCICSTRN クラスのプロファイルをリストします。

```
SEARCH  CLASS(TCICSTRN)
```

以下のコマンドは、GCICSTRN クラスのすべてのプロファイルおよびそれらの詳細情報をリストします。

```
SEARCH  CLASS(GCICSTRN)
RLIST GCICSTRN * ALL
```

リソース・クラスについては、 25 ページの『RACF 一般リソース・プロファイル』を参照してください。

注: グループ SPECIAL である (システム SPECIAL でない) ユーザーは、SEARCH コマンドを実行しても、クラスに存在するすべてのプロファイルはリストされない可能性があります。クラス内のプロファイルの完全なリストを取得するには、少なくとも各プロファイルをリストするための権限が必要です。詳しくは、z/OS Security Server RACF コマンド言語解説書の SEARCH コマンドに関する RACF 要件の説明、およびリソースの保護に使用されるプロファイルを参照してください。

クラスの保護のアクティブ化

RACF クラス内のプロファイルによって保護されるすべてのリソースの保護を開始するには、CLASSACT を指定して SETROPTS コマンドを発行することにより、そのクラスをアクティブ化します。

```
SETROPTS CLASSACT(class_name)
```

総称プロファイルの定義

RDEFINE を使用して総称プロファイル (つまり、アスタリスク (*)、二重アスタリスク (**)、アンパーサンド (&)、またはパーセント (%) 文字を使用したプロファイル) を定義する前に、まず以下のコマンドを発行します。

```
SETROPTS GENERIC(class_name)
```

クラスの保護の非アクティブ化

クラスを非アクティブ化すると、プロファイル自体を妨害することなく保護がオフになります。クラスが非アクティブ化されると、RACF はそのクラスのすべてのリソースについて、「非保護」の戻りコードを CICS に発行します。CICS はこの応答を「アクセス拒否」として扱います。RACF クラスを非アクティブ化するには、NOCLASSACT を指定して SETROPTS コマンドを発行します。

```
SETROPTS NOCLASSACT(class_name)
```

アクティブ・クラスの判別

現在アクティブである RACF クラスを判別するには、LIST を指定して SETROPTS コマンドを発行します。

```
SETROPTS LIST
```

大/小文字混合パスワードのサポートのアクティブ化

大/小文字混合パスワードのサポートをオンにするには、次のように PASSWORD を指定して SETROPTS コマンドを発行します。

```
SETROPTS PASSWORD(MIXEDCASE)
```

大/小文字混合パスワードのサポートをオフにするには、次の SETROPTS コマンドを発行します。

```
SETROPTS PASSWORD(NOMIXEDCASE)
```

大/小文字混合パスワードは、z/OS Security Server (RACF) 1.7 以上でサポートされます。

データおよびユーザーのセキュリティー分類

RACF は、システム上の一部またはすべてのリソースを分類する手段を提供します。セキュリティー・レベルまたはセキュリティー・カテゴリー (あるいはその両方) を使用して、CICS 関連リソースを保護することができます。

各リソース・プロファイル内でアクセス・リストを指定することなくリソースへのアクセスを制御したい場合は、リソースの分類を検討してください。リソースを分類する場合は、ユーザー・プロファイルが適切に分類されているユーザーのみがそのリソースにアクセスできるようになります。セキュリティー・レベルおよびセキュリティー・カテゴリーの使用については、z/OS Security Server RACF セキュリティー管理者のガイドを参照してください。 CICS は RACROUTE REQUEST=FASTAUTH 機能を使用するため、セキュリティー・ラベルやグローバル・アクセス検査など、一部のサービスは CICS では使用できません。FASTAUTH と共に使用できるものについては、z/OS Security Server RACF セキュリティー管理者のガイドを参照してください。

同じアクセス要件またはロギング要件を持つユーザーをグループに分類することもできます。ユーザーは 1 つ以上のグループに属することができ、そのうちの 1 つがデフォルトとなります。サインオン・プロセスでは、ユーザーはデフォルトの RACF ユーザー・グループ名をオーバーライドできます。「グループのリストの検査」が非アクティブである場合は、サインオン時に使用するグループ名によって異なる権限がユーザーに与えられる可能性があります。

独自のリソース・クラスの定義

各 CICS 領域のリソース・クラス名が固有となるように、独自のリソース・クラスを定義することができます。

独自のリソース・クラス名を定義すると、以下の利点が得られます。

他の領域からのアクセスの制御

ある CICS 領域で実行されているユーザーが、異なるクラス名が指定されている別の CICS 領域のリソースにアクセスするのを防ぐことができます。(接頭部を付けることでこの制御を行うこともできます。 63 ページの『セキュリティー関連システム初期設定パラメーター』の SECPRFX パラメーターの説明を参照してください。)

各領域のグループ管理者

インストール定義クラスを持つ CICS 領域ごとに、異なるグループ管理者に、その領域で使用されるプロファイルの作成権限を付与できます。

この利点を得るためには、インストール定義クラスを 5 (IBM 提供 CICS クラスの POSIT 番号) 以外の POSIT 番号で定義します。次に、それらのクラスの少なくとも 1 つについて、グループ管理者に CLAUTH (クラス権限) を付与します。

総称プロファイルを定義する前に、SETROPTS GENERIC コマンドを使用します (32 ページの『RACF コマンドの要約』を参照)。

接頭部の付加がアクティブな場合は、競合を気にせずに、異なる管理者を割り当てることもできます。これを行うには、接頭部を高位修飾子として使用して、各クラスに総称プロファイルを作成します。例えば、以下のような項目が含まれています。

```
RDEFINE TCICSTRN cics_region_id.** UACC(NONE)
        OWNER(cics_region_administrator_userid)
```

該当する各プロファイルの OWNER として指定された管理者は、より具体的なプロファイルを作成し、保守することができます。他の管理者がこれを行うことはできません。

インストール先定義クラスのセットアップ

インストール先定義クラスをセットアップするには、RACF システム・プログラマーと連携して、RACF クラス記述子テーブル (CDT) のインストール先定義部分 (モジュール ICHRRCDE) に新規のクラス記述子を追加します。

インストール先定義クラスを CDT に追加する方法の例については、セキュリティー処理のカスタマイズ を参照してください。

CDT で定義されたインストール先定義クラスはすべて、MVS ルーター・テーブルにも定義されている必要があります。これは、MVS ルーターが、ルーター要求で使用されるすべてのクラスをチェックして、そのクラスが存在するかどうかを判別するからです。存在しない場合は、RACF に要求が送信されることはありません。クラスを MVS ルーターに定義するには、z/OS Security Server RACROUTE マクロ解説書に記載されているとおりに、それらを ICHRFR01 (MVS ルーター・テーブルのユーザーが変更可能な部分) に追加します。 182 ページの『ユーザー定義リソースを RACF に指定する』も参照してください。

インストール先定義クラスをセットアップする際は、IBM 提供のデフォルトを CDT からコピーするようお勧めします。z/OS Security Server RACF マクロおよびインターフェースには、その例が記載されています。次に、名前、グループ名またはメンバー名、POSIT 番号および ID を変更する必要があります。これらのオペランドの有効な値の詳細については、z/OS Security Server RACF マクロおよびインターフェースの ICHERCDE マクロの説明を参照してください。インストール先定義のリソース・クラスの作成に関しては、同じ資料を参照してください。リソース・クラスの追加方法の例については、IBM 提供のサンプル、DFH$RACF を参照してください。これは、CICSTS55.CICS.SDFHSAMP に記載されています。

CICS リソースの場合は、リソース・クラス名の先頭文字が CICS によって定義済みであり、これはデフォルトのリソース・クラス名と整合しています。リソース・クラス名の 2 番目から 8 番目の文字を定義できますが、管理を容易にするために、メンバーおよびグループ・クラスの両方に同じ文字を指定することをお勧めします。メンバー・クラスに指定された 7 文字は、各種の *X*name パラメーターで CICS に定義するリソース・クラス名の一部ですが、次の場合を除きます。

- XDB2。これには、CICS 定義の接頭部文字がないので、1 から 8 文字の任意の定義クラス名を指定できます。これらのリソース専用の特定のクラス (複数も可) を使用することをお勧めします。
- XAPPC および XUSER。これらには、『name』 オプションがなく、セキュリティーがアクティブであるかどうかを指定するために YES または NO のいずれかを取ります。

定義する任意のクラス名の 2 から 5 文字目に 『CICS』 の文字を使用することは避けてください。 RACF では、クラス名のうち少なくとも 1 文字が国別文字または数字である必要があります。

RACF プロファイル内のフィールドへのアクセスの制御

RACF データベース内のフィールドへのアクセスを制御するプロファイルを定義するには、FIELD リソース・クラスを使用します。

RACF FIELD クラスでプロファイルを作成することにより、以下の形式で、ユーザー・プロファイル内の CICS セグメントまたは LANGUAGE セグメントと、パートナー LU プロファイルの適切なフィールドのリスト作成または更新を許可できます。

```
USER.CICS.OPIDENT
USER.CICS.OPCLASSN
USER.CICS.OPPRTY
USER.CICS.TIMEOUT
USER.CICS.XRFSOFF
USER.LANGUAGE.USERNL1
USER.LANGUAGE.USERNL2
APPCLU.SESSION.SESSKEY
APPCLU.SESSION.KEYINTVL
APPCLU.SESSION.SLSFLAGS
```

あるいは、総称プロファイル USER.CICS.** をセットアップして、CICS セグメント内のすべてのフィールドへのアクセスを制御することができます。総称プロファイルを定義する前に、 32 ページの『RACF コマンドの要約』で説明されているように、SETROPTS GENERIC コマンドを使用します。

これらのプロファイルをリストするには READ 権限、変更するには UPDATE 権限が必要です。詳しい説明については、z/OS Security Server RACF セキュリティー管理者のガイドのフィールド・レベル・アクセス検査に関するセクションを参照してください。

第 2 章 単一 CICS 領域での RACF 保護の実装

ここでは、単一の CICS 領域内で CICS リソースを保護する方法を示します。

CICS システム・リソースのセキュリティー

このトピックでは、CICS に必要なシステム・リソースを保護する方法について説明します。

RACF の CICS インストール要件

CICS 領域 (複数可) が使用するリソースへのアクセスを、RACF 機能を使用して制御できます。配布ボリュームで提供される CICS ライブラリーには、外部セキュリティー管理をサポートするために必要な CICS モジュールが含まれています。

CICS 提供の RACF 動的解析検証ルーチン

CICS 端末オペレーターのデータを定義するには、CICS 提供の RACF 動的解析検証ルーチンを使用します。これらのルーチンを、MVS リンク・リスト内で APF 許可ライブラリーにする必要がある SYS1.CICSTS55.CICS.SDFHLINK にインストールします。

詳しくは、CICS 必須モジュールを MVS リンク・リストにインストールするを参照してください。

該当するルーチンは、以下のとおりです。

DFHSNNFY
CICS セグメントの更新通知

DFHSNPTO
CICS セグメントの TIMEOUT 印刷形式設定

DFHSNVCL
CICS セグメントの OPCLASS キーワード検証

DFHSNVID
CICS セグメントの OPIDENT キーワード検証

DFHSNVPR
CICS セグメントの OPPRTY キーワード検証

DFHSNVTO
CICS セグメントの TIMEOUT キーワード検証

マルチ MVS 環境での RACF サポートの使用

共用 DASD があるマルチ MVS 環境を稼動している場合、同じ端末ユーザー特性へのアクセス権限を持つアクティブな代替 CICS システムが必要になることがあります。

これは、アクティブな代替 CICS システムに同じ RACF データベースを共用させることで実現できます。

© Copyright IBM Corp. 1989, 2018

MVS プログラム・プロパティー・テーブルでのオプションの設定

MVS プログラム・プロパティー・テーブル (PPT) 内に DFHSIP プログラムの項目がある場合は、SYS1.PARMLIB ライブラリーの SCHEDxx メンバーの PPT ステートメント内で、NOPASS オプションを DFHSIP に対して設定しないようにしてください。 NOPASS オプションを設定すると、CICS 領域によってアクセスされるデータ・セットに対する、パスワードおよび RACF 許可検査がバイパスされることになります。

CICS MVS PPT オプションの指定の詳細については、CICS 必須モジュールを MVS リンク・リストにインストールするを参照してください。

CICS ロード・ライブラリーの保護

通常、CICS は無許可の状態で稼働しますが、CICS 初期設定プログラム (DFHSIP) は、その一部を許可状態で実行する必要があります。このため、配布テープで提供されるバージョンの DFHSIP モジュールは、(リンケージ・エディター SETCODE AC(1) の制御ステートメントを使用して)「許可」属性を使用してリンク・エディットされ、CICSTS55.CICS.SDFHAUTH にインストールされています。このライブラリーは、APF 許可としてオペレーティング・システムに定義する必要があります。

CICSTS55.CICS.SDFHAUTH が、無許可あるいは偶発的に変更されないようにするには、このライブラリーを RACF 保護しなければなりません。このような保護がなければ、MVS システムの保全性およびセキュリティーが脅かされることになります。DFHSIP を使用して、CICS システムの無許可の始動を制御するために、下記を実施することも検討してください。

- DFHSIP が MVS リンク・リストに置かれたライブラリーにある場合は、PROGRAM リソース・クラスのプロファイルを使用して DFHSIP を保護します。このプロファイルへの READ アクセス権限は、CICS の実行を許可されたユーザーにのみ付与するようにします。
- DFHSIP がリンク・パック域 (LPA) に置かれている場合は、PROGRAM リソース・クラスでは保護できません。その代わりに、接尾部付きのあらゆる DFHSIT ロード・モジュールのロードを制御することによって、CICS の始動を制御します。DFHSIT ロード・モジュールが LPA に含まれていないことを確認してから、総称「DFHSIT*」プロファイルを PROGRAM リソース・クラスに作成することによって、DFHSIT のロードを制御します。このプロファイルへの READ アクセス権限は、CICS の実行を許可されたユーザーにのみ付与するようにします。

また、SYS1.CICSTS55.CICS.SDFHLINK および SYS1.CICSTS55.CICS.SDFHLPA も RACF 保護します。STEPLIB および DFHRPL ライブラリー連結を構成するその他のライブラリー (CICSTS55.CICS.SDFHLOAD を含む) も RACF 保護しなければなりません。

CICS データ・セットの保護と、適切なデータ・セット・セキュリティー・プロファイルの作成について詳しくは、49 ページの『CICS データ・セットへのアクセスの許可』を参照してください。

注: アプリケーション・プログラムのソース・ステートメントは機密事項です。この情報を含むデータ・セットは RACF 保護することを検討してください。

CICS 領域のユーザー ID の指定

RACF がインストールされている MVS 環境で CICS 領域を (ジョブとしてあるいは開始済みタスクとして) 開始すると、そのジョブまたはタスクは、**CICS 領域ユーザー ID** と呼ばれるユーザー ID に関連付けられます。

このユーザー ID に関連付けられた権限により、CICS 領域がアクセスできる RACF 保護付きリソースが決定されます。

各 CICS 領域は、実動用であれテスト用であれ、CICS 領域を実行する領域ユーザー ID に基づいて、通常の RACF データ・セット保護を受ける必要があります。CICS を実行する領域ユーザー ID は、次の 3 つの方法のいずれかで指定します。

開始済みタスクとして:

- RACF 開始済みプロシージャー・テーブル ICHRIN03 で指定します。これは、MVS START コマンドを使用して CICS を開始済みタスクとして開始する場合です。(CICS プロシージャーの RACF 下での実行の許可を参照してください。) ただし、「信頼」属性または「特権」属性を開始済みプロシージャー・テーブルで CICS エントリーに割り当てないでください。詳しくは、z/OS Security Server RACF システム・プログラマーのガイドで、MVS 開始済みプロシージャーをユーザー ID に関連付ける方法を参照してください。

開始済みジョブとして:

- 開始済みの一般リソース・クラス・プロファイルで、STDATA セグメントのユーザー・パラメーター上で指定します。

ジョブとして:

- CICS をジョブとして開始するときに、JOB ステートメントの USER パラメーターで指定します。

さまざまな **CICS 領域に対する許可が適切に差別化されるように、固有の領域ユーザー ID** を使用してそれぞれの領域を実行します。例えば、給与計算および人事アプリケーションを処理するために実動 CICS 領域を実行するのに使用するユーザー ID は、実動の給与計算データ・セットおよび人事データ・セットへのアクセスだけを許可された CICS ユーザー ID でなければなりません。

相互通信を使用している場合は、リンク・セキュリティー検査をバイパスしたいのでない限り、固有のユーザー ID を使用することがとりわけ重要です。詳しくは、使用している環境に応じて、Link security with LU6.2、LU6.1 でのリンク・セキュリティー または Link security with MRO を参照してください。

保護ユーザー ID の使用

CICS を開始タスクとして実行する場合は、CICS 領域ユーザー ID と CICS のデフォルト・ユーザー ID を保護ユーザー ID として定義することを検討します。

保護ユーザー ID を、パスワードを要求するシステムに入るために使用することは決してできません。これには TSO へのログオン、CICS へのサインオン、または JOB カード上でパスワードを指定するバッチ・ジョブの実行などが含まれます。ユーザーが、不注意または故意に誤ったパスワードを入力することで、保護ユーザー ID を取り消すことはできません。

保護ユーザー ID を作成するには、ユーザー・プロファイルに NOPASSWORD オプション属性および NOIDCARD オプション属性を指定します。保護ユーザー ID について詳しくは、z/OS Security Server RACF セキュリティー管理者のガイドを参照してください。

RACF での CICS プロシージャーの実行を許可する

CICS 始動プロシージャーを呼び出して、CICS を開始タスクまたは開始ジョブとして開始することができます。RACF は、開始タスクに対して ICHRIN03 プロシージャー・テーブルを提供し、開始ジョブに対して STARTED 一般リソース・クラスを提供します。この 2 つのオプションについて、以下で説明します。

開始タスクでの ICHRIN03 テーブルの使用

CICS を開始タスクとして実行する場合は、RACF テーブル ICHRIN03 を使用して、適切な権限を持つ RACF ユーザーにカタログ式プロシージャー名を関連付ける必要があります。

RACF ではデフォルトの ICHRIN03 テーブルが提供されますが、これは変更できます。このテーブルの詳細、および CICS を始動するためのカタログ式プロシージャー名のデフォルト項目を追加する方法については、z/OS Security Server RACF システム・プログラマーのガイドを参照してください。

ICHRIN03 テーブルにデフォルト項目が含まれている場合、テーブルを更新する必要はありません。ただし、カタログ式プロシージャーと同じ名前で RACF ユーザーを定義してください。

ICHRIN03 テーブルにデフォルト項目が含まれていない場合 (またはデフォルト項目を設定しないことを選択した場合) は、カタログ式プロシージャー名とその関連 RACF ユーザーを含む項目を使用してテーブルを更新します。この RACF ユーザーは、カタログ式プロシージャーと同じ名前である必要はありません。

ICHRIN03 テーブルにデフォルト項目が含まれているか、定義した特定の項目が含まれているかにかかわらず、ICHRIN03 を使用して識別される RACF ユーザーがカタログ式プロシージャー内のデータ・セットへの必要なアクセス権限を持っていることを確認してください。

例えば、DFHCICS というカタログ式プロシージャーを RACF ユーザー ID CICSR に関連付ける場合、ユーザー ID CICSR は、DFHCICS によって開始されるタスクがアクセスする CICS リソースへのアクセス権限を持っている必要があります。

開始ジョブの STARTED プロファイルの使用

複数の CICS 領域を開始ジョブとして開始する場合は、すべてを同じプロシージャーから開始するとしても、それぞれの開始領域に別個のユーザー ID を使用できます。また、同じユーザー ID を共用することになる CICS 領域のグループ (例えば、端末専有領域など、同じタイプのすべての領域) には、総称プロファイルを使用できます。

開始ジョブのサポートは、RACF STARTED 一般リソース・クラス、およびその関連 STDATA セグメントによって提供されます。固有のユーザー ID で実行する必要がある各ジョブまたはジョブのグループに対して、このクラスでプロファイルを定義します。

STDATA セグメントに指定されているユーザー ID が RACF に定義されていることを確認します。さらに、ユーザー ID に、それらの下で実行される CICS 領域のデータ・セット・プロファイルに対する適切な権限があることを確認します。

複数の **AOR** に対する総称プロファイルの例**:**

以下の例は、CICSTASK というプロシージャーを使用して開始されるジョブに対して総称プロファイルを定義する方法を示しています。

この例では、ジョブ名の先頭の文字は、CICS アプリケーション所有領域 (AOR) のグループを表す CICSDA となっています。

```
RDEFINE  STARTED  (CICSTASK.CICSDA*)  STDATA( USER(CICSDA##) )
```

例えば CICSDAA1 というジョブ名を持つ CICSTASK を開始するために START コマンドを発行すると、MVS は、この CICS アプリケーション所有領域が実行されるユーザー ID を取得するために、プロシージャー名 (CICSTASK) およびジョブ名 (CICSDAA1) を渡します。この例では、CICS 領域ユーザー ID は、総称プロファイル CICSTASK.CICSDA* の下で開始されるすべての領域に対して、CICSDA## と定義されています。

各領域の固有プロファイルの例**:** 以下の例は、CICSTASK というプロシージャーを使用して開始されるジョブ (それぞれの開始ジョブは固有の CICS 領域ユーザー ID で実行される) に対して固有プロファイルを定義する方法を示しています。

```
RDEFINE  STARTED  (CICSTASK.CICSDAA2)  STDATA( USER(CICSDAA2) )
```

例えば CICSDAA2 というジョブ名を持つ CICSTASK を開始するために START コマンドを発行すると、MVS は、この CICS アプリケーション所有領域が実行されるユーザー ID を取得するために、プロシージャー名 (CICSTASK) およびジョブ名 (CICSDAA2) を渡します。この例では、CICS 領域ユーザー ID は、APPLID と同じである、CICSDAA2 と定義されます。

CICS 領域ユーザー ID のユーザー・プロファイルの定義

CICS 領域を起動する前に、必要なユーザー ID (CICS 領域ユーザー ID および CICS デフォルト・ユーザー ID) が定義されていることを確認してください。

適切な許可があれば、ADDUSER コマンドを使用して CICS 領域用の RACF ユーザー・プロファイルを定義できます。例えば、CICS 領域のユーザー ID として CICSR を定義するには、TSO から次の RACF コマンドを入力します。

```
ADDUSER CICSR  NAME(user-name) DFLTGRP(cics_region_group)
```

この例では DFLTGRP が指定されているため、初期パスワードは DFLTGRP 名です。DFLTGRP を指定しない場合、パスワードはデフォルトでは、ADDUSER コマンドを発行するユーザーが属するグループの名前に設定されます。あるいは、ADDUSER コマンドの PASSWORD パラメーターでパスワードを明示的に指定す

ることもできます。新規ユーザー ID のパスワードの変更について詳しくは、『CICS JOB ステートメントでの USER パラメーターのコーディング』を参照してください

OPERATIONS 属性を **CICS** 領域ユーザー **ID** に割り当てないでください。 これを行うと、CICS 領域は、特定の許可が実行されていない RACF 保護データ・セットにアクセスできるようになります。適切な CONNECT または PERMIT コマンドが発行されていれば、CICS 領域ユーザー ID に OPERATIONS 属性は不要です。これらのコマンドは、各 CICS 領域の CICS 領域ユーザー ID に対し、その領域で必要となる特定のデータ・セットへのアクセス権限のみを付与します。

CICS JOB ステートメントでの USER パラメーターのコーディング

ジョブから CICS を開始する場合は、JOB ステートメントでパラメーター USER= および PASSWORD= を組み込みます。

以下に例を挙げます。

```
//CICSA   JOB ... ,USER=CICSR,PASSWORD=password
```

新規ユーザーを RACF に定義すると、パスワードは自動的に有効期限切れのフラグが立てられます。このため、新規ユーザー ID で CICS を初めて開始するときは、JOB ステートメントで PASSWORD パラメーターを変更します。以下に例を挙げます。

```
PASSWORD=(oldpassword,newpassword)
```

JOB ステートメントでパスワードを指定したくない場合は、代理ユーザーに CICS ジョブを実行依頼してもらうことができます。代理ユーザーとは、別のユーザー (オリジナル・ユーザー) のパスワードを指定せずに、そのオリジナル・ユーザーに代わってジョブを実行依頼することを許可された、RACF 定義のユーザーです。代理ユーザーが実行依頼するジョブは、オリジナル・ユーザーの ID で実行されます。詳しくは、 61 ページの『CICS 環境での代理ジョブ実行依頼』を参照してください。領域ユーザー ID は、デフォルト・ユーザーを使用するための代理権限も持っていなければなりません。詳しくは、 154 ページの『代理ユーザー・セキュリティー』を参照してください。

CICS 領域ユーザー ID に必要な権限

CICS 制御プログラムは、CICS 領域ユーザー ID で実行されます。

したがって、このユーザー ID は、CICS 自体が使用する必要があるすべてのリソースへのアクセスを必要とします。これらのリソースには、以下の 2 つのタイプがあります。

1. CICS の外部リソース。データ・セット、スプール・システム、SNA ネットワークなど。
2. CICS の内部リソース。システム・トランザクションや補助ユーザー ID など。

外部リソースの許可**:**

バッチ・ジョブと同様、各 CICS 領域は多くの外部リソースにアクセスできる必要があります。CICS がそれらのリソースにアクセスするための権限は、CICS 領域ユーザー ID から取得されます。それらのリソースにアクセスするアクションを実行

するように CICS にどのサインオン・ユーザーが要求したかは、問題ではありません。外部サービスは、領域ユーザー ID の権限の下で、CICS によって要求されている場合にのみ認識されます。

以下のリソースにアクセス権限を与えます。

- MVS システム・ロガー

  CICS には、MVS ロガーに定義されているログ・ストリームを使用する権限が必要です。 48 ページの『MVS ログ・ストリームへのアクセスを許可する』を参照してください。
- CICS によって使用される外部データ・セット

  CICS には、使用するすべてのデータ・セットを開く権限が必要です。 49 ページの『CICS データ・セットへのアクセスの許可』を参照してください。
- アプリケーション・プログラムによって使用される外部データ・セット。

  CICS には、ユーザー独自のアプリケーション・プログラムが必要とするすべてのデータ・セットを開く権限が必要です。 52 ページの『ユーザー・データ・セットへのアクセスを許可する』を参照してください。
- 一時記憶域サーバー

  いずれかの TS キューが共用として定義されている場合、CICS には、一時記憶域サーバーにアクセスする権限が必要です。 54 ページの『一時記憶域サーバーへのアクセスを許可する』を参照してください。
- SMSVSAM サーバー

  ユーザーが VSAM レコード・レベル共用 (RLS) を使用する場合、CICS には、SMSVSAM サーバーにアクセスする権限が必要です。 57 ページの『SMSVSAM サーバーへのアクセスを許可する』を参照してください。
- z/OS Communications Server (for SNA または IP) アプリケーション

  各プログラムを、z/OS Communications Server アプリケーションにすることができるかどうか注意深く検討します。これを実行する場合、CICS にはその z/OS Communications Server ACB を開く権限が必要です。 60 ページの『CICS 領域の z/OS Communications Server ACB オープンの制御』を参照してください。
- 内部読み取りプログラムに実行依頼されるジョブ

  いずれかのアプリケーションが JES 内部読み取りプログラムに JCL を実行依頼する場合、USERID パラメーターなしで実行依頼されることを許可することができます。 60 ページの『ユーザー ID 伝搬の制御』を参照してください。

  ただし通常は、実行依頼ジョブに関して PASSWORD パラメーターを指定するようにアプリケーションに要求するべきではありません。そのため、CICS に、実行依頼をする可能性があるすべてのユーザー ID の代理ユーザーとなることを許可する必要があります。 61 ページの『CICS 環境での代理ジョブ実行依頼』を参照してください。
- システム・スプール・データ・セット

CICS には、JES スプール・システム内のデータ・セットにアクセスする権限が必要です。 62 ページの『CICS 環境での JES スプールの保護』を参照してください。

内部リソースの許可:

CICS がアプリケーション・プログラムのように動作する (ただし、どのユーザーに対しても直接ではないハウスキーピング機能を実行している) いくつかの内部機能があります。関連するトランザクションは、CICS 領域ユーザー ID の制御下で実行します。それらのトランザクションは CICS 内部リソースにアクセスするので、CICS 領域ユーザー ID には、それらにアクセスするための権限を与える必要があります。

それらは以下のとおりです。

- CICS システム・トランザクション

  CICS には、使用するすべての内部ハウスキーピング・トランザクションを接続する権限が必要です。 Security for CICS-supplied transactionsを参照してください。
- 補助ユーザー ID

  CICS 代理ユーザー検査が **XUSER** システム初期設定パラメーター (デフォルト) とともに指定されている場合、CICS には、特定の追加ユーザー ID を使用する権限が必要です。それらは以下のとおりです。
  - デフォルトのユーザー ID

    155 ページの『CICS デフォルト・ユーザー』を参照してください。
  - 初期設定後の処理に使用されるユーザー ID (PLTPIUSR)

    155 ページの『初期設定後の処理』を参照してください。
  - 一時データ・トリガー・トランザクションに使用されるユーザー ID

    157 ページの『一時データのトリガー・レベル・トランザクション』を参照してください。
- PLTPI プログラムによって使用されるリソース

  PLTPIUSR システム初期設定パラメーターが省略されると、すべての PLTPI プログラムで CICS 領域ユーザー ID が使用されます。この場合は、それらのプログラムが使用するすべての CICS リソースに対するアクセス権限を、CICS 領域ユーザー ID に与えます。 99 ページの『PLT プログラム』を参照してください。

デフォルトの CICS ユーザー ID を RACF に定義する

SEC=YES と指定した CICS 領域ごとに、RACF ユーザーのプロファイルを定義します。そのユーザー ID は **DFLTUSER** システム初期設定パラメーターの値と一致したものとします。

例えば、DFLTUSER=NOTSIGND と指定した場合、NOTSIGND という名前の RACF ユーザー・プロファイルを定義します。**DFLTUSER** パラメーターに値を指定し

ない場合、CICS 提供のデフォルト・ユーザー ID は CICSUSER です。CICSUSER という名前の RACF ユーザー・プロファイルを定義してください。

以下の考慮事項のいずれかに当てはまる場合は、各 CICS 領域に異なるデフォルト CICS ユーザー ID を定義します。

- デフォルトの CICS ユーザー ID が、異なるセキュリティー属性 (RACF グループのメンバーシップなど) を必要としている。
- デフォルトの CICS ユーザー ID が、異なるオペレーター・データ (RACF ユーザー・プロファイルの CICS セグメント) を必要としている。
- デフォルトの CICS ユーザー ID が、異なるデフォルト言語 (RACF ユーザー・プロファイルの LANGUAGE セグメント) を必要としている。

システム初期設定パラメーターのデフォルト名 (CICSUSER) で CICS デフォルト・ユーザーを定義するには、ADDUSER コマンドに CICS オペランドを付けて、次のように使用します。

```
ADDUSER CICSUSER  DFLTGRP(group_id) NAME(user_name)
        OWNER(userid or group)
        PASSWORD(password)
        CICS(OPCLASS(1,2,...,n) OPIDENT(identifier) OPPRTY(priority)
             TIMEOUT(timeout_value)  XRFSOFF(xrf_sign-off_option))
```

セキュリティー管理者は、デフォルト・ユーザー ID と開始済みのタスクのパスワードをデフォルトのまま許可せずに、必ず定義する必要があります。

各 CICS 領域は、デバッグの補助として、それぞれ独自のデフォルト・ユーザーを使用する必要があります。定義が類似したものになるように、RACF デフォルト・ユーザー・グループをセットアップします。

システム初期設定パラメーター XUSER=YES (デフォルト) を指定した場合は、CICS 領域のユーザー ID がデフォルト・ユーザー ID の代理ユーザーとなる許可を与えてください。例えば、次のとおりです。

```
PERMIT CICSUSER.DFHINSTL CLASS(SURROGAT) ID(cics_region_userid)
```

始動中に、CICS はデフォルト・ユーザー ID を「サインオン」します。デフォルト・ユーザーが (例えばユーザー ID が RACF に定義されていないなどの理由で) サインオンに失敗すると、CICS からはメッセージ DFHXS1104 が発行され、CICS の初期設定は終了します。

CICS は、デフォルト・ユーザーとして有効な RACF ユーザー ID を正常にサインオンすると、下記のソースのいずれかからデフォルト・ユーザーに対して端末ユーザー・データを設定します。

- デフォルト・ユーザーの RACF ユーザー・プロファイルの CICS セグメント
- 組み込みの CICS システム・デフォルト値

CICS 端末オペレーター・データを取得するためのサインオン・プロセスの詳細については、90 ページの『ユーザーの CICS 関連データの取得』を参照してください。

CICS は、端末ユーザーがサインオンを開始する前に、デフォルト・ユーザー ID のセキュリティー属性をすべての CICS 端末に割り当てます。デフォルト・ユーザーのセキュリティー属性と端末ユーザー・データは、ユーザーがサインオンしない

端末にも適用されます (CICS 提供の CESN トランザクション、またはユーザー作成の同等物を使用することによって)。ただし、端末定義の USERID オプションに値を指定することによってセキュリティーが明示的に事前設定されている場合を除きます。

注: デフォルト・ユーザーの RACF プロファイルによってゼロ以外の TIMEOUT 値が指定される場合、その値はサインオンしない端末には適用されません。

USERID パラメーターのない、一時データ・キューのために開始された「トリガー・レベルのトランザクション」にも、CICS はデフォルト・ユーザー ID のセキュリティー属性を割り当てます。

少なくとも、他のあらゆる端末ユーザーに付与すべき最小限の権限が、デフォルト・ユーザー ID によって与えられるようにしてください。具体的には、以下のとおりです。

- デフォルト・ユーザー・アクセス権限を領域のアプリケーション ID に付与すること。 58 ページの『CICS 領域へのアクセスの許可』を参照してください。
- 誰でも使用できるよう意図された CICS 提供のトランザクションに、デフォルト・ユーザー・アクセスを付与すること。 72 ページの『CICS 端末ユーザーの識別』 の定義、特に ALLUSER のトランザクションのグループ例に含めることを推奨されたトランザクションを参照してください。

MVS ログ・ストリームへのアクセスを許可する

CICS 領域ユーザー ID に、そのシステム・ログおよび一般ログで使用されるログ・ストリームへの書き込み (および必要な場合は作成) を許可するようにします。

これを行うには、LOGSTRM 一般リソース・クラスのログ・ストリーム・プロファイルへの適切なアクセス許可を付与します。

必要な許可のレベルは、ログ・ストリームが常に MVS システム・ロガーに明示的に定義されているかどうかによって、次のように異なります。

- CICS がログ・ストリームを動的に作成する予定の場合は、CICS に対して、関連ログ・ストリーム・プロファイルへの **ALTER** 権限と、関連カップリング・ファシリティー構造への **UPDATE** 権限を付与します。
- CICS の書き込み先のログ・ストリームがすべて MVS に既に定義されている場合は、CICS に対して、ログ・ストリーム・プロファイルへの **UPDATE** 権限のみを付与します。
- CICS ログ・ストリームを読み取る必要のあるユーザーには、**READ** アクセスを許可します。

例えば、CICS 領域によって参照され、その領域ユーザー ID およびアプリケーション ID によって識別されるすべてのログ・ストリームをカバーするには、次の例のように総称プロファイルを定義します。

```
RDEFINE LOGSTRM region_userid.** UACC(NONE)
```

ただし、同じ領域ユーザー ID を共用する複数の CICS システムがあり、それぞれが別々のセキュリティー要件をもつ場合は、次のように総称プロファイルにアプリケーション ID を含めてください。

```
RDEFINE LOGSTRM region_userid.applid.* UACC(NONE)
```

以下の例では、CICS が実行されている CICS 領域ユーザー ID に、指定されたカップリング・ファシリティー構造内のログ・ストリームにジャーナル・レコードおよびログ・レコードを書き込む許可を与えます。

```
PERMIT IXLSTR.structurename CLASS(FACILITY) ACCESS(UPDATE)
       ID(region_userid)
```

以下の例では、3 つのカテゴリーのユーザーにアクセス権が与えられています。

```
PERMIT region_userid.applid.* CLASS(LOGSTRM) ACCESS(ALTER)
       ID(region_userid)
PERMIT region_userid.applid.* CLASS(LOGSTRM) ACCESS(READ)
       ID(authorized_browsers)
PERMIT region_userid.applid.* CLASS(LOGSTRM) ACCESS(UPDATE)
       ID(archive_userid)
```

これらの例で、region_userid は、開始タスクまたはバッチ・ジョブとしての CICS の実行に使用される CICS 領域ユーザー ID です。ID archive_userid は、データが不要になった場合に CICS ログから古いデータをパージするためのアプリケーション・プログラムの実行に使用されるユーザー ID です。ID authorized_browsers は、ログ・ストリームの読み取りは許可されるがデータのパージは許可されないユーザーのユーザー ID を指しています。

いくつかの CICS 領域が同じ CICS 領域ユーザー ID を共有する場合、アプリケーション *ID* 修飾子に * を指定することによって、プロファイルをより一般的なものにすることができます。

定義するプロファイルの数は、ログの命名規則と、どの程度まで総称プロファイル処理を使用するのかによって異なります。

CICS データ・セットへのアクセスの許可

CICS ジョブ (または開始済みタスク) に領域ユーザー ID を定義したら、そのユーザー ID に対し、必要な権限をもって CICS システム・データ・セットにアクセスする許可を与えます。

CICS システム・データ・セットへのアクセスを許可するときは、READ、UPDATE、および CONTROL という 3 つのアクセス・レベルから適切なものを選択します。 また、UACC(NONE) を持つデータ・セット・プロファイルを定義して、CICS 領域のユーザー ID しかこれらデータ・セットにアクセスできないようにします。 CICS 領域のユーザー ID の詳細については、 41 ページの『CICS 領域のユーザー ID の指定』を参照してください。

CICS ロード・ライブラリーの場合は、READ アクセスのみを許可します。

次の 4 つのデータ・セットには、CONTROL アクセス権限が必要です。
- 一時記憶データ・セット
- 一時データ区画内データ・セット
- CAVM 制御データ・セット (XRF)
- CAVM メッセージ・データ・セット (XRF)

それ以外のすべての CICS データ・セットには、UPDATE アクセスを許可します。

したがって、CICS システム・データ・セットに関しては、少なくとも 3 つの 総称プロファイルが必要です。それは、アクセスを適切なレベルに制限するためです。表 3を参照してください。

表 *3.* 総称データ・セット・プロファイルのまとめ

必要なアクセス・レベル	保護される **CICS** データ・セットの種類
READ	ロード・ライブラリー
UPDATE	補助トレース、トランザクション・ダンプ、システム定義、グローバル・カタログ、ローカル・カタログ、および再始動
CONTROL	一時記憶、区画内一時データ、XRF メッセージ、および XRF 制御

データ・セット・プロファイルの総称名を使用すると、CICS 領域に必要なプロファイルの数を大幅に削減することができます。 このポリシーを、図 1 に示す、多数のサンプル CICS 領域を使用した例で説明します。

例に示す RACF コマンドを TSO セッションから発行することもできますし、図 1 に示すようにバッチ・ジョブで TSO 端末モニター・プログラム IKJEFT01 を使用してコマンドを実行することもできます。 また別の方法として、RACF 提供の ISPF パネルを使用することもできます。 いずれの方法を使用しても、必要なプロファイルを作成し、各 CICS 領域ユーザー ID に対して、対応する CICS 領域に適切なデータ・セットにアクセスする権限を与えることができます。

```
//RACFDEF  JOB 'accounting information',
//             CLASS=A,MSGCLASS=A,MSGLEVEL=(1,1)
//DEFINE   EXEC PGM=IKJEFT01
//SYSPRINT DD SYSOUT=A
//SYSTSPRT DD SYSOUT=A
//SYSUDUMP DD SYSOUT=A
//SYSTSIN  DD *
ADDSD  'CICSTS55.CICS.SDFHLOAD'  NOTIFY(cics_sys_admin_id) UACC(NONE)
PERMIT 'CICSTS55.CICS.SDFHLOAD' ID(cics_id1,...,cics_group1,..,cics_groupn)
                          ACCESS(READ)
ADDSD  'CICSTS55.CICS.SDFHAUTH'  NOTIFY(cics_sys_admin_id) UACC(NONE)
PERMIT 'CICSTS55.CICS.SDFHAUTH' ID(cics_id1,...,cics_group1,..,cics_groupn)
                          ACCESS(READ)
ADDSD  'CICSTS55.CICS.applid.**'  NOTIFY(cics_sys_admin_id) UACC(NONE)
PERMIT 'CICSTS55.CICS.applid.**'  ID(applid_userid) ACCESS(UPDATE)
ADDSD  'CICSTS55.CICS.applid.DFHXR*'  NOTIFY(cics_sys_admin_id) UACC(NONE)
PERMIT 'CICSTS55.CICS.applid.DFHXR*'  ID(applid_userid) ACCESS(CONTROL)
ADDSD  'CICSTS55.CICS.applid.DFHINTRA' NOTIFY(cics_sys_admin_id) UACC(NONE)
PERMIT 'CICSTS55.CICS.applid.DFHINTRA'  ID(applid_userid) ACCESS(CONTROL)
ADDSD  'CICSTS55.CICS.applid.DFHTEMP' NOTIFY(cics_sys_admin_id) UACC(NONE)
PERMIT 'CICSTS55.CICS.applid.DFHTEMP'  ID(applid_userid) ACCESS(CONTROL)
ADDSD  'CICSTS55.CICS.DFHCSD'          NOTIFY(cics_sys_admin_id) UACC(NONE)
PERMIT 'CICSTS55.CICS.DFHCSD' ID(cics_group1,..,cics_groupn) ACCESS(UPDATE)
/*
//
```

図 *1. CICS* データ・セットへのアクセスを許可するジョブの例

注**:** すべての CICS 領域から同じ方法 (READ アクセスまたは UPDATE アクセスなど) でアクセスする必要のあるデータ・セットは、アプリケーション ID を含まないプロファイルによって保護される必要があります。 例えば、CICS ロード・モ

ジュールを含む区分データ・セットは、すべての CICS 領域グループ (またはユーザー ID) に READ アクセスを付与するプロファイルを使って定義してください。

これらすべてのデータ・セットを、「CICSTS55.CICS.**」という名の 1 つの総称プロファイルを使用して保護する方法も考えられます。ただし、CICSTS55.CICS.SDFHAUTH への読み取りアクセス権限を誰が持つかは、厳密に制御する必要があります。これには APF 許可プログラムが収められており、このデータ・セットを保護するプロファイルには UACC(NONE) を定義する必要があるためです。 50 ページの図 1 では、区分データ・セットにはすべて UACC(NONE) が定義されており、明示的なアクセス・リストがあります。

CICS モジュールはライブラリー SYS1.CICSTS55.CICS.SDFHLPA および SYS1.CICSTS55.CICS.SDFHLINK に存在しますが、これらのライブラリーへのアクセスを必要とする CICS 領域ユーザー ID はありません。

各領域に属するデータ・セットに命名規則を確立し、各 CICS 領域ごとに 1 つの総称プロファイルを設け、CICS z/OS Communications Server アプリケーション ID をデータ・セット修飾子の 1 つとすることにより、データ・セットにアクセス可能な CICS 領域を 1 つに限ることができます。 50 ページの図 1 に示した例では、すべての名前に CICSTS55.CICS の高位修飾子が付けられていますが、命名規則はインストール済み環境によって異なります。

CICS では、これらのプロファイルでカバーされるすべてのデータ・セットに UPDATE アクセス権限が必要です。 このカテゴリーに属するデータ・セットの CICS DDNAME は下記のとおりです。

DFHGCD
: グローバル・カタログ・データ・セット

DFHLCD
: ローカル・カタログ・データ・セット

DFHAUXT
: 補助トレース・データ・セット、A エクステント

DFHBUXT
: 補助トレース・データ・セット、B エクステント

DFHDMPA
: トランザクション・ダンプ・データ・セット、A エクステント

DFHDMPB
: トランザクション・ダンプ・データ・セット、B エクステント

注**:** 補助トレース・データ・セット、トランザクション・ダンプ・データ・セット、および MVS ダンプ・データ・セットには、機密情報が含まれることがあります。それらは、無許可アクセスから保護してください。

CICS では、区画内の一時データ、一時記憶、および CICS 可用性マネージャー (CAVM) データ・セットに CONTROL アクセス権限が必要です。

このカテゴリーに属するデータ・セットの CICS DDNAME は下記のとおりです。

DFHINTRA
: 一時データ区画内データ・セット

DFHTEMP
: 一時記憶域データ・セット

DFHXRMSG
XRF メッセージ・データ・セット

CICS システム定義データ・セット (CSD) は、すべての CICS グループがアクセスできる個別プロファイルにより保護されます。 このことは、すべての CICS 領域が共通の CSD を共用していることを前提としています。 ご使用の CICS 領域が共通 CSD を共用せず、各領域に固有の CSD がある場合、または領域のグループが CSD を共用している場合は、必要に応じて、個別データ・セット・プロファイルまたは総称データ・セット・プロファイルを定義してください。

CICS システム定義データ・セット (CSD) の DFHCSD ファイル用の VSAM カタログに対する読み取り権限を、CICS 領域ユーザー ID に付与する必要があります。

MVS ライブラリー・ルックアサイド (LLA) 機能によるアクセスの許可

MVS のライブラリー・ルックアサイド (LLA) 機能を使用している場合は、LLACOPY マクロを使用するプログラムの機能を制御できます。

CICS 領域のユーザー ID を以下のいずれかの 方法で許可します。

- これには、LLA モジュールが含まれているデータ・セットに対する UPDATE 権限が必要です。
- これには、CSVLLA リソースに対する UPDATE 権限が FACILITY クラス内になければなりません。 *datasetname*。ここで、*datasetname* は LLA モジュールが含まれているライブラリーの名前です。以下に例を挙げます。

```
RDEFINE FACILITY CSVLLA.datasetname UACC(NONE) NOTIFY
PERMIT  CSVLLA.datasetname CLASS(FACILITY) ID(....) ACCESS(UPDATE)
```

ユーザー・データ・セットへのアクセスを許可する

CICS 領域の RACF ユーザー ID を定義し、そのユーザーに CICS システム・データ・セットへのアクセス権限を付与した場合は、そのユーザー ID に対し、必要な権限を付与して CICS アプリケーション・データ・セットへのアクセスを許可します。

ファイル定義が CICS にインストールされていて、CICS の開始時またはその後の任意の時点で開かれることになるファイルに関し、それぞれの VSAM カタログへの読み取りアクセスを、CICS 領域ユーザー ID に対して付与する必要があります。

以下の RACF コマンドは、ID パラメーターに指定されたユーザー ID に対し、いくつかの CICS ユーザー・アプリケーション・データ・セットへのアクセスを許可します。その際、最初の 2 つのデータ・セットには READ 権限を付与し、最後の 2 つのデータ・セットには UPDATE 権限を付与します。

```
PERMIT 'CICSTS55.CICS.appl1.dataset1'  ID(user or group)  ACCESS(READ)
PERMIT 'CICSTS55.CICS.appl1.dataset2'  ID(user or group)  ACCESS(READ)
PERMIT 'CICSTS55.CICS.appl2.dataset3'  ID(user or group)  ACCESS(UPDATE)
PERMIT 'CICSTS55.CICS.appl2.dataset4'  ID(user or group)  ACCESS(UPDATE)
```

VSAM 入力順データ・セット (ESDS) に対し ACCESS(CONTROL) とする

CICS ファイル制御は、VSAM ESDS (非 RLS モードのみ) をオープンするときに制御間隔処理を使用します。つまり、該当するすべてのデータ・セットに対して ACCESS(CONTROL) を指定する必要があり、そうしないと OPEN コマンドは失敗し、メッセージ DFHFC0966 が出されます。

BWO を使用する場合は VSAM データ・セットに対し ACCESS(ALTER) とする

オープン中のバックアップ (BWO) を使用して、統合カタログ機能 (ICF) のカタログに BACKUPTYPE(DYNAMIC) または BWO(TYPECICS) として定義された、現在使用中の VSAM データ・セットをバックアップするには、CICS 領域ユーザー ID に対し、データ・セット、またはそのデータ・セットが定義された ICF カタログへの RACF ALTER 権限を付与します。これを行わないと、OPEN コマンドは失敗し、メッセージ DFHFC5803 が出されます。 BWO の使用については、Back-up-while-open (BWO) を参照してください。

SMS データ・クラス属性の動的ボリューム・カウントを指定する場合は VSAM データ・セットに対し ACCESS(ALTER) とする

動的ボリューム・カウント

SMS データ・クラス属性の動的ボリューム・カウントを使用して、VSAM データを複数のボリュームに拡張することができます。SMS のリリース・レベルによっては、EOV 拡張処理中に CICS 領域ユーザー ID が ICF カタログ・ボリューム・リストを更新するために ACCESS(ALTER) が必要となる可能性があります。ご使用の DFSMS リリースに対する正しいアクセス・レベルを特定するには、「z/OS DFSMS アクセス方式サービス・プログラム・コマンド」の『必須 RACF 許可テーブル』を参照してください。

一時記憶域プールへのアクセスを許可する

一時記憶域 (TS) サーバーによるカップリング・ファシリティーの TS プールへのアクセスを制御できます。

各 TS サーバーは、ジョブまたは開始タスクとして開始できます。TS サーバーの TS キュー・プールの名前は、サーバーの始動時に指定されます。TS プールごとに、シスプレックス内の各 MVS イメージ上で稼働できる TS サーバーは 1 つだけです。

TS サーバーのユーザー ID (つまり、ジョブまたは開始タスクが実行されているユーザー ID) に対して 2 つのセキュリティー検査が行われます。サーバーがこれらの検査に確実に合格するためには、次のようにします。

- TS サーバー領域が固有の TS プールに使用されるカップリング・ファシリティー・リスト構造に接続することを許可します。これを行うには、TS サーバーのユーザー ID が、FACILITY 一般リソース・クラスの IXLSTR.*structure_name* というカップリング・ファシリティー・リソース管理 (CFRM) RACF プロファイルへの ALTER 権限を持っている必要があります。

例えば、サーバーのユーザー ID が DFHXQTS1 で、リスト構造が DFHXQLS_TSPRODQS という名前である場合、以下の RACF コマンドはプロファイルを定義し、必要なアクセス権限を提供します。

```
RDEFINE FACILITY IXLSTR.DFHXQLS_TSPRODQS UACC(NONE)
PERMIT IXLSTR.DFHXQLS_TSPRODQS CLASS(FACILITY) ID(DFHXQTS1) ACCESS(ALTER)
```

セキュリティー管理を削減するには、同じ TS プールをサポートする各 TS サーバーを始動するのに同じ TS サーバー・ユーザー ID を使用します。

- TS サーバーのユーザー ID に対して、FACILITY 一般リソース・クラスの DFHXQ.*poolname* という CICS RACF プロファイルへの CONTROL アクセス権限を付与します。これにより TS サーバーは、指定された TS プールのサーバーとなることが許可されます。

  例えば、サーバーのユーザー ID が DFHXQTS1 で、プール名が TSPRODQS の場合、以下の RACF コマンドはプロファイルを定義し、必要なアクセス権限を提供します。

  ```
  RDEFINE FACILITY DFHXQ.TSPRODQS UACC(NONE)
  PERMIT DFHXQ.TSPRODQS CLASS(FACILITY) ID(DFHXQTS1) ACCESS(CONTROL)
  ```

  TS サーバーに対する応答について詳しくは、『TS サーバーに対する System Authorization Facility (SAF) の応答』を参照してください。

一時記憶域サーバーへのアクセスを許可する

TS サーバーへのアクセスを CICS 領域によって制御できます。

セキュリティー検査は CICS 領域ユーザー ID に対して行われ、領域が TS サーバーのサービスの使用を許可されているかどうかが確認されます。この検査は、CICS 領域が TS サーバーに接続するたびに行われます。

TS サーバーに接続する各 CICS 領域のユーザー ID に対して、FACILITY 一般リソース・クラスの DFHXQ.*poolname* という CICS RACF プロファイルへの UPDATE アクセス権限を付与します。これにより CICS 領域は、指定された TS プールに対して TS サーバーのサービスの使用を許可されます。

例えば、CICS 領域のユーザー ID が CICSDAA1 で、プール名が TSPRODQS の場合、以下の RACF コマンドによりプロファイルが定義され、必要なアクセス権限が提供されます。

```
RDEFINE FACILITY DFHXQ.TSPRODQS UACC(NONE)
PERMIT DFHXQ.TSPRODQS CLASS(FACILITY) ID(CICSDAA1) ACCESS(UPDATE)
```

CICS 領域が TS プールに接続されている場合は、サーバーによってさらにセキュリティー検査が実行されることなく、TS キューの書き込み、読み取り、および削除ができます。ただし、TS API 要求を発行する CICS アプリケーション所有領域は、TS リソース・セキュリティー検査の既存のメカニズムを使用できます。

TS サーバーに対する System Authorization Facility (SAF) の応答

TS プールのセキュリティー・プロファイルを取得できない場合、SAF はアクセス要求を認可することも拒否することもしません。このような状態では、次のようになります。

CICS 領域または TS サーバー自体のいずれかによる、TS プールへのアクセスは、以下の場合に拒否されます。

- セキュリティー・マネージャーはインストールされていますが、MVS イメージが持続している間、一時的に非アクティブまたは操作不能になります。これは、セキュリティー・マネージャーがアクティブであると TS プールへのアクセスを許可しないプロファイルを取得する可能性があるという理由に基づく、フェイルセーフ動作です。

CICS 領域または TS サーバー自体のいずれかによる、TS プールへのアクセスは、以下の場合に受け入れられます。

- インストールされているセキュリティー・マネージャーがありません。または、
- アクティブなセキュリティー・マネージャーがありますが、FACILITY クラスが非アクティブであるか、または FACILITY クラスにプロファイルがありません。 TS サーバーへのアクセスを制御することをユーザーが希望しているという証拠がないので、この場合にはアクセス要求は許可されます。

TS サーバーへのアクセスは、特定の DFHXQ.*poolname* プロファイルや適用可能な総称プロファイルがなくても許可されます。これを示すメッセージは発行されません。潜在的な機密漏れが発生しないようにするために、総称プロファイルを使用して、TS サーバーのすべてまたは特定のグループを保護することができます。例えば、以下のように指定します。

```
RDEFINE FACILITY (DFHXQ.*) UACC(NONE)
```

これにより、TS サーバーまたは CICS 領域が許可されている、より具体的なプロファイルを使用して、TS サーバーのみへのアクセスが許可されます。

名前付きカウンター・プールおよびサーバーへのアクセスの許可

アクセスは、次のもので制御できます。

- 名前付きカウンター・プールに対するカップリング・ファシリティー・データ・テーブル (名前付きカウンター) サーバー (『名前付きカウンター・プールへのアクセス』を参照)
- 名前付きカウンター・サーバーに対する CICS 領域 (56 ページの『名前付きカウンター・サーバーへのアクセス』を参照)

名前付きカウンター・プールへのアクセス

名前付きカウンター・サーバーによる、カップリング・ファシリティー内の名前付きカウンター・プールへのアクセスを制御できます。

それぞれの名前付きカウンター・サーバーは、ジョブまたは開始タスクとして開始できます。名前付きカウンター・サーバーの名前付きカウンター・プールの名前は、サーバーの始動時に指定されます。それぞれの名前付きカウンター・プールで、シスプレックス内の各 MVS イメージ上で実行するサーバーは 1 つのみ存在させることができます。

名前付きカウンター・サーバーのユーザー ID (ジョブまたは開始済みタスクの実行ユーザー ID) に対しては、2 つのセキュリティー検査が行われます。サーバーがこれらの検査に確実に合格するには、次のようにします。

- 名前付きカウンター・サーバー領域に、独自の名前付きカウンター・プールに使用されるカップリング・ファシリティー・リスト構造への接続を許可します。これには、名前付きカウンター・サーバー・ユーザー ID が、FACILITY 一般リソース・クラス内の IXLSTR.*structure_name* という、カップリング・ファシリティー・リソース管理 (CFRM) RACF プロファイルに対する ALTER 権限を持っていることが必要です。

  例えば、サーバーのユーザー ID が DFHNCSV1 であり、リスト構造が DFHNCLS_DFHNC001 である場合、以下の RACF コマンドはプロファイルを定義し、必要なアクセス権限を提供します。

  ```
  RDEFINE FACILITY IXLSTR.DFHNCLS_DFHNC001 UACC(NONE)
  PERMIT IXLSTR.DFHNCLS_DFHNC001 CLASS(FACILITY) ID(DFHNCSV1) ACCESS(ALTER)
  ```

セキュリティー管理を削減するには、同じ名前付きカウンター・サーバー・ユーザー ID を使用して、同じ名前付きカウンター・サーバー・プールをサポートするそれぞれの名前付きカウンター・サーバーを開始します。

- 名前付きカウンター・サーバーのユーザー ID に、FACILITY 一般リソース・クラス内の DFHNC.*poolname* という CICS RACF プロファイルに対する CONTROL 権限を付与します。これにより名前付きカウンター・サーバーは、名前付きカウンター・プールに対してサーバーとして機能することが許可されます。

  例えば、サーバーのユーザー ID が DFHNCSV1 であり、プール名が DFHNC001 である場合、以下の RACF コマンドはプロファイルを定義し、必要なアクセス権限を提供します。

  ```
  RDEFINE FACILITY DFHNC.DFHNC001 UACC(NONE)
  PERMIT DFHNC.DFHNC001 CLASS(FACILITY) ID(DFHNCSV1) ACCESS(CONTROL)
  ```

  CFDT サーバーへの応答については、 57 ページの『名前付きカウンター・サーバーに対する System Authorization Facility (SAF) の応答』を参照してください。

名前付きカウンター・サーバーへのアクセス

CICS 領域による名前付きカウンター・サーバーへのアクセスを制御できます。

CICS 領域が名前付きカウンター・サーバーに接続するたびに、CICS 領域ユーザー ID に対してセキュリティー検査が行われ、その領域が、その名前付きカウンター・サーバーのサービスの使用を許可されていることを確認します。

名前付きカウンター・サーバーに接続する各 CICS 領域ユーザー ID に、FACILITY 一般リソース・クラス内の DFHNC.*poolname* という CICS RACF プロファイルへの UPDATE 権限を付与します。これにより CICS 領域は、名前付きカウンター・プールに対して名前付きカウンター・サーバーのサービスを使用することが許可されます。

例えば、CICS 領域のユーザー ID が CICSDAA1 であり、プール名が DFHNC001 である場合、以下の RACF コマンドはプロファイルを定義し、必要なアクセス権限を提供します。

```
RDEFINE FACILITY DFHNC.DFHNC001 UACC(NONE)
PERMIT DFHNC.DFHNC001 CLASS(FACILITY) ID(CICSDAA1) ACCESS(UPDATE)
```

CICS 領域は、名前付きカウンター・プールに接続されると、サーバーによって実行される、それ以上のセキュリティー検査を受けることなく、名前付きカウンターを定義、更新、削除、取得、巻き戻し、および照会できます。

注: 共用一時記憶域プールおよびカップリング・ファシリティー・データ・テーブル・プールとは異なり、名前付きカウンターは、バッチ・アプリケーション領域によってアクセスすることもできます。バッチ・ジョブは、CICS 領域と同じセキュリティー・メカニズムに従います。

名前付きカウンター・サーバーに対する System Authorization Facility (SAF) の応答

名前付きカウンター・プールのセキュリティー・プロファイルを取得できない場合、SAF はアクセス要求を認可することも拒否することもしません。このような状態では、次のようになります。

CICS 領域または名前付きカウンター・サーバー自体のいずれかによる、名前付きカウンター・プールへのアクセスは、以下の場合に拒否されます。

- セキュリティー・マネージャーはインストールされていますが、MVS イメージが持続している間、一時的に非アクティブまたは操作不能になります。これは、セキュリティー・マネージャーがアクティブであると名前付きカウンター・プールへのアクセスを許可しないプロファイルを取得する可能性があるという理由に基づく、フェイルセーフ動作です。

CICS 領域または名前付きカウンター・サーバー自体のいずれかによる、名前付きカウンター・プールへのアクセスは、以下の場合に受け入れられます。

- インストールされているセキュリティー・マネージャーがありません。または、
- アクティブなセキュリティー・マネージャーがありますが、FACILITY クラスが非アクティブであるか、または FACILITY クラスにプロファイルがありません。名前付きカウンター・サーバーへのアクセスを制御することをユーザーが希望しているという証拠がないため、この場合にはアクセス要求は許可されます。

名前付きカウンター・サーバーへのアクセスは、特定の DFHCF.*poolname* プロファイルや適用可能な総称プロファイルがなくても許可されます。これを示すメッセージは発行されません。潜在的な機密漏れが発生しないようにするために、総称プロファイルを使用して、名前付きカウンター・サーバーのすべてまたは特定のグループを保護することができます。例えば、以下のように指定します。

```
RDEFINE FACILITY (DFHNC.*) UACC(NONE)
```

これにより、名前付きカウンター・サーバーまたは CICS 領域が許可されている、より具体的なプロファイルを使用して、名前付きカウンター・サーバーのみへのアクセスが許可されます。

SMSVSAM サーバーへのアクセスを許可する

SMSVSAM は、独自のアドレス・スペースで稼働して CICS が必要とする RLS サポートを提供する、データ共用サブシステムです。 VSAM レコード・レベル共用 (RLS) を使用する CICS 領域の場合は、CICS 領域ユーザー ID に対して行われる

RACF セキュリティー検査で、その領域が SMSVSAM サーバーへの登録を許可されているかどうかを確認することにより、SMSVSAM サーバーへのアクセスが制御されます。

テスト環境では、デフォルトのアクションを使用して、VSAM RLS を使用するすべての CICS 領域に SMSVSAM サーバーへの接続を許可できます。このアクセスを保護するには、RACF SUBSYSNM 一般リソース・クラスをアクティブにし、SMSVSAM サーバーに接続する各 CICS 領域にサーバーへのアクセスを許可する必要があります。つまり、RACF SUBSYSNM 一般リソース・クラスの該当のプロファイルへのアクセスを認可するということです。

一般リソース・クラス SUBSYSNM は、SMSVSAM に接続するサブシステムの許可をサポートします。SUBSYSNM プロファイル名は、特定のサブシステムが VSAM に認識される際の名前です。CICS はサブシステム名としてアプリケーション ID を使用します。CICS が制御 ACB を登録できるように、SUBSYSNM リソース内で CICS アプリケーション ID のプロファイルを定義してください。

CICS が初期設定時に制御 ACB を登録しようとすると、SMSVSAM は RACF を呼び出して、CICS 領域ユーザー ID が SUBSYSNM クラス内の CICS プロファイルに対して許可されているかどうかを検査します。CICS 領域ユーザー ID に READ 権限がない場合、オープン要求は失敗します。

例えば、CICS AOR のアプリケーション ID が CICSDAA1 で、CICS 領域ユーザー ID (複数の AOR で共有) が CICSDA# # の場合、次のようにプロファイルを定義し、許可を与えてください。

```
RDEFINE SUBSYSNM  CICSDAA1  UACC(NONE) NOTIFY(userid)
PERMIT  CICSDAA1  CLASS(SUBSYSNM)  ID(CICSDA##)  ACCESS(READ)
```

アプリケーション ID にワイルドカード文字を使用することで、複数の CICS 領域を指定できます。例:

```
PERMIT  CICSD%%%  CLASS(SUBSYSNM)  ID(CICSDGRP) ACCESS(READ)
```

CICS 領域へのアクセスの許可

CICS アプリケーション ID プロファイルを RACF APPL クラスに定義することによって、端末ユーザーによるアクセスを特定の CICS 領域に制限できます。

この目的で、CICS 領域のアプリケーション ID は下記のようになります。

- システム初期設定パラメーターとして GRNAME が指定されている場合は、z/OS Communications Server 汎用リソース名になる。
- アプリケーション ID システム初期設定パラメーターとして汎用アプリケーション ID が指定されている場合は、汎用アプリケーション ID になる。
- システム初期設定パラメーターとして特定のアプリケーション ID しか指定されていない場合は、その特定のアプリケーション ID になる。

CICS アプリケーション ID の APPL クラスにプロファイルを定義する場合、または UACC(NONE) が定義されている 1 つ以上の CICS アプリケーション ID に当てはまる総称プロファイルを定義する場合は、CICS 領域にサインオンしようとする

すべての端末ユーザーに、その領域のアプリケーション ID に該当するプロファイルへの明示的アクセスが (個別プロファイルまたはグループのメンバーとして) 必要です。例えば、次のとおりです。

```
RDEFINE APPL cics_region_applid UACC(NONE)  NOTIFY(sys_admin_userid)
```

同じ z/OS Communications Server 汎用リソース名のメンバーであるすべての CICS 領域に対して、RACF データベースには APPL プロファイル名を 1 つ定義するだけで済みます。CICSPlex では、すべての端末専有領域は同一の z/OS Communications Server 汎用リソース名を持つため、CICSPlex におけるすべてのサインオン検証は、同一の APPL プロファイルに対して行われます。

MRO の場合に限り、アプリケーション ID は端末所有領域 (TOR) からユーザーがアクセスするその他の領域へ伝搬されます。例えば TOR からアプリケーション所有領域 (AOR) へ、また AOR からファイル所有領域 (FOR) へなどという形があります。その結果として、次のようになります。

- これら領域の APPL プロファイルには、AOR および FOR のユーザーを含める必要がありません。
- 他のアプリケーション ID へのアクセスを拒否することによって、ユーザーに TOR 経由のサインオンを強制することができます。

CICS APPL プロファイルのアクセス・リストに許可されたユーザーを追加するには、RACF PERMIT コマンドを使用します。例えば、以下のような項目が含まれています。

```
PERMIT cics_region_applid CLASS(APPL) ID(group1,...,groupn) ACCESS(READ)
```

これにより、リストされたグループで定義されているすべてのユーザーが、cics_region_applid へのサインオンを許可されます。

この保護を有効にするには、APPL クラスがアクティブである必要があります。

```
SETROPTS CLASSACT(APPL)
```

また、パフォーマンス上の理由から、RACLIST を使用してプロファイルを APPL クラスで活動化することも検討してください。

```
SETROPTS RACLIST(APPL)
```

APPL クラスが既にアクティブになっている場合は、SETROPTS コマンドを使用して、ストレージ内の APPL プロファイルをリフレッシュします。

```
SETROPTS RACLIST(APPL) REFRESH
```

注**:**

1. CICS は RACF にユーザーのサインオン検査を実行するように要求するとき、必ずアプリケーション ID を RACF に渡します。CICS の内部には、これを防止する手段はありません。
2. RACF は、未定義の CICS アプリケーション ID を UACC(READ) として処理します。
3. APPL クラスがアクティブであり、APPL クラスに CICS 領域のプロファイルが存在する場合は、許可されたリモート CICS 領域が、この方法で保護される CICS 領域に必ずサインオンできるようにしてください。

アプリケーションへのアクセスの制御について詳しくは、「z/OS Security Server RACF セキュリティー管理者のガイド」を参照してください。

CICS 領域の z/OS Communications Server ACB オープンの制御

非 APF 許可プログラムを実行しているユーザーのうち、どのユーザーが CICS アドレス・スペース (CICS 領域) に関連した z/OS Communications Server SNA ACB を OPEN できるかを制御することができます。

これにより、許可された CICS 領域のみが、サービスにこの APPLID を提供する z/OS Communications Server アプリケーションとなることができるため、無許可のユーザーは実 CICS 領域を使用できなくなります。 **SET VTAM OPEN** コマンドの発行者ではなく、CICS 領域ユーザー ID に OPEN アクセス権が必要です。

アプリケーション ID ごとに VTAMAPPL プロファイルを作成し、CICS 領域ユーザー ID に READ アクセスを与えてください。例えば、以下のような項目が含まれています。

```
RDEFINE VTAMAPPL applid UACC(NONE)  NOTIFY(userid)
PERMIT applid CLASS(VTAMAPPL) ID(cics_region_userid) ACCESS(READ)
```

VTAMAPPL クラスに指定する正確な CICS アプリケーション ID は、CICS システム初期設定パラメーターに指定される特定のアプリケーション ID です。

この保護を有効にするには、RACLIST を使用して VTAMAPPL クラスをアクティブ化する必要があります。

```
SETROPTS CLASSACT(VTAMAPPL) RACLIST(VTAMAPPL)
```

VTAMAPPL クラスが既にアクティブな場合は、SETROPTS コマンドを使用して、ストレージ内の VTAMAPPL プロファイルをリフレッシュします。

```
SETROPTS RACLIST(VTAMAPPL) REFRESH
```

注**:** VTAM は、現在は z/OS Communications Server (for SNA または IP) です。

ユーザー ID 伝搬の制御

ジョブが CICS から JES 内部読み取りプログラムに実行依頼されるときに、JOB ステートメントに USER オペランドが指定されていない場合、そのジョブは CICS 領域ユーザー ID のもとで実行されます。これらのジョブは CICS 領域自体のアクセス権限を持っているため、MVS システム内の他のデータ・セットを公開する恐れがあります。

お客様 (または RACF セキュリティー管理者) は、プロファイル名が CICS 領域ユーザー ID である PROPCNTL クラスでプロファイルを定義することによって、これらのバッチ・ジョブに CICS 領域ユーザー ID が伝搬されるのを防止できます。例えば、CICS 領域ユーザー ID が CICS1 である場合は、CICS1 という名前の PROPCNTL プロファイルを定義します。

```
RDEFINE  PROPCNTL  CICS1
```

この保護を有効にするには、RACLIST を使用して PROPCNTL クラスをアクティブ化する必要があります。

```
SETROPTS CLASSACT(PROPCNTL) RACLIST(PROPCNTL)
```

PROPCNTL クラスが既にアクティブな場合は、SETROPTS コマンドを使用して、ストレージ内の PROPCNTL プロファイルをリフレッシュします。

```
SETROPTS RACLIST(PROPCNTL) REFRESH
```

お客様 (または RACF セキュリティー管理者) は、これらのプロファイルをリフレッシュするには SETROPTS コマンドを発行する必要があります。CICS PERFORM SECURITY REBUILD コマンドを発行しても、PROPCNTL クラスには影響しません。

CICS 環境での代理ジョブ実行依頼

CICS によって実行依頼されたバッチ・ジョブは、CICS 領域のユーザー ID 以外の USER パラメーターを使用するが、対応する PASSWORD は指定しない状態で、実行することができます。

これは、代理ジョブ実行依頼と呼ばれます。これらのジョブは、JOB ステートメントで指定された USER パラメーターのアクセス権限を持っています。JOB ステートメントに PASSWORD パラメーターが指定されている場合、代理処理は行われません。

お客様 (または RACF セキュリティー管理者) は、SURROGAT クラスのプロファイルを定義することで、この処理を許可できます。例えば、CICS 領域のユーザー ID が CICS1 で、ジョブがユーザー ID JOE に対して実行される場合は、JOE.SUBMIT という名前の SURROGAT プロファイルを次のように定義します。

```
RDEFINE  SURROGAT  JOE.SUBMIT  UACC(NONE)
         NOTIFY(JOE)
```

さらに、CICS 領域のユーザー ID が、定義したプロファイルの代理となることを許可する必要があります。

```
PERMIT JOE.SUBMIT CLASS(SURROGAT) ID(CICS1) ACCESS(READ)
```

この保護を有効にするには、RACLIST を使用して SURROGAT クラスをアクティブ化する必要があります。

```
SETROPTS CLASSACT(SURROGAT) RACLIST(SURROGAT)
```

重要:

すべての CICS ユーザーは、サインオンしているかどうかにかかわらず、その CICS ユーザー ID に SURROGAT の権限があれば、SURROGAT ユーザー ID を使用するジョブを実行依頼できます。インストール済み環境が一時データ・キューを使用してジョブを実行依頼する場合は、内部読み取りプログラムに送信される一時データ・キューへの書き込みが許可されるユーザーを制御できます。ただし、インストール済み環境が EXEC CICS SPOOLOPEN を使用してジョブを実行依頼する場合、(コマンドを検査する API グローバル・ユーザー出口プログラムを作成せずに) ジョブを実行依頼できるユーザーを制御することはできません。CICS スプール・コマンドは、CICS リソースまたはコマンドの検査を行いません。

EXEC CICS ASSIGN USERID コマンドを使用すると、アプリケーション・コードをトリガーしたユーザーのユーザー ID を検索できます。これでアプリケーション・プログラマーは、USER オペランドを編集して内部読み取りプログラム向けの JOB カードに記述するコードを提供することができます。

代理ジョブ実行依頼サポートの詳細な説明については、z/OS Security Server RACF セキュリティー管理者のガイドを参照してください。

CICS 領域ユーザー ID を代理ユーザーとして許可する

CICS が代理ユーザー検査を実行する場合、CICS 領域ユーザー ID は代理として許可されていなければなりません。

このタスクについて

CICS 領域ユーザー ID に対して、以下の代理ユーザーとなる権限を付与します。

- CICS デフォルト・ユーザー
- 初期設定後の処理に使用されるユーザー ID (PLTPIUSR)
- 一時データのトリガー・レベル・トランザクションに使用されるすべてのユーザー ID
- Db2 リソース定義の AUTHID または COMAUTHID パラメーターに指定されたすべてのユーザー ID

代理ユーザー検査について詳しくは、 154 ページの『代理ユーザー・セキュリティー』を参照してください。

CICS 環境での JES スプールの保護

インストール済み環境では、JESSPOOL クラスのプロファイルを使用して JES スプール・データ・セットを保護できます。

SPOOLOPEN コマンドによって作成されたスプール・ファイルは、そのセキュリティー・トークン内に、**SPOOLOPEN** コマンド発行者のユーザー ID ではなく CICS 領域のユーザー ID が含まれています。したがって、関連 JESSPOOL プロファイル内のユーザー ID 修飾子は、CICS 領域のユーザー ID です。

SPOOLOPEN INPUT コマンドを使用する場合、CICS は、APPLID の先頭の 4 文字がスプール・ファイルの外部書き出しプログラム名に対応するかどうかを検査します。この検査は、他にも実行される可能性があるどの RACF 検査とも無関係です。

SPOOLWRITE コマンドを使用して内部読み取りプログラムに書き込む場合、CICS は、ジョブ・カードに指定されたユーザー ID を使用してジョブを実行依頼することがユーザーに許可されているかどうかを検査するために、代理ユーザー検査を実行します。詳しくは、内部読み取りプログラムに JCL ジョブを実行依頼する場合のセキュリティーを参照してください。

EXEC PGM=DFHSIP ステートメントの PARM パラメーターまたは SYSIN での HPO 使用の許可

CICS 領域の JCL にある EXEC PGM=DFHSIP ステートメントの **PARM** パラメーター、または CICS 始動ジョブ・ストリームの SYSIN データ・セットで、**HPO** を指定する場合は、RACF プロファイルを使用して **HPO** の使用を制御し、このプロファイルへのアクセスに必要な CICS 領域ユーザー ID を付与する必要があります。

HPO 用 RACF プロファイルの作成

FACILITY クラスに RACF プロファイル DFHSIT.HPO を作成して、EXEC PGM=DFHSIP ステートメントの **PARM** パラメーターまたは SYSIN で **HPO** を使用する際に必要なセキュリティー許可を定義する必要があります。

例:

```
RDEFINE FACILITY DFHSIT.HPO UACC(NONE)
```

UACC(NONE) は、コマンド規則がデフォルトですべてのユーザーに適用されることを意味します。

CICS 領域ユーザー ID へのプロファイル DFHSIT.HPO に対する READ アクセス権限の付与

PERMIT コマンドを使用して、関連する領域ユーザー ID に、プロファイルに対する READ アクセス権限を付与する必要があります。

例:

```
PERMIT DFHSIT.HPO CLASS(FACILITY) ID(CICS_region_userid) ACCESS(READ)
```

セキュリティー関連システム初期設定パラメーター

システム・セキュリティー要件を指定するために、いくつかのシステム初期設定パラメーターが用意されています。

SEC

SEC システム初期設定パラメーターは、CICS 領域に必要なリソース・セキュリティー管理のレベルを指定します。以下の 2 つのオプションがあります。

YES

CICS 外部セキュリティー・インターフェースを初期化します。CICS セキュリティーの制御は、以下に示す他のセキュリティー関連システム初期設定パラメーターによって決まります。

CMDSEC	**XAPPC**	**XPPT**
DFLTUSER	**XCMD**	**XPSB**
ESMEXITS	**XDB2**	**XRES**

PSBCHK	**XDCT**	**XRFSOFF**
PLTPISEC	**XFCT**	**XRFSTME**
RESSEC	**XHFS**	**XTRAN**
SECPRFX	**XJCT**	**XTST**
SNSCOPE	**XPCT**	**XUSER**

NO CICS リソースおよびこの領域内のその他のリソースにアクセスしたいユーザーに対して、セキュリティー検査は実行されず、ユーザー・サインオンは使用できません。

注: SEC=NO を指定した場合でも、MRO のバインド時のセキュリティーを使用すると、CICS 領域ユーザー ID は 2 次システムに送信され、2 次システムでバインド時の検査が実行されます。詳細については、 336 ページの『MRO でのバインド時のセキュリティー』を参照してください。

SECPRFX

このパラメーターは、SEC=YES も指定する場合にのみ有効です。SECPRFX システム初期設定パラメーターは、CICS が許可を得るために RACF に渡すリソース名に接頭部を付けるかどうかを指定します。CICS が使用する接頭部は、CICS 領域が実行されている RACF ユーザー ID です。

接頭部を付けることは、主に、複数の CICS 領域がある場合に役立ちます。これにより、ある CICS 領域上のユーザーが、異なる接頭部を持つ別の CICS 領域のリソースにアクセスするのを防ぐことができます。例えば、ある CICS 領域が接頭部 CICSPROD を使用し、別の領域は接頭部 CICSTEST を使用するとします。CICSTEST システムのユーザーは、CICSTEST 接頭部を含むプロファイルを使用でき、CICSPROD システムのユーザーは、CICSPROD 接頭部を含むプロファイルを使用できます。どちらのシステムのユーザーも、CICS* を含むプロファイルによって保護されたリソースを使用できます。

SECPRFX パラメーターには、以下のオプションがあります。

NO CICS は、この CICS 領域から RACF に渡す許可要求のリソース名に接頭部を付けません。

YES
CICS は、許可要求を RACF に渡すときに、リソース名の前に CICS 領域ユーザー ID を付けます。

この値を変更するには、ICHRTX00 SAF 事前処理出口を使用します。詳しくは、 196 ページの『CICS が CICS 領域のユーザー ID を特定する状況と方法』を参照してください。

prefix
CICS は、許可要求を RACF に渡すときに、指定された接頭部をリソース名の前に付けます。

例えば、CICS ジョブが JOB ステートメントで USER=CICSREG と指定し、なおかつ SECPRFX=YES が指定されている場合は、以下のように TCICSTRN リソース・クラスの CICS マスター端末トランザクション (CEMT) へのアクセスを定義および許可できます。

```
RDEFINE  TCICSTRN  CICSREG.CEMT
         UACC(NONE)   NOTIFY(sys_admin_userid)
PERMIT  CICSREG.CEMT  CLASS(TCICSTRN)
        ID(groupid1,...,groupidn) ACCESS(READ)
```

GCICSTRN リソース・クラスのリソース・グループ・プロファイルを使用することもできます。その場合は、ADDMEM オペランドで接頭部を指定します。次の例は、CICSTRANS という名前のプロファイルに指定された CICSREG を示します。

```
RDEFINE  GCICSTRN  CICSTRANS
         ADDMEM(CICSREG.CEMT)
         UACC(NONE)   NOTIFY(sys_admin_userid)
PERMIT  CICSTRANS  CLASS(GCICSTRN)
        ID(groupid1,...,groupidn) ACCESS(READ)
```

注: あるリソース・グループ・プロファイルを使用してリソースを保護する場合、別のプロファイルを使用して同じリソースを保護することは避けてください。プロファイルが異なる (例えば、異なるアクセス・リストを持つ) 場合、RACF は許可検査中に使用できるようにプロファイルをマージします。このマージ処理はパフォーマンスに影響する恐れがあるだけでなく、特定のユーザーに適用されるアクセス権限を正確に判別するのが困難になる可能性があります。詳しくは、z/OS Security Server RACF セキュリティー管理者のガイドを参照してください。

CMDSEC

CMDSEC は、トランザクションのリソース定義で指定された CMDSEC オプションを CICS が使用するかどうかを指定します。CMDSEC でオプション ASIS が指定された場合は、CICS が CMDSEC オプションに従うことを意味します。CMDSEC でオプション ALWAYS が指定された場合は、CICS が CMDSEC オプションを無視し、常にコマンド検査を実行することを意味します。これらのオプションについて詳しくは、CMDSEC システム初期設定パラメーターを参照してください。

DFLTUSER

RACF にデフォルト・ユーザー ID として定義した名前を CICS に対して示すには、DFLTUSER の値を指定します。このパラメーターを省略すると、名前はデフォルトで CICSUSER に設定されます。 46 ページの『デフォルトの CICS ユーザー ID を RACF に定義する』を参照してください。

ESMEXITS

ESMEXITS は、RACF インストール・システム出口で使用するためのインストール・データを CICS が渡すかどうかを指定します。

PLTPISEC

PLTPISEC は、CICS 初期設定時に実行される PLT プログラムに対して CICS がコマンド・セキュリティー検査またはリソース・セキュリティー検査を実行するかどうかを指定します。

PLTPIUSR

PLTPIUSR は、CICS 初期設定時に実行される PLT プログラムのセキュリティー検査で CICS が使用するユーザー ID を指定します。

PSBCHK

(接続先の IMS™ システムにアクセスするために) トランザクション・ルーティングを使用してこの CICS 領域でトランザクションを開始するリモート端末ユ

ーザーに対して、CICS が PSB 許可検査を実行するよう指定するには、PSBCHK をコーディングします。デフォルトの PSBCHK=NO は、CICS はリモート・リンクを検査するがリモート・ユーザーは検査しないことを示します。リモート・ユーザーを検査するには、PSBCHK=YES を指定します。

RESSEC

RESSEC は、トランザクションのリソース定義で指定された RESSEC オプションを CICS が順守するかどうかを指定します。RESSEC でオプション ASIS が指定された場合は、CICS が RESSEC オプションに従うことを意味します。RESSEC でオプション ALWAYS が指定された場合は、CICS が RESSEC オプションを無視し、常にリソース検査を実行することを意味します。これらのオプションについて詳しくは、「*CICS* システム定義ガイド」を参照してください。

SNSCOPE

SNSCOPE (サインオン SCOPE) は、明示的なサインオン要求 (例えば、EXEC CICS SIGNON コマンドや CESN トランザクションなど) によってサインオンしているすべてのユーザー ID に適用されます。SNSCOPE は、1 つのユーザー ID において同時に複数の CICS セッションがアクティブになることができるかどうかを指定します。

サインオン SCOPE は、MVS ENQ マクロによって適用されます。SNSCOPE 値は、ENQ スコープの STEP、SYSTEM、および SYSTEMS レベルに対応します。これは、SNSCOPE に対してまったく同じ値を指定している CICS システムのみが、互いのスコープを検査できることを意味します。

SNSCOPE は、ローカル端末でサインオンしているユーザー、または CRTE トランザクションを使用して別のシステムに接続した後にサインオンしているユーザーにのみ影響を及ぼします。

CICS リソース・クラスのシステム初期設定パラメーター

CICS で RACF を使って CICS リソースへのアクセスを許可することをシステム・レベルで (SEC=YES パラメーターによって) 指定します。また、X*name* システム初期設定パラメーターを使用して、CICS のどの特定リソースを CICS で検査するかについてもシステム・レベルで指定します。

CICS リソース・クラスの完全なリストを表 4 に示します。各クラスは、対応する X*name* システム初期設定パラメーターで示されます。

表 *4. CICS* リソース・クラスのシステム初期設定パラメーター

システム初期設定パラメーター	リソース
XAPPC={**NO**|YES}	APPC パートナー - LU 検査
XCMD= {**YES**|name|NO}	EXEC CICS システム・コマンド EXEC CICS FEPI システム・コマンド
XDB2={**NO**|name}	CICS Db2 リソース
XDCT={**YES**|name|NO}	一時データ・キュー
XFCT= {**YES**|name|NO}	ファイル
XHFS={**YES**|NO}	z/OS UNIX システム・サービスによって管理される z/OS UNIX ファイル

表 4. *CICS* リソース・クラスのシステム初期設定パラメーター (続き)

システム初期設定パラメーター	リソース
XJCT={**YES**|name|NO}	ジャーナルおよびログ
XPCT={**YES**|name|NO}	開始されたトランザクションおよび EXEC CICS コマンド: COLLECT STATISTICS TRANSACTION DISCARD TRANSACTION INQUIRE TRANSACTION SET TRANSACTION
XPPT= {**YES**|name|NO}	Programs (プログラム)
XPSB={**YES**|name|NO}	DL/I プログラム仕様ブロック (PSB)
XRES={**YES**|name|NO}	XRES セキュリティー検査の対象となる CICS リソース。XRES セキュリティー検査の対象となるリソースのリストについては、 101 ページの『XRES リソース・セキュリティー・パラメーターを使用したセキュリティー』を参照してください。
XTRAN={**YES**|name|NO}	接続されたトランザクション
XTST={**YES**|name|NO}	一時記憶域キュー
XUSER={**YES**|NO}	代理ユーザー検査 Db2 AUTHTYPE 検査

注**:**

1. パラメーターは SEC=YES が指定されている場合のみ有効です。
2. どのパラメーターもコンソールのオーバーライドとして入力することはできません。

RACF を使用してリソース・セキュリティーを管理する X*name* システム初期設定パラメーターのいずれかに YES を指定した場合、CICS はそのパラメーターのデフォルト・クラス名を使用します。これらのリストについては、 27 ページの『RACF classes for CICS リソース』を参照してください。

一例として、SEC=YES を指定し、さらに 3 つのリソース・クラス・パラメーターで X*name*=YES と指定した場合の効果を次の表に示します。

表 5. デフォルト・リソース・クラスによる外部セキュリティーの指定

システム初期設定パラメーター	効果
SEC=YES	CICS は外部セキュリティー・インターフェースを初期設定します。
XTRAN=YES	CICS は、トランザクション接続セキュリティー検査に TCICSTRN および GCICSTRN リソース・クラス・プロファイルを使用します。
XFCT=YES	CICS は、ファイル・アクセス・セキュリティー検査に FCICSFCT および HCICSFCT リソース・クラス・プロファイルを使用します。

表 5. デフォルト・リソース・クラスによる外部セキュリティーの指定 (続き)

システム初期設定パラメーター	効果
XPSB=YES	CICS は、PSB アクセス・セキュリティー検査に PCICSPSB および QCICSPSB リソース・クラス・プロファイルを使用します。

もう一つの例として、SEC=YES を指定し、同じ 3 つの関連リソース・クラス・パラメーターで X*name=username* と指定した場合の効果を表 6 に示します。

表 6. ユーザー定義リソース・クラスの外部セキュリティーの指定

システム初期設定パラメーター	効果
SEC=YES	CICS は、完全な RACF セキュリティー・サポートを使用します。
XTRAN=$usrtrn	CICS は、トランザクション接続セキュリティー検査に T$usrtrn および G$usrtrn ユーザー定義リソース・クラス・プロファイルを使用します。
XFCT=$usrfct	CICS は、ファイル・アクセス・セキュリティー検査に F$usrfct および H$usrfct ユーザー定義リソース・クラス・プロファイルを使用します。
XPSB=$usrpsb	CICS は、PSB アクセス・セキュリティー検査に P$usrpsb および Q$usrpsb ユーザー定義リソース・クラス・プロファイルを使用します。

CICS が初期設定されるときに、リソース・プロファイルを主記憶域に組み込んで、システム初期設定パラメーターで指定するすべてのリソース・クラスが一致するように、CICS は RACF に対して要求します。このシステム初期設定パラメーターでは (XAPPC と XDB2 を除き) X*name*=YES がデフォルトであり、CICS は GCICSTRN などのデフォルト・クラス名を使用することに注意してください。X*name*=NO を明示的に指定しないすべてのリソースについて、RACF プロファイルを用意してください。CICS が、存在しない一般リソース・クラスまたは正しく定義されていないリソース・クラスをロードするように RACF に要求する場合、CICS は外部セキュリティー初期設定が失敗したことを示すメッセージを発行し、CICS 初期設定を終了します。

システム初期設定パラメーター XHFS は、このプロセスの例外です。z/OS UNIX ファイルのアクセス制御は RACF によって直接管理されるわけではないため、XHFS=YES が指定されていても、個々の RACF プロファイルを必要としません。z/OS UNIX ファイルのアクセス制御は、z/OS UNIX システム・サービスで指定されます。このサービスは、RACF を利用してユーザー ID とグループを管理しますが、ファイルに設定された許可の制御を保持しています。z/OS UNIX ファイルに対してアクセス制御リスト (ACL) を使用する場合は、RACF クラス FSSEC がアクティブでなければなりません。

外部セキュリティー・システム初期設定パラメーターの構文については、システム初期設定パラメーターの説明と要約を参照してください。

CSD での個々のトランザクション定義によって、トランザクションで使用されるリソースおよびコマンドに RACF セキュリティーを使用するかどうかが決まります。トランザクションのリソース・セキュリティーおよびコマンド・セキュリティーの指定について詳しくは、 72 ページの『CICS ユーザーの検証』および 94 ページの『トランザクション・セキュリティー』を参照してください。

XAPPC、**XHFS**、および **XUSER:**

XAPPC、**XHFS**、および **XUSER** システム初期設定パラメーターの構文は、他の X*name* パラメーターの構文とは少し異なっています。 YES または NO のみを指定できます。

XAPPC=YES は、APPC セッションのセッション・セキュリティーを必要としていることを示します。 XAPPC=YES が指定されているが、APPCLU クラスが RACF でアクティブ化されていない場合、CICS の初期設定は失敗します。詳しくは、セキュリティーに関連する CICS 初期設定の失敗を参照してください。

XAPPC により、RACF LU6.2 バインド時 (APPC とも呼ばれる) セキュリティーが有効になります。詳細については、 275 ページの『LU6.2 でのバインド時のセキュリティー』を参照してください。

詳細については、 277 ページの『APPCLU 一般リソース・クラスでのプロファイルの定義』を参照してください。

XHFS は、CICS 領域における z/OS UNIX ファイルへの Web クライアント・アクセスのアクセス制御をアクティブ化します。詳しくは、 123 ページの『z/OS UNIX ファイルのセキュリティー』を参照してください。

XUSER は、代理ユーザー・セキュリティーと、Db2 に対する AUTHTYPE 検査をアクティブ化します。詳しくは、 154 ページの『代理ユーザー・セキュリティー』を参照してください。XUSER=YES が指定されているが、SURROGAT クラスが RACF でアクティブ化されていない場合、CICS の初期設定は失敗します。

プレフィックス変換のない IBM 提供クラスの使用

プレフィックス変換のない IBM 提供リソース・クラスを使用して、トランザクション、ファイル、および PSB の外部セキュリティーをセットアップするには、このトピックで説明されている手順を実行します。

プロファイルを定義する前に、 32 ページの『RACF コマンドの要約』で説明されているように、SETROPTS CLASSACT コマンドおよび SETROPTS GENERIC コマンドを使用して、関連するクラスをアクティブ化します。

実際のビジネス・プロセスの中断を最小限に抑えるには、まずテスト領域で作業します。

1. 関連するクラスで RACF プロファイルを以下のように計画および作成します。

```
RDEFINE  TCICSTRN  transaction-name  UACC(NONE)  NOTIFY(userid)
RDEFINE  FCICSFCT  file-name         UACC(NONE)  NOTIFY(userid)
RDEFINE  PCICSPSB  PSB-name          UACC(NONE)  NOTIFY(userid)
```

2. 適切なユーザーまたはグループ (グループを推奨) に、プロファイルに対するアクセス権限を持つことを許可します。

```
PERMIT  transaction-name  CLASS(TCICSTRN)  ACCESS(READ)
        ID(userid or groupid)
PERMIT  file-name         CLASS(FCICSFCT)  ACCESS(READ)
        ID(userid or groupid)
PERMIT  PSB-name          CLASS(PCICSPSB)  ACCESS(READ)
        ID(userid or groupid)
```

3. 以下の CICS システム初期設定パラメーターを指定します。

```
SEC=YES           XTRAN=YES        XCMD=NO
SECPRFX=NO        XFCT=YES         XDB2=NO
                  XPSB=YES         XDCT=NO
                                   XHFS=NO
                                   XJCT=NO
                                   XPCT=NO
                                   XPPT=NO
                                   XRES=NO
                                   XTST=NO
                                   XUSER=NO
                                   XAPPC=NO
```

4. 外部セキュリティーを使用する CICS 領域を開始します。
5. 関連するクラスで RACF プロファイルを追加、変更、または削除した場合は、ストレージ内のプロファイルをリフレッシュします。 (詳しくは、 26 ページの『主記憶域内のリソース・プロファイルのリフレッシュ』を参照。)

プレフィックス変換がある IBM 提供クラスの使用

プレフィックス変換がある IBM 提供リソース・クラスを使用して、トランザクション、ファイル、および PSB の外部セキュリティーをセットアップするには、このセクションで説明されている手順を実行します。

プロファイルを定義する前に、 32 ページの『RACF コマンドの要約』で説明されているように、SETROPTS CLASSACT コマンドおよび SETROPTS GENERIC コマンドを使用して、関連するクラスをアクティブ化する必要があります。

実際のビジネス・プロセスで中断を最小限に抑えるには、まずテスト領域で作業します。

注**:** 以下の例では、CICS 領域ユーザー ID が CICS1 であること、さらに SECPRFX=YES であることを想定しています。

1. 関連するクラスで RACF プロファイルを以下のように計画および作成します。

```
RDEFINE  TCICSTRN  CICS1.transaction-name  UACC(NONE)  NOTIFY(userid)
RDEFINE  FCICSFCT  CICS1.file-name         UACC(NONE)  NOTIFY(userid)
RDEFINE  PCICSPSB  CICS1.PSB-name          UACC(NONE)  NOTIFY(userid)
```

2. 適切なユーザーまたはグループ (グループを推奨) に、プロファイルに対するアクセス権限を持つことを許可します。

```
PERMIT  CICS1.transaction-name  CLASS(TCICSTRN)  ACCESS(READ)
        ID(userid or groupid)
PERMIT  CICS1.file-name         CLASS(FCICSFCT)  ACCESS(READ)
        ID(userid or groupid)
PERMIT  CICS1.PSB-name          CLASS(PCICSPSB)  ACCESS(READ)
        ID(userid or groupid)
```

3. 以下のシステム初期設定パラメーターを指定します。

```
SEC=YES             XTRAN=YES        XCMD=NO
SECPRFX=YES         XFCT=YES         XDB2=NO
                    XPSB=YES         XDCT=NO
                                    XHFS=NO
                                     XJCT=NO
```

```
XPCT=NO
XPPT=NO
XRES=NO
XTST=NO
XUSER=NO
XAPPC=NO
```

4. 外部セキュリティーを使用する CICS 領域を開始します。
5. 関連するクラスで RACF プロファイルを追加、変更、または削除した場合は、ストレージ内のプロファイルをリフレッシュします。 (詳しくは、 26 ページの『主記憶域内のリソース・プロファイルのリフレッシュ』を参照。)

接頭部を付けずにインストール定義クラスを使用する

インストール定義クラスのトランザクション、ファイル、および PSB に対して、接頭部を付けずに外部セキュリティーをセットアップするには、このトピックで説明されているステップを実行します。

XTRAN パラメーターのインストール定義クラス (T$USRTRN および G$USRTRN) を定義する方法の例については、CICSTS55.CICS.SDFHSAMP で IBM 提供のサンプル DFH$RACF を参照してください。 182 ページの『ユーザー定義リソースを RACF に指定する』も参照してください。

プロファイルを定義する前に、SETROPTS CLASSACT コマンドおよび SETROPTS GENERIC コマンドを使用して、関連するクラスをアクティブ化します (32 ページの『RACF コマンドの要約』を参照)。

実際のビジネス・プロセスの中断を最小限に抑えるためには、まずテスト領域で作業します。

1. 以下のインストール定義クラスをセットアップします。
   - T$USRTRN (TCICSTRN と同様) および G$USRTRN (GCICSTRN と同様)
   - F$USRFCT (FCICSFCT と同様) および H$USRFCT (HCICSFCT と同様)
   - P$USRPSB (PCICSPSB と同様) および Q$USRPSB (QCICSPSB と同様)

インストール定義クラスのセットアップに関する具体的な情報については、z/OS Security Server RACF システム・プログラマーのガイドを参照してください。

1. 関連するクラスの RACF プロファイルを計画および作成します。

```
RDEFINE  T$USRTRN  transaction-name  UACC(NONE)  NOTIFY(userid)
RDEFINE  F$USRFCT  file-name         UACC(NONE)  NOTIFY(userid)
RDEFINE  P$USRPSB  PSB-name          UACC(NONE)  NOTIFY(userid)
```

2. 適切なユーザーまたはグループ (できればグループ) がプロファイルへのアクセス権限を持つことを許可します。

```
PERMIT  transaction-name  CLASS(T$USRTRN)  ACCESS(READ)
        ID(userid or groupid)
PERMIT  file-name         CLASS(F$USRFCT)  ACCESS(READ)
        ID(userid or groupid)
PERMIT  PSB-name          CLASS(P$USRPSB)  ACCESS(READ)
        ID(userid or groupid)
```

3. 次のシステム初期設定パラメーターを指定します。

```
SEC=YES          XTRAN=$USRTRN        XCMD=NO
SECPRFX=NO       XFCT=$USRFCT         XDB2=NO
                 XPSB=$USRPSB         XDCT=NO
                                      XHFS=NO
```

```
XJCT=NO
XPCT=NO
XPPT=NO
XRES=NO
XTST=NO
XUSER=NO
XAPPC=NO
```

4. 外部セキュリティーを使用する CICS 領域を開始します。
5. 関連するクラスで RACF プロファイルを追加、変更、または削除した場合は、ストレージ内のプロファイルをリフレッシュします。(詳しくは、 26 ページの『主記憶域内のリソース・プロファイルのリフレッシュ』を参照。)

CICS ユーザーの検証

無許可アクセスからリソースを保護するには、システムのユーザーがトランザクションを呼び出す際に、CICS でそれらのユーザーを識別できる必要があります。

CICS 端末ユーザーの識別

RACF セキュリティー検査を備えた CICS を実行する場合は、RACF 管理リソース・プロファイルに定義する許可レベルを通して、CICS リソースへのユーザー・アクセスを制御します。

これらの許可を特定のユーザーに対して定義するには、個々の RACF ユーザー ID (または RACF グループ ID) をリソース・アクセス・リストに追加します。また、サインオンしていないユーザーに対して定義するには、選択したリソース・アクセス・リストにデフォルトの CICS ユーザー ID を追加します。

すべての CICS 端末ユーザー・データは、RACF ユーザー・プロファイルの CICS セグメントに定義されます。CICS 端末ユーザー・データ、および CICS がそのデータを取得する方法について詳しくは、 90 ページの『ユーザーの CICS 関連データの取得』を参照してください。

セキュア・サインオンのためのパスチケットの実装

パスワードの代わりにパスチケットを使用すれば、サインオン・プロセスでアプリケーションがパスワードを保管する (またはユーザーにパスワードの再入力を求める) 必要がなくなるため、ネットワークを介してパスワードを送信する必要はありません。 CICS 環境でセキュア・サインオンのためのパスチケットを実装する場合は、以下の手順に従います。

この資料では、発信システム とはパスチケットが生成されるシステムを指し、宛先システム とはサインオン・ユーザー ID がパスチケットを使用してアクセスを試みたときにそのパスチケットが認証されるシステムを指します。

始める前に

パスチケットを実装するには、関係するシステムが次の要件を満たしている必要があります。

- パスチケットの生成および検証アルゴリズムは、パスチケットを生成するシステムとパスチケットを認証するシステムのどちらも、パスチケットをサポートする一定レベルの外部セキュリティー・マネージャーを使用する必要があることを意味します。
- エンド・ユーザーは、発信システムで使用するユーザー ID と同じものを宛先システムで使用する必要があります。
- パスチケットにはタイム・スタンプが付くため、宛先システムと発信システムのシステム・クロックが有効な時刻範囲内になるように同期化しておく必要があります。パスチケットが有効な時刻範囲にあると見なされるのは、パスチケット生成時刻 (生成するコンピューターでのクロックにおける) が、パスチケット評価時刻 (評価コンピューターでのクロックにおける) のプラスまたはマイナス 10 分以内である場合です。システム時刻の差および同期について詳しくは、『z/OS Security Server RACF Security Administrator's Guide』の『Using the Secured Signon Function』を参照してください。

手順

注: 以下の手順では、RACF が実装で使用される外部セキュリティー・マネージャーであることを前提としています。他の外部セキュリティー・マネージャーのユーザーは、それぞれの製品資料を参照してください。

セキュア・サインオン・キーを定義します。

パスチケットを処理するために、外部セキュリティー・マネージャーは、発信システムと宛先システムによって共用されるセキュア・サインオン・キーを使用します。宛先システムごとにセキュア・サインオン・キーを定義する必要があります。RACF を使用して PTKTDATA リソース・クラスにプロファイルを定義することによりこれを行う方法については、『z/OS Security Server RACF Security Administrator's Guide』の『Using the Secured Signon Function』を参照してください。

RACF プロファイルを定義します。

発信システムでパスチケットを生成できるようにするには、以下のようにします。

注: パスチケットの生成は、それが必要とされる領域でのみ行うことを強くお勧めします。その領域は、システム初期化パラメーター XPTKT=YES (デフォルト) を使用して設定する必要があります。

特定の発信領域にあるすべてのユーザーのプロファイルは、以下のようになります。

```
RDEFINE PTKTDATA IRRPTAUTH.applid.* UACC(UPDATE)
```

特定の発信領域にある特定のユーザーのプロファイルは、以下のようになります。

```
RDEFINE PTKTDATA IRRPTAUTH.applid.* UACC(NONE)
RDEFINE PTKTDATA IRRPTAUTH.applid.userid UACC(UPDATE)
```

applid は、発信領域の汎用アプリケーション ID です。

宛先領域でパスチケットを受け入れるようにするには、以下のようにします。

```
RDEF PTKTDATA applid SSIGNON(KEYMASKED(key)) UACC(NONE)
```

applid は、宛先領域の汎用アプリケーション ID です。

PTKTDATA で **RACLIST** を使用する場合は、定義をリフレッシュします。

`SETR RACLIST(PTKTDATA) REFRESH` を発行します。

同じユーザー ID に対する並列パスチケット要求のサポート

パスチケットは 1 回限りの使用チケットです。生成メカニズムの細分度は 1 秒です。したがって、特定のユーザー ID に対してパスチケットを生成する場合に同じコマンドを 1 秒以内に 2 回発行すると、同じパスチケットが生成されます。このため、パスチケットを使用する要求は、アプリケーション ID ごとに 1 秒当たり 1 つに制限されます。ただし、同じユーザー ID に対する並列パスチケット要求をサポートすることができます。

1 つのアプリケーション ID について、1 ユーザーにつき 1 秒当たり複数の要求を許可するには、ユーザーに対して、グループ別に個別に鍵を定義します。

```
RDEF PTKTDATA applid.group1.userid SSIGNON(KEYMASKED(key1)) UACC(NONE)
RDEF PTKTDATA applid.group2.userid SSIGNON(KEYMASKED(key2)) UACC(NONE)
```

適切な鍵を使用してパスチケットを生成するカスタム・コードを作成する必要があります。

宛先領域でパスチケットを受け入れるようにするには、この領域で、パスワードの認証時に (例えば **SIGNON** または **VERIFY PASSWORD** コマンドで)、鍵に関連付けられたグループを使用する必要があります。グループが指定されていない場合は、デフォルト・グループが使用されます。

FEPI アプリケーションでのパスチケットの使用

FEPI アプリケーションは、パスワードとユーザー ID を送信するかのように、パスチケットとユーザー ID を使用してバックエンド・システムでのサインオンを実行します。

始める前に

パスチケットを処理するために、外部セキュリティー・マネージャーはセキュア・サインオン・キーと呼ばれるキーを使用します。このキーは、フロントエンド・システムとバックエンド・システムが共用します。 FEPI が通信する各ターゲット・システムに対してセキュア・サインオン・キーを定義する必要があります。RACF を使用して PTKTDATA リソース・クラスにプロファイルを定義することによりこれを行う方法については、『z/OS Security Server RACF Security Administrator's Guide』の『Using the Secured Signon Function』を参照してください。その他の ESM ユーザーは、それぞれの製品資料を参照してください。

手順

1. エンド・ユーザーは、通常の方法で、フロントエンド CICS システムへのサインオンによって検査されます。

2. エンド・ユーザーが、FEPI を使用するトランザクションを実行すると、アプリケーションは **FEPI REQUEST PASSTICKET** コマンドを発行してパスチケットを取得します。 EDF が使用されていると、パスチケットは表示されません。パスチケットが生成されるユーザー ID は、現在サイオンしているユーザーのユーザー ID です。FEPI アプリケーションは **EXEC CICS ASSIGN** コマンドを使用して、現在サイオンしているユーザーのユーザー ID を検査できます。
3. FEPI アプリケーションは、パスチケットとユーザー ID を使用してバックエンド・システムでのサインオンを実行します。 例:

```
EXEC CICS FEPI SEND FORMATTED
        CONVID(convid) FROM(CESN userid PassTicket)
        FROMLENGTH(length_of_data)
```

   CICS はバックエンド・システムのタイプ (CICS または IMS) も、サインオン処理に使用されるバックエンド・プログラムも識別できないため、アプリケーション側でサインオン処理の役割を担う必要があります。
4. バックエンド・システムは、未変更のインターフェースを使用して、サインオンを実行します。ユーザー ID とパスチケットを受け取る CICS システムは、既存のプロシージャーを使用してそのユーザー ID をサインオンできます。RACF は、パスワードではなく、パスチケットが渡される場合に対応します。

タスクの結果

パスチケットがタイムアウトになる (セッションが失敗するなどの理由で) 場合、アプリケーションは別のパスチケットを生成して、サインオンを再試行する必要があります。引き続きサインオンが失敗し、フロントエンドとバックエンドが異なる別の MVS システムにある場合、システムクロックが適切に同期されていることを確認します。サインオン試行の失敗回数が多すぎると、そのユーザー ID が取り消される可能性があります。

次のタスク

パスチケットについて詳しくは、『z/OS Security Server RACF Security Administrator's Guide』の『Using the Secured Signon Function』を参照してください。

サインオン・プロセス

ユーザーは z/OS Communications Server を介して CICS にログオンしたけれどもサインオンしていない場合、そのユーザーは、CICS デフォルト・ユーザーが使用を許可されているトランザクションのみを使用できます。

これらは厳密に制限される可能性が高いため、ユーザーは、使用を許可されているトランザクションの実行権限を取得するにはサインオンしなければなりません。

明示的サインオン

明示的サインオンは、CICS 提供のサインオン・トランザクション CESL または CESN の いずれかを使用するか、あるいは **EXEC CICS SIGNON** コマンドを使用するインストール済み環境提供のサインオン・トランザクションを使用して実行できます。

カード・リーダーが DFHOPID ID (AID) をサポートしている場合、OIDCARD ユーザーは CESL または CESN を使用してサインオンできます。サポートしていない場合は、独自のインストール済み環境提供のサインオン・トランザクションを使用します。 CICS にサインオンする場合、サインオン・プロセスには以下のフェーズが含まれます。

有効範囲
: サインオン・パネルを完了して送信すると、CICS は、ユーザーが入力したユーザー ID が、CICS システムの **SNSCOPE** 定義のスコープ内で既にサインオンしているユーザー ID と一致しないことを確認します。

識別
: CICS は RACF を呼び出して、ユーザー ID に対してプロファイルが定義済みであることを確認します。

検査
: CICS は RACF に情報を渡し、ユーザー ID が本物であることを検査します。 RACF の場合、この情報はパスワードまたは OIDCARD (あるいはその両方) です。入力したパスワードの長さが 9 文字から 100 文字の場合、RACF はそれを、ユーザー ID に関連付けられたパスワード・フレーズを使用して検査します。パスワードの長さが 8 文字以下の場合、RACF はそれを、標準パスワードを使用して検査します。ユーザー ID は、標準パスワードとパスワード・フレーズの両方を持つことができます。パスワードまたはパスワード・フレーズの有効期限が切れると、CICS により新規パスワードまたはパスワード・フレーズの入力が求められます。新規パスワードがインストール済み環境の RACF パスワード・フォーマット設定規則に準拠している場合、その新規パスワードまたはパスワード・フレーズとその変更日時が RACF ユーザー・プロファイルに記録されます。

  パスワードとパスワード・フレーズは互いに独立して機能します。標準パスワードの有効期限が切れた場合でも、有効なパスワード・フレーズを使用してサインオンできます。同様に、パスワード・フレーズの有効期限が切れた場合、有効なパスワードを使用してサインオンできます。

  RACF にユーザー ID とパスワードの検査を要求した直後に、CICS は内部パスワード・フィールドをクリアします。これにより、CICS アドレス・スペースのダンプが取られるとしても、ダンプからパスワードまたはパスワード・フレーズが漏えいする可能性が最小限に抑えられます。

  CESL トランザクションと CESN トランザクションは、パスワードまたはパスワード・フレーズの有効期限が切れようとしているとき、ユーザーに警告を出しません。ユーザーに有効期限切れの警告を表示するには、CESL または CESN を、カスタム・サインオン・プログラムを呼び出す独自のトランザクションに置き換える必要があります。このプログラムでは、**EXEC CICS SIGNON** コマンドを使用してユーザー ID の完全検証要求を行う前に、**EXEC CICS VERIFY PASSWORD** コマンドまたは **EXEC CICS VERIFY PHRASE** コマンドを使用して有効期限切れ情報を RACF から返すことができます。

Authorization
: RACF は、アプリケーション名およびエントリー・ポートに対する検査を実行して、ユーザーが CICS システムの使用を許可されていることを確認します。アプリケーション名検査では、RACF は、APPLID または GRNAME システム初期設定パラメーターで指定されたアプリケーションへのアクセスを、ユーザーが許可されているかどうかを判別します。 RACF はこれを行うために、RACF APPL リソース・クラスで定義されている

CICS アプリケーション・プロファイルのアクセス・リストを検査します。(APPL リソース・クラスのプロファイルを定義する方法については、 58 ページの『CICS 領域へのアクセスの許可』を参照してください。)

エントリー・ポート検査を使用して、RACF は、ユーザーがそのエントリー・ポートを使用したサインオンを許可されていることを検査します。定義されている端末の使用は、特定の時刻、および特定の曜日に制限できます。 79 ページの『特定のエントリー・ポートから CICS へのアクセスの制御』を参照してください。

これらの検査により、CICS ユーザーは、使用が許可されている端末のみから、許可されている CICS 領域のみへのサインオンに制限されます。

CESL または CESN トランザクションを使用するか、または SIGNON コマンドを使用する明示的サインオンは、エントリー・ポートで実行されます。

表 7. 明示的サインオンおよび暗黙的サインオン

フェーズ	明示的	暗黙的
有効範囲	はい	いいえ
識別	はい	はい
検査	はい	いいえ (ATTACHSEC (IDENTIFY) を使用した場合を除く)
Authorization	はい	はい

ユーザー属性

CICS は、RACF データベースの CICS セグメントおよび LANGUAGE セグメントから CICS ユーザー属性を取得します。

サインオフ・プロセス

サインオフ・プロセスは、事前にユーザーがサインオンした端末からそのユーザーの関連付けを解除します。ユーザーを明示的にサインオフするには、CESF トランザクションを使用する方法と、SIGNOFF API コマンドを発行するインストール済み環境提供のトランザクションを使用する方法があります。サインオン・ユーザーの属性に TIMEOUT 値としてゼロ以外が含まれる場合は、この端末でトランザクション終了後にこの間隔が経過すると、暗黙的なサインオフが実行されます。

デフォルトの GNTRAN=NO が指定されている場合、タイムアウト期間が満了すると、CICS は直ちにサインオフを実行します。 GNTRAN にはスケジュールされるトランザクション ID が指定されていて、そのトランザクションがサインオフを実行する場合、CICS が実行するアクションは、端末の TYPETERM リソース定義に指定された SIGNOFF オプションによって決まります。

例外的なケースは、CRTE セッションのユーザーにグッドナイト・トランザクションが使用されないことです。時間が有効期限切れになった代理ユーザーはサインオフされ、端末が以前に持っていたセキュリティー機能は失われます。メッセージ DFHSN1200 が CSCS ログに送信され、何が起こったかを示します。

システム初期設定パラメーター GNTRAN の使用について詳しくは、『グッドナイト・トランザクション』を参照してください。指定できるサインオフ・オプションおよび関連アクションは以下のとおりです。

SIGNOFF(YES)
CICS は、オペレーターを CICS からサインオフしますが、端末は接続したままです。

SIGNOFF(LOGOFF)
CICS はオペレーターを CICS からサインオフし、さらに z/OS Communications Server から端末をログオフします。

また、端末が自動インストールされた場合は、システム初期設定パラメーターの AILDELAY オペランドによって指定された遅延期間が開始します。一方、端末が再ログオンを試行する前に遅延期間が満了した場合、CICS は端末エントリー (TCTTE) を TCT から削除します。

SIGNOFF(NO)
CICS はユーザーをサインオン状態のままにし、端末はログオンしたままとなるため、タイムアウト期間は事実上オーバーライドされます。

明示的サインオフ

明示的サインオフを実行すると、ユーザーの有効範囲は除去されます。 CESF トランザクションまたは SIGNOFF コマンドを使用してサインオフするには、その前にユーザーが明示的にサインオンしている必要があります。ユーザーはデフォルトのセキュリティー・レベルに戻ります。

注: CESL パネルまたは CESN パネルを使用するための有効な試行が行われるまで、CESL または CESN はユーザーをサインオフしません。後続のサインオン試行が失敗する場合でもそのようになります。 CESL および CESN をグッドナイト・トランザクションに使用することは推奨されません。

暗黙的サインオンおよび暗黙的サインオフ

暗黙的サインオンとは、CICS によってシステムに追加される他のすべてのユーザー ID が、パスワードおよびパスワード・フレーズなしで暗黙的にサインオンするものと見なされることを意味します。

データを基本機能に送信しようとしているときに、トランザクションが TERMERR 状態になると、ユーザーは暗黙的にサインオフされます。ただし、ユーザーは USRDELAY の対象ではなく、即時にサインオフされます。 SNSCOPE が使用中の場合、スコープはサインオフ時に解放されます。トランザクションが ABEND を処理する場合、開始済みユーザーの権限で非端末タスクとして実行を継続します。

グッドナイト・トランザクション

独自の GNTRAN トランザクションを指定することにより、CICS API を使用して TIMEOUT 操作を制御できます。例えば、トランザクションは、パスワードの入力を求める画面を表示できます。

ASSIGN コマンドで USERID オプションを指定することによりユーザー ID が取得され、**VERIFY PASSWORD** コマンドによってその入力が検証されます。応答に基づいて、ユーザーをサインオンしたままにするか、サインオフすることができます。

デフォルトでは、CICS は CESF トランザクションを使用してユーザー端末をサインオフします。グッドナイト・トランザクションは、CRTE セッション中にタイムアウトになった代理端末には使用できません。サインオフが発生し、端末が以前に持っていたセキュリティー機能は失われて、ログに DFHSN1200 メッセージが記録されます。

特定のエントリー・ポートから CICS へのアクセスの制御

サインオン処理中、CICS は、RACF に要求を発行して、ユーザーのパスワードを検証し、ユーザーがその端末へのアクセスを許可されているかを検査します。

この検査は、事前設定セキュリティー端末定義に指定されたユーザー ID にも実行されます。自動インストールされ、自動サインオンを使用しているコンソールは、事前設定セキュリティー定義があるものとして処理されます (83 ページの『Preset terminal security』を参照)。端末が RACF に定義されていない場合、RACF は、SETROPTS コマンドで指定されたシステム全体の RACF オプションに従って CICS に応答します。オプションは以下のとおりです。

TERMINAL(READ)
このオプションが有効な場合、端末ユーザーは、アクセスを許可されているプロファイルの対象となる端末、または RACF による保護が定義されていない端末で、サインオンすることができます。

TERMINAL(NONE)
このオプションが有効な場合、端末ユーザーは、RACF に定義され、使用を許可されている特定の端末プロファイルを持つ端末でのみ、サインオンすることができます。

注: TERMINAL クラスは、MVS コンソールからのアクセスを制御しません。これらは CONSOLE リソース・クラスによって制御されます。 82 ページの『コンソール・プロファイル』を参照してください。

グループ端末オプション TERMUACC または NOTERMUACC を使用して、システム全体の端末オプションを RACF グループ・レベルでオーバーライドできます。

端末の SETROPTS コマンド、およびグループの TERMUACC|NOTERMUACC オプションについて詳しくは、 81 ページの『未定義端末に対する汎用アクセス権限』を参照してください。

エントリー・ポート・プロファイルの定義

エントリー・ポート は、ユーザーがサインオンする装置の総称です。CICS の場合、エントリー・ポートは端末またはコンソールのいずれかです。関連付けられたエントリー・ポート・プロファイルを使用すると、特定の装置でユーザーがサインオンできるかどうかを制御できます。

端末プロファイル

TERMINAL および GTERMINAL リソース・クラスのプロファイルを使用して、端末へのユーザー・アクセスを制御できます。

このセクションでは、CICS ユーザーにとって関心のある端末プロファイルのいくつかの側面について簡単に説明します。MVS システムでの端末の定義および保護の詳細 (特に以下のトピック) については、z/OS Security Server RACF セキュリティー管理者のガイドを参照してください。

- TERMINAL クラスまたは GTERMINL クラスでプロファイルを作成する
- 未定義の端末の使用を防止する
- 特定のユーザー・グループを特定の端末に制限する
- 端末を使用できる日数または時間を制限する
- セキュリティー・ラベルを使用して端末を制御する

端末へのユーザー・アクセスを制御するには、端末を RACF に定義します。 (ユーザー・アクセスは、CICS サインオン時に決定します。) RACF では、端末に関して IBM が提供する次の 2 つのリソース・クラス名がサポートされます。

TERMINAL
個々の端末のプロファイル定義に使用します。

GTERMINL
端末グループのプロファイル定義に使用します。

注: GTERMINL プロファイルの場合、RACF は常にストレージ内のプロファイルを使用するため、このプロファイルを手動でリフレッシュする必要があります。GTERMINL プロファイルを作成、変更、または削除するたびに、お客様 (または RACF セキュリティー管理者) は SETROPTS RACLIST(TERMINAL) REFRESH コマンドを発行して、変更を有効にしなければなりません。

個々の端末のプロファイルの定義

TERMINAL リソース・クラス内で NETNAME の NETID1、NETID2、および NETID3 を使用して端末を定義するには、次のコマンドを使用します。

```
RDEFINE TERMINAL (NETID1, NETID2, NETID3) UACC(NONE)
        NOTIFY(sys_admin_userid)
```

端末 ID が同じ文字で始まっている場合は、同じイニシャル文字を持つ端末のグループをカバーする総称プロファイルを作成できます。総称プロファイルを定義する前に、 32 ページの『RACF コマンドの要約』で説明されているように、SETROPTS GENERIC コマンドを使用する必要があります。これにより、アクセス・リストを作成するために必要な労力が削減されます。以下に例を挙げます。

```
RDEFINE TERMINAL NETID* UACC(NONE)
        NOTIFY(sys_admin_userid)
PERMIT netid* CLASS(TERMINAL)
       ID(group1, group2,.., groupn) ACCESS(READ)
```

プロファイルのグループのプロファイルの定義

リソース・グループ・クラス内で同じ端末を定義できるようにするには、端末を適切な端末グループのメンバーとして含めます。例えば、次のとおりです。

```
RDEFINE GTERMINL term_groupid
        ADDMEM(NETID1, NETID2, NETID3) UACC(NONE)
        NOTIFY(sys_admin_userid)
```

リソース・グループ・プロファイルから端末を除去するには、RALTER コマンドに DELMEM オペランドを指定します。例えば、以下のような項目が含まれています。

```
RALTER  GTERMINL term_groupid
        DELMEM(NETID3)
```

特定の部門内のユーザー・グループにこれらの端末へのアクセス権限を持つことを許可するには、PERMIT コマンドを以下のように使用します。

```
PERMIT term_groupid CLASS(GTERMINL) ID(dept_groupid) ACCESS(READ)
```

TERMINAL クラスまたは GTERMINAL クラス内のプロファイル

CICS の場合、TERMINAL クラスまたは GTERMINL クラス内で RACF に対して定義する端末プロファイルは、SNA LU に対してのみ使用されます。

このプロファイルの名前は、RDO 端末定義または自動インストールに指定されている NETNAME の値です。 TERMINAL プロファイルを非 SNA LU とともに使用することはできません。

未定義端末に対する汎用アクセス権限

RACF では未定義端末に対する汎用アクセス機能がサポートされており、(必要な権限があれば) SETROPTS TERMINAL コマンドを使用して未定義端末を制御できます。SETROPTS TERMINAL(NONE|READ) が有効になっている場合、その影響はすべての MVS 端末サブシステムに及びます。

SETROPTS TERMINAL(READ) が有効になっている場合、RACF では、どのユーザーがどの未定義端末 (つまり、TERMINAL または GTERMINL リソース・クラスに定義されていない端末) でログオンすることも許可されます。SETROPTS TERMINAL(NONE) が有効になっている場合、RACF では、どのユーザーがどの未定義端末でログオンすることも許可されません。

注**:** SETROPTS TERMINAL(NONE) コマンドを発行する前に、いくつかの TERMINAL または GTERMINL クラス・プロファイルを定義し、十分な権限を指定して、少なくともいくつかの端末を使用できるようにします。そうしないと、だれも端末にアクセスできなくなります。

SETROPTS TERMINAL コマンドのオーバーライド**:**

ADDGROUP コマンドまたは ALTGROUP コマンドに TERMUACC オプションまたは NOTERMUACC オプションを指定することで、グループ・レベルで SETROPTS TERMINAL コマンドをオーバーライドできます。

TERMUACC パラメーターの効果は、汎用アクセス・オプションを適用することです。例えば、SETROPTS TERMINAL(READ) がアクティブの場合、TERMUACC オプションは、グループ内のどのユーザーでも未定義の端末にアクセスできることを意味します。ただし、グループに対して NOTERMUACC を指定した場合、SETROPTS TERMINAL コマンドはそのグループに対して効果を持たず、グループ内のユーザーは端末を使用するための明示的な許可が必要です。 NOTERMUACC オプションが指定されたグループが端末にアクセスできるようにするには、グループのユーザー ID を適切な TERMINAL プロファイルまたは GTERMINL プロファイルのアクセス・リストに追加する必要があります。

条件付きアクセス処理

RACF を使用すると、ユーザーが特定の端末またはコンソールにサインオンしているときはそのユーザーにリソースへのアクセスを許可するが、それ以外の場合は許可しないようにすることができます。このように制限されるアクセスは、条件付きアクセス と呼ばれます。

条件付きアクセス権限をリソースに付与するには、

```
WHEN(TERMINAL(netname))
```

または

```
WHEN(CONSOLE(console-name))
```

を PERMIT コマンドに追加します。

次の例では、PAYROLL グループのメンバーは、どこにサインオンしていても SALARY ファイルを読み取ることができます。以下のコマンドを発行することで、ネット名が PAY001 の端末からのみこのファイルを更新できます。

```
RDEFINE FCICSFCT SALARY UACC(NONE)
PERMIT SALARY CLASS(FCICSFCT) ID(PAYROLL) ACCESS(READ)
PERMIT SALARY CLASS(FCICSFCT) ID(PAYROLL)
       (WHEN(TERMINAL(PAY001)) ACCESS(UPDATE)
```

操作グループ OPS のメンバーがコンソール名 MVS1MAST からのみ CEMT トランザクションを使用できるようにするには、以下のコマンドを発行します。

```
RDEFINE TCICSTRN CEMT UACC(NONE)
PERMIT CEMT CLASS(TCICSTRN) ID(OPS) WHEN(CONSOLE(MVS1MAST)) AC(READ)
```

注:

1. CONSOLE 条件付きアクセス・リストを使用するには、CONSOLE クラスがアクティブでなければなりません。
2. 条件付きアクセス・リストは権限の拡大のみを行うものであり、権限を縮小することはできません。

   条件付きアクセス・リストに関するその他の考慮事項については、z/OS Security Server RACF セキュリティー管理者のガイドを参照してください。

コンソール・プロファイル

コンソールへのユーザー・アクセスを制御するプロファイルを定義するには、CONSOLE リソース・クラスを使用します。

CONSOLE クラスがアクティブ化されている場合は、ユーザーがコンソールへのサインオンを許可されるかどうかを制御できます。コンソール保護は、端末の保護と同様の方法で実施されます。ただし、 81 ページの『SETROPTS TERMINAL コマンドのオーバーライド』で説明されている以下の例外を除きます。

1. SETROPTS TERMINAL コマンドはコンソールに適用されません。
2. TERMUACC グループ属性はコンソールに適用されません。

CONSOLE クラスをアクティブ化する前に、MVS コンソールに対するコンソール保護の効果について、z/OS MVS 計画: 操作 を確認してください。

コンソール・クラスで使用されるプロファイルは、コンソールの名前または番号です。以下に例を示します。

```
RDEFINE CONSOLE CICSCONS UACC(NONE)
```

ユーザーは、コンソールでサインオンするには、コンソール名の READ 権限が必要です。以下の例は、ユーザー CICSOPR が、CONCICS1 という名前のコンソールへのサインオンをどのように許可されるかを示しています。

```
RDEFINE CONSOLE CONCICS1 UACC(NONE)
PERMIT CONCICS1 CLASS(CONSOLE) ID(CICSOPR) ACCESS(READ)
```

TERMINAL 保護の場合と異なり、サインオンの試行は、アクティブ化された CONSOLE クラス内で定義されていないコンソールで行われると失敗することに注意してください。未定義のコンソールへのアクセス権限は NONE です。

サインオンおよびサインオフ・アクティビティーの監査

RACF は、無効または失敗したサインオン試行を含む、SMF へのすべてのサインオンおよびサインオフ・アクティビティーをログに記録することができます。この情報はさまざまな方法で使用できます。例えば、監査証跡として、セキュリティー侵害の可能性がある試行を突き止めたり、キャパシティー・プランニングに役立てたりできます。

失敗したサインオン試行のロギングを正しく解釈するには、成功したサインオンも記録しなければなりません。例えば、ユーザーが試行を 1、2 回失敗した直後にサインオンに成功した場合、失敗したサインオンは、端末での誤ったキー入力が原因であると解釈できます。ただし、短時間にさまざまなユーザー ID で試行が何回か失敗し、それ以降は成功したサインオン・アクティビティーが記録されていない場合は、調査を必要とするセキュリティー上の問題である可能性があります。

成功したサインオンおよびサインオフ・アクティビティーを記録することで、端末ユーザー母集団による特定システムへのアクセスの監査証跡が作成されます。これはシステムのキャパシティー・プランニングにも役立つ可能性があり、通常は、SMF に記録される情報のほんのわずかな部分を占めています。

CICS は、セキュリティー・メッセージ用に CSCS 一時データ宛先を使用します。CICS 領域のセキュリティー管理者にとって関心のあるメッセージは、この宛先に送信されます。場合によっては、セキュリティー関連のメッセージが端末ユーザーに送信されると、対応するメッセージが CSCS 一時データ宛先に書き込まれます。例えば、端末ユーザーに送信される DFHCE3544 メッセージおよび DFHCE3545 メッセージの場合は、対応するメッセージ DFHSN1118 および DFHSN1119 が CSCS に送信されます。DFHSNxxxx メッセージには、無効なサインオン試行の正確な性質を示す理由コードが含まれます。

Preset terminal security

一部の端末、および CICS 端末として使用される場合は MVS コンソールでは、端末ユーザー・セキュリティーの代わりとして、事前設定端末セキュリティーを使用するのが適切です。

TERMINAL リソース定義で USERID 属性を指定すると、端末は事前設定セキュリティー端末になります。

コンソールには、次の 2 種類の事前設定セキュリティーがあります。

1. 通常の事前設定セキュリティー (他の端末の事前設定セキュリティーと同じ)
2. 自動事前設定セキュリティー

通常の事前設定セキュリティー

CICS 事前設定端末セキュリティーにより、ユーザー ID を、CICS に定義されている端末またはコンソールに永続的に関連付けることができます。

これは、CICS がデバイスの設置時にそのデバイスに暗黙的にサインオンするということです。これはその後のユーザーによるその端末へのサインオンに代わるものになります。一般に、事前設定セキュリティーは、プリンターなどの、キーボードを持たず、ユーザーがサインオンできないデバイスに対して定義されます。

さらに、通常の事前設定セキュリティーは、端末ユーザー・セキュリティーに代わるものとして、普通のディスプレイ端末で使用することもできます。これにより、事前設定セキュリティーが設定された端末に物理的にアクセスできるユーザーは誰でも、その端末に許可されているトランザクションに入ることが許可されます。端末は設置されている限りサインオンの状態を維持し、端末に対して明示的なサインオフを実行することはできません。事前設定セキュリティーが設定されたディスプレイ端末に関連付けられているユーザー ID で、機密トランザクションの使用が許可されている場合は、必ず、アクセスが制限されている安全な場所に端末を設置してください。事前設定セキュリティーは、例えば、CICS ネットワーク・コントロール・センター内に物理的に設置されている端末に適しています。

事前設定セキュリティーを使用して、立ち入り制限がないエリアに設置された端末に対して、デフォルトよりも低い権限を持つユーザー ID を割り当てることができます。

例えば、端末を事前設定セキュリティーで定義する場合は、RACF コマンドと CICS (CEDA) コマンドを以下のように使用します。

```
ADDUSER userid NAME(preset_terminal_user_name) OWNER(owner_userid or group_id)
               DFLTGRP(group_name)
CEDA DEFINE TERMINAL(cics_termid) NETNAME(vtam_termid) USERID(userid)
            TYPETERM(cics_typeterm)
```

注: VTAM は現在のところ、z/OS Communications Server (for SNA または IP) です。

トランザクション・ルーティング環境内の事前設定セキュリティー端末の詳細については、 292 ページの『事前設定セキュリティー端末およびトランザクション・ルーティング』を参照してください。

コンソールの自動事前設定セキュリティー

自動事前設定セキュリティーは、コンソール定義にのみ適用されます。 CICS 自動事前設定セキュリティーを使用すると、MVS が RACF を介して検査済みのユーザー ID を、コンソールの CICS 定義に関連付けることができます。

TERMINAL 定義に実際のユーザー ID を指定する代わりに、特殊値 (*FIRST または *EVERY) を指定して、MVS によって MODIFY コマンドで渡されたユーザー ID を使用するように CICS に指示します。これは、CICS がコンソールのインスト

ール時、およびオプションで各入力メッセージの発行時に、そのコンソールに暗黙的にサインオンするということです。これはその後のユーザーによるそのコンソールへのサインオンに代わるものになります。これにより、特に自動インストール済みコンソールの場合、ユーザーは CICS 端末定義でユーザー ID とコンソールの関係を定義する必要なしに、事前設定セキュリティーのメリットを得ることができます。したがって、コンソール・ユーザーは各 CICS 領域に、平文パスワードを使用してサインオンする必要はありません。

この自動形式の事前設定セキュリティーは、事前定義されたコンソール、自動インストールされたコンソール、および **CREATE TERMINAL** コマンドを使用してインストールされたコンソールで使用できます。

例えば、検査済みであり、必要な場合にはすべての MODIFY で変更されたコンソールを、自動事前設定セキュリティーを使用して定義するには、CICS (CEDA) コマンドを次のように使用します。

```
CEDA DEFINE TERMINAL(cics_termid)
            CONSNAME(consolename) USERID(*EVERY)
            TYPETERM(cics_typeterm)
```

最初の有効な MODIFY コマンドのみで定義されているコンソールを、自動事前設定セキュリティーを使用して定義するには、CICS (CEDA) コマンドを次のように使用します。

```
CEDA DEFINE TERMINAL(cics_termid)
            CONSNAME(consolename) USERID(*FIRST)
            TYPETERM(cics_typeterm)
```

事前設定セキュリティーの使用の制御

事前設定セキュリティー端末がインストールされると、指定されたユーザー ID がその端末で暗黙的にサインオンされます。

端末に指定されたユーザー ID は、インストールの実行者が使用できない CICS リソースにアクセスできる可能性があるため、事前設定セキュリティーを備えた端末の定義およびインストールは、信頼できるユーザーのみが許可されるようにしてください。コンソール・ユーザーは (RACF によって検証された) 身元の情報に関連付けられているため、コンソールの自動事前設定セキュリティーでは同じようなリスクを伴いません。このため、USERID(*EVERY) または USERID(*FIRST) のいずれかを使用してコンソール装置が CICS に定義された場合、検査は実行されません。

代理ユーザー検査により、ユーザーが他のユーザーの代理をする権限があるかを確認します。ユーザーが別のユーザー ID 用に事前設定された端末をインストールするときに、その端末が RACF SURROGAT リソース・クラスによって指定されている場合は、代理ユーザー検査を実施できます。SURROGAT リソース・クラスに CICS *userid*.DFHINSTL リソースを定義すると、その特定ユーザー ID 用に事前設定された端末のインストールを許可できます。

事前設定されたユーザー ID を持つ端末がインストールされる場合、代理ユーザーは、インストールを実行しているユーザー ID です。詳細については、 154 ページの『代理ユーザー・セキュリティー』を参照してください。

CEDA コマンドは、事前設定された端末をインストールするためにユーザーの権限を検査します。そのため、事前設定セキュリティーを備えた端末を定義およびインストールできるユーザーを制御する目的で以下の機能を制限するかどうかを検討してください。

- CEDA トランザクション:

  CEDA トランザクションの使用を許可ユーザーに制限すると、端末などのリソースを CICS に定義できるユーザーを制御できるようになります。CICS 提供トランザクションの保護については、 186 ページの『CICS トランザクションのセキュリティー』を参照してください。
- SURROGAT リソース・クラス:

  事前設定セキュリティーを備えた端末をインストールできるユーザーを制限すると、該当する端末が CSD に定義されていても、その端末を CICS にインストールできるのは許可ユーザーだけです。この権限は CEDA の実行に必要な権限に追加されるものです。ユーザーは CEDA トランザクションの実行権限を既に持っていなければなりません。

  代理プロファイルを定義し、事前設定セキュリティーを備えた端末定義のインストール権限をユーザーに付与するには、以下のコマンドを使用します。

  ```
  RDEFINE userid1.DFHINSTL SURROGAT UACC(NONE)
  PERMIT userid1.DFHINSTL CLASS(SURROGAT) ID(userid2) ACCESS(READ)
  ```

  これにより、*userid2* は *userid1* で事前設定された端末をインストールできます。
- XUSER システム初期設定パラメーター:

  事前設定セキュリティーを備えた端末に対する CEDA INSTALL コマンドの使用に関して CICS が代理ユーザー・セキュリティー検査を実行できるようにするには、**XUSER** システム初期設定パラメーターを定義します。このパラメーターの定義について詳しくは、 66 ページの『CICS リソース・クラスのシステム初期設定パラメーター』を参照してください。
- DFHCSDUP ユーティリティーを使用した CSD へのバッチ・アクセス:

  CSD バッチ・ユーティリティー・プログラム DFHCSDUP を使用して、CSD にリソースを定義することができます。許可ユーザーのみが実動 CSD を更新できるようにするため、CSD データ・セット・プロファイルのアクセス・リストを CICS 領域ユーザー ID およびその他の許可ユーザーのみに制限してください。DFHCSDUP では INSTALL コマンドは使用できません。
- CSD 定義をロックするための LOCK コマンド

  CICS は、初期始動時またはコールド・スタート時に、**GRPLIST** システム初期設定パラメーターに定義されているグループのリストから、リソース定義を CSD にインストールします。CICS 始動グループ・リストへのリソース・グループの追加を制御するには、CEDA LOCK コマンドを使用してリストをロックします。このロックによりグループ・リストは、許可されていない追加操作から保護されます。また、このリストに指定されているすべてのグループをロックします。

注: サインオンしているユーザーの OPIDENT は、CEDA LOCK コマンドおよび CEDA UNLOCK コマンドのキーとして使用されます。LOCK コマンドと UNLOCK コマンドについて詳しくは、および を参照してください。

注: CICS が事前設定端末の定義を含む GRPLIST をインストールする場合、初期設定時に検査は行われません。ただし、CEDA LOCK コマンドを使用して GRPLIST グループの内容を制御することにより、事前設定セキュリティーを備えた端末やセッションを定義およびインストールできるユーザーを確実に制御できます。

事前設定セキュリティーに関するその他の考慮事項

事前設定セキュリティーを使用する場合は、いくつかの追加の考慮事項に注意する必要があります。

- 自動インストール・モデル:

  自動インストール・モデルを事前設定セキュリティーと共に使用している場合、CICS はモデルのインストール時に通常の端末と同じ代理許可検査を行います。自動インストール・モデルを使用して装置の自動インストールが実行された場合、代理許可は検査されません。コンソール用の自動事前設定セキュリティーを使用して定義されたモデルをインストールする場合も、CICS は代理許可検査を行いません。

  事前設定されたユーザー ID を持つ自動インストール・モデルが無効になった場合 (例えば、ユーザー ID が取り消された場合)、モデルによる端末のインストール試行は失敗します。
- 事前設定セキュリティーによるセッション:

  セッション定義でユーザー ID オペランドを指定した場合、セッションは事前設定セキュリティーによって制御されます。事前設定セキュリティー・セッションをインストールすると、同じ検査が実行されます。
- TCT に定義された端末:

  DFHTCT マクロにより端末管理テーブル (TCT) に定義された端末 (BSAM 端末など) の場合は、ユーザー ID も TCT に定義され、CICS の初期設定時にそのユーザー ID がこれらの端末にサインオンします。サインオンが失敗した場合 (例えば、ユーザー ID が取り消された場合)、端末はサービス休止になります。ユーザー ID が後で有効になった場合 (例えば、再開された場合) は、端末をサービス中に設定すると、正常にサインオンされます。CICS は、これらの端末の代理ユーザー検査を実行しません。

MVS システム・コンソールを CICS 端末として使用する

CICS 端末として MVS システム・コンソールを使用する予定の場合、OPERCMDS リソース・クラスを使って MVS MODIFY コマンドを使用する許可が必要になる場合があります。

コンソールの CICS 端末定義で自動事前設定セキュリティーを指定できるため、コンソール・ユーザーは、(パスワードを公開する) CICS サインオンを明示的に実行することなく、正しいレベルの権限を取得します。

事前設定セキュリティーが定義されていない場合、コンソール・ユーザーは、デフォルト・ユーザーとは異なる権限を取得するためにサインオンする必要がありま

す。この場合、パスワードまたはパスワード・フレーズは通常はコンソールおよびシステム・ログで確認できます。ただし、CICS が JES2 システムの MVS サブシステムとして定義されている場合は、SYS1.PARMLIB の DFHSSI*xx* メンバーの HIDEPASSWORD=YES オプションを使用でき、これにより CICS はコマンドを代行受信し、パスワードまたはパスワード・フレーズをアスタリスクで上書きします。CICS を MVS サブシステムとして定義する方法について詳しくは、CICS の MVS サブシステムとしての定義を参照してください。

CESL または CESN コマンドを使用して、コンソールから CICS にサインオンすることができます。CESL では、許可として標準パスワードまたはパスワード・フレーズを使用できます。CESL は、8 文字を超えるパスワードをパスワード・フレーズとして処理します。CESN は、パスワード・フレーズの使用をサポートしていません。コンソールから入力した場合の CESL および CESN サインオン・コマンドのフォーマットは、以下のとおりです。

```
MODIFY jobname,command_name [USERID=userid][,PS=password]
       [,NEWPS=newpassword][,GROUPID=groupid]
       [,LANGUAGE=language-code]
```

パスフレーズに大/小文字混合文字、ブランク、またはその両方が含まれている場合は、PS パラメーターの値を単一引用符で囲む必要があります。以下に例を示します。

```
F JATP3250,CESL USERID=JAT232,PS='MOND July 17th'
```

MODIFY コマンドの最大長は、F *taskname* を含めて 126 文字です。パスフレーズの最大長は、次のように指定されている場合は 91 文字です。

```
F JATP3250,CESL USERID=JAT284,PS='This is a valid long passphrase of
length 91 chars for CESL JAT testing 9XYZ@,#,.12345678!i'
```

詳しくは、z/OS でコンソール・コマンドを発行する場合の指針を参照してください。

CESL では、PS (パスワード) パラメーターと NEWPS (新規パスワード) パラメーターは一致しなければなりません。それらは、両方ともパスワード・フレーズであるか、両方とも標準パスワードでなければなりません。パスワードを使用した新規パスワード・フレーズを許可することはできず、パスワード・フレーズを使用して新規パスワードを許可することもできません。 PS パラメーターと NEWPS パラメーターではスペースや句読文字 (単一引用符を含む) を使用できますが、これらの文字はそれぞれ 2 つの単一引用符で囲む必要があります。

コマンドで入力されたデータのいずれかが無効の場合、あるいは、パスワードまたはパスワード・フレーズが欠落しているか有効期限切れの場合、CICS はサインオン試行を終了します。

TSO ユーザーに TSO CONSOLE コマンドの使用権限を付与できます。(このコマンドについて詳しくは、z/OS TSO/E システム・プログラミング コマンド解説書を参照。) これらのユーザーは、DEFINE TERMINAL コマンドの CONSNAME オプションを使用して CICS にコンソールとして定義されているか、あるいはコンソールの自動インストールによってサポートされていなければなりません。詳しくは、Autoinstalling MVS consolesを参照してください。

CESL または CESN コマンドから PS パラメーターを省略すると、RACF はセキュリティー違反メッセージ ICH408I を生成する可能性があります。CESL および CESN は、OIDCARD、NOPASSWORD を使用して定義されたユーザーと、PASSWORD オーセンティケーターを使用して定義されたが意図的にパスワードを省略したユーザーとを区別できません。パスワードの入力を求めるプロンプトを出すか、サインオンを拒否する (OIDCARD で定義されたユーザーはコンソールでサインオンできない) かを確認するには、サインオンを試行する必要があります。サインオンが失敗した場合、メッセージ ICH408I が発行され、CICS は RACF からの戻りコードを解釈して、PASSWORD または OIDCARD のいずれのオーセンティケーターが必要かを判別します。

ユーザーは、CESL または CESN を使用してサインオンできます。あるいは、事前設定セキュリティー (CICS 端末の通常の事前設定セキュリティー、またはコンソールの自動事前設定セキュリティー) を使用することもできます。TSO ユーザーが CONSOLE コマンドを使用する場合は、そのユーザーのユーザー ID がデフォルトでコンソール名になります。ただし、TSO CONSOLE コマンドで CONSNAME(*name*) オプションを使用して、コンソール名を他の名前に変更することができます。このコンソール名は、その後、CICS の CONSNAME オプションで対応する TERMINAL 定義がある場合 (あるいは、端末定義を自動インストールする場合) に、CICS 端末として使用できます。別の名前が指定されている場合、その名前は、CICS がコンソールとの通信に使用する名前です。例えば、ある TSO ユーザーが別の TSO ユーザーの ID と同じ名前を使用することが可能です。

また、CONSOLE コマンドを使用して TSO オペレーターが CESL または CESN トランザクションで CICS にサインオンできるようにする場合、それらのパスワードは TSO 画面や MVS システム・ログで公開される可能性があります。事前設定セキュリティーを保有するように端末を定義すると、これらの潜在的な公開を防ぐことができます。以下の理由により、自動化された事前設定セキュリティーを使用することをお勧めします。

- TSO ユーザーはサインオンする必要はありません。サインオンしないことで、自分の ID やパスワードがログに公開されるのを回避できます。
- CICS 定義でコンソール名とユーザーの間の関係を定義する必要はありません。関係は頻繁に変化したり、無効になったりする可能性があります。
- コンソール定義のほとんどをカバーし、各ユーザーに正しいレベルの事前設定セキュリティーを提供する自動インストール・モデルを 1 つ定義することができます。

自動事前設定セキュリティーを定義するには、USERID(*EVERY) を指定することで、コマンドが発行されるたびに正しいユーザー ID でサインオンされるようにします。あるいは、USERID(*FIRST) を指定することで、CICS に MVS MODIFY コマンドを最初に発行するユーザー ID を使用してコンソールにサインオンし、これを保持してその後のコマンドで使用するようにします。

- コンソールの使用を、RACF を使用する CICS と似たセキュリティー特性を持つ 1 人以上のユーザーに制限し、ユーザー ID をアプリケーション内の ID として使用しない場合は、USERID(*FIRST) を選択します。
- 各入力要求をテストして、コンソール・ユーザーが正しいセキュリティー・レベルを保持するようにしなければならない場合は、USERID(*EVERY) を使用します。ユーザー ID を検査すると MODIFY でオーバーヘッドがかかり、事前設定

ユーザー ID を変更すると、コンソール・ユーザーが CESL または CESN を使用してサインオンするのと同等のオーバーヘッドがさらにかかるため、注意してください。

ユーザーの CICS 関連データの取得

CICS は、CICS 関連データを次のいずれかのソース、すなわち RACF プロファイルの CICS および LANGUAGE セグメントか、または組み込み CICS システム・デフォルト値から取得します。このセクションでは、デフォルト・ユーザーおよびサインオンする端末ユーザーのデータが取得される方法を説明します。

デフォルト・ユーザーの CICS 関連データの取得

初期設定時に CICS デフォルト・ユーザーを暗黙的にサインオンする場合、CICS は以下の方法で属性を取得します。

1. CICS は RACF を呼び出して、CICS セグメントおよび LANGUAGE セグメントからの CICS デフォルト・ユーザーのユーザー・データを要求します。デフォルト・ユーザー ID の CICS セグメント・データまたは LANGUAGE セグメント・データがある場合、RACF はこのデータを CICS に返します。CICS セグメントで定義できる情報の詳細については、 17 ページの『CICS セグメント』を参照してください。 LANGUAGE セグメントの詳細については、 22 ページの『LANGUAGE セグメント』を参照してください。
2. RACF がデフォルト・ユーザー ID の CICS セグメント・データも LANGUAGE セグメント・データも返さない場合、CICS は以下の組み込みシステム・デフォルト値を割り当てます。

   各国語
   : NATLANG システム初期設定パラメーターの最初のオペランドから取得されます。指定されない場合、これはデフォルトで米国英語になります。

   オペレーター・クラス
   : 1 (OPCLASS=1)

   オペレーター **ID**
   : ブランク (OPIDENT=' ')

   オペレーター優先順位
   : ゼロ (OPPRTY=0)

   Timeout
   : ゼロ (TIMEOUT=0)

   XRF サインオフ
   : 強制サインオフなし (XRFSOFF=NOFORCE)

サインオン時の CICS 関連データの取得

CICS 端末ユーザーの明示的サインオンを処理する場合、CICS は以下の方法で端末ユーザー属性を取得します。

1. CICS は RACF を呼び出して、CICS セグメントおよび LANGUAGE セグメントからの CICS 端末ユーザーに関するデータを要求します。端末ユーザーの CICS セグメント・データまたは LANGUAGE セグメント・データがある場合、RACF はこのデータを CICS に返します。CICS セグメントで定義できる

情報の詳細については、CICS セグメントを参照してください。 LANGUAGE セグメントの詳細については、LANGUAGE セグメントを参照してください。

2. RACF がユーザーの CICS セグメント・データも LANGUAGE セグメント・データも返さない場合、CICS は、システム初期設定中に定義された CICS デフォルト・ユーザーのユーザー属性を使用します。(デフォルト・ユーザーの CICS 関連データの取得を参照してください。)

CICS は、各国語属性を以下の順序で取得します。

1. CICS 提供の CESN トランザクションでの LANGUAGE オプション。あるいは、**SIGNON** コマンドの LANGUAGECODE または NATLANG オプション (CICS によってサポートされる場合)。サポートされる各国語とは、**NATLANG** システム初期設定パラメーターに指定されていて、対応するメッセージ定義が用意されている有効な各国語のことです。
2. ユーザーの RACF プロファイルの LANGUAGE セグメント内の PRIMARY *primary-language* パラメーター (CICS によってサポートされる場合)。
3. ユーザーの RACF プロファイルの LANGUAGE セグメント内の SECONDARY *secondary-language* パラメーター (CICS によってサポートされる場合)。
4. ユーザーの端末の CSD 定義内にある **NATLANG** パラメーター。
5. デフォルト・ユーザー用に設定された言語 (デフォルト・ユーザーの CICS 関連データの取得を参照)。

有効な各国語のリストについては、各国語コードを参照してください。

注**:** CICS は、以下のコマンドによって定義された RACF のデフォルト言語を無視します。

```
SETROPTS LANGUAGE(PRIMARY(...) SECONDARY(...))
```

RACF への端末ユーザーおよびユーザー・グループの定義**:**

CICS 端末ユーザーをグループで定義する計画を立てる必要があります。

この目的で、CICS システムのユーザーを管理しやすいようにグループに分類してみてください。例えば、同じマネージャーを持つすべてのユーザー、または 1 つの受注機能内のすべてのユーザーは、管理単位であると考えることができます。このようなユーザーを、CICS システム・リソースへのアクセス要件が類似している個々のユーザーのグループとして、RACF に定義できます。以下について詳しくは、z/OS Security Server RACF セキュリティー管理者のガイドを参照してください。

- システム管理者のアクセス制御および操作の柔軟性
- グループ SPECIAL 属性の使用およびその制御範囲
- ストレージ内のプロファイルをリフレッシュする必要性の低減

グループを定義し、次にそのグループのメンバーとしてユーザーを定義すると、グループ内のすべてのユーザーは、グループがアクセス権限を付与されたリソースにアクセスできます。

選択するグループ構造は、お客様のインストール済み環境の要件によって異なります。新規グループを作成するには、以下の RACF コマンド ADDGROUP を使用します。

```
ADDGROUP groupname OWNER(userid)
```

新規ユーザーをグループに追加し、そのユーザーのデフォルト・グループとしてグループ名を定義するには、以下の ADDUSER コマンドを使用します。

```
ADDUSER userid NAME(username) DFLTGRP(group_id)
        CICS(OPCLASS(1,2,..,n) OPIDENT(abc) OPPRTY(255) TIMEOUT(minutes)
             XRFSOFF(NOFORCE) LANGUAGE(PRIMARY(language))
```

端末ユーザーを複数のグループのメンバーにすることができます。それには以下の CONNECT コマンドを使用して、ユーザーを、そのユーザーのデフォルト・グループ以外のグループに追加します。

```
CONNECT userid GROUP(groupname)
```

ユーザーのデフォルト・グループを変更するには、以下のように ALTUSER コマンドを使用します。

```
ALTUSER userid DFLTGRP(groupname)
```

既存のユーザー ID の CICS データを追加するには、ALTUSER コマンドを使用します。CICS オプション・データの詳細については、 17 ページの『CICS セグメント』を参照してください。

これらのコマンドの完全な構文については、z/OS Security Server RACF コマンド言語解説書を参照してください。

RACF に対する端末ユーザーとユーザー・グループの定義の例**:** 次を担当するカスタマー・サービス部門があるとします。

- 注文を受ける
- それらの注文に関する問い合わせに回答する
- 新規顧客をつなぎ止める

次のようなカスタマー・サービス・グループを作成することを考慮します。

```
ADDGROUP custserv OWNER(grpmangr)
```

この例で、*grpmangr* は、カスタマー・サービス部門システムを担当する人物の RACF ユーザー ID です。

grpmangr によって表される人物、または RACF セキュリティー管理者は次に、グループ CUSTSERV 内に次のようにして追加のグループを作成できます。

```
ADDGROUP ORDERS OWNER(SUP1) SUPGROUP(CUSTSERV)
ADDGROUP ORDINQ OWNER(SUP2) SUPGROUP(CUSTSERV)
ADDGROUP NEWCUST OWNER(SUP3) SUPGROUP(CUSTSERV)
```

グループ所有者、*grpmangr* によって表される人物、または RACF セキュリティー管理者は次に、グループ内のユーザーを定義できます。例えば、SUP1 によって表される人物は、グループ ORDERS のユーザーを次のように定義できます。

```
ADDUSER AARCHER NAME('ANNE ARCHER') DFLTGRP(ORDERS)
ADDUSER JBRACER NAME('JOHN BRACER') DFLTGRP(ORDERS) PASSWORD(XPRDTD)
           CICS(OPCLASS(1) OPIDENT(JBR) OPPRTY(0) TIMEOUT(15) XRFSOFF(FORCE))
           LANGUAGE(PRIMARY(ENU))
```

注:

1. ユーザー Anne Archer のパスワードのデフォルトは ORDERS ですが、ユーザー John Bracer のパスワードは最初は XPRDTD と設定されます。
2. ユーザー John Bracer は、CICS セグメントおよび LANGUAGE セグメントを使用して定義されます。

大/小文字混合パスワードのサポート

大/小文字混合パスワードのサポートは、ご使用の外部セキュリティー・マネージャーによって異なります。パスワード・フレーズ (長さが 9 から 100 文字までのパスワード) をサポートするすべてのセキュリティー・マネージャーで大/小文字混合がサポートされます。しかし、標準パスワード (長さが 8 文字までのパスワード) をサポートするすべてのセキュリティー・マネージャーで大/小文字混合がサポートされるわけではありません。

CICS で使用されるセキュリティー・マネージャーが大/小文字混合の標準パスワードの使用をサポートする場合 (例えば z/OS Security Server (RACF) for z/OS 1.7 など)、CICS はセキュリティー・マネージャーにパスワードを渡す前に、パスワードを大文字に変換しません。

パスワードを入力するには、2 つのサインオン・トランザクション CESL および CESN のいずれか、または次の API コマンドのいずれかを使用します。

CHANGE PASSWORD
CHANGE PHRASE
VERIFY PASSWORD
VERIFY PHRASE
SIGNON

CESL では、パスワード・フレーズおよび標準パスワードがサポートされます。CESN では、標準パスワードのみがサポートされます。これらのトランザクションには、パスワードを入力できるフィールドが 2 つあります。

パスワード
新規パスワード

CICS で使用される外部セキュリティー・マネージャーが大/小文字混合をサポートするかどうかに応じて、CICS は 2 とおりの方法でパスワードを処理します。

- セキュリティー・マネージャーが大/小文字混合をサポートする場合、CICS はパスワードをそのままセキュリティー・マネージャーに渡します。
- 大/小文字混合をサポートしない場合、CICS は、パスワードをセキュリティー・マネージャーに渡す前にパスワードを大文字に変換します。

大/小文字混合パスワードのサポートをオンにする方法については、32 ページの『RACF コマンドの要約』を参照してください。

各国語および非端末トランザクション

ユーザーがサインオン時に各国語を指定すると、サインオン・オプションはユーザーの RACF CICS セグメントに指定された言語をオーバーライドします。

このように指定された言語は、ユーザーが端末にサインオンしている間、設定されます。サインオンしているユーザーによって呼び出されるトランザクションは、サインオン時に指定された各国語で実行されます。

ただし、トランザクションが START コマンドを使用して別のトランザクションを開始する場合、開始されたトランザクションの各国語属性は次のようにして得られます。

1. START コマンドで USERID パラメーターが指定された場合、各国語は、指定されたユーザー ID の RACF CICS セグメントから取得されます。
2. 事前設定の各国語が端末定義に指定されている端末でユーザーがサインオンしている場合は、この事前設定の各国語が、開始されたトランザクションに割り当てられます。
3. START コマンドにユーザー ID がなく、端末に各国語が事前設定されていない場合、開始されたトランザクションは、サインオンしているユーザーの RACF CICS セグメントで指定された各国語を継承します (サインオン時に使用された各国語ではありません)。

元の端末の各国語が必要な場合は、START コマンドが発行される前に、端末の各国語を照会できます。その後、この情報は、開始されたトランザクション用に START コマンドのデータとして渡すことができます。

トランザクション・セキュリティー

トランザクション・セキュリティー (接続時セキュリティー や、トランザクション接続セキュリティー ともいう) とは、トランザクションを実行しようとしているユーザーにその資格があるのか確認するものです。

トランザクション接続セキュリティーを制御する CICS パラメーター

CICS トランザクション接続セキュリティー検査は、CICS システム初期設定パラメーターを通じて制御されます。

該当するパラメーターは、以下のとおりです。

SEC RACF サービスを使用して CICS リソース (特に CICS トランザクション) へのアクセスを制御するには、SEC=YES を指定します。(詳しくは、 63 ページの『セキュリティー関連システム初期設定パラメーター』を参照。)

SECPRFX

CICS 領域のユーザー ID に対応する接頭部を使用してトランザクション・プロファイルが RACF に定義されている場合は、SECPRFX=YES を指定します。

その他の接頭部を使用してトランザクション・プロファイルが RACF に定義されている場合は、SECPRFX=*prefix* を指定します。 (詳しくは、 63 ページの『セキュリティー関連システム初期設定パラメーター』を参照。)

XTRAN

トランザクションを開始できるユーザーを CICS で制御したい場合は、XTRAN=YES または XTRAN=*resource_class_name* を指定します。YES を

指定すると、CICS は RACF のデフォルト・リソース・クラス TCICSTRN および GCICSTRN に定義されたプロファイルを使用します。(これらのリソース・クラスの詳細については、 27 ページの『RACF classes for CICS リソース』を参照してください。)

リソース・クラス名を指定した場合、CICS は、指定された名前に、接頭部としてリソース・クラスの場合は T、グループ・クラスの場合は G を付けたものを使用します。

XTRAN=NO を指定した場合、CICS はトランザクションを開始するユーザーに対して許可検査を実行しません。

デフォルトは YES です。したがって、SEC=YES を指定し、XTRAN パラメーターを省略すると、トランザクション接続セキュリティーは有効になり、デフォルト・リソース・クラス名が使用されます。

トランザクション接続セキュリティーを個々のトランザクション・レベルで制御できるようにする CICS パラメーターはありません。SEC=YES および XTRAN=YES (または XTRAN=resource_class_name) を指定すると、CICS はあらゆるトランザクションに関して許可要求を発行します。トランザクションが端末から開始されたか、EXEC CICS START コマンドを使用して開始されたか、あるいは一時データ・キューからトリガーされたかに関係なく、また termid オペランドの有無にかかわらず、この処理は行われます。サインオンしているユーザーがいない場合でも、CICS はこのセキュリティー検査を実行します。サインオンしないユーザーは、デフォルト・ユーザーに許可されているトランザクションのみを使用できます。

96 ページの図 2 は、CICS トランザクション・セキュリティーの主なエレメントを示す例です。

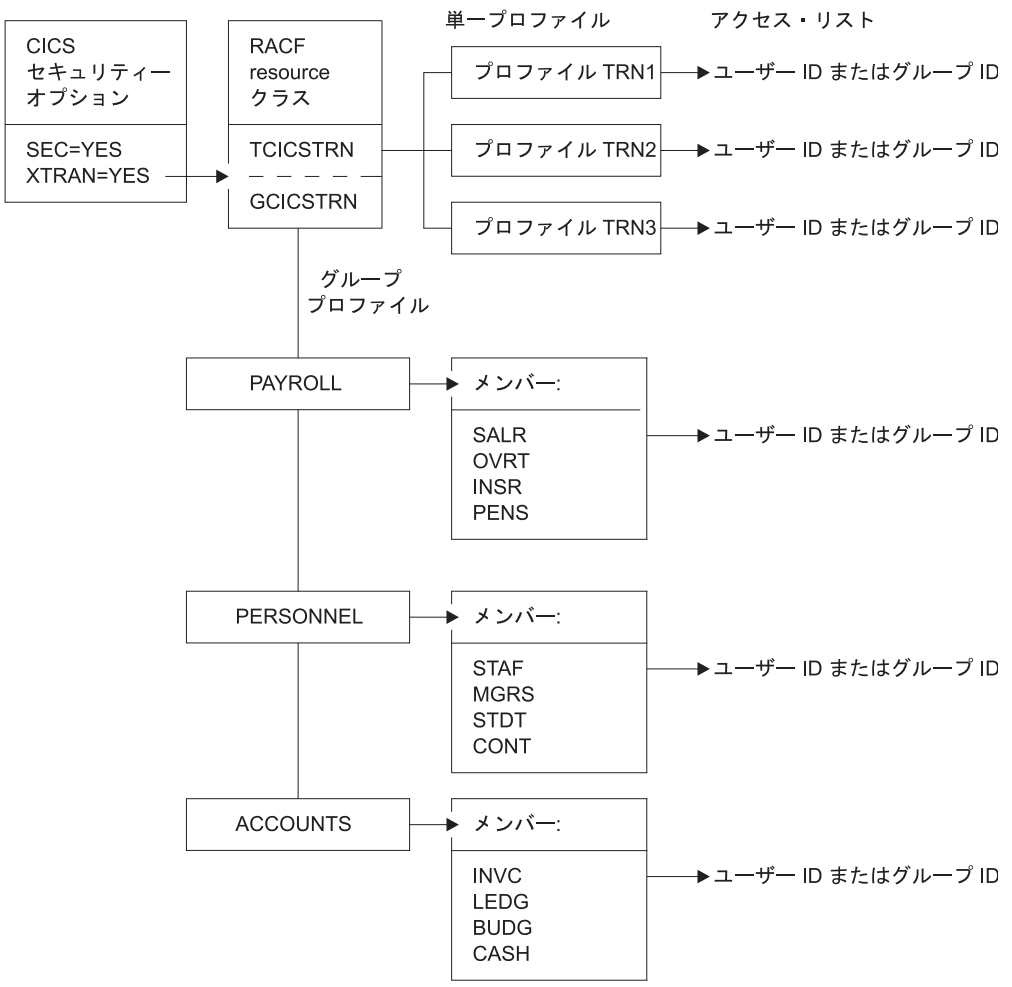


図 2. *CICS* トランザクション・セキュリティーの主なエレメントの例：

この例では、次のようになっています。

- 以下のシステム初期設定パラメーターが指定されました。

  ```
  SEC=YES
  XTRAN=YES
  ```

  XTRAN=YES が指定されているため、リソース・クラス名は TCICSTRN、グループ・クラス名は GCICSTRN です。
- リソース・クラス TCICSTRN には、プロファイル TRN1、TRN2、および TRN3 が含まれます。
- グループ・クラス GCICSTRN には、以下のグループ・プロファイルが含まれます。
  - PAYROLL (メンバー SALR、OVRT、INSR、および PENS)
  - PERSONNEL (メンバー STAF、MGRS、STDT、および CONT)
  - ACCOUNTS (メンバー INVC、LEDG、BUDG、および CASH)

SEC=YES および XTRAN=YES の場合のトランザクション接続処理

CICS 端末でトランザクションが開始されるたびに、CICS は許可要求を発行して、端末に関連付けられているユーザーがそのトランザクションに対して許可されているかどうかを確認します。

CICS および RACF は、XTRAN SIT パラメーターによって識別される RACF クラス内の現在アクティブなトランザクション・プロファイルを使用して、許可要求を処理します。(詳しくは、 26 ページの『主記憶域内のリソース・プロファイルのリフレッシュ』を参照。)

RACF へのトランザクション・プロファイルの定義

トランザクション・セキュリティー検査を適用した状態で実行される CICS 領域の場合は、無許可アクセスから保護する必要があるすべてのトランザクションについてトランザクション・プロファイルを定義します。

このタスクについて

デフォルトのトランザクション・リソース・クラス、または RACF クラス記述子テーブルに追加したインストール定義クラスのいずれかに、これらのプロファイルを定義できます。(トランザクション・リソース・クラスについて詳しくは、 27 ページの『RACF classes for CICS リソース』を参照してください。)

いくつかの推奨事項

以下の推奨事項は、関係する作業の量を削減することを目的としています。

- トランザクションは、リソース・グループ・クラス GCICSTRN 内で定義します。これにより、トランザクション・プロファイルおよび関連するアクセス・リストを定義および維持するために必要な労力の量を最小限に抑え、ストレージ内のプロファイルのサイズを低く抑えておくこともできます。ただし、リソース・グループを使用しても、重複メンバー名を定義しないようにしている場合には、必要なストレージ量を削減するだけであることに注意してください。
- ユーザーは個別のユーザーとしてではなく、グループでアクセス・リストに追加し、権限は READ と定義します。
- 可能な限り、総称プロファイルまたはメンバー名を使用します。

例えば、以下の RDEFINE コマンドと PERMIT コマンドは、給与計算部門のメンバーに付与された権限がある、一部の給与計算トランザクションを示しています。

```
RDEFINE  GCICSTRN  salarytrans
         NOTIFY(pay_manager)
         UACC(NONE)  ADDMEM(Pay1, Pay2, Pay3,..., Payn)
PERMIT  salarytrans CLASS(GCICSTRN)
        ID(paydept_group_userid) ACCESS(READ)
```

この例では、メンバーを `P*` または `Pay*` などのように総称的に定義できます。

ただし、総称プロファイルを定義する前に、以下のコマンドを発行する必要があります。

```
SETROPTS GENERIC(TCICSTRN)
```

GCICSTRN クラスは、SETROPTS GENERIC コマンドでクラスをグループ化できないので、指定できません。

他のどのユーザーでも使用できるトランザクションがある場合、UACC(READ) を使用してそれらのトランザクションの RACF トランザクション・プロファイルを定義することで、それらのアクセス・リストを維持しないようにすることができます。例えば、以下のような項目が含まれています。

```
RDEFINE TCICSTRN tranid UACC(READ)
```

いずれのトランザクションも RACF に対して定義されないようにする場合は、以下のように汎用アクセス権限を指定できます。

```
RDEFINE TCICSTRN ** UACC(READ)
```

これにより、RACF に対して定義する必要があるのは、さらに制限が多いセキュリティーを必要とするトランザクションのみとなります。

注: 本資料で説明されているようなプロファイルを使用する場合は、新規 CICS リソースをインストールする前に、新規プロファイルを RACF に対して定義します。

トランザクション・プロファイルの条件付きアクセス・リストの使用

アクセス・リストを特定の端末またはコンソールにサインオンするユーザーに基づく条件付きにすることで、セキュリティーの別のエレメントを追加できます。

例えば、以前の給与計算の例が TCICSTRN クラスの汎用トランザクションとして定義されている場合、次のように条件付きアクセスを定義できます。

```
RDEFINE  TCICSTRN  PAY*
         NOTIFY(pay_manager)  UACC(NONE)
PERMIT  pay*  CLASS(TCICSTRN)  ID(userid) ACCESS(READ)
              WHEN(TERMINAL(termid))
              WHEN(CONSOLE(*))
```

注:

1. このサポートを有効にするには、TERMINAL クラスまたは CONSOLE クラスがアクティブでなければなりません。
2. WHEN(TERMINAL(*termid*)) は、明示的にサインオンしたユーザー、ユーザーが明示的にサインオンした領域内、および MRO リンクによって接続されている領域内にのみ適用されます。
3. CICS は、コンソールおよび端末のエントリー・ポートのみを使用します。

許可障害とエラーメッセージ

端末ユーザーが無許可のトランザクションを開始しようとすると、CICS は端末にセキュリティー違反メッセージ (DFHAC2033) を発行します。

その後 CICS は、対応するメッセージ (DFHAC2003) を CSMT 一時データ宛先に送信し、DFHXS1111 メッセージを CSCS に送信します。RACF は、通常は ICH408I メッセージを CICS 領域のジョブ・ログ、およびセキュリティー・コンソール (宛先コード 9 メッセージに定義されたコンソール) に発行します。ただし、RACF に対するトランザクションの定義で LOG(NONE) が指定されている場合、ICH408I メッセージは出されません。ICH408I メッセージの説明については、z/OS Security Server RACF メッセージおよびコードを参照してください。

許可の問題の解決について詳しくは、CICS-RACF セキュリティー環境での問題判別を参照してください。

このアクセスに対して監査 (AUDIT オペランドによって要求された監査など) が要求された場合、RACF は SMF タイプ 80 ログ・レコードを書き込みます。RACF 監査員は、RACF 報告書作成プログラムを使用して、これらのレコードに基づくレポートを生成することができます。詳しくは、「 z/OS Security Server RACF 監査担当者のガイド」を参照してください。

非端末トランザクションの保護

すべてのリソース・セキュリティー検査について、CICS は、リソースへのユーザーのアクセス権限を検査するため、ユーザー ID、またはユーザー ID にマップする ICRX を必要とします。

端末に関連付けられていないトランザクションでリソースが使用される場合、CICS はそれらのリソースを無許可の使用から保護できます。

非端末トランザクションのタイプには、以下のものがあります。

- 端末 ID が指定されずに **EXEC CICS START** コマンドによって開始されたトランザクション
- **RUN TRANSID** コマンドによって開始されたトランザクション
- 区画内一時データ・キューのトリガー・レベルに達したときに端末なしで開始されたトランザクション
- CICS の始動時にプログラム・リスト・テーブル (PLT) の第 2 フェーズから実行されたプログラム

注**: START** コマンドがパラメーター **USERID** または **TERMID** のいずれかを指定して実行される場合、あるいは、LU61 または LU62 接続を介してシップされる場合、分散 ID 情報は保存されません。

トリガーされたトランザクション

CEDA トランザクション、CEDA DEFINE TDQUEUE、EXEC CICS CREATE、および EXEC CICS SET コマンドの ATIUSERID オプションは、一時データ・トリガー・レベルによって開始された非端末トランザクションのセキュリティーを確立します。

SET コマンド、INSTALL コマンド、または CREATE コマンドを発行するユーザーには、ATIUSERID オプションで指定されたユーザー ID の代理権限が必要です。トリガーされたトランザクションに関連付けられるユーザーは、一時データ・キュー定義の USERID 属性で指定されます。

PLT プログラム

PLT プログラムが CICS の始動時に実行される場合、CICS は領域ユーザー ID に対して代理ユーザー・セキュリティー検査を実行します。 43 ページの『CICS 領域ユーザー ID のユーザー・プロファイルの定義』を参照してください。この検査は、CICS ジョブが PLTPIUSR パラメーターで指定されたユーザー ID の代理となることが許可されているかどうかを判別します。 PLTPISEC および PLTPIUSR システム初期設定パラメーターは、CICS 始動の第 3 ステージから実行される PLT プログラムに対するセキュリティー・オプションを指定します (これは PLTP 初期設定の第 2 フェーズです)。

PLTPIUSR パラメーターが指定されていない場合、PLTPISEC=NONE オプションが定義されていれば、PLT プログラムは CICS 領域ユーザー ID で実行されます。これに対する代理検査は必要ありません。 PLT プログラムが START コマンドを発行した場合、ジョブ・ステップ・ユーザー ID は、ユーザー ID がコーディングされないときにそのコマンドを開始する代理権限を持ちます。スターターは常にそれ自体に対する代理権限を持っていることに注意してください。開始済みトランザクションが始動すると、トランザクションの接続の権限、および TCICSTRN クラス内のそのトランザクションへのアクセスの権限がユーザー ID にあるかどうかを確認するための別の検査が行われます。追加リソースにジョブ・ステップ・アクセス権限を付与するのではなく、PLTPIUSR パラメーターと PLTPISEC パラメーターを使用できます。

シャットダウン時に、シャットダウンを要求したトランザクションのユーザー ID の権限の下で、CICS は PLT プログラムを実行します。そのトランザクションの RESSEC オプションと CMDSEC オプションの値は、PLT プログラムにも適用されます。 EXEC CICS PERFORM SHUTDOWN コマンドを発行するトランザクションの定義に RESSEC(YES) および CMDSEC(YES) が指定されている場合は、シャットダウンの最初の段階でセキュリティー検査が行われます。

リソース・セキュリティー

リソース・セキュリティーは、CICS リソースを無許可アクセスから保護します。

リソース・セキュリティーの概要

リソース・セキュリティーは、CICS トランザクションが使用するリソースへのアクセスを制御することで、トランザクション・セキュリティーにさらなるセキュリティー・レベルを提供します。特定の CICS トランザクションの呼び出し権限を持つユーザーが、そのトランザクション内で使用されるファイル、PSB、またはその他の一般リソースへのアクセス権限を持っていない場合があります。個々のトランザクションについてオフにすることができないトランザクション・セキュリティーとは異なり、リソース・セキュリティー検査は個々のトランザクション・レベルで制御できます。

リソースまたはコマンドのセキュリティー検査が指定されている場合は、アプリケーション・プログラミング言語をサポートするために CICS に定義されたリソースもセキュリティー検査の対象となります。

CICS トランザクションで使用される一般リソースにアクセスできるユーザーを制御するには、以下の方法があります。

- システム初期設定パラメーターとして SEC=YES を指定する
- システム初期設定パラメーターとして RESSEC=ALWAYS を指定する
- TRANSACTION リソース定義に RESSEC(YES) を指定する
- RACF 一般リソース・クラスの CICS システム初期設定パラメーターを定義することによって、保護するリソースのタイプを指定する
- リソース・クラス・プロファイルで、適切なアクセス・リストを指定して CICS リソースを RACF に定義する

システム・プログラミング・コマンドのアクセス許可レベルについて詳しくは、137 ページの『リソースおよびコマンドの検査の相互参照』を参照してください。

XRES リソース・セキュリティー・パラメーターを使用したセキュリティー

セキュリティー・プロファイルおよび **XRES** システム初期設定パラメーターを使用して、CICS リソースのサブセットをセキュリティー検査します。CICS 文書テンプレートにリソース・セキュリティーを実装する方法の例を示します。

このタスクについて

CICS セキュリティー・プロファイル名は 3 つの部分から成っており、*security_prefix.resource_type.resource_name* という形式です。 CICS プロファイルは、検査のためにセキュリティー・マネージャーに渡されます。セキュリティー検査は、大文字小文字の区別があります。

セキュリティーの接頭部

security_prefix は、**SECPRFX** システム初期設定パラメーターで指定された値です。 **SECPRFX** パラメーターのデフォルト値は `NO` です。これは、*security_prefix* が省略されることを意味します。 **SECPRFX** パラメーターの値が `YES` の場合、*security_prefix* は領域ユーザー ID の名前です。あるいは、*security_prefix* に 1 から 8 文字の値を指定することもできます。

リソース・タイプ

resource_type は、検査が実行されるリソースのタイプを指定します。ほとんどの場合、各 CICS リソースには対応するセキュリティー・プロファイルがあります。例えば、ATOMSERVICE リソースには以下のように、対応するリソース・タイプに関するセキュリティー・プロファイルがあります。

security_prefix.ATOMSERVICE.*resource_name*

ただし、ある特定の CICS リソースには対応するセキュリティー・プロファイルがありません。例えば、BUNDLEPART、OSGIBUNDLE、および OSGISERVICE リソースは、BUNDLE セキュリティー・プロファイルを使用して検査されます。

以下のリソース・タイプを使用するセキュリティー・プロファイルを作成できます。

- ATOMSERVICE
- BUNDLE
- DOCTEMPLATE
- EPADAPTER
- EPADAPTERSET
- EVENTBINDING
- JVMSERVER
- XMLTRANSFORM

セキュリティー検査に関連付けられた CICS リソース、リソース・タイプ、およびコマンドの完全なリストについては、 137 ページの『リソースおよびコマンドの検査の相互参照』を参照してください。

注: あるユーザーにプラットフォームまたはアプリケーションに対してアクションを実行する権限を付与すると、そのプラットフォームまたはアプリケーションに対して動的に生成されるリソースについても、そのユーザーが同じアクションを実行できる権限を付与することになります。アプリケーションまたはプラットフォームを介して CICS バンドルを作成または操作する場合、CICS リソース・セキュリティー検査は実行されません。ただし、個々の BUNDLE リソース、または CICS バンドルに定義されたリソースに対してアクションを直接実行するときは、バンドルがプラットフォームまたはアプリケーションのインストール時に作成された場合でも、CICS リソース・セキュリティー検査が適用されます。詳しくは、バンドルのセキュリティーを参照してください。

リソース名
resource_name は、CICS リソースの名前を指定します。

例

このタスク例は、CICS 文書テンプレートにリソース・セキュリティーを実装する方法を示します。

CICS 文書テンプレートは、以下の場合に制御されます。

- Web クライアントの要求に対する静的応答として送信された文書テンプレート (要求に対する URIMAP 定義の TEMPLATENAME 属性上で指定)。
- Web クライアントの要求に対するアプリケーション生成の応答の一部として送信された文書テンプレート (要求を処理するアプリケーション・プログラムによって使用されます)。
- すべての **EXEC CICS CREATE**、**INQUIRE**、**DISCARD**、および **SET DOCTEMPLATE** コマンド。
- すべての **EXEC CICS DOCUMENT INSERT** コマンドおよび **CREATE** コマンド (TEMPLATE オプションを指定)。

EXEC CICS DOCUMENT コマンドは、DOCTEMPLATE リソースの TEMPLATENAME 属性で指定された 48 文字のテンプレート名を使用して、文書テンプレートを参照します。ただし、これらのコマンドに対するセキュリティー検査では、TEMPLATENAME 属性に対応する DOCTEMPLATE リソースの名前が使用されます。そのため、TEMPLATENAME 属性ではなく DOCTEMPLATE リソースの名前を使用して、文書テンプレートごとに 1 つのプロファイル名をセットアップできます。

注: 区分データ・セット、CICS プログラム、CICS ファイル、z/OS UNIX システム・サービスのファイル、一時記憶キュー、一時データ・キュー、出口プログラムなど、さまざまなソースから文書テンプレートを取得することができます。リソース・セキュリティー検査が文書テンプレートに対して行われる場合、文書テンプレートを提供するリソースについては、CICS は追加のセキュリティー検査を実行しません。CICS 領域で、そのタイプのリソースに対してリソース・セキュリティーが指定されていても、同様です。

CICS 文書テンプレートにリソース・セキュリティーを実装するには、以下のステップを実行します。

1. デフォルトの RCICSRES リソース・クラスまたは WCICSRES グループ化クラス、あるいはユーザー定義のリソース・クラス名があればそれらに相当するクラスで、プロファイルを RACF に定義する。

   プロファイル名については、DOCTEMPLATE リソース定義の名前に、接頭部としてリソース・タイプ DOCTEMPLATE を付け、CICS 領域用の **SECPRFX** システム初期設定パラメーターで指定された任意の追加の接頭部を付けたものを使用します。

   例えば、RCICSRES クラスの文書テンプレートを定義し、ユーザーがそれらを使用して文書をアセンブルすることを許可するには、次のコマンドを使用します。

```
RDEFINE  RCICSRES  (DOCTEMPLATE.doc1, DOCTEMPLATE.doc2, .., DOCTEMPLATE.docn) UACC(NONE)
                  NOTIFY(sys_admin_userid)
PERMIT DOCTEMPLATE.doc1 CLASS(RCICSRES) ID(group1, group2) ACCESS(READ)
```

   適切なアクセス・リストを使用し、WCICSRES リソース・グループ化クラスのプロファイルのメンバーとして文書テンプレートを定義するには、次のコマンドを使用します。

```
RDEFINE  WCICSRES  (doc_groupname) UACC(NONE)
                  ADDMEM(DOCTEMPLATE.doca, DOCTEMPLATE.docb) NOTIFY(sys_admin_userid)
PERMIT doc_groupname CLASS(WCICSRES) ID(group_userid) ACCESS(READ)
```

   RCICSRES または WCICSRES クラスに対して RDEFINE コマンドを発行した後、クラスがまだ活動化されていない場合は、SETROPTS コマンドを発行して活動化する必要があります。以下に例を示します。

```
SETROPTS CLASSACT(RCICSRES) RACLIST(RCICSRES)
```

   クラスがアクティブである場合は、SETROPTS コマンドを使用してクラスをリフレッシュします。

```
SETROPTS RACLIST(RCICSRES) REFRESH
```

2. CICS システム初期設定パラメーターとして **SEC**=YES を指定する (接頭部付きのプロファイルを定義する場合は、さらに **SECPRFX**=YES を指定する)。
3. RCICSRES および WCICSRES のデフォルトのリソース・クラス名の場合は XRES=YES、ユーザー定義のリソース・クラス名の場合は XRES=*name* を指定する。

   XRES=YES がデフォルトです。
4. CICS 文書テンプレートにアクセスするトランザクションの TRANSACTION リソース定義で RESSEC(YES) を指定する。

   CICS Web サポートの場合、すべての静的応答用のトランザクションは CWXN であるか、または TCPIPSERVICE 定義で TRANSACTION 属性を使用して CWXN の代わりに指定した代替トランザクションです。アプリケーション生成の応答のトランザクションは、別名トランザクションです。これは要求に応じて URIMAP 定義で指定するか、アナライザー・プログラムによって設定することができ、デフォルトは CWBA です。

CICS で提供されたとおり、CWXN の定義は RESSEC(YES) を指定しますが、CWBA の定義は RESSEC(NO) を指定し、一般的に TRANSACTION リソース定義の場合、デフォルトは RESSEC(NO) です。

CICS Web サポートによって使用される文書テンプレートおよび z/OS UNIX ファイルにアクセスするためのユーザー ID

あるトランザクションに関してリソース・セキュリティーがアクティブになっている場合、外部セキュリティー・マネージャーは、トランザクションに関連付けられたユーザー ID が必要なリソースへのアクセスを許可されているかどうかを確認します。CICS Web サポートの場合は、特定の Web 要求のトランザクションに関連付けられたユーザー ID を複数のソースから入手できます。必要なセキュリティーのレベルに応じて、保護された文書テンプレートまたは z/OS UNIX ファイルに対するリソース・セキュリティー検査に使用するユーザー ID を決定するように、CICS Web サポートのアーキテクチャーを調整できます。

アプリケーションが生成する応答。

CICS Web サポートの場合、アプリケーション生成の応答のトランザクションは、別名トランザクションです。これは要求に応じて URIMAP 定義で指定するか、アナライザー・プログラムによって設定することができ、デフォルトは CWBA です。CWBA は RESSEC(NO) として定義されているため、別名トランザクションに関してリソース・セキュリティーが必要な場合は、CWBA 定義をユーザー独自のグループにコピーし、RESSEC 属性を変更するか、別の別名トランザクションを使用する必要があります。

Web クライアントが CICS Web サポートに対して要求を送り、アプリケーションから応答が提供された場合、CICS は次の優先順位に従って、別名トランザクションのユーザー ID を選択します。

1. アナライザー・プログラムを使用して設定したユーザー ID。 このユーザー ID は、Web クライアントから取得された、または URIMAP 定義によって提供されたユーザー ID をオーバーライドできます。
2. 基本認証を使用して Web クライアントから取得したユーザー ID、または Web クライアントによって送信されたクライアント証明書に関連付けられたユーザー ID。 接続に認証が必要なのに、クライアントが認証済みユーザー ID を提供していない場合、要求はリジェクトされます。
3. 要求の URIMAP 定義で指定されたユーザー ID。
4. 他に判別できるユーザー ID がない場合、CICS デフォルト・ユーザー ID。

アプリケーションから生成された応答については、Web クライアントに関して選択されたユーザー ID が別名トランザクション全体に適用されます。トランザクションの接続、Web アプリケーション・プログラムの使用、および保護された文書テンプレートの使用のためには、このユーザー ID が認可されている必要があります。

アプリケーションから生成された応答については、Web クライアントのユーザー ID は、 z/OS UNIX ファイルに関する特別の許可を必要としません。これは、ファイルが CICS 文書テンプレートとして定義されている場合、アプリケーションは EXEC CICS コマンドを使用する以外 z/OS UNIX ファイルを操作できないためです。セキュリティー検査は CICS 文書テンプレートについてのみ実施され、 z/OS

UNIX ファイルが再度検査されることはありません。

静的応答

URIMAP 定義に指定されたすべての静的応答のトランザクションは、デフォルトの Web 生成タスク CWXN であるか、 CWXN の代わりに TCPIPSERVICE 定義の TRANSACTION 属性を使用してユーザーが指定した任意の別名トランザクションです。

Web クライアントが CICS Web サポートに要求を送り、その応答が URIMAP 定義に指定された静的応答であった場合、この Web クライアントのユーザー ID として、基本認証を使用して Web クライアントから取得されたユーザー ID、または Web クライアントによって送信されたクライアントの証明書に関連付けられたユーザー ID が使用されます。

文書テンプレートのリソース・セキュリティー検査は、XRES システム初期設定パラメーターおよびトランザクションの RESSEC 属性によって制御されます (CWXN またはその別名)。 z/OS UNIX ファイルに関するアクセス制御は、XHFS システム初期設定パラメーターのみで制御されます。

Web クライアントのユーザー ID は、静的応答として提供される文書テンプレートまたは z/OS UNIX ファイルに関するリソース・セキュリティーの検査プロセスでのみ使用されます。文書テンプレートまたは z/OS UNIX ファイルにアクセスするためには、ユーザー ID が認可されている必要があります。

CICS および RACF によるリソース・セキュリティー検査

CICS は RACF を使用して、CICS アプリケーション・プログラムからアクセスできるリソースを保護します。

CICS および RACF リソース検査の要約

RACF クラス名を指定するには、CICS リソース・システム初期設定パラメーターを使用します。

表 8 では各リソースが簡単に説明されており、RACF クラス名を指定するために使用する関連の CICS システム初期設定パラメーターが記載されています。各システム初期設定パラメーターに関連したアプリケーション・プログラミング・コマンドおよびシステム・プログラミング・コマンドに関する総合的な情報については、137 ページの『リソースおよびコマンドの検査の相互参照』を参照してください。

BMS コマンドに対しては、許可処理は実行されません。

表 *8. CICS* による一般リソース検査

CICS パラメーター	保護されるリソース	詳細情報
XAPPC	パートナー論理装置 (LU6.2)。	275 ページの『LU6.2 セキュリティーの実装』.

表 8. *CICS* による一般リソース検査 (続き)

CICS パラメーター	保護されるリソース	詳細情報
XCMD	コマンド・セキュリティー検査の対象である CICS アプリケーション・プログラミング・コマンドのサブセット。 EXEC CICS FEPI システム・コマンドも、このパラメーターによって制御されます。	162 ページの『CICS コマンド・セキュリティー』
XDB2	DB2ENTRY の Db2 リソース・クラスは、XDB2 システム初期設定パラメーターで CICS に対して指定されます。	32 ページの『DB2ENTRY リソースのリソース・クラス』
XDCT	CICS 区画外および区画内の一時データ・キュー。一時データ・クラス内でプロファイルを定義して、CICS 一時データ・キューへのアクセスが許可されるユーザーを制御します。	110 ページの『一時データのセキュリティー』.
XFCT	CICS ファイル制御管理対象の VSAM ファイルおよび BDAM ファイル。ファイル・クラス内でプロファイルを定義して、CICS VSAM ファイルおよび BDAM ファイルへのアクセスが許可されるユーザーを制御します。	112 ページの『ファイルのセキュリティー』.
XHFS	z/OS UNIX システム・サービスによって管理される z/OS UNIX ファイル。 z/OS UNIX ファイルのアクセス制御は z/OS UNIX システム・サービスで指定され、z/OS UNIX ファイルには個々の RACF プロファイルは不要であるため、これは特殊なケースです。このリソースに関連付けられているアプリケーション・プログラミング・コマンドおよびシステム・プログラミング・コマンドはありません。	125 ページの『z/OS UNIX ファイルのセキュリティーのインプリメント』.
XJCT	CICS システム・ログおよび一般ログ。ジャーナル・クラス内でプロファイルを定義して、CICS ログ・ストリーム上の CICS ジャーナルへのアクセスが許可されるユーザーを制御します。	114 ページの『ジャーナルとログ・ストリームのセキュリティー』.
XPCT	CICS 開始済みトランザクションおよび **EXEC CICS** コマンドは、**COLLECT STATISTICS TRANSACTION**、**DISCARD TRANSACTION**、**INQUIRE TRANSACTION**、**INQUIRE REQID**、**SET TRANSACTION**、および **CANCEL** です。開始済みトランザクション・クラス内でプロファイルを定義して、開始済み CICS トランザクションへのアクセスが許可されるユーザーを制御します。	115 ページの『開始されたトランザクションのセキュリティー』.
XPPT	CICS アプリケーション・プログラム。プログラム・クラス内でプロファイルを定義して、CICS アプリケーション・プログラムへのアクセスが許可されるユーザーを制御します。	121 ページの『アプリケーション・プログラムのセキュリティー』.

表 *8. CICS* による一般リソース検査 *(*続き*)*

CICS パラメーター	保護されるリソース	詳細情報
XPSB	DL/I プログラム仕様ブロック (PSB)。プログラム仕様ブロック・クラス内でプロファイルを定義して、CICS アプリケーション・プログラムで使用される DL/I PSB へのアクセスが許可されるユーザーを制御します。	128 ページの『プログラム仕様ブロックのセキュリティー』.
XRES	**XRES** パラメーターを使用する CICS リソースは、ATOMSERVICE、BUNDLE、DOCTEMPLATE、EPADAPTER、EPADAPTERSET、EVENTBINDING、JVMSERVER、および XMLTRANSFORM です。 例えば、DOCTEMPLATE リソース・クラス内でプロファイルを定義して、文書テンプレートへのアクセスが許可されるユーザーを制御します。	101 ページの『XRES リソース・セキュリティー・パラメーターを使用したセキュリティー』.
XTRAN	CICS トランザクション。	94 ページの『トランザクション・セキュリティー』.
XTST	CICS 一時記憶域キュー。一時記憶域クラス内でプロファイルを定義して、CICS 一時記憶域キューへのアクセスが許可されるユーザーを制御します。	122 ページの『一時記憶域のセキュリティー』.
XUSER	代理ユーザー・セキュリティー。	154 ページの『代理ユーザー・セキュリティー』.

RESSEC トランザクション・リソース・セキュリティー・パラメーター

システム初期設定パラメーターで定義されている適切なリソース・クラスとともに、トランザクションの定義に RESSEC(YES) を指定すると、トランザクション接続セキュリティーに加えてセキュリティー検査の別の層が導入されます。

最も単純な (つまり単一機能の) トランザクションの場合、この追加層のセキュリティーは必要ありません。例えば、端末ユーザーは個人ファイルの更新はできるが、それ以外は何もできないように、トランザクションが設計されている場合、ファイルへのアクセスを制御しなくても、トランザクションへのアクセスを許可するだけで十分ということになります。ただし、ユーザーが機能の選択をできる複雑なトランザクションであったり、トランザクション内で使用可能なすべてのオプションについてユーザーが分からなかったりする場合は、追加層のセキュリティーを加えて、トランザクションだけでなくデータへのアクセスも制限することをお勧めします。リソース・セキュリティー検査を実装する前に、リソース・セキュリティー検査が関係する余分のオーバーヘッドを考慮に入れ、その余分のコストを費やす価値があると確信する場合にのみ実装してください。

トランザクション定義で RESSEC(YES) を指定した場合、CICS は、X*name* リソース・クラス・パラメーターを使用して、ユーザーがセキュリティーを要求した以下

のいずれかのリソースに適用される各 CICS コマンドに対して (または z/OS UNIX ファイルの場合は各操作に対して) RACF を呼び出します。

- 区画外および区画内の一時データ・キュー (XDCT パラメーター)
- ファイル制御管理対象 VSAM ファイルおよび BDAM ファイル (XFCT パラメーター)
- システム・ログおよび一般ログ (XJCT パラメーター)
- 開始済みトランザクションおよび EXEC CICS コマンド: COLLECT STATISTICS TRANSACTION、DISCARD TRANSACTION、INQUIRE TRANSACTION、INQUIRE REQID、SET TRANSACTION、および CANCEL (XPCT パラメーター)
- アプリケーション・プログラム (XPPT パラメーター)
- XRES セキュリティー検査 (XRES パラメーター) の対象となるリソース (101 ページの『XRES リソース・セキュリティー・パラメーターを使用したセキュリティー』にリストがある)
- 一時記憶域宛先 (XTST パラメーター)

システム初期設定パラメーターには、以下のものが含まれます。

```
SEC=YES
XDCT=NO
XFCT=YES
XTRAN=YES
XTST=YES
```

トランザクション TRN1 には、以下の EXEC CICS コマンドが含まれます。

- 4 つのファイル制御 READ コマンド
- 1 つのファイル制御 WRITE コマンド
- 2 つの一時データ WRITEQ コマンド
- 1 つの一時記憶域 WRITEQ コマンド

TRN1 を実行すると、以下の 7 つの呼び出しが RACF に対して行われます。

- XTRAN=YES が指定されているため、トランザクション接続で 1 つ
- XFCT=YES が指定されているため、ファイル制御アクセスに対して 5 つ
- XTST=YES が指定されているため、一時記憶域アクセスに対して 1 つ

XDCT=NO が指定されているため、RACF は一時データ・キュー・アクセスに対して呼び出されません。

RESSEC システム初期設定パラメーター

RESSEC=ALWAYS システム初期設定パラメーターを指定することで、RESSEC=YES の効果をすべての CICS トランザクションに強制できます。

一般に、これは以下の理由から推奨されていません。

- ほとんどの単純なトランザクションでは、トランザクションへのアクセスを制御するだけで、トランザクションが実行できるすべてのことを制御するには十分です。

- すべての CICS リソースに対するリソース検査を呼び出すと、追加のオーバーヘッドが消費され、すべてのトランザクションのパフォーマンスを低下させます。
- 一部の CICS 提供のトランザクションは、ユーザーが認識していないリソースにアクセスする場合があります。これらのトランザクションのユーザーに、トランザクションで作業を続行できる十分な権限が付与されていることの確認は、ユーザー自身で行うことになります。

許可障害

端末ユーザーが CICS コマンドで指定されたリソースへのアクセスを許可されていない場合、CICS はアプリケーション・プログラムに NOTAUTH 条件を返します。

CICS はこの許可障害を、EXEC インターフェース・ブロック (DFHEIBLK) の EIBRESP フィールドを値 70 に設定することによって (およびバイト 0 の EIBRCODE フィールドの X'46' によって)、この許可障害を示します。制御を適切なルーチンに渡すことでセキュリティー違反を処理するように、CICS アプリケーションを設計します。これらは、次のいずれかの方法で行うことができます。

- NOTAUTH 条件を受け取る可能性がある各コマンドに RESP オプションを追加して、EIBRESP 条件をテストします。例えば、以下のようにします (COBOL の場合)。

```
EXEC CICS FILE('FILEA')
   INTO(REC) RIDFLD(KEY)
        RESP(COMMAND-RESPONSE)
END-EXEC.


EVALUATE COMMAND-RESPONSE
   WHEN DFHRESP(NORMAL)
        CONTINUE (続行)
   WHEN DFHRESP(NOTAUTH)
    PERFORM SECURITY-ERROR
END-EVALUATE.
```

- EXEC CICS HANDLE CONDITION NOTAUTH(*label*) コマンドをコーディングします。ここで *label* は、セキュリティー違反ルーチンの名前です。

アプリケーションがセキュリティー違反に対応しない場合、CICS はトランザクションを AEY7 異常終了コードで異常終了します。

SMF への RACF 監査メッセージのロギング

特定のセキュリティー・コマンドの処理時を除いて (174 ページの『**QUERY SECURITY** コマンドを使用したセキュリティー検査』を参照)、CICS はセキュリティー許可要求をロギング・オプションとともに発行します。これは、RACF が SMF タイプ 80 のログ・レコードを SMF に書き込むことを意味します。どのイベントがログに記録されるかは、有効である監査によって異なります。例えば、リソース・プロファイル内の AUDIT または GLOBALAUDIT オペランドによって要求されたイベント、あるいは SETROPTS AUDIT コマンドまたは SETROPTS LOGOPTIONS コマンドによって要求されたイベントをログに記録できます。

SMF TYPE 80 ログ・レコードに加え、RACF は指定されたコンソールに対して ICH408I メッセージを発行し、宛先コード 9 のメッセージを受け取ります。

RACF 報告書作成プログラムを使用して SMF タイプ 80 ログ・レコードを確認する方法を含め、監査の詳細については、 z/OS Security Server RACF 監査担当者のガイドを参照してください。

WARNING オプションの使用**:**

RACF WARNING オプションは、RACF プロファイルで使用される場合は、CICS によって受け入れられます。 WARNING オプションにより、ユーザーは他の方法では拒否されるリソースにアクセスできます。 RACF は、WARNING が有効でなかった場合には失敗したアクセスを SMF に記録します。

アプリケーションのリソース・セキュリティーの初期実装時には、WARNING の選択的な使用が、RACF セキュリティー定義内のエラーまたは省略の検査手段としてとりわけ役に立ちます。 WARNING の結果として SMF タイプ 80 レコードが記録される場合は、ユーザーをリソースのアクセス・リストに追加するかどうかを確認し、それに応じて RACF プロファイルを変更する必要があります。警告オプションが有効な状態でリソースがアクセスされる期間を厳密に制限し、警告期間中のロギングを最小限に抑える必要があります。

注**:** ユーザーへのアクセスが拒否されたときにすぐに通知を受け取りたい場合は、NOTIFY オプションを指定します。

一時データのセキュリティー

一時データ・キューのセキュリティーを実装するには、以下の手順に従います。

手順

1. 適切なトランザクションのリソース定義で RESSEC(YES) を指定する。
2. DCICSDCT または ECICSDCT リソース・クラス (あるいはユーザー定義のリソース・クラス名がある場合はそれらに相当するクラス) で、適切なアクセス・リストを指定して、プロファイルを RACF に定義する。 一時データ・キュー名の長さは最大 4 文字です (CSMT、L86O、L86P など)。

   例えば、DCICSDCT クラスにキューを定義し、ユーザーに対してこれらのキューの読み取りと書き込みの両方を許可するには、以下のコマンドを使用します。

   ```
   RDEFINE  DCICSDCT  (qid1, qid2, ..., qidn) UACC(NONE)
                      NOTIFY(sys_admin_userid)
   PERMIT qid1 CLASS(DCICSDCT) ID(group1, group2) ACCESS(UPDATE)
   PERMIT qid2 CLASS(DCICSDCT) ID(group1, group2) ACCESS(UPDATE)
   ```

   CICS 一時データ・リソース・グループ・クラスのプロファイルのメンバーとして、一時データ・キューを適切なアクセス・リストと共に定義するには、以下のコマンドを使用します。

   ```
   RDEFINE  ECICSDCT  (queue_groupname) UACC(NONE)
                      ADDMEM(qida, qidb, ..., qidz) NOTIFY(sys_admin_userid)
   PERMIT queue_groupname CLASS(ECICSDCT) ID(group_userid) ACCESS(UPDATE)
   ```

   111 ページの『一時的データ・キューのプロファイル定義に関する考慮事項』も参照してください。
3. CICS システム初期設定パラメーターとして SEC=YES を指定する (接頭部付きのプロファイルを定義する場合は、さらに SECPRFX を指定する)。

4. デフォルト・リソース・クラス名 DCICSDCT および ECICSDCT の場合は XDCT=YES を指定する (ユーザー定義のリソース・クラス名の場合は XDCT=class_name を指定する)。

一時的データ・キューのプロファイル定義に関する考慮事項

一時データ・キューへのアクセスを制御するために RACF にプロファイル名を定義するときは、以下の一時データ・キューのプロファイルのみを定義します。

- 区画内一時データ・キュー
- 区画外一時データ・キュー

間接一時データ・キューのプロファイルは定義しないでください。CICS は、間接キューのすべての要求を別のキューに送信しますが、そのキューは区画外、区画内、またはリモート・キューである可能性があります。リダイレクト先は、別の間接キューの場合もあります。

一時データ・キューのセキュリティー検査を適用して CICS を実行している場合、CICS は、キュー名を指定するコマンドごとに RACF の呼び出しを発行します。ただし、CICS が RACF に渡すリソース名は、最後のキューのキュー名であり、これは必ずしもコマンドに指定されたキューの名前とは限りません。

例えば、キュー QID2 が QID1 の間接キューとして定義されていて、EXEC CICS コマンドで QID2 が指定された場合、CICS は RACF を呼び出して、QID2 ではなく QID1 の許可検査を行います。この例を以下に示します。

```
TDQ definition: DEFINE TDQUEUE(QID1)
                       TYPE(EXTRA)
                       TYPEFILE(OUTPUT)
                       RECORDSIZE(132)
                       BLOCKSIZE(136)
                       RECORDFORMAT(VARIABLE)
                       BLOCKFORMAT(UNBLOCKED)
                       DDNAME(CICSMSGS)
                       GROUP(DFHDCTG)

                DEFINE TDQUEUE(QID2)
                       TYPE(INDIRECT)
                       INDIRECTNAME(QID1)
                       GROUP(DFHDCTG)

 CICS transaction:   EXEC CICS WRITEQ TD
                          QUEUE(QID2)
                          FROM(data_area)
                          LENGTH(length)
 CICS calls RACF:    Does the terminal user of the CICS transaction
                     have UPDATE authorization for QID1?
```

アクセス許可レベル

一時データ・キューから項目を読み取ることができるのは一度だけです。一時データ・キューで項目の読み取りが行われた場合は必ず、CICS がその項目を (「破壊的読み取り」を実行して) 削除するからです。

したがって、システム初期設定パラメーターとして SEC=YES を指定してセキュリティーを指定した場合、CICS はすべての TD コマンド (DELETEQ、WRITEQ、および READQ) に対して、最小許可レベルとして UPDATE を必要とします。

一時データ・キューに適用されるシステム・プログラミング・コマンドのアクセス許可レベルについては、 137 ページの『リソースおよびコマンドの検査の相互参照』を参照してください。

CICS 必須の一時データ・キュー・リソース定義

CICS 自体は、多数の一時データ・キューを使用します。これらのキューは、グループ DFHDCTG (DFHLIST の一部) に定義されています。ユーザー・アプリケーション・プログラムからこれらの定義へのアクセスを保護する場合は、アクセス・リストを使用せずに、UACC(NONE) を使用して RACF に対してそれらを定義します。

提供されているリソース定義内では、ほとんどの一時データ・キューは間接的であり、一時データ・キュー CSSL または CCSO を指します。提供されたままの状態で定義を使用する場合は、RACF に対してキュー名 CSSL および CCSO を以下のように定義します。

```
RDEFINE  ECICSDCT  CICSQUEUES UACC(NONE)
                   ADDMEM(CSSL, CCSO)
                   NOTIFY(sys_admin_userid)
```

トリガーされるトランザクションについての考慮事項

トリガー・レベルがゼロより大きい区画内一時データ・キューの場合、CICS は、トリガーされたトランザクションに関連付けられたユーザー ID を以下のソースから取得します。

- ATIFACILITY(FILE) を使用して定義された一時データ・キューの場合、CICS は、一時データ・キューのリソース定義で指定された USERID パラメーターを使用します。
- ATIFACILITY(TERMINAL) を使用して定義された一時データ・キューの場合、CICS は端末に関連付けられたユーザー ID を使用します。端末にサインオンしているユーザーがいない場合は、CICS デフォルト・ユーザー ID が使用されます。
- ATIFACILITY(SYSTEM) を使用して定義された一時データ・キューの場合、CICS は、CONNECTION リソース定義によって取得されるリンク・ユーザー ID を使用します。

ファイルのセキュリティー

CICS アプリケーション・プログラムは、CICS にとって物理 VSAM または BDAM データ・セットの論理ビューであるファイルを処理します。

ファイルは CICS に対して 8 文字のファイル名で識別されます。同じ物理データ・セットを参照する多数のファイルを CICS に定義することができ、その物理データ・セットは 44 文字のデータ・セット名 (DSNAME) で別個に識別されます。例えば、FILEA、FILEB、および FILEC というファイル・リソース定義で、それらはすべて 1 つの物理 VSAM データ・セットを参照するが、各ファイル定義は異なる属性を指定するように定義できます。

CICS トランザクションは、CICS ファイル制御名を使用して物理データ・セットのデータにアクセスします。したがって、CICS 管理ファイルへのアクセスを制御するには、RACF データ・セット・クラス内ではなく、CICS ファイルの RACF 一般リ

ソース・クラス内にプロファイルを定義します。リソースを識別するため、CICS の 8 文字のファイル名を使用してプロファイルを定義します。(44 文字のデータ・セット名に基づく RACF データ・セット許可は、OPEN 処理時に、その CICS 領域ユーザー ID が OPEN 要求されたデータ・セットへのアクセス権限を持っているかどうかを判断するためにのみ使用されます。これは、OPEN の実行を引き起こしたトランザクションを実行しているユーザー ID には依存しません。)

CICS ファイル制御によって管理されるファイルのセキュリティーを実装するには、次のようにします。

1. ファイルにアクセスするトランザクションの CSD リソース定義で RESSEC(YES) を指定する。
2. FCICSFCT または HCICSFCT リソース・クラス (あるいはユーザー定義のリソース・クラス名がある場合はそれらに相当するクラス) で、プロファイルを RACF に定義する。その際、CICS ファイル名を使用してプロファイルを識別します。例えば、FCICSFCT クラスにファイルを定義し、ユーザーに対してこれらのファイルの読み取りまたは書き込みを許可するには、以下のコマンドを使用します。

   ```
   RDEFINE  FCICSFCT  (file1, file2, .., filen) UACC(NONE)
                    NOTIFY(sys_admin_userid)
   PERMIT file1 CLASS(FCICSFCT) ID(group1, group2) ACCESS(UPDATE)
   PERMIT file2 CLASS(FCICSFCT) ID(group1, group2) ACCESS(READ)
   ```

   ファイルを CICS ファイル・リソース・グループ・クラス内のプロファイルのメンバーとして、適切なアクセス・リストを指定して定義するには、次のコマンドを使用します。

   ```
   RDEFINE  HCICSFCT  (file_groupname) UACC(NONE)
                     ADDMEM(filea, fileb, .., filez) NOTIFY(sys_admin_userid)
   PERMIT file_groupname CLASS(HCICSFCT) ID(group_userid) ACCESS(UPDATE)
   ```

3. CICS システム初期設定パラメーターとして SEC=YES を指定する (接頭部付きのプロファイルを定義する場合は、さらに SECPRFX を指定する)。
4. デフォルト・リソース・クラス名 FCICSFCT および HCICSFCT の場合は XFCT=YES を指定する (ユーザー定義のリソース・クラス名の場合は XFCT=class_name を指定する)。

RDO トランザクションは CSD へのアクセスにファイル・コマンドを使用しないため、これらのメカニズムの対象にはならないことに注意してください。

アクセス許可レベル

システム初期設定パラメーターとして SEC=YES を指定してセキュリティーを指定する場合、CICS は、意図したファイル・アクセスに適した許可レベルを必要とします。これは読み取りインテントの場合は最小で READ、更新または削除インテントの場合は最小で UPDATE です。

ファイルに適用されるシステム・プログラミング・コマンドのアクセス許可レベルについては、137 ページの『リソースおよびコマンドの検査の相互参照』を参照してください。

ジャーナルとログ・ストリームのセキュリティー

CICS ログ・マネージャーは、CICS システム・ログと CICS 一般ログの書き込みおよび読み取り機能を提供します。一般ログは、ユーザー・ジャーナル、順方向リカバリー・ログ、および自動ジャーナルから成ります。ジャーナルとログ・ストリームのセキュリティーを実装すると、それらを無許可アクセスから保護できます。

システム・ログは、リカバリーの目的でのみ使用されます (例えば、動的トランザクション・バックアウトや緊急時再始動のときなど)。他の目的には使用できません。したがって、ユーザー・アプリケーションから WRITE JOURNALNAME コマンドを使用してシステム・ログに書き込まないでください。

CICS は、その 1 次システム・ログにジャーナル ID **DFHLOG** を使用します。ユーザー・トランザクションがこのシステム・ログに書き込むことは許可されません。このような書き込みを防ぐには、以下のコマンドを使用して、システム・ログをアクセス・リストなしで JCICSJCT クラスに定義します。

```
RDEFINE  JCICSJCT  DFHLOG  UACC(NONE) NOTIFY(sys_admin_userid)
```

ユーザー・トランザクションで (ファイル・リソース定義でのオプションに応じて) CICS が実行する自動ジャーナル処理および順方向リカバリー・ロギングに加えて、ユーザー・アプリケーションも **WRITE JOURNALNAME** コマンドを使用してユーザー・ジャーナル・レコードに書き込むことができます。

ジャーナル・レコードに書き込む必要があるユーザーは、(JCICSJCT で定義された) JOURNALNAME への書き込み権限を持っていなければなりません。CICS は RACF を呼び出してセキュリティー検査を実行しますが、検査の対象となるは CICS API コマンドによるユーザー・ジャーナルへのアクセス試行のみであり、ファイル・リソース定義のジャーナル処理オプションに応じて実行されるジャーナル処理は対象外です。CICS API では、CICS トランザクションからジャーナルを読み取るための READ コマンドは提供されません。このため、ご使用の CICS システムへのアプリケーションのインストールを適切に管理すれば、CICS 内から読み取れないジャーナルに対して RACF 保護を追加する必要はないと考えることができます。

CICS ジャーナルのセキュリティーを実装する場合は、次のようにしてください。

1. ジャーナルに書き込むトランザクションの CSD リソース定義で RESSEC(YES) を指定します。
2. JCICSJCT または KCICSJCT リソース・クラス (あるいはユーザー定義のリソース・クラス名がある場合はそれらに相当するクラス) で、プロファイルを RACF に定義します。その際、CICS ジャーナル名を使用してプロファイルを識別します。

   ジャーナルをジャーナル・リソース・グループ・クラス内のプロファイルのメンバーとして、適切なアクセス・リストを指定して定義するには、次のコマンドを使用します。

   ```
   RDEFINE  KCICSJCT  userjnls UACC(NONE)
                      ADDMEM(JRNL001, JRNL002, ....)
                      NOTIFY(sys_admin_userid)
   PERMIT   userjnls  CLASS(KCICSJCT) ID(group_userid) ACCESS(UPDATE)
   ```

3. **SEC** システム初期設定パラメーターに YES を指定します。さらに、接頭部付きのプロファイルを定義する場合は **SECPRFX** に YES を設定します。CICS は、ジャーナルにアクセスするためには少なくとも UPDATE 権限が必要です。ジャーナルに適用されるシステム・プログラミング・コマンドのアクセス許可レベルについて詳しくは、 137 ページの『リソースおよびコマンドの検査の相互参照』を参照してください。
4. デフォルト・リソース・クラス名 JCICSJCT および KCICSJCT の場合は **XJCT** システム初期設定パラメーターに YES を指定し、ユーザー定義のリソース・クラス名の場合は XJCT=*class_name* を指定します。詳しくは、XJCT システム初期設定パラメーターを参照してください。

WRITE JOURNALNUM コマンドを使用するトランザクション

WRITE JOURNALNUM コマンドは、CICS Transaction Server for z/OS, バージョン 5 リリース 5 で、以前のリリースとの互換性のためにサポートされています。新規アプリケーションには、WRITE JOURNALNAME コマンドが推奨されます。リソース・セキュリティーが WRITE JOURNALNUM を実行しているトランザクションに適用される場合、セキュリティー検査が適用される前に、ジャーナル番号には「DFHJ」という接頭部が付きます。したがって、ジャーナル番号 2 への書き込みには、リソース DFHJ02 に対する UPDATE 権限が必要です。

開始されたトランザクションのセキュリティー

CICS トランザクションは、**EXEC CICS START** コマンドを使用して他のトランザクションを開始できます。この方法で始動したトランザクションは開始されたトランザクションと呼ばれており、CICS RACF セキュリティーを使用すると、**START** コマンドで他のトランザクションを開始できるユーザーを制御することができます。

トランザクションが **EXEC CICS START TRANSID** コマンドを発行すると、CICS は RACF を呼び出して、コマンドを発行したトランザクションのユーザーに、開始されるトランザクションに対する許可があるか検査します。

開始されたトランザクション、および XPCT クラスに対して検査されるトランザクションのセキュリティーを実装するには、次のようにします。

1. CICS システム初期設定パラメーターとして SEC=YES を指定する (接頭部付きのプロファイルを定義する場合は、さらに SECPRFX を指定する)。
2. **START** コマンドを発行するトランザクションの CSD リソース定義で RESSEC(YES) を指定する。
3. ACICSPCT および BCICSPCT のデフォルトのリソース・クラス名の場合は XPCT=YES を指定する (ユーザー定義のリソース・クラス名の場合は XPCT=class_name を指定する)。

   これにより、トランザクションが **START** コマンドによって開始されると、CICS は RACF を呼び出して、トランザクションに関連付けられているユーザー ID がトランザクションの接続を許可されているかを確認します。
4. ACICSPCT または BCICSPCT リソース・クラス (あるいはユーザー定義のリソース・クラス名がある場合はそれらに相当するクラス) で、プロファイルを RACF に定義する。その際、開始されたトランザクションの名前を使用して、プロファイルを識別します。

例えば、ACICSPCT クラスにトランザクションを定義し、1 人のユーザーのみを許可するには、次のコマンドを使用します。

```
RDEFINE  ACICSPCT (tran1, tran2, ..., trann) UACC(NONE)
                  NOTIFY(sys_admin_userid)
PERMIT  tran1 CLASS(ACICSPCT)  ID(userid) ACCESS(READ)
PERMIT  tran2 CLASS(ACICSPCT)  ID(userid) ACCESS(READ)
```

開始されたトランザクションを、開始されたトランザクション・リソース・グループ・クラス内のプロファイルのメンバーとして、適切なアクセス・リストを指定して定義するには、次のコマンドを使用します。

```
RDEFINE  BCICSPCT  started_trans  UACC(NONE)
                   ADDMEM(trana, tranb, ..., tranx)
                   NOTIFY(sys_admin_userid)
PERMIT started_trans  CLASS(BCICSPCT) ID(group_userid) ACCESS(READ)
```

5. TCICSTRN および GCICSTRN のデフォルトのリソース・クラス名の場合は XTRAN=YES を指定する (ユーザー定義のリソース・クラス名の場合は XTRAN=class_name を指定する)。
6. TCICSTRN または GCICSTRN リソース・クラス (あるいはユーザー定義のリソース・クラス名がある場合はそれらに相当するクラス) で、プロファイルを RACF に定義する。その際、開始されたトランザクションの名前を使用して、プロファイルを識別します。

端末で開始されるトランザクション

START により、CICS アプリケーション・プログラムは、START コマンドの発行元とは別の端末に関連付けられている、別のトランザクションを開始できます。例えば、termid1 で呼び出された CICS トランザクション tranid1 で発行された以下のコマンドは、termid2 で tranid2 という別のトランザクションを開始します。

```
EXEC CICS START
     TRANSID(tranid2)
     AT HOURS('18') MINUTES('50')
     TERMID(termid2)
```

TERMID が開始済みトランザクションに対して指定されている場合、CICS は、端末 (この例では termid2) に関連付けられているユーザー ID でクラス TCICSTRN およびクラス GCICSTRN を使用して、トランザクション接続セキュリティー検査を実行します。したがって、端末 (termid2) に関連付けられているユーザー ID が、トランザクションの呼び出しを許可されていることを確認する必要があります。このユーザー ID は、サインオン・ユーザーのユーザー ID であるか、サインオン・ユーザーがいない場合は CICS のデフォルトのユーザー ID です。 termid2 が許可されていない場合、メッセージ DFHAC2033 が termid2 のユーザーに対して発行されます。 START コマンドを発行した端末のユーザーは、「通常の」応答を受け取ります。開始済みトランザクションが RESSEC(YES) で定義されている場合、端末 (この例では termid2) に関連付けられているユーザー ID が、保護リソースへのアクセスを適切に許可されていることも確認します。

事前設定セキュリティーで定義されている端末でのタスクの開始:

一般に、端末に関連付けられている開始済みトランザクションは印刷タスクであり、その場合の指定された端末はプリンターです。

この場合、特定のユーザー ID を端末に関連付けるには、事前設定セキュリティーで端末を定義します。詳細については、『 83 ページの『Preset terminal security』』を参照してください。

端末なしで開始されるトランザクション

EXEC CICS START コマンドにより、CICS アプリケーション・プログラムは、どの端末とも関連付けられていない別のトランザクションを開始できます。 TERMID が開始済みトランザクションに指定されていない場合、新規トランザクションに関連付けられるユーザー ID は、USERID オプションも指定しているかどうかに応じて異なります。

非端末開始トランザクションのユーザー **ID:** START コマンドの USERID オプション (START コマンドに TERMID および USERID が含まれていない場合は端末ユーザー) は、非端末開始トランザクションのユーザー ID を決定します。
USERID オプションがない場合、非端末開始トランザクションは、START コマンドを実行したトランザクションと同じユーザー ID を持ちます。 USERID オプションが START コマンドに指定されている場合、指定されたユーザー ID が代わりに使用されます。

START コマンドが TERMID オプションの指定なしで実行された場合、CICS は代理ユーザー検査を実行して、トランザクションが非端末開始トランザクションによって使用されるユーザー ID に対して許可されていることを確認します。代理ユーザーのリンク許可については、 272 ページの『相互通信リンク・セキュリティー』を参照してください。代理ユーザーの EDF 許可については、 82 ページの『条件付きアクセス処理』を参照してください。

注**: START** コマンドをパラメーター **USERID**、**TERMID**、または **RTERMID** のいずれかを指定して実行した場合、またはそのコマンドが LU61 接続あるいは LU62 接続を介してシップされた場合、配布された ID 情報は保持されません。

非端末開始済みトランザクションによるリソースへのアクセス**:** USERID オプションが STARTBROWSE PROCESS コマンドで指定されていない場合、非端末開始済みトランザクションは、そのコマンドを実行したトランザクションのすべてのセキュリティーを必ずしも継承するわけではありません。さらにこれは、リンク・セキュリティーによって決定されたリソース・アクセス権限や、二重画面モードで使用されるときに EDF のユーザー ID によって決定されたリソース・アクセス権限も継承しません。これは、次のことを意味します。

- トランザクション・ルーティングされたトランザクションが START コマンドを実行する場合、または START コマンドが機能シップ済みである場合は、非端末開始済みトランザクションはリンク・セキュリティーの対象になりません。
- START コマンドを発行するトランザクションに対して EDF を二重画面モードで使用する場合、非端末開始トランザクションは、EDF 端末のユーザー ID によって決定されるリソース・アクセス権限の対象になりません。

開始済みトランザクションが、これから開始するトランザクションとまったく同じセキュリティー機能を持つようにしたい場合は、USERID オプションを省略します。 USERID オプションを指定しない場合、非端末開始済みトランザクションによるリソース・アクセス権限は、端末トランザクションのサインオン・パラメーターによって決定されます。これには RACF グループと、端末ユーザーがサインオンし

たエントリー・ポートが含まれます。エントリー・ポートとは、以下の例に示すように、サインオンに使用される端末またはコンソールです。

端末ユーザーは、NETNAMEX というネット名を持つ端末で、CESN トランザクションを使用してサインオンします。したがって、RACF の場合、エントリー・ポートは NETNAMEX です。 CESN 画面で、端末ユーザーはユーザー ID USERID1 およびグループ ID GROUPID2 を入力します。次に端末ユーザーは、TERMID オプションおよび USERID オプションを指定せずに EXEC CICS START コマンドを実行する端末トランザクションを実行します。非端末開始済みトランザクションには、ユーザー ID USERID1、グループ ID GROUPID2、およびエントリー・ポート NETNAMEX によって決定されるリソース・アクセス権限があります。

非端末トランザクションがリソースへのアクセスを RACF によって拒否された場合、生成されるエラー・メッセージには、端末サインオン・パラメーター、ユーザー ID、およびグループ ID が含まれることがあります。これにはエントリー・ポートが含まれることもあります。ユーザー ID、グループ ID、およびエントリー・ポートは、非端末トランザクションを開始した端末トランザクションから継承したものにできます。

USERID オプションが START コマンドで指定されている場合、非端末開始済みトランザクションは、USERID オプションで指定されたユーザー ID によって決定されるリソースへのアクセス権限を持ちます。

USERID オプションには、端末トランザクションの現行ユーザー ID を指定しないことをお勧めします。非端末開始トランザクションが、端末トランザクションと同じリソース・アクセス権限を持つことはできません。以下の例は、非端末開始済みトランザクションが異なるリソース・アクセス権限を持つことができる方法を示しています。

例 **1** RACF 条件付きアクセス・リストは、RACF PERMIT コマンドで WHEN(TERMINAL(...)) または WHEN(CONSOLE(...)) を指定して、特定のリソースへの端末トランザクション・アクセスを許可することによって使用できます。これは、指定されたエントリー・ポートが使用中であるためです。 82 ページの『条件付きアクセス処理』を参照してください。

START TRANSID USERID コマンドが、CESN でのサインオン時に端末ユーザーが入力したものと同じユーザー ID を指定する端末トランザクションによって実行される場合、開始済みトランザクションは、指定されたユーザー ID によって決定されるリソースへのアクセス権限を持ちますが、エントリー・ポートによって決定されるリソースへのアクセス権限は持ちません。

開始済みトランザクションは、EXEC CICS START USERID コマンドを実行した端末トランザクションに有効な条件付きアクセス・リストには従いません。

例 **2** RACF を使用すると、RACF 保護リソースへのグループ・アクセスを認可 (または拒否) することができます。

端末ユーザーは CESN でサインオンするときに、グループ ID とユーザー ID を入力できます。端末ユーザーが端末トランザクションを実行するときは、グループ ID がリソース・アクセス権限を決定できます。

START TRANSID USERID コマンドが、CESN でのサインオン時に端末ユーザーが入力したものと同じユーザー ID を指定する端末トランザクション

によって実行される場合、開始済みトランザクションは、指定されたユーザー ID によって決定されるリソースへのアクセス権限を持ちます。リソース・アクセス権限は、CESN でのサインオン時に端末ユーザーが入力したグループ ID では決定されません。非端末開始済みトランザクションのリソース・アクセス権限は、指定されたユーザー ID のデフォルト・グループ ID によって決定される場合があります。

非端末開始済みトランザクションは、START USERID コマンドを実行した端末トランザクションに対して有効なグループ・アクセスの対象にはなりません。

アクセス許可レベル

CICS には、開始済みトランザクションに対する最小権限である READ が必要です。

トランザクションに適用されるシステム・プログラミング・コマンドのアクセス許可レベルについては、137 ページの『リソースおよびコマンドの検査の相互参照』を参照してください。

EXEC CICS RUN TRANSID によって開始されたトランザクションのセキュリティー

CICS トランザクションでは、**EXEC CICS RUN TRANSID** コマンドを使用して他のトランザクションを開始できます。CICS RACF セキュリティーを使用すると、**RUN TRANSID** コマンドを使用して他のトランザクションを開始できるユーザーを制御することができます。

トランザクションが **EXEC CICS RUN TRANSID** コマンドを発行すると、CICS は RACF を呼び出して、コマンドを発行しているトランザクションのユーザーが、開始されたトランザクションに対して許可されているかどうかを確認します。

非同期トランザクションに対してセキュリティーを実装するには、以下を行う必要があります。

1. CICS システム初期設定パラメーターとして SEC=YES を指定する (接頭部付きのプロファイルを定義する場合は、さらに SECPRFX を指定する)。
2. **EXEC CICS RUN TRANSID** コマンドを発行する親トランザクションの CSD リソース定義で RESSEC(YES) を指定する。

   これにより、子トランザクションが **EXEC CICS RUN TRANSID** コマンドによって開始されると、CICS は RACF を呼び出して、トランザクションに関連付けられているユーザー ID がトランザクションの接続を許可されているかどうかを確認します。
3. ACICSPCT および BCICSPCT のデフォルトのリソース・クラス名の場合は XPCT=YES を指定する (ユーザー定義のリソース・クラス名の場合は XPCT=class_name を指定する)。
4. ACICSPCT または BCICSPCT リソース・クラス (あるいはユーザー定義のリソース・クラス名がある場合はそれらに相当するクラス) で、プロファイルを RACF に定義する。その際、開始された子トランザクションの名前を使用して、プロファイルを識別する。

5. TCICSTRN および GCICSTRN のデフォルトのリソース・クラス名の場合は XTRAN=YES を指定する (ユーザー定義のリソース・クラス名の場合は XTRAN=class_name を指定する)。
6. TCICSTRN または GCICSTRN リソース・クラス (あるいはユーザー定義のリソース・クラス名がある場合はそれらに相当するクラス) で、プロファイルを RACF に定義する。その際、開始された子トランザクションの名前を使用して、プロファイルを識別します。

EXEC CICS RUN TRANSID を使用して開始されたトランザクションのユーザー ID

EXEC CICS RUN TRANSID コマンドによって開始された子トランザクションは、このコマンドを発行した親トランザクションの USERID のもとで実行されます。

EXEC CICS RUN TRANSID を使用して開始されたトランザクションによるリソースへのアクセス

- トランザクション・ルーティングを行った親トランザクションが **EXEC CICS RUN TRANSID** コマンドを実行する場合、開始された子トランザクションはリンク・セキュリティーの対象になりません。
- **EXEC CICS RUN TRANSID** コマンドを発行するトランザクションに対して EDF が二重画面モードで使用される場合、開始されたトランザクションには、EDF 端末のユーザー ID によって決まるリソース・アクセスは適用されません。

アクセス許可レベル

CICS では、**EXEC CICS RUN TRANSID** コマンドによって開始されたトランザクションに対し、少なくとも READ 権限が必要です。

XPCT 検査トランザクションのセキュリティー

XPCT システム初期設定パラメーターは、**EXEC CICS** コマンド (**EXEC CICS START** および **EXEC CICS RUN TRANSID** など) によって開始されたトランザクションに対して、CICS がトランザクション・リソース・セキュリティー検査を実行するかどうかを指定します。 ACICSPCT リソース・クラス・プロファイルおよび BCICSPCT リソース・クラス・プロファイルで定義されたトランザクションは、XPCT セキュリティー検査の対象になります。

コマンドを発行するトランザクションが RESSEC(YES) で定義されている場合、これらのプロファイルはさらに、他の特定の **EXEC CICS** コマンドで指定されたトランザクションへのアクセスを制御します。 XPCT 検査の影響を受ける **EXEC CICS** コマンドは、以下のとおりです。

- START
- COLLECT STATISTICS TRANSACTION
- DISCARD TRANSACTION
- INQUIRE TRANSACTION
- SET TRANSACTION
- INQUIRE REQID
- CANCEL
- RUN TRANSID

アプリケーション・プログラムのセキュリティー

トランザクション・リソース定義に指定された初期プログラムへのアクセスを制御するには、トランザクションの開始をユーザーに許可します (トランザクション接続セキュリティー)。

一方で、CICS アプリケーション・プログラムは、LINK、LOAD、および XCTL コマンドを使用して他のプログラムを呼び出すことができます。また、プログラムのロード状況は、CICS RELEASE、ENABLE、および DISABLE コマンドによって変更される場合があります。ただし、タスクの存続期間中にロードされたプログラムの RELEASE に関する個別のセキュリティー検査はないため、注意してください。これは、対応する LOAD に対して行われます。

これらのコマンドを使用して呼び出されたプログラムへのアクセスを制御するには、CICS アプリケーション・プログラム・クラスにプロファイルを定義します。このクラスは XPPT システム初期設定パラメーターで CICS に定義されます。

他のプログラムのロード状況を呼び出しまたは変更できるユーザーを制御するには、次のようにします。

1. LINK、LOAD、XCTL、CICS RELEASE、ENABLE、または DISABLE コマンドを使用するトランザクションの CSD リソース定義に RESSEC(YES) を指定する。
2. MCICSPPT または NCICSPPT リソース・クラス (あるいはユーザー定義のリソース・クラス名がある場合はそれらに相当するクラス) で、プロファイルを RACF に定義する。その際、LINK、LOAD、または XCTL コマンドで呼び出されるプログラムの名前を使用して、プロファイルを識別します。

   例えば、MCICSPPT クラスにプログラムを定義し、1 人のユーザーのみを許可するには、次のコマンドを使用します。

   ```
   RDEFINE  MCICSPPT  (prog1, prog2, ..., progn)  UACC(NONE)
                    NOTIFY(sys_admin_userid)
   PERMIT  prog1 CLASS(MCICSPPT)  ID(userid) ACCESS(READ)
   PERMIT  prog2 CLASS(MCICSPPT)  ID(userid) ACCESS(READ)
   ```

   プログラムをアプリケーション・プログラム・リソース・グループ・クラス内のプロファイルのメンバーとして、適切なアクセス・リストを指定して定義するには、次のコマンドを使用します。

   ```
   RDEFINE  NCICSPPT  cics_programs  UACC(NONE)
                      ADDMEM(proga, progb, ..., progx)
                      NOTIFY(sys_admin_userid)
   PERMIT cics_programs CLASS(NCICSPPT) ID(group_userid) ACCESS(READ)
   ```
3. CICS システム初期設定パラメーターとして SEC=YES を指定する (接頭部付きのプロファイルを定義する場合は、さらに SECPRFX を指定する)。
4. デフォルト・リソース・クラス名 MCICSPPT および NCICSPPT の場合は、CICS システム初期設定パラメーターとして XPPT=YES を指定する (ユーザー定義のリソース・クラス名の場合は XPPT=class_name を指定する)。

分散プログラム・リンク (DPL) コマンドの例外

CICS は、**LINK** コマンドで参照されるプログラムがリモート・プログラムであることが判明すると、リンク・コマンドが発行される領域でセキュリティー検査を実行しません。セキュリティー検査は、リンク先プログラムが最終的に実行される CICS 領域でのみ実行されます。

例えば、CICSA が DPL コマンドを CICSB に機能シップし、CICSB でプログラムが実行される場合は、CICSB がセキュリティー検査を発行します。DPL 要求が CICSC に再び機能シップされた場合、セキュリティー検査を発行するのは CICSC です。

アクセス許可レベル

CICS には、プログラムに対する最小権限である READ が必要です。

プログラムに適用されるシステム・プログラミング・コマンドのアクセス許可レベルについては、137 ページの『リソースおよびコマンドの検査の相互参照』を参照してください。

一時記憶域のセキュリティー

RESSEC(YES) を指定するその他のリソースとは異なり、RACF 保護を必要とする一時記憶域キューは、適切な TSMODEL リソース定義のセキュリティー属性も必要とします。

CSD に TSMODEL 定義を指定します。TSMODEL リソース定義について詳しくは、TSMODEL リソースを参照してください。

一時記憶域キューのセキュリティーの実装

一時記憶域キューのセキュリティーを実装するには、次のようにします。

1. 適切なトランザクションの CSD リソース定義で RESSEC(YES) を指定します。
2. 適切な TSMODEL リソース定義でセキュリティー属性を指定します。 CICS は、対応する TSMODEL 定義で SECURITY=NO を指定する一時記憶域キューに対しては、どのようなセキュリティー検査も実行しません。
3. プロファイルを SCICSTST または UCICSTST リソース・クラス (またはユーザー定義のリソース・クラス名があればこれと同等のもの) 内の RACF に対して、必要に応じてアクセス・リストを使用して定義します。例えば、SCICSTST クラスにキューを定義し、それらのキューに対する読み書き両方の権限をユーザーに付与するには、次のコマンドを使用します。

   ```
   RDEFINE  SCICSTST  (tsqueue1, tsqueue2, ..., tsqueuen) UACC(NONE)
                      NOTIFY(sys_admin_userid)
   PERMIT tsqueue1 CLASS(SCICSTST) ID(group1, group2) ACCESS(UPDATE)
   PERMIT tsqueue2 CLASS(SCICSTST) ID(group1, group2) ACCESS(UPDATE)
   ```

   適切なアクセス・リストを使用して、CICS 一時記憶域リソース・グループ・クラス内のプロファイルのメンバーとして一時記憶域キューを定義するには、次のコマンドを使用します。

```
RDEFINE  UCICSTST  tsqueue_group  UACC(NONE)
                   ADDMEM(tsqueuea, tsqueueb, ..., tsqueuex)
                   NOTIFY(sys_admin_userid)
PERMIT tsqueue_group CLASS(UCICSTST) ID(group_userid) ACCESS(UPDATE)
```

一時記憶域プロファイルの定義の詳細については、『一時記憶域のセキュリティーに関するその他の考慮事項』を参照してください。

4. CICS システム初期設定パラメーターとして SEC=YES を指定する (接頭部付きのプロファイルを定義する場合は、さらに SECPRFX を指定する)。
5. デフォルトのリソース・クラス名 SCICSTST および UCICSTST の CICS システム初期設定パラメーターとして、XTST=YES を指定します (またはユーザー定義リソース・クラス名に対して XTST=class_name を指定します)。

一時記憶域のセキュリティーに関するその他の考慮事項

TSMODEL リソース定義の PREFIX 属性で、次のようにキュー名を定義できます。

- READQ TS または WRITEQ TS コマンドで指定されたキュー名と正確に一致した、完全に特定される名前を指定します。これは、1 から 16 文字の英数字です。
- 一連のキュー名の先頭の英数字に対応する総称名 (接頭部) を指定します。

  したがって接頭部は、1 から 15 文字までということになります。なぜなら、キュー名として 16 文字すべてを指定した場合、それは特定の一時記憶域キューの名前であるからです。

一時記憶域セキュリティーが有効である場合に CICS アプリケーションが一時記憶域コマンド (例えば、DELETEQ TS、READQ TS、または WRITEQ TS) を発行すると、CICS は、キュー名の先行文字に対応する TSMODEL リソース定義を検索します。

一時記憶域キュー名に 16 進文字を入れると、予測できない結果が生じる可能性があるため、注意してください。また、一時記憶域キュー名に組み込みブランクが含まれている場合、RACF はリソース名をそのブランクで切り捨てます。

アクセス許可レベル

システム初期設定パラメーターとして SEC=YES を指定してセキュリティーを指定する場合、CICS は、意図した一時記憶域キュー・アクセスに適した許可レベルを必要とします。例えば、READQ TS の場合は最小で READ、DELETEQ TS および WRITEQ TS の場合は最小で UPDATE です。

一時記憶域キューに適用されるシステム・プログラミング・コマンドのアクセス許可レベルについては、 137 ページの『リソースおよびコマンドの検査の相互参照』を参照してください。

z/OS UNIX ファイルのセキュリティー

z/OS UNIX システム・サービス・ファイル・システムに保管されているファイルを使用すると、URIMAP 定義によって提供される静的応答として、CICS Web サポートを介して Web ページを提供できます。これらのファイルに対するアクセス制御が指定されている場合は、個々の Web クライアントのユーザー ID に基づい

て、これらのファイルへのアクセスを制御できます。デフォルトでは、z/OS UNIX ファイルのアクセス制御が有効になっています。

z/OS UNIX ファイルのアクセス制御は、XHFS システム初期設定パラメーターによってアクティブ化されます。このパラメーターのデフォルトは YES で、z/OS UNIX ファイルのリソース・セキュリティーがアクティブになることを意味します。こうしたファイルのリソース・セキュリティーをアクティブにしない場合には、このシステム初期設定パラメーターを NO に設定してください。

z/OS UNIX ファイルのアクセス制御は、基本認証を使用して Web クライアントから取得したユーザー ID、または Web クライアントにより送信されたクライアント証明書に関連付けられたユーザー ID に基づいています。ユーザー ID が使用されるのは、セキュリティー検査プロセス中のみです。

z/OS UNIX ファイルのアクセス制御は、X*name* システム初期設定パラメーターによって制御される他のリソース・タイプの標準リソース・セキュリティーとは、以下の幾つかの重要な点で異なります。

- z/OS UNIX ファイルに対するアクセス制御は、RACF によっては直接に管理されません。このアクセス制御は、RACF を使用してユーザー ID およびユーザー ID のグループを管理する z/OS UNIX システム・サービスで指定されますが、ファイルおよびディレクトリーに対する許可セットの制御を保持します。このため、個々のファイルに RACF プロファイルを定義する必要がなく、これらのプロファイルへのアクセスを検査するために QUERY SECURITY コマンドを使用することはできません。 z/OS UNIX ファイルおよびディレクトリーに対する許可の検査と指定は、z/OS UNIX システム・サービスのシェル環境で、z/OS UNIX コマンドを使用して行います。 RACF は、ユーザー・プロファイル、グループ、およびアクセス制御リスト (ACL) の管理に使用されます。 ACL を使用する場合は、検査される ACL に対する FSSEC クラスをアクティブにする必要があります。
- z/OS UNIX ファイルに対するセキュリティー検査は、ファイルにアクセスするトランザクションの TRANSACTION リソース定義にある RESSEC 属性に影響されません。 CICS 領域のシステム初期設定パラメーターとして XHFS=YES を指定すると、CICS Web サポートによって静的応答として使用されるすべての z/OS UNIX ファイル (およびそのディレクトリー) は、そうしたファイルやディレクトリーにアクセスするトランザクションの RESSEC 属性にかかわりなく、セキュリティー検査の対象となります。(ただし、SEC システム初期設定パラメーターは、すべてのリソースに関してセキュリティー検査を実行するかどうかに影響を及ぼします。)
- z/OS UNIX ファイルが CICS アプリケーション・プログラミング・コマンドまたはシステム・プログラミング・コマンドによって直接参照されることはありません。こうしたファイルを参照できるのは、ファイルが CICS 文書テンプレートとして定義されている場合の EXEC CICS コマンドに限られます。その場合は、XRES システム初期設定パラメーターによって指定する CICS 文書テンプレートのリソース・セキュリティーによって、それらへのユーザー・アクセスが制御されます。 CICS は、Web クライアントのユーザー ID を使用して z/OS UNIX ファイルの追加の許可検査を実行することはありません。アクセス制御が CICS 領域の z/OS UNIX ファイルに対して指定されている場合であっても、またはリソース・セキュリティーが文書テンプレートでアクティブになっていない場合

であっても、これは当てはまります。z/OS UNIX ファイルが CICS 文書テンプレートとして定義されている場合、Web クライアントのユーザー ID アクセス制御をセットアップする必要があるのは、z/OS UNIX ファイルの z/OS UNIX システム・サービスではなく、CICS 文書テンプレートの RACF です。(ただし、CICS 領域のユーザー ID には、文書テンプレートとして定義されている場合であっても、z/OS UNIX ファイルでの読み取り権限が必ず必要になります。) これらの事柄は CICS Web サポートからのすべてのアプリケーション生成の応答、および HFSFILE 属性ではなく TEMPLATENAME 属性が使用されている静的応答の URIMAP 定義すべてに当てはまることに特に注意してください。

z/OS UNIX ファイルのセキュリティーのインプリメント

CICS Web サポートで使用される z/OS UNIX ファイルが、HFSFILE 属性を使用して URIMAP 定義で静的応答として指定されている場合に、これらのファイルに関するアクセス制御をインプリメントするには、このトピックに示されたステップに従います。

始める前に

CICS 領域ユーザー ID は、CICS Web サポートに使用するすべての z/OS UNIX ファイル、およびそれらが含まれるディレクトリーに対して、常に読み取りおよび実行以上の権限を持っていなければなりません。Web クライアントのユーザー ID が使用されるのは、z/OS UNIX ファイルに静的応答としてアクセスしている場合のみです。ファイルにアクセスしようとするその他のすべての試行には、CICS 領域ユーザー ID が適用されます。CICS 領域ユーザー ID にファイルへのアクセス権限がない場合は、権限のある Web クライアントであっても、ファイルを表示することができません。ファイルが CICS 文書テンプレートとして定義されている場合も、同様です。

このタスクについて

手順

1. z/OS UNIX ファイルおよびディレクトリーにアクセスする権限を Web クライアントに付与するために、適切な方式を選択します。 ファイルおよびディレクトリーに対してグループ権限を使用するのか、またはアクセス制御リスト (ACL) を使用するのかを選択することができます。 グループ権限を使用できる場合でも、Web クライアントのユーザー ID に権限を付与する場合は、ACL を使用することをお勧めします。ACL を使用すると、複数のユーザー・グループにアクセスを許可することができます。そのアクセスは、1 つのファイルに限定したり、ディレクトリー内のすべてのファイルに 1 回ずつ設定することができます。ディレクトリー権限は同じ方法で設定することができます。また、ACL コマンドを使用して、ファイルおよびディレクトリーの権限を変更することもできます。 z/OS UNIX システム・サービス・シェル環境で ACL を使用していても、ACL は RACF によって作成およびチェックされるため、別のセキュリティー製品を使用している場合は、対応する文書を参照して、ACL がサポートされているかどうか確認してください。 推奨方式が選択されている場合は、この手順の残りの関連ステップに従ってください。
2. Web クライアントが使用する認証済みユーザー ID を識別します。これらはアクセス制御に基づいている必要があります。 (アプリケーションが生成する応答の場合とは異なり、アナライザー・プログラムを使用してオーバーライド

を提供することはできません。) 認証済みユーザー ID は、すでにセキュリティー・マネージャーにユーザー・プロファイルが定義されています。

3. Web クライアント・ユーザー ID ごとに、適切な z/OS UNIX ユーザー ID (UID) を選択し、割り当てます。UID は、0 から 16 777 216 までの範囲の数値です。UID を割り当てるには、ユーザー ID ごとに、ユーザー・プロファイルの OMVS セグメントで UID 値を指定します。 ALTUSER コマンドを使用して RACF ユーザー・プロファイルを更新する方法については、『 15 ページの『RACF ユーザー・プロファイル』』を参照してください。 例えば、Web クライアントのユーザー ID が WEBUSR1 で、割り当てる UID が 2006 の場合は、以下のコマンドを使用します。

   ```
   ALTUSER WEBUSR1 OMVS(UID(2006))
   ```

   z/OS UNIX ユーザー ID (UID) に基づいて権限を割り当てていない場合でも、z/OS UNIX 機能を使用するためには、すべてのユーザーのユーザー・プロファイルに UID が設定されていなければなりません。z/OS UNIX システム・サービス計画には、z/OS UNIX システムの UID と GID を管理する方法が説明されています。

4. 同じ権限を持つ Web クライアント・グループで使用できる RACF グループを選択するか、または作成します。 パフォーマンスを最適化するには、ACL を使用している場合でも、個別のユーザーでなくグループに権限を割り当てる必要があります。

5. RACF グループごとに、適切な z/OS UNIX グループ ID (GID) を選択し、RACF に GID を割り当てます。GID を割り当てるには、RACF グループ・プロファイルの OMVS セグメントに GID 値を指定します。 例えば、RACF グループが CICSWEB1 で、割り当てる GID が 9 の場合は、以下のコマンドを使用します。

   ```
   ALTGROUP CICSWEB1 OMVS(GID(9))
   ```

6. Web クライアントのユーザー ID それぞれが、z/OS UNIX グループ ID (GID) を割り当てた RACF グループに接続するようにします。 ご使用の Web クライアントを複数の RACF グループに接続する必要がある場合は、RACF のグループ・リストをシステムでアクティブにする必要があります。

7. z/OS UNIX ファイルおよびディレクトリーの権限を変更する前に、ユーザー ID が z/OS UNIX のスーパーユーザーになっているか、または作業する各 z/OS UNIX ファイルおよびディレクトリーの所有者になっていることを確認します。グループで作業する場合は、ファイルおよびディレクトリーの所有者を、使用中の RACF グループに接続する必要もあります。

8. オプション: ACL を使用するように選択した場合は、z/OS UNIX システム・サービス・シェル環境で `setfacl` コマンドを使用して、静的応答のために CICS Web サポートで使用されるすべての z/OS UNIX ファイルおよびディレクトリーに適用する ACL をセットアップします。 ACL の使用に関する情報、および `setfacl` コマンドの使用法を示す例については、z/OS UNIX システム・サービス計画を参照してください。

   a. ファイルに対しては、各ファイルに適用されるアクセス ACL をセットアップしたり、ディレクトリー内およびサブディレクトリー内のすべてのファイルに適用されるファイル・デフォルト ACL を設定したりすることができます。

b. ディレクトリーに対しては、各ディレクトリーに適用されるアクセス ACL をセットアップしたり、ディレクトリー内のサブディレクトリーに適用されるディレクトリー・デフォルト ACL を設定したりすることができます。

c. パフォーマンスへの影響を最小化するには、個別のユーザー ID を使用しないで、Web クライアントのユーザー ID が接続される RACF グループにファイルのグループ権限を割り当てます。 (ACL で指定できる項目数にも制限があります)。

d. グループに付与された権限 (読み取り、書き込み、および実行アクセスを指定する基本権限ビット) を変更する必要がある場合は、`setfacl` コマンドを同様に使用して指定することができます。 Web クライアントには、z/OS UNIX ファイルおよびディレクトリーへの読み取りアクセス権限が必要です。

e. ACL を使用している場合は、FSSEC クラスがアクティブであることを確認します。このためには、RACF コマンド `SETROPTS CLASSACT(FSSEC)` を使用します。 FSSEC クラスをアクティブにする前に ACL を定義することができますが、アクセス判断で ACL を使用する前に、FSSEC クラスをアクティブにする必要があります。

9. オプション: ACL を使用しないでグループ権限を使用するように選択した場合は、Web クライアントが接続されたグループに、各 z/OS UNIX ファイルおよびディレクトリーに対するグループ権限を割り当てて、グループ 読み取り権限を付与します。このためには、UNIX コマンド `chmod` を使用します。このコマンドの使用法については、z/OS UNIX システム・サービス コマンド解説書および z/OS UNIX システム・サービス ユーザーズ・ガイドを参照してください。 (この方法を使用する場合、グループ権限は 1 つのグループにしか割り当てることができないため、正しい権限をすべて獲得するには、Web クライアントのユーザー ID のいくつかを複数のグループに接続する必要があるかもしれません)。
10. CICS システム初期化パラメーターとして SEC=YES を指定します (SECPRFX には RACF プロファイルがないため、z/OS UNIX ファイルに関連しません)。
11. CICS システム初期化パラメーターとして XHFS=YES を指定します。 このステップを実行すると、CICS 領域内のすべての z/OS UNIX ファイルのアクセス制御がアクティブになります。

タスクの結果

セットアップ手順を完了すると、この時点以降、以下のようになります。

- 基本認証またはクライアント証明書認証による接続を使用し、HFS ファイルへのアクセスを試行するすべての Web クライアントでは、セキュリティー・マネージャー内にユーザー・プロファイルが必要です。ただし、このセキュリティー・マネージャーに、有効な z/OS UNIX UID を格納し、有効な z/OS UNIX GID を使用して RACF グループに接続する必要があります。
- z/OS UNIX ファイルから取得された Web ページを表示するには、基本認証またはクライアント証明書認証による接続を使用する Web クライアントに対して、このファイルおよびファイルが格納されたディレクトリーへの読み取り権限を個別に、または接続先の RACF グループを介して設定する必要があります。

これらの条件が適切に設定されていない場合、Web クライアントは 403 (Forbidden) 状況コードを受け取り、CICS は メッセージ DFHXS1116 を発行します。

アクセス許可レベル

Web クライアントのユーザー ID、および CICS 領域ユーザー ID は、CICS Web サポートにおいて URIMAP 定義で提供される静的応答として使用される (HFSFILE 属性により指定される) z/OS UNIX ファイル、およびそれらが含まれるディレクトリーに対して、読み取り以上のアクセス権限が必要です。

z/OS UNIX ファイルが CICS 文書テンプレートとして定義されていて、文書テンプレートが URIMAP 定義で静的応答として使用される (TEMPLATENAME 属性により指定される) か、またはアプリケーションによって使用される場合、CICS はこのファイルに対し、Web クライアントのユーザー ID を使用して追加の許可検査を実行することはありません。ただし CICS 領域ユーザー ID は、ファイルが文書テンプレートとして定義されている場合でも、ファイルに対する読み取り権限が必要です。

z/OS UNIX ファイルは、CICS アプリケーション・プログラミング・コマンドまたはシステム・プログラミング・コマンドによって直接操作されることはありません。これらのファイルは、CICS 文書テンプレートとして定義されている場合にのみ、EXEC CICS コマンドによって操作されます。この場合は、CICS 文書テンプレートのリソース・セキュリティーによって、ファイルへのユーザーのアクセスが制御されます。

プログラム仕様ブロックのセキュリティー

DL/I プログラム仕様ブロック (PSB) は、アプリケーション・プログラムによって使用されるデータベースおよび論理メッセージ宛先を記述する IMS 制御ブロックです。PSB は 1 つ以上のプログラム連絡ブロック (PCB) で構成されており、これらは IMS データベースに対するアプリケーション・プログラムのインターフェースを記述しています。

CICS アプリケーションにスケジュールされた PSB のセキュリティーを実装するには、次のようにします。

1. PCICSPSB または QCICSPSB リソース・ (あるいはユーザー定義のリソース・クラス名がある場合はそれらに相当するクラス) で、適切なアクセス・リストを指定して、プロファイルを RACF に定義する。 RACF に定義するリソース・プロファイル名は、CICS PSB スケジュール・コマンドで指定された PSB の名前に対応しなければなりません。例えば、PCICSPSB クラスに PSB を定義し、ユーザーに対してこれらのキューへのアクセスを許可するには、以下のコマンドを使用します。

   ```
   RDEFINE  PCICSPSB  (psbname1, psbname2, ..., psbnamen) UACC(NONE)
                      NOTIFY(sys_admin_userid)
   PERMIT psbname1 CLASS(PCICSPSB) ID(group1, group2) ACCESS(READ)
   PERMIT psbname2 CLASS(PCICSPSB) ID(group1, group2) ACCESS(READ)
   ```

   PSB を CICS PSB リソース・グループ・クラス内のプロファイルのメンバーとして、適切なアクセス・リストを指定して定義するには、次のコマンドを使用します。

```
RDEFINE  QCICSPSB  psbname_group UACC(NONE)
                   ADDMEM(psbnamea, psbnameb, ..., psbnamex)
                     NOTIFY(sys_admin_userid)
PERMIT psbname_group CLASS(QCICSPSB) ID(group_userid) ACCESS(UPDATE)
```

2. CICS システム初期設定パラメーターとして SEC=YES を指定する (接頭部付きのプロファイルを定義する場合は、さらに SECPRFX を指定する)。
3. PCICSPSB および QCICSPSB のデフォルト・リソース・クラス名の CICS システム初期化パラメーターとして、XPSB=YES を指定する (あるいは、ユーザー定義のリソース・クラス名の場合は XPSB=class_name を指定する)。
4. トランザクション・ルーティングされたトランザクションでアクセスされる PSB に対して完全なセキュリティーが必要な場合は、PSBCHK=YES を指定する。これは、DL/I インターフェースの両方のタイプ (リモートおよび DBCTL) に適用されます。PSBCHK=NO を指定すると、トランザクション・ルーティングされたトランザクションではリモート・ユーザーの権限は使用されません。

注: CICS は、PSB に対して少なくとも READ 権限が必要です。

DBCTL を使用している場合、CICS-DBCTL 環境でのセキュリティーの定義については、DBCTL のセキュリティー検査を参照してください。

CEDF、CEDG、CEDX、または CEDY の下で実行されるトランザクションのセキュリティー検査

この 4 つの EDF トランザクションのいずれかの下でトランザクションが実行される場合、CICS はターゲット・トランザクションのセキュリティー設定を検査します。

IBM が提供する DFHEDFグループ内の CEDF、CEDG、CEDX、および CEDY の定義では、RESSEC(YES) が指定されます。IBM 提供のグループ内の定義は変更できません。定義を変更するには、トランザクションを別のグループにコピーします。

CEBR および CECI が CEDF 内から呼び出される場合は、トランザクション接続検査が実行されます。

4 つの EDF トランザクションのいずれかを使用してトランザクションがテストされる場合は、テスト対象のトランザクションを実行するユーザーの権限が検査されます。テスト対象のトランザクションがアクセスする各リソースに対して、ユーザーはアクセス権限を持っている必要があります。アクセス権限がないと、NOTAUTH 状態が発生します。この要件は、以下のすべてのリソース検査に適用されます。

- トランザクション接続
- CICS リソース
- CICS コマンド
- QUERY SECURITY コマンドを通してアクセスされる非 CICS リソース
- 代理ユーザー

リソースの総称プロファイルの定義

トランザクション接続セキュリティーによって CICS トランザクションへのアクセスを制御する場合、RACF 保護のレベルをさらに引き上げる必要がある他のリソース・タイプのサブセットはおそらくごくわずかです。

例えば、CICS アプリケーション・プログラム・リソース・クラス内でとりわけ重要であるプログラムはごくわずかであり、それよりはるかに多くのプログラムには重大なリスクはないという場合があります。この場合には、重要なプログラムのみを対象とした特定の RACF プロファイルを定義することで、そのわずかなプログラムを保護できます。以下のように、完全な総称リソース・プロファイルを定義することによって、すべてのユーザーはその他の重要でないプログラムに確実にアクセスできるようになります。

```
RDEFINE  MCICSPPT  *  UACC(READ) ...
```

このプロファイルは、特定のプロファイルのいずれによってもカバーされていない、プログラムのすべての許可要求に適用されます。 RACF 処理ロジックはそのようなものであるので、特定のリソース名の最も具体的なプロファイルが必ず使用されます。

プロファイルが総称であるかどうかは、プロファイル名を RLIST または SEARCH を使用してリストしたときに、プロファイル名の後ろに「G」が付いているかどうかを確認するだけで分かります。以下に例を示します。

SEARCH CLASS(TCICSTRN)

次の出力が表示されるとします。

```
C*
CED% (G)
** (G)
```

この出力は、CED% と ** の両方が総称プロファイルであることを示しています。C* プロファイルは後ろに (G) が付いていないので、総称プロファイルではありません。総称プロファイルが SETROPTS コマンドで有効にされるよりも前に C* プロファイルが作成された場合、このようになる可能性があります。 C* プロファイルは削除して、以下のようにして適正な総称プロファイルとして再定義できます。

```
SETROPTS NOGENERIC(TCICSTRN)
SETROPTS NOGENCMD(TCICSTRN)
RDEL TCICSTRN C*
SETROPTS GENERIC(TCICSTRN)
RDEFINE TCICSTRN C* UACC(NONE)
```

すべてにアクセスするかまたはどれにもアクセスしないかの選択

RACF は、特定のプロファイルと総称プロファイルのどちらも検出できない場合には、「プロファイルが見つかりません」条件を返します。

CICS は、この戻りコードを「ユーザーは使用を許可されていません」戻りコードとまったく同じように扱い、NOTAUTH 条件を CICS アプリケーション・プログラムに返します。 RACF が APPL クラスを検出できない場合は、「READ アクセス・インテント」条件が返されます。

完全な総称プロファイルを使用して、その他のより具体的なプロファイルでカバーされていないすべてのリソースへのアクセスを許可するか、または UACB(READ|UPDATE) オプションあるいは UACC(NONE) オプションを使用して、アクセスを防止することができます。例えば、以下のようにします。

```
RDEFINE  DCICSDCT  *  UACC(NONE)
```

これにより、RACF に対して定義されている他のどのプロファイルにもカバーされていない一時データ・キューにアクセスできなくなり、結果として RACF は SMF レコードを書き込みます。

一方、以下のコマンドによって、ファイルを「公開」として定義できます。

```
RDEFINE  FCICSFCT  *  UACC(READ)
```

総称プロファイルを使用する場合は、同じ POSIT 値を持つ CICS クラスのいずれか 1 つに対して SETROPTS GENERIC(*classname*) コマンドを発行することによって、総称プロファイル検査が、CICS RACF リソース・クラス (IBM 提供クラス、および RACF クラス記述子テーブルに追加されている任意のインストール定義クラスの両方) に対してアクティブ化されていることを確認します。これにより、同じ POSIT 値を持つ他のすべての CICS クラスの総称検査を確実に実行できます。総称プロファイルを変更する場合、SETROPTS GENERIC(*classname*) REFRESH コマンドを発行する必要があります。 POSIT 値および総称クラスの定義について詳しくは、z/OS Security Server RACF システム・プログラマーのガイドを参照してください。

XRES リソース・セキュリティー・パラメーターを使用したセキュリティー

セキュリティー・プロファイルおよび **XRES** システム初期設定パラメーターを使用して、CICS リソースのサブセットをセキュリティー検査します。CICS 文書テンプレートにリソース・セキュリティーを実装する方法の例を示します。

このタスクについて

CICS セキュリティー・プロファイル名は 3 つの部分から成っており、*security_prefix.resource_type.resource_name* という形式です。 CICS プロファイルは、検査のためにセキュリティー・マネージャーに渡されます。セキュリティー検査は、大文字小文字の区別があります。

セキュリティーの接頭部
: *security_prefix* は、**SECPRFX** システム初期設定パラメーターで指定された値です。 **SECPRFX** パラメーターのデフォルト値は NO です。これは、*security_prefix* が省略されることを意味します。 **SECPRFX** パラメーターの値が YES の場合、*security_prefix* は領域ユーザー ID の名前です。あるいは、*security_prefix* に 1 から 8 文字の値を指定することもできます。

リソース・タイプ
: *resource_type* は、検査が実行されるリソースのタイプを指定します。ほとんどの場合、各 CICS リソースには対応するセキュリティー・プロファイルがあります。例えば、ATOMSERVICE リソースには以下のように、対応するリソース・タイプに関するセキュリティー・プロファイルがあります。

  security_prefix.ATOMSERVICE.*resource_name*

ただし、ある特定の CICS リソースには対応するセキュリティー・プロファイルがありません。例えば、BUNDLEPART、OSGIBUNDLE、および OSGISERVICE リソースは、BUNDLE セキュリティー・プロファイルを使用して検査されます。

以下のリソース・タイプを使用するセキュリティー・プロファイルを作成できます。

- ATOMSERVICE
- BUNDLE
- DOCTEMPLATE
- EPADAPTER
- EPADAPTERSET
- EVENTBINDING
- JVMSERVER
- XMLTRANSFORM

セキュリティー検査に関連付けられた CICS リソース、リソース・タイプ、およびコマンドの完全なリストについては、137 ページの『リソースおよびコマンドの検査の相互参照』を参照してください。

注: あるユーザーにプラットフォームまたはアプリケーションに対してアクションを実行する権限を付与すると、そのプラットフォームまたはアプリケーションに対して動的に生成されるリソースについても、そのユーザーが同じアクションを実行できる権限を付与することになります。アプリケーションまたはプラットフォームを介して CICS バンドルを作成または操作する場合、CICS リソース・セキュリティー検査は実行されません。ただし、個々の BUNDLE リソース、または CICS バンドルに定義されたリソースに対してアクションを直接実行するときは、バンドルがプラットフォームまたはアプリケーションのインストール時に作成された場合でも、CICS リソース・セキュリティー検査が適用されます。詳しくは、バンドルのセキュリティーを参照してください。

リソース名

resource_name は、CICS リソースの名前を指定します。

例

このタスク例は、CICS 文書テンプレートにリソース・セキュリティーを実装する方法を示します。

CICS 文書テンプレートは、以下の場合に制御されます。

- Web クライアントの要求に対する静的応答として送信された文書テンプレート (要求に対する URIMAP 定義の TEMPLATENAME 属性上で指定)。
- Web クライアントの要求に対するアプリケーション生成の応答の一部として送信された文書テンプレート (要求を処理するアプリケーション・プログラムによって使用されます)。
- すべての **EXEC CICS CREATE**、**INQUIRE**、**DISCARD**、および **SET DOCTEMPLATE** コマンド。

- すべての **EXEC CICS DOCUMENT INSERT** コマンドおよび **CREATE** コマンド (TEMPLATE オプションを指定)。

EXEC CICS DOCUMENT コマンドは、DOCTEMPLATE リソースの TEMPLATENAME 属性で指定された 48 文字のテンプレート名を使用して、文書テンプレートを参照します。ただし、これらのコマンドに対するセキュリティー検査では、TEMPLATENAME 属性に対応する DOCTEMPLATE リソースの名前が使用されます。そのため、TEMPLATENAME 属性ではなく DOCTEMPLATE リソースの名前を使用して、文書テンプレートごとに 1 つのプロファイル名をセットアップできます。

注: 区分データ・セット、CICS プログラム、CICS ファイル、z/OS UNIX システム・サービスのファイル、一時記憶キュー、一時データ・キュー、出口プログラムなど、さまざまなソースから文書テンプレートを取得することができます。リソース・セキュリティー検査が文書テンプレートに対して行われる場合、文書テンプレートを提供するリソースについては、CICS は追加のセキュリティー検査を実行しません。CICS 領域で、そのタイプのリソースに対してリソース・セキュリティーが指定されていても、同様です。

CICS 文書テンプレートにリソース・セキュリティーを実装するには、以下のステップを実行します。

1. デフォルトの RCICSRES リソース・クラスまたは WCICSRES グループ化クラス、あるいはユーザー定義のリソース・クラス名があればそれらに相当するクラスで、プロファイルを RACF に定義する。

   プロファイル名については、DOCTEMPLATE リソース定義の名前に、接頭部としてリソース・タイプ DOCTEMPLATE を付け、CICS 領域用の **SECPRFX** システム初期設定パラメーターで指定された任意の追加の接頭部を付けたものを使用します。

   例えば、RCICSRES クラスの文書テンプレートを定義し、ユーザーがそれらを使用して文書をアセンブルすることを許可するには、次のコマンドを使用します。

```
RDEFINE  RCICSRES  (DOCTEMPLATE.doc1, DOCTEMPLATE.doc2, .., DOCTEMPLATE.docn) UACC(NONE)
                  NOTIFY(sys_admin_userid)
PERMIT DOCTEMPLATE.doc1 CLASS(RCICSRES) ID(group1, group2) ACCESS(READ)
```

   適切なアクセス・リストを使用し、WCICSRES リソース・グループ化クラスのプロファイルのメンバーとして文書テンプレートを定義するには、次のコマンドを使用します。

```
RDEFINE  WCICSRES  (doc_groupname) UACC(NONE)
                  ADDMEM(DOCTEMPLATE.doca, DOCTEMPLATE.docb) NOTIFY(sys_admin_userid)
PERMIT doc_groupname CLASS(WCICSRES) ID(group_userid) ACCESS(READ)
```

   RCICSRES または WCICSRES クラスに対して RDEFINE コマンドを発行した後、クラスがまだ活動化されていない場合は、SETROPTS コマンドを発行して活動化する必要があります。以下に例を示します。

```
SETROPTS CLASSACT(RCICSRES) RACLIST(RCICSRES)
```

   クラスがアクティブである場合は、SETROPTS コマンドを使用してクラスをリフレッシュします。

```
SETROPTS RACLIST(RCICSRES) REFRESH
```

2. CICS システム初期設定パラメーターとして **SEC**=YES を指定する (接頭部付きのプロファイルを定義する場合は、さらに **SECPRFX**=YES を指定する)。
3. RCICSRES および WCICSRES のデフォルトのリソース・クラス名の場合は XRES=YES、ユーザー定義のリソース・クラス名の場合は XRES=*name* を指定する。

   XRES=YES がデフォルトです。
4. CICS 文書テンプレートにアクセスするトランザクションの TRANSACTION リソース定義で RESSEC(YES) を指定する。

   CICS Web サポートの場合、すべての静的応答用のトランザクションは CWXN であるか、または TCPIPSERVICE 定義で TRANSACTION 属性を使用して CWXN の代わりに指定した代替トランザクションです。アプリケーション生成の応答のトランザクションは、別名トランザクションです。これは要求に応じて URIMAP 定義で指定するか、アナライザー・プログラムによって設定することができ、デフォルトは CWBA です。

   CICS で提供されたとおり、CWXN の定義は RESSEC(YES) を指定しますが、CWBA の定義は RESSEC(NO) を指定し、一般的に TRANSACTION リソース定義の場合、デフォルトは RESSEC(NO) です。

リソース・セキュリティー

リソース・セキュリティーは、CICS リソースを無許可アクセスから保護します。

CICS および RACF リソース検査の要約

RACF クラス名を指定するには、CICS リソース・システム初期設定パラメーターを使用します。

105 ページの表 8 では各リソースが簡単に説明されており、RACF クラス名を指定するために使用する関連の CICS システム初期設定パラメーターが記載されています。各システム初期設定パラメーターに関連したアプリケーション・プログラミング・コマンドおよびシステム・プログラミング・コマンドに関する総合的な情報については、 137 ページの『リソースおよびコマンドの検査の相互参照』を参照してください。

BMS コマンドに対しては、許可処理は実行されません。

表 *9. CICS* による一般リソース検査

CICS パラメーター	保護されるリソース	詳細情報
XAPPC	パートナー論理装置 (LU6.2)。	275 ページの『LU6.2 セキュリティーの実装』.

表 9. *CICS* による一般リソース検査 (続き)

CICS パラメーター	保護されるリソース	詳細情報
XCMD	コマンド・セキュリティー検査の対象である CICS アプリケーション・プログラミング・コマンドのサブセット。 EXEC CICS FEPI システム・コマンドも、このパラメーターによって制御されます。	162 ページの『CICS コマンド・セキュリティー』
XDB2	DB2ENTRY の Db2 リソース・クラスは、XDB2 システム初期設定パラメーターで CICS に対して指定されます。	32 ページの『DB2ENTRY リソースのリソース・クラス』
XDCT	CICS 区画外および区画内の一時データ・キュー。一時データ・クラス内でプロファイルを定義して、CICS 一時データ・キューへのアクセスが許可されるユーザーを制御します。	110 ページの『一時データのセキュリティー』.
XFCT	CICS ファイル制御管理対象の VSAM ファイルおよび BDAM ファイル。ファイル・クラス内でプロファイルを定義して、CICS VSAM ファイルおよび BDAM ファイルへのアクセスが許可されるユーザーを制御します。	112 ページの『ファイルのセキュリティー』.
XHFS	z/OS UNIX システム・サービスによって管理される z/OS UNIX ファイル。 z/OS UNIX ファイルのアクセス制御は z/OS UNIX システム・サービスで指定され、z/OS UNIX ファイルには個々の RACF プロファイルは不要であるため、これは特殊なケースです。このリソースに関連付けられているアプリケーション・プログラミング・コマンドおよびシステム・プログラミング・コマンドはありません。	125 ページの『z/OS UNIX ファイルのセキュリティーのインプリメント』.
XJCT	CICS システム・ログおよび一般ログ。ジャーナル・クラス内でプロファイルを定義して、CICS ログ・ストリーム上の CICS ジャーナルへのアクセスが許可されるユーザーを制御します。	114 ページの『ジャーナルとログ・ストリームのセキュリティー』.
XPCT	CICS 開始済みトランザクションおよび **EXEC CICS** コマンドは、**COLLECT STATISTICS TRANSACTION**、**DISCARD TRANSACTION**、**INQUIRE TRANSACTION**、**INQUIRE REQID**、**SET TRANSACTION**、および **CANCEL** です。開始済みトランザクション・クラス内でプロファイルを定義して、開始済み CICS トランザクションへのアクセスが許可されるユーザーを制御します。	115 ページの『開始されたトランザクションのセキュリティー』.
XPPT	CICS アプリケーション・プログラム。プログラム・クラス内でプロファイルを定義して、CICS アプリケーション・プログラムへのアクセスが許可されるユーザーを制御します。	121 ページの『アプリケーション・プログラムのセキュリティー』.

表 9. CICS による一般リソース検査 (続き)

CICS パラメーター	保護されるリソース	詳細情報
XPSB	DL/I プログラム仕様ブロック (PSB)。プログラム仕様ブロック・クラス内でプロファイルを定義して、CICS アプリケーション・プログラムで使用される DL/I PSB へのアクセスが許可されるユーザーを制御します。	128 ページの『プログラム仕様ブロックのセキュリティー』.
XRES	**XRES** パラメーターを使用する CICS リソースは、ATOMSERVICE、BUNDLE、DOCTEMPLATE、EPADAPTER、EPADAPTERSET、EVENTBINDING、JVMSERVER、および XMLTRANSFORM です。 例えば、DOCTEMPLATE リソース・クラス内でプロファイルを定義して、文書テンプレートへのアクセスが許可されるユーザーを制御します。	101 ページの『XRES リソース・セキュリティー・パラメーターを使用したセキュリティー』.
XTRAN	CICS トランザクション。	94 ページの『トランザクション・セキュリティー』.
XTST	CICS 一時記憶域キュー。一時記憶域クラス内でプロファイルを定義して、CICS 一時記憶域キューへのアクセスが許可されるユーザーを制御します。	122 ページの『一時記憶域のセキュリティー』.
XUSER	代理ユーザー・セキュリティー。	154 ページの『代理ユーザー・セキュリティー』.

リソースの総称プロファイルの定義

トランザクション接続セキュリティーによって CICS トランザクションへのアクセスを制御する場合、RACF 保護のレベルをさらに引き上げる必要がある他のリソース・タイプのサブセットはおそらくごくわずかです。

例えば、CICS アプリケーション・プログラム・リソース・クラス内でとりわけ重要であるプログラムはごくわずかであり、それよりはるかに多くのプログラムには重大なリスクはないという場合があります。この場合には、重要なプログラムのみを対象とした特定の RACF プロファイルを定義することで、そのわずかなプログラムを保護できます。以下のように、完全な総称リソース・プロファイルを定義することによって、すべてのユーザーはその他の重要でないプログラムに確実にアクセスできるようになります。

```
RDEFINE  MCICSPPT  *  UACC(READ) ...
```

このプロファイルは、特定のプロファイルのいずれによってもカバーされていない、プログラムのすべての許可要求に適用されます。 RACF 処理ロジックはそのようなものであるので、特定のリソース名の最も具体的なプロファイルが必ず使用されます。

プロファイルが総称であるかどうかは、プロファイル名を RLIST または SEARCH を使用してリストしたときに、プロファイル名の後ろに「G」が付いているかどうかを確認するだけで分かります。以下に例を示します。

SEARCH CLASS(TCICSTRN)

次の出力が表示されるとします。

```
C*
CED% (G)
** (G)
```

この出力は、CED% と ** の両方が総称プロファイルであることを示しています。C* プロファイルは後ろに (G) が付いていないので、総称プロファイルではありません。総称プロファイルが SETROPTS コマンドで有効にされるよりも前に C* プロファイルが作成された場合、このようになる可能性があります。 C* プロファイルは削除して、以下のようにして適正な総称プロファイルとして再定義できます。

```
SETROPTS NOGENERIC(TCICSTRN)
SETROPTS NOGENCMD(TCICSTRN)
RDEL TCICSTRN C*
SETROPTS GENERIC(TCICSTRN)
RDEFINE TCICSTRN C* UACC(NONE)
```

リソースおよびコマンドの検査の相互参照

相互参照を使用して、リソースに対するコマンドを検査します。

表 *10.* リソースおよびコマンドの検査の相互参照

EXEC CICS COMMAND	リソース検査			**CLASS=XCMD** の検査	
	CLASS	**ACCESS**	**RESOURCE**	**ACCESS**	**RESOURCE**
ABEND					
ACQUIRE				UPDATE	TERMINAL
ACQUIRE FOR BTS 注 9 (153 ページ) を参照	XFCT	UPDATE	BTS REPOSITORY FILE		
ADDRESS					
ALLOCATE					
ASKTIME					
BIF DEEDIT					
BIF DIGEST					
BUILD ATTACH					
CANCEL 注 1 (152 ページ) を参照	XPCT	READ	TRANSID		
CANCEL FOR BTS	XFCT	UPDATE	BTS REPOSITORY FILE		
CHANGE PASSWORD					
CHANGE TASK					

表 *10.* リソースおよびコマンドの検査の相互参照 (続き)

EXEC CICS COMMAND	リソース検査			**CLASS=XCMD** の検査	
	CLASS	**ACCESS**	**RESOURCE**	**ACCESS**	**RESOURCE**
COLLECT STATISTICS				READ	STATISTICS
COLLECT STATISTICS FILE	XFCT	READ	FILE	READ	STATISTICS
COLLECT STATISTICS JOURNALNAME	XJCT	READ	JOURNAL	READ	STATISTICS
COLLECT STATISTICS JOURNALNUM	XJCT	READ	JOURNAL	READ	STATISTICS
COLLECT STATISTICS PROGRAM	XPPT	READ	PROGRAM	READ	STATISTICS
COLLECT STATISTICS TDQUEUE	XDCT	READ	TDQUEUE	READ	STATISTICS
COLLECT STATISTICS TRANSACTION	XPCT	READ	TRANSID	READ	STATISTICS
CONNECT PROCESS					
CONVERSE					
CREATE ATOMSERVICE	XRES	ALTER	ATOMSERVICE.*resource_name*	ALTER	ATOMSERVICE
CREATE BUNDLE	XRES	ALTER	BUNDLE.*resource_name*	ALTER	BUNDLE
CREATE CONNECTION 注 2 (152 ページ) を参照				ALTER	CONNECTION
CREATE DB2CONN 注 3 (153 ページ) を参照		ALTER			DB2CONN
CREATE DB2ENTRY 注 3 (153 ページ) を参照	XDB2	ALTER	DB2ENTRY	ALTER	DB2ENTRY
CREATE DB2TRAN 注 3 (153 ページ) を参照	XDB2	ALTER	DB2TRAN	ALTER	DB2TRAN
CREATE DOCTEMPLATE	XRES	ALTER	DOCTEMPLATE.*resource_name*	ALTER	DOCTEMPLATE

表 10. リソースおよびコマンドの検査の相互参照 (続き)

EXEC CICS COMMAND	リソース検査			**CLASS=XCMD** の検査	
	CLASS	**ACCESS**	**RESOURCE**	**ACCESS**	**RESOURCE**
CREATE ENQMODEL				ALTER	ENQMODEL
CREATE FILE	XFCT	ALTER	FILE	ALTER	FILE
CREATE IPCONN				ALTER	IPCONN
CREATE JOURNALMODEL				ALTER	JOURNALMODEL
CREATE JVMSERVER	XRES	ALTER	JVMSERVER.*resource_name*	ALTER	JVMSERVER
CREATE LIBRARY				ALTER	LIBRARY
CREATE LSRPOOL				ALTER	LSRPOOL
CREATE MAPSET	XPPT	ALTER	MAPSET	ALTER	MAPSET
CREATE MQCONN				ALTER	MQCONN
CREATE MQMONITOR				ALTER	MQMON
CREATE PARTITIONSET	XPPT	ALTER	PARTITIONSET	ALTER	PARTITIONSET
CREATE PARTNER				ALTER	PARTNER
CREATE PIPELINE				ALTER	PIPELINE
CREATE PROCESSTYPE 注 10 (153 ページ) を参照				ALTER	PROCESSTYPE
CREATE PROFILE				ALTER	PROFILE
CREATE PROGRAM	XPPT	ALTER	PROGRAM	ALTER	PROGRAM
CREATE SESSIONS 注 3 (153 ページ) を参照				ALTER	セッション
CREATE TCPIPSERVICE				ALTER	TCPIPSERVICE
CREATE TDQUEUE 注 3 (153 ページ) を参照	XDCT	ALTER	TDQUEUE	ALTER	TDQUEUE
CREATE TERMINAL 注 3 (153 ページ) を参照				ALTER	TERMINAL
CREATE TRANCLASS				ALTER	TCLASS
CREATE TRANSACTION	XPCT	ALTER	TRANSID	ALTER	TRANSACTION
CREATE TSMODEL				ALTER	TSMODEL

表 *10.* リソースおよびコマンドの検査の相互参照 (続き)

EXEC CICS COMMAND	リソース検査			**CLASS=XCMD** の検査	
	CLASS	**ACCESS**	**RESOURCE**	**ACCESS**	**RESOURCE**
CREATE TYPETERM				ALTER	TYPETERM
CREATE URIMAP				ALTER	URIMAP
CREATE WEBSERVICE				ALTER	WEBSERVICE
CSD ADD				UPDATE	CSD
CSD ALTER				UPDATE	CSD
CSD APPEND				UPDATE	CSD
CSD COPY				UPDATE	CSD
CSD DEFINE				UPDATE	CSD
CSD DELETE				UPDATE	CSD
CSD DISCONNECT				READ	CSD
CSD ENDBRGROUP				READ	CSD
CSD ENDBRLIST				READ	CSD
CSD ENDBRRSRCE				READ	CSD
CSD GETNEXTGROUP				READ	CSD
CSD GETNEXTLIST				READ	CSD
CSD GETNEXTRSRCE				READ	CSD
CSD INQUIREGROUP				READ	CSD
CSD INQUIRELIST				READ	CSD
CSD INQUIRERSRCE				READ	CSD
CSD INSTALL				ALTER	CSD
CSD LOCK				UPDATE	CSD
CSD REMOVE				UPDATE	CSD
CSD RENAME				UPDATE	CSD
CSD STARTBRGROUP				READ	CSD
CSD STARTBRLIST				READ	CSD
CSD STARTBRRSRCE				READ	CSD
CSD UNLOCK				UPDATE	CSD
CSD USERDEFINE				UPDATE	CSD
DEFINE ACTIVITY 注 7 (153 ページ) および 9 (153 ページ) を参照	XFCT	UPDATE	BTS REPOSITORY FILE		

表 10. リソースおよびコマンドの検査の相互参照 (続き)

EXEC CICS COMMAND	リソース検査			**CLASS=XCMD** の検査	
	CLASS	**ACCESS**	**RESOURCE**	**ACCESS**	**RESOURCE**
DEFINE PROCESS 注 7 (153 ページ) および 9 (153 ページ) を参照	XFCT	UPDATE	BTS REPOSITORY FILE		
DELAY					
DELETE	XFCT	UPDATE	FILE		
DELETE ACTIVITY 注 9 (153 ページ) を参照	XFCT	UPDATE	BTS REPOSITORY FILE		
DELETEQ TD	XDCT	UPDATE	TDQUEUE		
DELETEQ TS 注 4 (153 ページ) を参照	XTST	UPDATE	TSQUEUE		
DEQ					
DISABLE PROGRAM	XPPT	UPDATE	PROGRAM	UPDATE	EXITPROGRAM
DISCARD ATOMSERVICE	XRES	ALTER	ATOMSERVICE.*resource_name*	ALTER	ATOMSERVICE
DISCARD AUTINSTMODEL				ALTER	AUTINSTMODEL
DISCARD BUNDLE	XRES	ALTER	BUNDLE.*resource_name*	ALTER	BUNDLE
DISCARD CONNECTION					
DISCARD DB2CONN				ALTER	DB2CONN
DISCARD DB2ENTRY	XDB2	ALTER	DB2ENTRY	ALTER	DB2ENTRY
DISCARD DB2TRAN	XDB2	ALTER	DB2TRAN	ALTER	DB2TRAN
DISCARD DOCTEMPLATE	XRES	ALTER	DOCTEMPLATE.*resource_name*	ALTER	DOCTEMPLATE
DISCARD ENQMODEL				ALTER	ENQMODEL
DISCARD FILE	XFCT	ALTER	FILE	ALTER	FILE
DISCARD IPCONN				ALTER	IPCONN
DISCARD JOURNALMODEL				ALTER	JOURNALMODEL
DISCARD JOURNALNAME	XJCT	ALTER	JOURNAL	ALTER	JOURNALNAME
DISCARD JVMSERVER	XRES	ALTER	JVMSERVER.*resource_name*	ALTER	JVMSERVER
DISCARD LIBRARY				ALTER	LIBRARY
DISCARD MQCONN				ALTER	MQCONN

表 *10.* リソースおよびコマンドの検査の相互参照 (続き)

EXEC CICS COMMAND	リソース検査			**CLASS=XCMD** の検査	
	CLASS	**ACCESS**	**RESOURCE**	**ACCESS**	**RESOURCE**
DISCARD MQMONITOR				ALTER	MQMON
DISCARD PARTNER				ALTER	PARTNER
DISCARD PIPELINE				ALTER	PIPELINE
DISCARD PROCESSTYPE 注 10 (153 ページ) を参照				ALTER	PROCESSTYPE
DISCARD PROFILE				ALTER	PROFILE
DISCARD PROGRAM	XPPT	ALTER	PROGRAM	ALTER	PROGRAM
DISCARD TCPIPSERVICE				ALTER	TCPIPSERVICE
DISCARD TDQUEUE	XDCT	ALTER	TDQUEUE	ALTER	TDQUEUE
DISCARD TERMINAL				ALTER	TERMINAL
DISCARD TRANCLASS				ALTER	TCLASS
DISCARD TRANSACTION	XPCT	ALTER	TRANSID	ALTER	TRANSACTION
DISCARD TSMODEL				ALTER	TSMODEL
DISCARD URIMAP				ALTER	URIMAP
DISCARD WEBSERVICE				ALTER	WEBSERVICE
DOCUMENT CREATE 注 14 を参照	XRES	READ	DOCTEMPLATE.*resource_name*		
DOCUMENT INSERT 注 14 を参照	XRES	READ	DOCTEMPLATE.*resource_name* 注 13 を参照		
DUMP TRANSACTION					
ENABLE PROGRAM	XPPT	UPDATE	PROGRAM	UPDATE	EXITPROGRAM
ENDBR 注 5 (153 ページ) を参照					
ENQ					
ENTER TRACENUM					
EXTRACT					
EXTRACT EXIT	XPPT	READ	PROGRAM	UPDATE	EXITPROGRAM

表 10. リソースおよびコマンドの検査の相互参照 (続き)

EXEC CICS COMMAND	リソース検査			**CLASS=XCMD** の検査	
	CLASS	**ACCESS**	**RESOURCE**	**ACCESS**	**RESOURCE**
EXTRACT STATISTICS				READ	STATISTICS
FEPI					FEPI
FORMATTIME					
FREE					
FREEMAIN					
GDS					
GETMAIN					
HANDLE ABEND PROGRAM	XPPT	READ	PROGRAM		
HANDLE AID					
HANDLE CONDITION					
IGNORE CONDITION					
INQUIRE ACTIVITYID 注 9 (153 ページ) を参照	XFCT	READ	BTS REPOSITORY FILE		
INQUIRE ASSOCIATION				READ	ASSOCIATION
INQUIRE ATOMSERVICE	XRES	READ	ATOMSERVICE.*resource_name*	READ	ATOMSERVICE
INQUIRE AUTINSTMODEL				READ	AUTINSTMODEL
INQUIRE AUTOINSTALL				READ	AUTOINSTALL
INQUIRE BRFACILITY				READ	BRFACILITY
INQUIRE BUNDLE	XRES	READ	BUNDLE.*resource_name*	READ	BUNDLE
INQUIRE BUNDLEPART	XRES	READ	BUNDLE.*resource_name*	READ	BUNDLEPART
INQUIRE CAPDATAPRED	XRES	READ	EVENTBINDING.*resource_name*	READ	CAPDATAPRED
INQUIRE CAPINFOSRCE	XRES	READ	EVENTBINDING.*resource_name*	READ	CAPINFOSRCE
INQUIRE CAPOPTPRED	XRES	READ	EVENTBINDING.*resource_name*	READ	CAPOPTPRED
INQUIRE CAPTURESPEC	XRES	READ	EVENTBINDING.*resource_name*	READ	CAPTURESPEC
INQUIRE CFDTPOOL				READ	CFDTPOOL

表 10. リソースおよびコマンドの検査の相互参照 (続き)

EXEC CICS COMMAND	リソース検査			**CLASS=XCMD** の検査	
	CLASS	**ACCESS**	**RESOURCE**	**ACCESS**	**RESOURCE**
INQUIRE CONNECTION				READ	CONNECTION
INQUIRE CONTAINER 注 9 (153 ページ) を参照	XFCT	READ	BTS REPOSITORY FILE		
INQUIRE DB2CONN				READ	DB2CONN
INQUIRE DB2ENTRY	XDB2	READ	DB2ENTRY	READ	DB2ENTRY
INQUIRE DB2TRAN	XDB2	READ	DB2TRAN	READ	DB2TRAN
INQUIRE DELETSHIPPED				READ	DELETSHIPPED
INQUIRE DOCTEMPLATE	XRES	READ	DOCTEMPLATE.*resource_name*	READ	DOCTEMPLATE
INQUIRE DSNAME				READ	DSNAME
INQUIRE DUMPDS				READ	DUMPDS
INQUIRE ENQMODEL				READ	ENQMODEL
INQUIRE EPADAPTER	XRES	READ	EPADAPTER.*resource_name*	READ	EPADAPTER
INQUIRE EPADAPTERSET	XRES	READ	EPADAPTERSET.*resource_name*	READ	EPADAPTERSET
INQUIRE EPADAPTINSET	XRES	READ	EPADAPTERSET.*resource_name*	READ	EPADAPTINSET
INQUIRE EVENT 注 9 (153 ページ) を参照	XFCT	READ	BTS REPOSITORY FILE		
INQUIRE EVENTBINDING	XRES	READ	EVENTBINDING.*resource_name*	READ	EVENTBINDING
INQUIRE EVENTPROCESS				READ	EVENTPROCESS
INQUIRE EXCI				READ	EXCI
INQUIRE EXITPROGRAM	XPPT	READ	PROGRAM	READ	EXITPROGRAM
INQUIRE FEATUREKEY				READ	SYSTEM
INQUIRE FILE	XFCT	READ	FILE	READ	FILE
INQUIRE HOST				READ	HOST
INQUIRE IPCONN				READ	IPCONN
INQUIRE IRC				READ	IRC

表 10. リソースおよびコマンドの検査の相互参照 (続き)

EXEC CICS COMMAND	リソース検査			CLASS=XCMD の検査	
	CLASS	**ACCESS**	**RESOURCE**	**ACCESS**	**RESOURCE**
INQUIRE JOURNALMODEL				READ	JOURNALMODEL
INQUIRE JOURNALNAME	XJCT	READ	JOURNAL	READ	JOURNAL
INQUIRE JVMSERVER	XRES	READ	JVMSERVER.*resource_name*	READ	JVMSERVER
INQUIRE LIBRARY				READ	LIBRARY
INQUIRE MODENAME				READ	MODENAME
INQUIRE MONITOR				READ	MONITOR
INQUIRE MQCONN				READ	MQCONN
INQUIRE MQINI				READ	MQMON
INQUIRE MQMONITOR				READ	MQMON
INQUIRE MVSTCB				READ	MVSTCB
INQUIRE NETNAME				READ	TERMINAL
INQUIRE NODEJSAPP	XRES	READ	BUNDLE.*resource_name*	READ	NODEJSAPP
INQUIRE OSGIBUNDLE	XRES	READ	BUNDLE.*resource_name*	READ	OSGIBUNDLE
INQUIRE OSGISERVICE	XRES	READ	BUNDLE.*resource_name*	READ	OSGISERVICE
INQUIRE PARTNER				READ	PARTNER
INQUIRE PIPELINE				READ	PIPELINE
INQUIRE PROCESS 注 9 (153 ページ) を参照	XFCT	READ	BTS REPOSITORY FILE		
INQUIRE PROCESSTYPE 注 10 (153 ページ) を参照	XPTT			READ	PROCESSTYPE
INQUIRE PROFILE				READ	PROFILE
INQUIRE PROGRAM	XPPT	READ	PROGRAM	READ	PROGRAM
INQUIRE REQID 注 8 (153 ページ) を参照	XPCT	READ	TRANSID	READ	REQID
INQUIRE RRMS				READ	RRMS
INQUIRE STATISTICS				READ	STATISTICS

表 *10.* リソースおよびコマンドの検査の相互参照 (続き)

EXEC CICS COMMAND	リソース検査			**CLASS=XCMD** の検査	
	CLASS	**ACCESS**	**RESOURCE**	**ACCESS**	**RESOURCE**
INQUIRE STORAGE				READ	STORAGE
INQUIRE STREAMNAME				READ	STREAMNAME
INQUIRE SUBPOOL				READ	SUBPOOL
INQUIRE SYSDUMPCODE				READ	SYSDUMPCODE
INQUIRE SYSTEM				READ	SYSTEM
INQUIRE TASK				READ	TASK
INQUIRE TCLASS				READ	TCLASS
INQUIRE TCPIP				READ	TCPIP
INQUIRE TCPIPSERVICE				READ	TCPIPSERVICE
INQUIRE TDQUEUE	XDCT	READ	TDQUEUE	READ	TDQUEUE
INQUIRE TEMPSTORAGE				READ	TEMPSTORAGE
INQUIRE TERMINAL				READ	TERMINAL
INQUIRE TIMER 注 9 (153 ページ) を参照	XFCT	READ	BTS REPOSITORY FILE		
INQUIRE TRACEDEST				READ	TRACEDEST
INQUIRE TRACEFLAG				READ	TRACEFLAG
INQUIRE TRACETYPE				READ	TRACETYPE
INQUIRE TRANCLASS				READ	TCLASS
INQUIRE TRANDUMPCODE				READ	TRANDUMPCODE
INQUIRE TRANSACTION	XPCT	READ	TRANSID	READ	TRANSACTION
INQUIRE TSMODEL				READ	TSMODEL
INQUIRE TSPOOL				READ	TSPOOL
INQUIRE TSQNAME 注 4 (153 ページ) を参照	XTST	READ	TSQNAME	READ	TSQUEUE
INQUIRE TSQUEUE 注 4 (153 ページ) を参照	XTST	READ	TSQUEUE	READ	TSQUEUE

表 *10.* リソースおよびコマンドの検査の相互参照 (続き)

EXEC CICS COMMAND	リソース検査			**CLASS=XCMD** の検査	
	CLASS	**ACCESS**	**RESOURCE**	**ACCESS**	**RESOURCE**
INQUIRE UOW				READ	UOW
INQUIRE UOWDSNFAIL				READ	UOWDSNFAIL
INQUIRE UOWENQ				READ	UOWENQ
INQUIRE UOWLINK				READ	UOWLINK
INQUIRE URIMAP				READ	URIMAP
INQUIRE VTAM				READ	VTAM (現在は z/OS Communications Server)
INQUIRE WEB				READ	WEB
INQUIRE WEBSERVICE				READ	WEBSERVICE
INQUIRE WLMHEALTH				READ	WLMHEALTH
INQUIRE XMLTRANSFORM	XRES	READ	XMLTRANSFORM.*resource_name*	READ	XMLTRANSFORM
ISSUE					
LINK	XPPT	READ	PROGRAM		
LINK ACQPROCESS 注 9 (153 ページ) を参照	XFCT	UPDATE	BTS REPOSITORY FILE		
LINK ACTIVITY / ACQACTIVITY 注 9 (153 ページ) を参照	XFCT	UPDATE	BTS REPOSITORY FILE		
LOAD	XPPT	READ	PROGRAM		
MONITOR					
PERFORM DELETSHIPPED				UPDATE	DELETSHIPPED
PERFORM DUMP				UPDATE	DUMP
PERFORM PIPELINE				UPDATE	PIPELINE
PERFORM RESETTIME				UPDATE	RESETTIME
PERFORM SECURITY REBUILD				UPDATE	SECURITY
PERFORM SHUTDOWN				UPDATE	SHUTDOWN
PERFORM SSL REBUILD				UPDATE	SECURITY

表 10. リソースおよびコマンドの検査の相互参照 (続き)

EXEC CICS COMMAND	リソース検査			**CLASS=XCMD** の検査	
	CLASS	**ACCESS**	**RESOURCE**	**ACCESS**	**RESOURCE**
PERFORM STATISTICS				UPDATE	STATISTICS
POINT					
POP HANDLE					
POST					
PURGE MESSAGE					
PUSH HANDLE					
QUERY SECURITY 注 6 (153 ページ) を参照					
READ	XFCT	READ	FILE		
READ NEXT 注 5 (153 ページ) を参照					
READ PREV 注 5 (153 ページ) を参照					
READQ TD	XDCT	UPDATE	TDQUEUE		
READQ TS 注 4 (153 ページ) を参照	XTST	READ	TSQUEUE		
RECEIVE					
RELEASE	XPPT	READ	PROGRAM		
RESET ACTIVITY 注 9 (153 ページ) を参照	XFCT	UPDATE	BTS REPOSITORY FILE		
RESET ACQPROCESS 注 9 (153 ページ) を参照	XFCT	UPDATE	BTS REPOSITORY FILE		
RESETBR 注 5 (153 ページ) を参照					
RESYNC ENTRYNAME				UPDATE	EXITPROGRAM
RETRIEVE					

表 *10.* リソースおよびコマンドの検査の相互参照 *(*続き*)*

EXEC CICS COMMAND	リソース検査			**CLASS=XCMD** の検査	
	CLASS	**ACCESS**	**RESOURCE**	**ACCESS**	**RESOURCE**
RETURN / RETURN ENDACTIVITY 注 9 (153 ページ) および 11 (153 ページ) を参照	XFCT	UPDATE	BTS REPOSITORY FILE		
REWRITE	XFCT	UPDATE	FILE		
ROUTE					
RUN 注 9 (153 ページ) および 11 (153 ページ) を参照	XFCT	UPDATE	BTS REPOSITORY FILE		
RUN / ASYNCH SYNC 注 9 (153 ページ) を参照	XFCT	UPDATE	DFHLRQ FILE		
SEND					
SET ATOMSERVICE	XRES	UPDATE	ATOMSERVICE.*resource_name*	UPDATE	ATOMSERVICE
SET AUTOINSTALL				UPDATE	AUTOINSTALL
SET BRFACILITY				UPDATE	BRFACILITY
SET BUNDLE	XRES	UPDATE	BUNDLE.*resource_name*	UPDATE	BUNDLE
SET CONNECTION				UPDATE	CONNECTION
SET DB2CONN				UPDATE	DB2CONN
SET DB2ENTRY	XDB2	UPDATE	DB2ENTRY	UPDATE	DB2ENTRY
SET DB2TRAN	XDB2	UPDATE	DB2TRAN	UPDATE	DB2TRAN
SET DELETSHIPPED				UPDATE	DELETSHIPPED
SET DOCTEMPLATE	XRES	UPDATE	DOCTEMPLATE.*resource_name*	UPDATE	DOCTEMPLATE
SET DSNAME				UPDATE	DSNAME
SET DUMPDS				UPDATE	DUMPDS
SET ENQMODEL				UPDATE	ENQMODEL
SET EPADAPTER	XRES	UPDATE	EPADAPTER.*resource_name*	UPDATE	EPADAPTER
SET EPADAPTERSET	XRES	UPDATE	EPADAPTERSET.*resource_name*	UPDATE	EPADAPTERSET
SET EVENTBINDING	XRES	UPDATE	EVENTBINDING.*resource_name*	UPDATE	EVENTBINDING
SET EVENTPROCESS	XRES	UPDATE	EVENTPROCESS.*resource_name*	UPDATE	EVENTPROCESS
SET FILE	XFCT	UPDATE	FILE	UPDATE	FILE
SET HOST				UPDATE	HOST
SET IPCONN				UPDATE	IPCONN
SET IRC				UPDATE	IRC
SET JOURNALNAME	XJCT	UPDATE	JOURNAL	UPDATE	JOURNAL

表 *10.* リソースおよびコマンドの検査の相互参照 (続き)

EXEC CICS COMMAND	リソース検査			**CLASS=XCMD** の検査	
	CLASS	**ACCESS**	**RESOURCE**	**ACCESS**	**RESOURCE**
SET JVMSERVER	XRES	UPDATE	JVMSERVER.*resource_name*	UPDATE	JVMSERVER
SET LIBRARY				UPDATE	LIBRARY
SET MODENAME				UPDATE	MODENAME
SET MONITOR				UPDATE	MONITOR
SET MQCONN 注 13 (153 ページ) を参照				UPDATE	MQCONN
SET MQMONITOR				UPDATE	MQMON
SET NETNAME				UPDATE	TERMINAL
SET PIPELINE				UPDATE	PIPELINE
SET PROCESSTYPE 注 10 (153 ページ) を参照	XPTT			UPDATE	PROCESSTYPE
SET PROGRAM	XPPT	UPDATE	PROGRAM	UPDATE	PROGRAM
SET PROGRAM REPLICATION 注 16 (154 ページ) を参照				ALTER	REPLICATION
SET STATISTICS				UPDATE	STATISTICS
SET SYSDUMPCODE 注 15 を参照				UPDATE	SYSDUMPCODE
SET SYSTEM				UPDATE	SYSTEM
SET TASK				UPDATE	TASK
SET TCLASS				UPDATE	TCLASS
SET TCPIP				UPDATE	TCPIP
SET TCPIPSERVICE				UPDATE	TCPIPSERVICE
SET TDQUEUE 注 3 (153 ページ) を参照	XDCT	UPDATE	TDQUEUE	UPDATE	TDQUEUE
SET TEMPSTORAGE				UPDATE	TEMPSTORAGE
SET TERMINAL				UPDATE	TERMINAL
SET TRACEDEST				UPDATE	TRACEDEST
SET TRACEFLAG				UPDATE	TRACEFLAG
SET TRACETYPE				UPDATE	TRACETYPE
SET TRANCLASS				UPDATE	TCLASS
SET TRANDUMPCODE				UPDATE	TRANDUMPCODE

表 10. リソースおよびコマンドの検査の相互参照 (続き)

EXEC CICS COMMAND	リソース検査			**CLASS=XCMD** の検査	
	CLASS	**ACCESS**	**RESOURCE**	**ACCESS**	**RESOURCE**
SET TRANSACTION	XPCT	UPDATE	TRANSID	UPDATE	TRANSACTION
SET TSQNAME 注 4 (153 ページ) を参照	XTST	UPDATE	TSQUEUE	UPDATE	TSQUEUE
SET TSQUEUE 注 4 (153 ページ) を参照	XTST	UPDATE	TSQNAME	UPDATE	TSQUEUE
SET UOW				UPDATE	UOW
SET UOWLINK				UPDATE	UOWLINK
SET URIMAP				UPDATE	URIMAP
SET VTAM				UPDATE	VTAM
SET WEB				UPDATE	WEB
SET WEBSERVICE				UPDATE	WEBSERVICE
SET WLMHEALTH				UPDATE	WLMHEALTH
SET XMLTRANSFORM	XRES	UPDATE	XMLTRANSFORM.*resource_name*	UPDATE	XMLTRANSFORM
SIGNOFF					
SIGNON					
SPOOLCLOSE					
SPOOLOPEN					
SPOOLREAD					
SPOOLWRITE					
START 注 7 (153 ページ) を参照	XPCT	READ	TRANSID		
STARTBR	XFCT	READ	FILE		
STARTBROWSE ACTIVITY 注 9 (153 ページ) を参照	XFCT	READ	BTS REPOSITORY FILE		
STARTBROWSE CONTAINER (BTS) 注 9 (153 ページ) を参照	XFCT	READ	BTS REPOSITORY FILE		
STARTBROWSE EVENT 注 9 (153 ページ) を参照	XFCT	READ	BTS REPOSITORY FILE		

表 *10.* リソースおよびコマンドの検査の相互参照 (続き)

EXEC CICS COMMAND	リソース検査			**CLASS=XCMD** の検査	
	CLASS	**ACCESS**	**RESOURCE**	**ACCESS**	**RESOURCE**
STARTBROWSE PROCESS 注 9 (153 ページ) を参照	XFCT	READ	BTS REPOSITORY FILE		
SUSPEND 注 9 (153 ページ) を参照	XFCT	UPDATE	BTS REPOSITORY FILE		
SYNCPOINT					
TCPIP					
UNLOCK					
VERIFY PASSWORD					
WAIT					
WAIT JOURNALNAME	XJCT	READ	JOURNAL		
WAIT JOURNALNUM	XJCT	READ	JOURNAL		
WAITCICS					
WEB					
WRITE	XFCT	UPDATE	FILE		
WRITE JOURNALNAME	XJCT	UPDATE	JOURNAL		
WRITE JOURNALNUM	XJCT	UPDATE	DFHJNN		
WRITE OPERATOR					
WRITEQ TD	XDCT	UPDATE	TDQUEUE		
WRITEQ TS 注 4 (153 ページ) を参照	XTST	UPDATE	TSQUEUE|TSQNAME		
XCTL	XPPT	READ	PROGRAM		

注:

1. **CANCEL** コマンドは 2 つの検査を実行できます。1 つの検査は、**CANCEL** コマンドで指定されたトランザクションに対して実行されます。もう 1 つの検査は、該当する場合、キャンセル対象の REQID に関連付けられたトランザクションに対して実行されます。
2. **CREATE CONNECTION** コマンドは、接続を定義するときにコマンド・セキュリティー検査の対象となります。例えば、**CREATE CONNECTION(CON1)** *attribute*(...) などです。ただし、**CREATE CONNECTION COMPLETE** または **CREATE CONNECTION DISCARD** コマンドを使用する場合は、**COMPLETE** および **DISCARD** コマンドの使

用権限を持っていない限り、コマンド・セキュリティー検査は実行されません。**COMPLETE** および **DISCARD** コマンドを使用できるのは、**CREATE CONNECTION(CON1)** および **CREATE SESSIONS(SES1)** コマンドの実行権限を持つユーザーのみです。それ以外の場合は、ILLOGIC が返されます。

3. インストールの代理ユーザー検査も発生する可能性があります。詳細については、 154 ページの『代理ユーザー検査が適用される状態』を参照してください。
4. 一時記憶域キューに対してセキュリティー検査が実行されるのは、関連する TSMODEL リソース定義で SECURITY(YES) が指定されている場合、あるいは、一時記憶域テーブル (TST) を使用していて、一時記憶域キュー用に DFHTST TYPE=SECURITY マクロがコーディングされている場合です。
5. このコマンドの前に **STARTBR** コマンドが発行されていなければならないため、セキュリティー検査は実行されません。セキュリティー検査は、**STARTBR** コマンドで発行されます。
6. **QUERY SECURITY** コマンドはリソース検査またはコマンド検査によって制御されませんが、このコマンドによってこれらの検査が発行される可能性があります。
7. 開始の代理ユーザー検査も発生する可能性があります。
8. REQID がトランザクションに関連付けられている場合にのみ、transid のリソース検査が実行されます。
9. CICS BTS API コマンドです。
10. コマンド・セキュリティーの対象となる CICS BTS コマンドです。他の CICS BTS コマンドはいずれも、コマンド・レベル・セキュリティーの対象になりません。
11. タイミング・オペランドを使用して BTS LRQ ファイルにアクセスする BTS コマンドです。
12. **EXEC CICS DOCUMENT CREATE** および **INSERT** コマンドは、DOCTEMPLATE リソース定義の TEMPLATENAME 属性に指定された 48 文字のテンプレート名を使用して、文書テンプレートを参照します。ただし、これらのコマンドのセキュリティー検査では、TEMPLATENAME 属性に対応する DOCTEMPLATE リソース定義の名前を使用します。
13. CICS 領域に対してコマンド・セキュリティーがアクティブである場合、**EXEC CICS SET MQCONN** コマンドを使用して IBM MQ への接続を開始または停止するためには、ユーザーは **EXEC CICS SET MQCONN** コマンドだけでなく、**EXEC CICS EXTRACT EXIT** コマンドの使用権限も必要です。ユーザーが **EXEC CICS EXTRACT EXIT** コマンドの使用権限なしで接続を開始または停止しようとすると、CICS はメッセージ DFHXS1111 および DFHMQ0302 を発行します。CICS 領域に対してプログラムのリソース・セキュリティー (XPPT) がアクティブであり、SET を使用して「接続状況」を「接続済み」に設定した場合、ユーザーは、提供された CICS MQ プログラムを実行するための十分な権限が必要です。
14. 静的 Web ページとして使用される CICS 文書テンプレート、アプリケーション生成の応答の一部としてアプリケーション・プログラムが使用する文書テンプレート、および TEMPLATE オプションを指定したすべての **EXEC CICS**

DOCUMENT INSERT コマンドおよび **EXEC CICS DOCUMENT CREATE** コマンドにより使用される文書テンプレートの場合は、少なくとも READ アクセス権限が必要です。

15. CHECK CLASS=XCMD の場合、**SET SYSDUMPCODE** の **JOBLIST** オプションまたは **DSPLIST** オプションを更新するには、少なくとも CONTROL アクセス権限が必要です。
16. コマンド・セキュリティー検査がアクティブである場合は、**SET PROGRAM** で既に必要となる UPDATE アクセス権限に加えて、**SET PROGRAM REPLICATION** を発行するために、RACF REPLICATION リソースへの ALTER アクセス権限も持っている必要があります。REPLICATION は RACF 定義で CICS リソースとして扱われます。

代理ユーザー・セキュリティー

代理ユーザーは、別のユーザー (元のユーザー) の代理になることが許可されている、RACF 定義のユーザーです。CICS はさまざまに異なる状況で代理ユーザー・セキュリティーを使用します。

代理ユーザー検査が適用される状態

代理ユーザーとは、他のユーザーに代わって作業を開始する権限を持っているユーザーのことです。代理ユーザーには、他のユーザーのパスワードを知らなくても、そのユーザーに代わって操作することが許可されます。代理ユーザー検査を有効にするには、システム初期設定パラメーターとして XUSER=YES を指定する必要があります。

CICS はさまざまな状況で代理ユーザー・セキュリティー検査を行います。そのために、RACF などの外部セキュリティー・マネージャー (ESM) の代理ユーザー機能を使用します。代理ユーザー検査が有効である場合、以下の項目に適用されます。

- CICS のデフォルトのユーザー
- PLT の初期設定後の処理
- Preset terminal security
- 開始済みトランザクション
- RUN コマンドによって開始された CICS Business Transaction Services (BTS) のプロセスまたはアクティビティーに関連付けられているユーザー ID。
- 一時データ宛先に関連付けられているユーザー ID
- EXCI 呼び出しでパラメーターとして指定されたユーザー ID
- DB2CONN および DB2ENTRY リソース定義の AUTHID および COMAUTHID 属性に指定されたユーザー ID
- URIMAP リソース定義の USERID 属性に指定されたユーザー ID
- イベント処理トランザクション開始アダプターのトランザクション・ユーザー ID に指定されたユーザー ID
- COLM トランザクションで開始された CICSPlex SM MAS エージェント
- CORM トランザクションで開始された CICSPlex SM ローカル MAS エージェント
- JES 内部読み取りプログラムへの JCL ジョブ実行依頼に関係するユーザー ID

注: CICSPlex SM インターフェースを使用して代理ユーザー・セキュリティーに従うリソース定義をインストールする場合、CICS によって検査される代理ユーザーは、リソースがインストールされている領域内で CICSPlex SM エージェントを開始したユーザー ID です。

CICS デフォルト・ユーザー

CICS は、独自のユーザー ID (CICS 領域ユーザー ID) に対して代理ユーザー・セキュリティー検査を実行し、DFLTUSER システム初期設定パラメーターで指定されたデフォルト・ユーザー ID の代理として正しく許可されていることを確認します。

初期設定後の処理

PLTPI システム初期設定パラメーターにプログラム・リスト・テーブルを指定すると、CICS は、領域ユーザー ID が PLTPIUSR システム初期設定パラメーターに指定されたユーザー ID の代理ユーザーとして許可されているかどうかを検査します。

PLTPIUSR システム初期設定パラメーターは、CICS 初期設定中に実行される PLT プログラムで CICS が使用するユーザー ID を指定します。すべての PLT プログラムは、指定されたユーザー ID の権限のもとで実行されます (CPLT トランザクションを含む)。このユーザー ID は、プログラムによって参照されるすべてのリソースについての権限を持っていなければなりません。

PLT セキュリティー検査の範囲は、PLTPISEC システム初期設定パラメーターによって定義されます。このパラメーターは、コマンド・セキュリティー検査およびリソース・セキュリティー検査が PLTPI プログラムに適用されるかどうかを指定します。

PLTPIUSR パラメーターを指定しない場合、CICS は CICS 領域ユーザー ID の権限のもとで PLTPI プログラムを実行します。この場合、CICS は代理ユーザー検査を実行しません。ただし、CICS 領域ユーザー ID は、PLT プログラムによって参照されるすべてのリソースについての権限を持っていなければなりません。さらに、CICS 領域ユーザー ID は PLT プログラムによって開始されるすべてのトランザクションに関連付けられるため、これらのトランザクションの実行権限を持っていなければなりません。

事前設定端末セキュリティー

事前設定セキュリティー・ユーザー ID で定義された端末をインストールする場合、CICS は、インストールを実行しているユーザー ID が事前設定されたユーザー ID の代理ユーザーとして許可されていることを確認します。

これについては、 83 ページの『Preset terminal security』で説明しています。

開始済みトランザクション

CICS は、START コマンドを使用して端末と関連付けられていないトランザクションを開始するときに、代理ユーザー検査を実行します。

以下では、START コマンドを発行するトランザクションのユーザー ID は *starting-userid* と呼ばれており、開始済みトランザクションの実行ユーザー ID は *started-userid* と呼ばれています。

- TERMID オプションが START コマンドに指定されている場合、代理ユーザー検査は適用されません。 *started-userid* は、トランザクションが実行される端末から継承されます。
- USERID オプションが START コマンドに指定されている場合、*started-userid* はその指定されたユーザー ID に設定されます。
- TERMID および USERID のどちらも START コマンドに指定されていない場合、*started-userid* は *starting-userid* と同じ設定になります。

CICS は、START を発行するトランザクションに関連付けられているすべてのユーザー ID が、*started-userid* の代理であることを要求します。 CICS はさらに、すべてのユーザー ID が常にそのものの代理であると見なしています。そのため、*started-userid* と同じユーザー ID は既に代理と見なされており、それらに対して外部セキュリティー・マネージャーは呼び出されません。

トランザクションは、CICS 相互通信を使用している場合、および 2 端末モードで EDF を使用している場合に、*starting-userid* とは異なるユーザー ID に関連付けることができます。

USERID および **TERMID** のどちらも **START** コマンドに指定されていない場合、*starting-userid* と *started-userid* は同じであると見なされるため、代理検査は実行されません。 ICRX が使用可能な場合、CICS はそれを開始タスクに渡し、開始タスクは ICRX によって使用される分散 ID を継承します。

相互通信と開始トランザクション**:** START コマンド (TERMID の指定なし) がトランザクション・ルーティングされたトランザクションから機能シップまたは実行される場合、そのコマンドをリンク・セキュリティーの対象にすることができます。リンク・セキュリティーが有効である場合、CICS は代理ユーザー検査を実行して、リンク・セキュリティーのユーザー ID が、開始済みトランザクションのユーザー ID の代理ユーザーとして許可されることの確認も行います。この代理検査は、USERID が省略されている場合でも (*started-userid* がリンク・ユーザー ID と異なる場合でも) このステージで実行されます。詳細については、 272 ページの『相互通信リンク・セキュリティー』を参照してください。

二重画面モードでの **EDF** と開始済みトランザクション**:**

EXEC CICS START コマンド (TERMID の指定なし) が二重画面モードで EDF の制御下で実行されている場合、CICS は代理ユーザー検査も実行して、EDF 端末のユーザー ID が、開始済みトランザクションのユーザー ID の代理ユーザーとして許可されていることを確認します。

この検査は、*started-userid* が EDF ユーザー ID と異なっている場合、USERID が省略されていても実行されます。

代理ユーザー検査は、リンク・セキュリティーの対象にすることができます。 EDF が二重画面モードで使用中である場合、EDF を実行しているユーザーのセキュリテ

ィーも検査されます。 EXEC CICS START コマンドで NOTAUTH 条件が発生した場合、リンク・セキュリティーまたは EDF ユーザー・セキュリティーが原因である可能性があります。

BTS のプロセスおよびアクティビティー

CICS Business Transaction Services (BTS) のプロセスまたはアクティビティーが RUN コマンドによってアクティブ化されると、RUN を発行したトランザクションのユーザー ID とは異なるユーザー ID で実行できます。

アプリケーション・プログラマーは、DEFINE PROCESS または DEFINE ACTIVITY の USERID オプションをコーディングすることで、プロセスまたはアクティビティーが RUN コマンドによってアクティブ化されるときに、それを実行する権限を持つユーザーを指定できます。 USERID オプションを省略すると、デフォルトでその値は DEFINE コマンドを発行するトランザクションのユーザー ID になります。

USERID オプションが指定されている場合、CICS は (定義時に) 代理セキュリティー検査を実行して、DEFINE コマンドを発行したトランザクションのユーザー ID が、USERID によって指定されたユーザー ID の使用を許可されていることを確認します。

一時データのトリガー・レベル・トランザクション

一時データ・キューが、非端末トリガー・レベル・トランザクションと USERID パラメーターが指定された RDO によって定義されると、定義をインストールするユーザーが検査されます。

同様に、このような一時データ・キューが CREATE TDQUEUE コマンドで作成されると、そのコマンドを実行するユーザーが検査されます。

端末に関連付けられていない一時データ・トリガー・レベル・トランザクションのユーザー ID は、TDQUEUE リソース定義または SET TDQUEUE コマンドに指定できます。

区画内一時データ・リソース:

CICS は一時データ・キュー定義に指定されたユーザー ID を使用して、端末に関連付けられていないトリガー・レベル・トランザクションのセキュリティー検査を行います。

TRANSID オペランドで指定されたトリガー・レベル・トランザクションのセキュリティー検査で CICS に使用させるユーザー ID を指定して、USERID オペランドをコーディングします。USERID は、宛先ファシリティーがファイルの場合にのみ有効です。

トリガー・レベル・トランザクションは、指定されたユーザー ID の権限のもとで実行されます。このユーザー ID は、トランザクションによって使用されるすべてのリソースについての権限を持っていなければなりません。

適格なトリガー・レベルのエントリーからユーザー ID を省略した場合、CICS は DFLTUSER システム初期設定パラメーターに指定されているデフォルト・ユーザー ID を使用します。一時データ・キュー定義がインストールされている CICS 領域

のユーザー ID が、一時データ宛先定義で指定されたすべてのユーザー ID の代理として定義されるようにしてください。これは、コールド・スタート中に CICS が、インストールされている一時データ・キュー定義に指定されたすべてのユーザー ID に対して、CICS 領域ユーザー ID の代理ユーザー・セキュリティー検査を実行するためです。代理セキュリティー検査が失敗すると、一時データ・キュー定義はインストールされません。

EXEC CICS SET TDQUEUE ATIUSERID: ATIUSERID オプションが指定されたシステム・プログラミング・コマンド SET TDQUEUE は、端末に関連付けられていない、一時データのトリガー・レベル・トランザクションのユーザー ID を指定します。宛先機構はファイルでなければなりません。

CICS は、SET TDQUEUE コマンドを発行するトランザクションのユーザー ID に対して代理ユーザー・セキュリティー検査を行い、そのトランザクション・ユーザー ID が、ATIUSERID パラメーターに指定されたユーザー ID の代理ユーザーとして許可されていることを確認します。

TDQ のデフォルト・ジョブ・ユーザー ID

一時データ・キューが **JOBUSERID** パラメーターで定義されると、その定義をインストールしたユーザーに対して、代理ユーザー・セキュリティー検査が実行されます。

EXCI 呼び出しの代理ユーザー検査

代理ユーザー検査は、バッチ領域のユーザー ID が、別のユーザーに対して DPL 呼び出しを発行することを許可されているかどうか (つまり、DPL_request 呼び出しで指定されたユーザー ID の代理として許可されているかどうか) を確認するために実行されます。

外部 CICS インターフェース (EXCI) クライアント・ジョブが代理ユーザー検査を受けるようにするには、EXCI オプション・テーブルの DFHXCOPT で SURROGCHK=YES を指定します。 SURROGCHK=YES を指定する場合は、バッチ領域のユーザー ID を、すべての DPL_request 呼び出しで指定されたユーザー ID の代理として許可します。つまり、バッチ領域のユーザー ID は、SURROGAT 一般リソース・クラスの *userid*.DFHEXCI というプロファイルに対する READ 権限を持っている必要があります (ここで *userid* は、DPL 呼び出しで指定されたユーザー ID です)。例えば、次のコマンドは DPL ユーザー ID の代理プロファイルを定義し、EXCI バッチ領域に READ 権限を付与します。

```
RDEFINE  SURROGAT dpl_userid.DFHEXCI UACC(NONE) OWNER(DPL_userid)
PERMIT   userid.DFHEXCI CLASS(SURROGAT) ID(batch_region_userid)
                    ACCESS(READ)
```

代理ユーザー検査が有効である (SURROGCHK=YES) が、ユーザー ID が DPL 呼び出しで指定されていない場合、DPL 呼び出しのユーザー ID はデフォルトでバッチ領域のユーザー ID になるため、代理ユーザー検査は実行されません。

代理ユーザー・セキュリティー検査を行わない場合は、DFHXCOPT オプション・テーブルで SURROGCHK=NO を指定します。

代理ユーザー検査は、バッチ領域のユーザー ID が CICS サーバーの領域ユーザー ID と同じである場合に役立ちます。この場合、リンク・セキュリティー検査はバイ

パスされます。この場合は、DPL 呼び出しで指定された USERID が認証済みユーザー ID でない (パスワードがパスしていない) ため、代理ユーザー検査が推奨されます。

バッチ領域のユーザー ID と CICS 領域のユーザー ID が異なっている場合、リンク・セキュリティー検査が実施されます。リンク・セキュリティーを使用する場合、DPL 呼び出しで渡される非認証ユーザー ID は、リンク・セキュリティー検査で許可されているよりも高い権限を獲得することはできません。それは、リンク・セキュリティー検査で許可されているのと同じ (またはそれより低い) 権限しか獲得できません。

Db2 の AUTHID パラメーターおよび COMAUTHID パラメーターに対するユーザー ID

AUTHID 属性、SIGNID 属性、または COMAUTHID 属性を指定する DB2CONN リソースをインストールする場合、AUTHID を指定する DB2ENTRY リソースをインストールする場合、またはそれらの属性のいずれかを変更する場合、CICS は、操作を実行するユーザー ID が、AUTHID、COMAUTHID、または SIGNID の代理ユーザーとして許可されていることを確認します。この検査は、CICS コールド・スタートまたは初期始動でのグループ・リスト・インストールの際に、CICS 領域ユーザー ID にも適用されます。

これらの属性について詳しくは、DB2CONN リソースおよびDB2ENTRY リソースを参照してください。

XUSER システム初期設定パラメーターも AUTHTYPE 属性および COMAUTHTYPE 属性へのアクセスを制御するために使用されますが、それらのパラメーターのセキュリティー管理は FACILITY 一般リソース・クラスによって実行されます。詳細については、CICS DB2 環境でのセキュリティーを参照してください。

URIMAP リソース定義のユーザー ID

USERID 属性を指定する URIMAP リソース定義をインストールする場合、またはこの属性を変更する場合、CICS は、操作を実行するユーザー ID が、USERID 属性に指定されたユーザー ID の代理ユーザーとして許可されているかどうかを検査します。これは、CICS のコールド・スタートまたは初期始動でグループ・リストをインストールする際の CICS 領域ユーザー ID にも適用されます

URIMAP リソース定義は、CICS Web サポートに使用されます。このリソース定義について詳しくは、URIMAP リソースを参照してください。

代理ユーザー検査の RACF 定義

CICS 代理ユーザー検査を使用可能にするには、RACF データベース内で CICS 用に適切な SURROGAT クラス・プロファイルを定義し、適切な SURROGAT プロファイルに対して CICS 代理ユーザーを許可します。

CICS 代理ユーザー検査には、3 つの形式の代理クラス・プロファイル名を定義できます。これらの SURROGAT クラス・プロファイルの名前は、以下の命名規則に準拠している必要があります。

***userid*.DFHSTART**

userid は、次のいずれかを表します。

- 開始されたトランザクションの実行に使用されるユーザー ID。
- RUN コマンドによって開始された CICS Business Transaction Services (BTS) プロセスまたはアクティビティーに関連付けられているユーザー ID。

***userid*.DFHINSTL**

userid は、次のいずれかを表します。

- PLTPIUSR システム初期設定パラメーターで指定された PLT ユーザー ID
- トリガー・レベル・トランザクションに関連付けられているユーザー ID
- DFLTUSER システム初期設定パラメーターで指定された CICS デフォルト・ユーザー ID
- 事前設定端末セキュリティーに指定されたユーザー ID
- Db2 リソース定義の AUTHID または COMAUTHID パラメーターに指定されたユーザー ID
- URIMAP リソース定義の USERID 属性で指定されたユーザー ID
- イベント処理トランザクション開始アダプターのトランザクション・ユーザー ID に指定されたユーザー ID

CREATE IPCONN または CREATE CONNECTION コマンドのいずれかを発行するタスクに関連付けられているユーザー ID が、SECURITYNAME オプションで指定されたユーザーの許可された代理でない場合は、NOTAUTH エラーが返されます。

***userid*.DFHQUERY**

userid は、リソースへのアクセス権限を照会する対象ユーザーのユーザー ID を表します。

外部 CICS インターフェース (EXCI) セキュリティー検査用の代理クラス・プロファイルの形式を定義することもできます。

***userid*.DFHEXCI**

userid は、クライアント・バッチ領域の DPL 呼び出しで指定されたユーザーを表します。

この EXCI プロファイルに対して代理を許可するには、EXCI バッチ領域のユーザー ID に READ アクセス権限を付与します。

EXCI バッチ領域の代理セキュリティー検査は、ターゲット CICS 領域のセキュリティー定義とは無関係です。EXCI オプション・テーブル (DFHXCOPT) に SURROGCHK が指定されている場合は、CICS セキュリティー設定にかかわらず、EXCI クライアント・プログラムのアドレス・スペースで代理セキュリティー検査が実行されます。

これらのプロファイルの 1 つに対して代理ユーザーを許可するには、READ アクセス権限を付与する必要があります。

ユーザーをそれ自身の代理として定義する必要はありません。この場合、CICS は代理検査をバイパスします。

代理リソース・クラスの定義については、「z/OS Security Server RACF セキュリティー管理者のガイド」に詳しく記載されています。多数の RACF 定義を行うのに役立つ、総称リソース・クラスや RACFVARS プロファイルなどの RACF 機能を使用する必要がある場合は、この資料を参照してください。

代理ユーザー検査の RACF 定義の例

RACF に対する代理ユーザーは、以下のようにして定義します。

- 代理ユーザーに代理の役割を果たすことを要求する各ユーザーに対して、SURROGAT 一般リソース・クラス内で *userid.resource_name* プロファイルを定義します。この目的には、RACF RDEFINE SURROGAT コマンドを使用します。
- SURROGAT クラス・プロファイル内で定義されたユーザーの代理の役割を果たす各ユーザー ID を許可します。この目的には、RACF PERMIT コマンドを使用します。

PLT セキュリティー

PLT セキュリティー検査の場合、CICS 領域ユーザー ID は、**PLTPIUSR** システム初期設定パラメーターで定義されている PLT ユーザー ID の代理として許可される必要があります。

CICS 領域ユーザー ID には、PLT ユーザー ID によって所有されている SURROGAT リソース・クラス・プロファイルへのアクセス権限を付与する必要があります。これについては、以下の例で示しています。この例では、CICS 領域ユーザー ID は CICSHT01 であり、PLT セキュリティー・ユーザー ID は PLTUSER です。

```
RDEFINE  SURROGAT  PLTUSER.DFHINSTL  UACC(NONE) OWNER(PLTUSER)
PERMIT   PLTUSER.DFHINSTL  CLASS(SURROGAT) ID(CICSHT01)  ACCESS(READ)
```

SURROGAT プロファイルを定義して PLT セキュリティーを有効にすることに加え、PLT セキュリティーが (**PLTPISEC** システム初期設定パラメーターの使用により) アクティブな場合は、PLT プログラムがアクセスするすべてのリソースのアクセス・リストに PLT ユーザー ID も追加します。例えば、PLTPISEC=RESSEC を指定する場合は、PLT ユーザー ID が、セキュリティーがアクティブであるすべての CICS リソースに対して許可されていることを確認します。

開始済みトランザクション:

開始済みトランザクションの場合、CICS は 3 つのレベルの代理ユーザーを必要とすることがあります。

(START コマンドに必要となる可能性がある、さまざまな代理ユーザーの詳細については、 155 ページの『開始済みトランザクション』を参照してください。)

第 1 レベルでの開始済みトランザクションのセキュリティーでは、START コマンドを発行するトランザクションのユーザー ID は、START コマンドで指定されたユーザー ID の代理として許可される必要があります。

例えば、USERID2 で実行しているトランザクションは以下を発行します。

```
EXEC CICS START TRANSID('TBAK') USERID('USERID1')
```

USERID2 は、RACF に、USERID1 (READ 権限を持つ) の代理として定義されていなければなりません。これは、以下の RACF コマンドで示されています。

```
RDEFINE  SURROGAT  USERID1.DFHSTART  UACC(NONE)  OWNER(USERID1)
PERMIT   USERID1.DFHSTART  CLASS(SURROGAT) ID(USERID2)  ACCESS(READ)
```

代理セキュリティーについて詳しくは、 184 ページの『ユーザーの代理権限の照会』を参照してください。

CICS コマンド・セキュリティー

CICS コマンド・セキュリティーはシステム・プログラミング・コマンドの使用状況を管理します。そのコマンドは、特別な CICS 変換プログラム・オプションが必要なコマンド、SP です。

コマンド・セキュリティーの概要

CICS コマンド・セキュリティーは、システム・プログラミング・コマンド (特別な CICS 変換プログラム・オプション SP を必要とするコマンド) に適用されます。セキュリティー検査は、これらのコマンド (コマンドが CICS アプリケーション・プログラムから発行される場合) および CEMT マスター端末トランザクションで発行できる同等のコマンドに対して実行されます。

表 11 は、コマンド・セキュリティー検査の対象となるコマンドを示します。

表 11. システム・プログラミング・コマンドに必要なアクセス権限

コマンド名	必要なアクセス権限
COLLECT CSD DISCONNECT CSD ENDBRGROUP CSD ENDBRLIST CSD ENDBRRSRCE CSD GETNEXTGROUP CSD GETNEXTLIST CSD GETNEXTRSRCE CSD INQUIREGROUP CSD INQUIRELIST CSD INQUIRERSRCE CSD STARTBRGROUP CSD STARTBRLIST CSD STARTBRRSRCE EXTRACT STATISTICS INQUIRE	READ

表 11. システム・プログラミング・コマンドに必要なアクセス権限 (続き)

コマンド名	必要なアクセス権限
DISABLE CSD ADD CSD ALTER CSD APPEND CSD COPY CSD DEFINE CSD DELETE CSD LOCK CSD REMOVE CSD RENAME CSD UNLOCK CSD USERDEFINE ENABLE EXTRACT (ただし EXTRACT STATISTICS 以外) PERFORM RESYNC SET	UPDATE
CREATE CSD INSTALL DISCARD	ALTER
注: PERFORM CORBASERVER SCAN を実行すると、DJAR リソースが動的に作成およびインストールされる可能性があるため、PERFORM CORBASERVER SCAN コマンドは CORBASERVER リソースへの UPDATE 権限だけでなく、DJAR コマンド・セキュリティー・リソースへの ALTER アクセス権限も必要です。	

コマンド・セキュリティーは、トランザクションに対して定義されたトランザクション・セキュリティーまたはリソース・セキュリティーに加えて作動します。例えば、ユーザーが FILA というトランザクションの使用を許可されている場合に、ユーザーが使用を許可されていない EXEC CICS INQUIRE FILE コマンドを FILA が発行すると、CICS はコマンドに応答して「許可されていない」(NOTAUTH) 状態を発行し、コマンドは失敗します。

フロントエンド・プログラミング・インターフェース・セキュリティーは、FEPIRESOURCE リソース名を使用して、システム・プログラミング・コマンドと同じ許可メカニズムを使用します。

注: CICS 変換プログラムで SP オプションを使用できるユーザーを決定するには、RACF を使用して、変換時に DFHEITBS テーブルをロードできるユーザーを制御します。RACF プログラム制御について詳しくは、z/OS Security Server RACF セキュリティー管理者のガイドを参照してください。 DFHEITBS は、システム・プログラミング・コマンドを定義する言語定義テーブルです。これはオンデマンドでのみロードされます。

コマンド・セキュリティー検査の対象となる CICS リソース

トランザクションおよびリソースのセキュリティー検査の場合、RACF のリソースを、それらに割り当て済みの ID (例えば、ファイル名、キュー名、トランザクション名) を使用して識別します。ただし、コマンド・セキュリティーの場合、リソース ID は、CICS によってすべて事前定義されており、リソース・プロファイルを RACF に定義するとき、これらの事前定義された名前を使用します。

コマンド・セキュリティー検査の対象となるリソース ID の完全なリストを、関連するコマンドとともに、表 12に示します。これらのコマンドのほとんどは、CEMT および EXEC の両方の CICS インターフェースに共通しています。CEMT に固有のコマンドには、その前に CEMT が付けられています。

表 *12.* コマンド・セキュリティー検査の対象となる *CICS* リソースのリソース *ID* および関連コマンド

リソース **ID** **1**	関連する **CICS** コマンド
ASSOCIATION	**INQUIRE ASSOCIATION**
ATOMSERVICE	**CREATE ATOMSERVICE** **DISCARD ATOMSERVICE** **INQUIRE ATOMSERVICE** **SET ATOMSERVICE**
AUTINSTMODEL	**DISCARD AUTINSTMODEL** **INQUIRE AUTINSTMODEL**
AUTOINSTALL	**INQUIRE AUTOINSTALL** **SET AUTOINSTALL**
BRFACILITY	**INQUIRE BRFACILITY** **SET BRFACILITY**
BUNDLE	**CREATE BUNDLE** **DISCARD BUNDLE** **INQUIRE BUNDLE** **SET BUNDLE**
BUNDLEPART	**INQUIRE BUNDLEPART**
CAPDATAPRED	**INQUIRE CAPDATAPRED**
CAPINFOSRCE	**INQUIRE CAPINFOSRCE**
CAPOPTPRED	**INQUIRE CAPOPTPRED**
CAPTURESPEC	**INQUIRE CAPTURESPEC**
CFDTPOOL	**INQUIRE CFDTPOOL**
CONNECTION	**CREATE CONNECTION** **DISCARD CONNECTION** **INQUIRE CONNECTION** **SET CONNECTION**

表 *12.* コマンド・セキュリティー検査の対象となる *CICS* リソースのリソース *ID* および関連コマンド *(*続き*)*

リソース **ID** 1	関連する **CICS** コマンド
CSD	**CSD ADD** **CSD ALTER** **CSD APPEND** **CSD COPY** **CSD DEFINE** **CSD DELETE** **CSD DISCONNECT** **CSD ENDBRGROUP** **CSD ENDBRLIST** **CSD ENDBRRSRCE** **CSD GETNEXTGROUP** **CSD GETNEXTLIST** **CSD GETNEXTRSRCE** **CSD INQUIREGROUP** **CSD INQUIRELIST** **CSD INQUIRERSRCE** **CSD INSTALL** **CSD LOCK** **CSD REMOVE** **CSD RENAME** **CSD STARTBRGROUP** **CSD STARTBRLIST** **CSD STARTBRRSRCE** **CSD UNLOCK** **CSD USERDEFINE**
DB2CONN	**CREATE DB2CONN** **DISCARD DB2CONN** **INQUIRE DB2CONN** **SET DB2CONN**
DB2ENTRY	**CREATE DB2ENTRY** **DISCARD DB2ENTRY** **INQUIRE DB2ENTRY** **SET DB2ENTRY**
DB2TRAN	**CREATE DB2TRAN** **DISCARD DB2TRAN** **INQUIRE DB2TRAN** **SET DB2TRAN**
DELETSHIPPED	**INQUIRE DELETSHIPPED** **PERFORM DELETSHIPPED** **SET DELETSHIPPED**
DISPATCHER	**INQUIRE DISPATCHER** **SET DISPATCHER**
DOCTEMPLATE	**CREATE DOCTEMPLATE** **DISCARD DOCTEMPLATE** **INQUIRE DOCTEMPLATE** **SET DOCTEMPLATE**

表 12. コマンド・セキュリティー検査の対象となる *CICS* リソースのリソース *ID* および関連コマンド *(*続き*)*

リソース **ID** [1]	関連する **CICS** コマンド
DSNAME	**INQUIRE DSNAME** **SET DSNAME**
DUMP	**CEMT PERFORM SNAP** **PERFORM DUMP**
DUMPDS	**INQUIRE DUMPDS** **SET DUMPDS**
ENQMODEL	**CREATE ENQMODEL** **INQUIRE ENQMODEL** **SET ENQMODEL**
EPADAPTER	**INQUIRE EPADAPTER** [4] **SET EPADAPTER** [4]
EPADAPTERSET	**INQUIRE EPADAPTERSET** [4] **SET EPADAPTERSET** [4]
EPADAPTINSET	**INQUIRE EPADAPTINSET** [4]
EVENTBINDING	**INQUIRE EVENTBINDING** [4] **SET EVENTBINDING** [4]
EVENTPROCESS	**INQUIRE EVENTPROCESS** **SET EVENTPROCESS**
EXCI	**INQUIRE EXCI**
EXITPROGRAM	**DISABLE PROGRAM** **ENABLE PROGRAM** **EXTRACT EXIT** **RESYNC ENTRYNAME** **INQUIRE EXITPROGRAM**
FEPIRESOURCE	特定の **FEPI** コマンド
FILE	**CREATE FILE** **DISCARD FILE** **INQUIRE FILE** **SET FILE**
HOST	**INQUIRE HOST** **SET HOST**
IPCONN	**CREATE IPCONN** **DISCARD IPCONN** **INQUIRE IPCONN** **SET IPCONN**
IRC	**INQUIRE IRC** **SET IRC**
JOURNALMODEL	**CEMT INQUIRE JMODEL** **CREATE JOURNALMODEL** **DISCARD JOURNALMODEL** **INQUIRE JOURNALMODEL**
JOURNALNAME	**INQUIRE JOURNALNAME** **SET JOURNALNAME**

表 12. コマンド・セキュリティー検査の対象となる CICS リソースのリソース ID および関連コマンド (続き)

リソース ID [1]	関連する CICS コマンド
JVMSERVER	CREATE JVMSERVER DISCARD JVMSERVER INQUIRE JVMSERVER SET JVMSERVER
LIBRARY	CREATE LIBRARY [4] DISCARD LIBRARY [4] INQUIRE LIBRARY [4] SET LIBRARY [4]
LINE	CEMT INQUIRE LINE CEMT SET LINE
LSRPOOL	CREATE LSRPOOL
MAPSET	CREATE MAPSET
MODENAME	INQUIRE MODENAME SET MODENAME
MONITOR	INQUIRE MONITOR SET MONITOR
MQCONN	CREATE MQCONN DISCARD MQCONN INQUIRE MQCONN SET MQCONN
MQMON	CREATE MQMONITOR DISCARD MQMONITOR INQUIRE MQMONITOR SET MQMONITOR
MVSTCB	COLLECT STATISTICS INQUIRE MVSTCB
NODEJSAPP	INQUIRE NODEJSAPP [4]
OSGIBUNDLE	INQUIRE OSGIBUNDLE [4]
OSGISERVICE	INQUIRE OSGISERVICE [4]
PARTITIONSET	CREATE PARTITIONSET
PARTNER	CREATE PARTNER DISCARD PARTNER INQUIRE PARTNER
PIPELINE	CREATE PIPELINE DISCARD PIPELINE INQUIRE PIPELINE PERFORM PIPELINE SET PIPELINE
PROCESSTYPE	CEMT INQUIRE PROCESSTYPE CEMT SET PROCESSTYPE CREATE PROCESSTYPE DISCARD PROCESSTYPE

表 12. コマンド・セキュリティー検査の対象となる *CICS* リソースのリソース *ID* および関連コマンド *(*続き*)*

リソース **ID** [1]	関連する **CICS** コマンド
PROFILE	**CREATE PROFILE** **DISCARD PROFILE** **INQUIRE PROFILE**
PROGRAM	**CREATE PROGRAM** [4] **DISCARD PROGRAM** [4] **INQUIRE PROGRAM** [4] **SET PROGRAM** [4] **SET PROGRAM REPLICATION** [5]
REQID	**INQUIRE REQID**
RESETTIME	**PERFORM RESETTIME** [3]
RRMS	**INQUIRE RRMS**
SECURITY	**PERFORM SECURITY REBUILD** **PERFORM SSL REBUILD**
セッション	**CREATE SESSIONS**
SHUTDOWN	**PERFORM SHUTDOWN** [2]
STATISTICS	**COLLECT STATISTICS** **EXTRACT STATISTICS** **PERFORM STATISTICS RECORD** **INQUIRE STATISTICS** **SET STATISTICS**
STORAGE	**INQUIRE STORAGE**
STREAMNAME	**INQUIRE STREAMNAME**
SUBPOOL	**INQUIRE SUBPOOL**
SYSDUMPCODE	**INQUIRE SYSDUMPCODE** [3] **SET SYSDUMPCODE** [3]
SYSTEM	**INQUIRE SYSTEM** **SET SYSTEM** **INQUIRE FEATUREKEY**
TASK	**INQUIRE TASK** **SET TASK**
TCLASS	**CREATE TRANCLASS** **DISCARD TRANCLASS** **INQUIRE TRANCLASS** **SET TRANCLASS** **INQUIRE TCLASS** **SET TCLASS**
TCPIP	**INQUIRE TCPIP** **SET TCPIP**
TCPIPSERVICE	**CREATE TCPIPSERVICE** **DISCARD TCPIPSERVICE** **INQUIRE TCPIPSERVICE** **SET TCPIPSERVICE**

表 12. コマンド・セキュリティー検査の対象となる CICS リソースのリソース ID および関連コマンド (続き)

リソース ID [1]	関連する CICS コマンド
TDQUEUE	CREATE TDQUEUE DISCARD TDQUEUE INQUIRE TDQUEUE SET TDQUEUE
TEMPSTORAGE	INQUIRE TEMPSTORAGE SET TEMPSTORAGE
TERMINAL	INQUIRE NETNAME SET NETNAME CREATE TERMINAL DISCARD TERMINAL INQUIRE TERMINAL SET TERMINAL
TRACEDEST	INQUIRE TRACEDEST SET TRACEDEST
TRACEFLAG	INQUIRE TRACEFLAG SET TRACEFLAG
TRACETYPE	INQUIRE TRACETYPE SET TRACETYPE
TRANDUMPCODE	INQUIRE TRANDUMPCODE [3] SET TRANDUMPCODE [3]
TRANSACTION	CREATE TRANSACTION [4] DISCARD TRANSACTION [4] INQUIRE TRANSACTION [4] SET TRANSACTION [4]
TSMODEL	CREATE TSMODEL DISCARD TSMODEL INQUIRE TSMODEL
TSPOOL	INQUIRE TSPOOL
TSQUEUE	INQUIRE TSQUEUE
TSQNAME	INQUIRE TSQNAME SET TSQNAME
TYPETERM	CREATE TYPETERM
UOW	INQUIRE UOW SET UOW
UOWDSNFAIL	INQUIRE UOWDSNFAIL
UOWENQ	INQUIRE UOWENQ
UOWLINK	SET UOWLINK INQUIRE UOWLINK
URIMAP	CREATE URIMAP [4] DISCARD URIMAP [4] INQUIRE URIMAP [4] SET URIMAP [4]
VTAM	INQUIRE VTAM SET VTAM

表 12. コマンド・セキュリティー検査の対象となる CICS リソースのリソース ID および関連コマンド (続き)

リソース ID 1	関連する CICS コマンド
WEB	**INQUIRE WEB** **SET WEB**
WEBSERVICE	**CREATE WEBSERVICE** **DISCARD WEBSERVICE** **INQUIRE WEBSERVICE** **SET WEBSERVICE**
WLMHEALTH	**INQUIRE WLMHEALTH** **SET WLMHEALTH**
XMLTRANSFORM	**INQUIRE XMLTRANSFORM** **SET XMLTRANSFORM**

注:

1. プレフィックス変換を使用する場合は、SECPRFX パラメーターによって指定された値を、コマンド・リソース名の前に付ける必要があります。
2. SHUTDOWN オプションを含むこれらおよびその他の CICS コマンドへのアクセスを許可する場合は、特に慎重に行います。
3. 173 ページの『CEMT の考慮事項』を参照してください。
4. バンドル・コマンド・セキュリティーは、SPI コマンドを使用して BUNDLE リソースに対するアクションを実行する場合、およびそのプロセスで CICS バンドル内で定義されたこのタイプの動的生成リソースをインストール、有効化、無効化、または破棄する場合に適用されます。アプリケーションまたはプラットフォームからこのタイプの動的生成リソースをインストール、有効化、無効化、または破棄する場合、CICS コマンド・セキュリティーは適用されません。詳しくは、バンドルのセキュリティーを参照してください。
5. **SET PROGRAM REPLICATION** には、**SET PROGRAM** 以外に、追加のコマンド・セキュリティー検査があります。詳しくは、137 ページの『リソースおよびコマンドの検査の相互参照』を参照してください。

リソース・プロファイル例

コマンド・セキュリティーを使用して CICS を実行する場合は、164 ページの表 12 内のリソース名をプロファイル名として使用し、適切なアクセス・リストを使用して、リソース・プロファイルを RACF に定義します。代替方法として、VCICSCMD クラス内にリソース・グループ・プロファイルを作成することができます。

次の例では、**RDEFINE** コマンドで、CMDSAMP という名前のプロファイルを定義しています。このプロファイルによって保護されているコマンドは、ADDMEM オペランドに指定されます。 PERMIT コマンドは、ユーザーのグループに、INQUIRE に対するコマンドの実行を次のように許可します。

```
RDEFINE  VCICSCMD CMDSAMP UACC(NONE)
                  NOTIFY(sys_admin_userid)
                  ADDMEM(AUTINSTMODEL, AUTOINSTALL, CONNECTION,
                         DSNAME, TRANSACTION, TRANDUMPCODE, VTAM)
PERMIT CMDSAMP CLASS(VCICSCMD) ID(operator_group) ACCESS(READ)
```

2 つ目の例では、前の例における ADDMEM オペランド内の同じコマンドを使用して、CMDSAMP1 と呼ばれるプロファイルを定義します。 **PERMIT** コマンドは、ユーザーのグループに、次のコマンドに対する PERFORM、SET、および DISCARD の実行を許可します。

```
RDEFINE  VCICSCMD CMDSAMP1 UACC(NONE)
                  NOTIFY(sys_admin_userid)
                  ADDMEM(AUTINSTMODEL, AUTOINSTALL, CONNECTION,
                         DSNAME, TRANSACTION, TRANDUMPCODE, VTAM)
PERMIT CMDSAMP1 CLASS(VCICSCMD) ID(op_group_2) ACCESS(UPDATE)
```

SEC=YES を使用して CICS を実行する場合、ユーザーには、 137 ページの『リソースおよびコマンドの検査の相互参照』に示されているアクセス・レベルが必要になります。

コマンド・セキュリティーを指定するためのパラメーター

CICS は、コマンド・セキュリティーが必要であることを示せるように、**SEC** システム初期設定パラメーターと **SECPRFX** システム初期設定パラメーターに加え、**XCMD** システム初期設定パラメーターと、TRANSACTION リソース定義オプションの CMDSEC 属性を提供しています。

これらのパラメーターは、以下のように要約されます。

XCMD システム初期設定パラメーター

XCMD システム初期設定パラメーターは、CICS 領域でコマンド・セキュリティーをアクティブにするかどうかを指定するために使用します。オプションで、コマンド・セキュリティー・プロファイルを定義した RACF リソース・クラス名を指定するために使用します。

CICS コマンド・プロファイル (CCICSCMD および VCICSCMD) に IBM 提供の RACF リソース・クラス名を使用する場合は、XCMD=YES を指定します。次に、CICS は RACF に、それらのデフォルト・リソース・クラスからストレージ内プロファイルを作成するように要求します。

CICS コマンド・プロファイルにインストール定義リソース・クラス名を使用する場合は、XCMD=*user_class* を指定します。こうすると CICS は RACF に、ユーザー固有のインストール定義リソース・クラスからストレージ内プロファイルを作成するように要求します。

CICS 領域でコマンド・セキュリティーを必要としない場合は、XCMD=NO を指定します。

CMDSEC システム初期設定パラメーター

CMDSEC=ALWAYS システム初期設定パラメーターを指定することで、CMDSEC=YES の効果をすべての CICS トランザクションに強制できます。システム・プログラミング・コマンドを総合的に制御する必要があるインストール済み環境には、CMDSEC オプションを使用することをお勧めします。

CMDSEC トランザクション定義属性

TRANSACTION リソース定義上で CMDSEC 属性を以下のように使用して、コマンド・セキュリティーを適用するトランザクションを指定します。

CMDSEC(NO)
トランザクションのコマンド・セキュリティー検査を要求しません。

CMDSEC(YES)
162 ページの表 11 のシステム・プログラミング・コマンドに対するコマンド・セキュリティー検査を要求します。

ユーザー・アプリケーションで発行されるか、または CICS 提供トランザクションの CEMT および CECI によって発行されるこれらの各コマンドに対して、CICS は RACF を呼び出して、トランザクションを開始した端末オペレーターが、指定されたリソースに対してコマンドを使用する権限を持っているかどうかを検査します。

このリソースの属性をすべて表示するには、TRANSACTION リソースを参照してください。

CEDF の下で実行されるトランザクションのセキュリティー検査

CEDF トランザクションの下でトランザクションが実行される場合、CICS はターゲット・トランザクションのセキュリティー設定のみを考慮に入れます。

CEDF トランザクションの下でトランザクションが実行される場合、CICS はターゲット・トランザクションの定義内の CMDSEC 属性を使用します。CEDF の CMDSEC および RESSEC の値は考慮されません。

IBM が提供する DFHEDF グループ内の CEDF の定義では CMDSEC(YES) が指定されます。IBM 提供のグループ内の定義は変更できません。定義を変更するには、その定義を別のグループにコピーします。

CEBR または CECI が EDF 内から呼び出される場合は、トランザクション接続検査が実行されます。

CEDF を使用してトランザクションがテストされる場合は、テスト対象のトランザクションを実行するユーザーの権限が検査されます。テスト対象のトランザクションがアクセスする各リソースに対して、ユーザーはアクセス権限を持っている必要があります。アクセス権限がないと、NOTAUTH 状態が発生します。これは、以下のすべてのリソース検査に適用されます。

- トランザクション接続
- CICS リソース
- CICS コマンド
- QUERY SECURITY コマンドを通してアクセスされる非 CICS リソース
- 代理ユーザー

注**:** CEDF の下で実行されるトランザクションによって次のいずれかの EXEC CICS コマンドが発行されると、パスワードまたはパスワード・フレーズ (該当する場合は、さらに新規パスワードまたはパスワード・フレーズ) がブランクになります。

CHANGE PASSWORD

CHANGE PHRASE

SIGNON

VERIFY PASSWORD

VERIFY PHRASE

CEMT の考慮事項

一般に、CICS 提供の CEMT マスター端末トランザクションが操作するリソースは、同等のシステム・プログラミング・コマンドと同じです。

同等のシステム・プログラミング・コマンドは、 162 ページの表 11 に示されています。通常のトランザクション接続セキュリティーに加え、コマンド・セキュリティーを使用している場合は、必要に応じて、CEMT の許可ユーザーも CICS コマンドに対して許可されていることを確認する必要があります。ユーザーが、CEMT トランザクションの開始を許可されているが、 162 ページの表 11 のシステム・プログラミング・コマンドが依存するリソースに対して許可されていない場合、CICS は NOTAUTH 条件を返します。システム・プログラマーにコマンド・セキュリティー環境内で CEMT コマンドを使用することを許可するには、システム・プログラマーに PERFORM、SET、および DISCARD の各コマンドを発行させるコマンドを保護するグループ・プロファイルに対する UPDATE 権限をシステム・プログラマーに付与します。ユーザーが CEMT SET PROG(xxx) コマンドを発行するときは、XPPT=YES および XCMD=YES を指定して、ユーザーに UPDATE 権限を付与する必要があります。また、システム・プログラマーに INQUIRE コマンドと COLLECT コマンドのみを発行させるコマンドを保護するグループ・プロファイルに対する READ 権限を付与する必要もあります。

```
PERMIT profile_name CLASS(VCICSCMD) ID(user または group) ACCESS(READ)
PERMIT profile_name CLASS(VCICSCMD) ID(user または group) ACCESS(UPDATE)
```

アプリケーション・プログラムの許可障害

CICS は、コマンド・セキュリティー障害の場合は RESP2 で値 100 を返し、リソース・セキュリティー障害の場合は値 101 を返し、代理セキュリティー障害の場合は値 102 を返します。

CICS コマンド・セキュリティーで実行している場合、CICS は NOTAUTH 条件、RESP 値 70 をアプリケーションに返します。これはリソース・セキュリティー障害の場合と同じ条件です。さらに、CICS はメッセージ DFHXS1111 を、CICS セキュリティー一時データ宛先 CSCS に発行します。アプリケーション内のこの値のテストをするには、値を明示的にコーディングするよりも、DFHRESP(NOTAUTH) をコーディングすることをお勧めします。

コマンド・セキュリティー障害、リソース・セキュリティー障害、および代理セキュリティー障害を見分けるには、RESP2 値を検査します。

- コマンド・セキュリティー障害の場合、CICS は、RESP2 で値 100 を返します。
- リソース・セキュリティー障害の場合、RESP2 で値 101 を返します。
- 代理セキュリティー障害の場合、RESP2 で値 102 を返します。

QUERY SECURITY コマンドを使用したセキュリティー検査

アプリケーション・プログラムで **QUERY SECURITY** コマンドを使用して、ユーザーが特定のリソースに対して持つアクセスのレベルを判別します。 **QUERY SECURITY** コマンドは、リソースへのアクセス権限の付与や拒否は行いません。 代わりに、アプリケーション・プログラムが、コマンドによって返される値を使用して、取るべきアクションを判別します。

QUERY SECURITY の動作

QUERY SECURITY の動作は、複数の要因によって決まります。

- システム初期設定パラメーターに SEC=YES が指定されているか SEC=NO が指定されているか
- SECPRFX システム初期設定パラメーターに指定された値
- どのリソース・クラスがアクティブであるか
- 要求を発行するトランザクションがトランザクション・ルーティングの対象となっているかどうか。対象となっている場合は
  - 接続定義でどの ATTACHSEC パラメーターが指定されたか
  - RESTYPE('PSB') の場合のみ、PSBCHK システム初期設定パラメーターが YES と指定されているか NO と指定されているか

注**: QUERY SECURITY** は、トランザクション定義の RESSEC および CMDSEC キーワードによる影響を受けません。

QUERY SECURITY コマンドには、選択するオプションに応じて、異なる 2 つの形式があります。

- **QUERY SECURITY RESTYPE**
- **QUERY SECURITY RESCLASS**

SEC システム初期設定パラメーター

SEC システム初期設定パラメーターは、CICS で使用する外部セキュリティーのレベルを指定します。

表 13 は、関連するリソース・クラスがアクティブであることを前提としています。例えば、QUERY SECURITY RESTYPE('FILE') を発行する場合は XFCT=YES が指定されていると想定します。

表 *13. QUERY SECURITY* コマンドでの *SEC* パラメーターの効果

SEC	RACF アクセス	Query Security Read	Query Security Update	Query Security Control	Query Security Alter
YES	NONE READ UPDATE CONTROL ALTER	notreadable readable readable readable readable	notupdatable notupdatable updatable updatable updatable	notctrlable notctrlable notctrlable ctrlable ctrlable	notalterable notalterable notalterable notalterable alterable
NO	適用なし	readable	updatable	ctrlable	alterable

SECPRFX システム初期設定パラメーター

SECPRFX システム初期設定パラメーターは、CICS が QUERY SECURITY コマンドの RESID オプションに接頭部を適用するかどうかを決定します。例えば、以下のコマンドを発行する場合を考えます。

```
QUERY SECURITY RESTYPE('FILE') RESID('PAYFILE')
```

- SECPRFX=YES が指定されている場合、CICS は CICS 領域ユーザー ID を接頭部として適用し、RACF を呼び出して、*cics_region_userid*.PAYFILE へのユーザーのアクセスを検査します。
- SECPRFX=*prefix* が指定されている場合、CICS は、指定された接頭部を適用し、RACF を呼び出して、*prefix*.PAYFILE へのユーザーのアクセスを検査します。
- SECPRFX=NO が指定されている場合、CICS は接頭部を適用せず、RACF を呼び出して、PAYFILE へのユーザーのアクセスを検査します。

リソース・クラスのシステム初期設定パラメーター

174 ページの表 13 は、関連するリソース・クラスのシステム初期設定パラメーター (例えば、XFCT) がアクティブである場合の **QUERY SECURITY RESTYPE** コマンドの動作を示します。ただし、関連する *X**name*** パラメーターがアクティブでない場合 (例えば、XFCT=NO が指定された場合)、リソースは READABLE、UPDATABLE、CTRLABLE、および ALTERABLE です。

トランザクション・ルーティング

リモート・システムにルーティングされたトランザクションから、USERID オプションなしの **QUERY SECURITY** コマンドが発行される場合、CICS は、指定されたリソースへのリンク・ユーザーのアクセスを検査し、該当する場合はリソースへの端末ユーザーのアクセスも検査します。

ただし、USERID オプションありの **QUERY SECURITY** コマンドが発行される場合、CICS は、代理ユーザー検査を実行します。この検査では、リンク・ユーザーと端末ユーザーが対象となる可能性があります。この場合、リンク・ユーザーと端末ユーザーは、USERID オプションに指定されているユーザー ID に対する、代理ユーザー権限を持っている必要があります。代理ユーザー検査が成功すると、CICS は、USERID オプションに指定されているユーザー ID でリソースにアクセスできるかどうかのみを検査します。

詳しくは、使用している環境に応じて、IPIC link security、Link security with LU6.2 または Link security with MRO を参照してください。

ルーティングされたトランザクションが QUERY SECURITY RESTYPE('PSB') RESID(*psb_name*) を発行するときに、リンク・ユーザーだけでなく端末ユーザーに対しても検査を実行するには、次の条件を両方とも満たす必要があります。

- 接続定義の ATTACHSEC は LOCAL であってはならない (つまり、IDENTIFY、PERSISTENT、MIXIDPE、または VERIFY のいずれかになる)。
- リモート・システムのシステム初期設定パラメーターとして、PSBCHK=YES が指定されている。

RESTYPE オプション

初期設定時に RACLIST によってアクティブ化されたクラスに含まれている CICS リソース (Db2 リソース定義を含む) へのアクセス・レベルを照会するには、**QUERY SECURITY** コマンドで RESTYPE オプションを指定します。

QUERY SECURITY コマンドに対する応答は、このリソースでのリソース検査の結果を示します。リソースが RACF に定義されていない場合、CICS はアクセス権限を付与せず、応答は NOTREADABLE になります。RESTYPE 要求により RACF に渡されるリソース名の長さは、そのリソース・タイプの実際の最大長となるようにしてください。

RESTYPE 値

RESTYPE は、*X*name システム初期設定パラメーターの 1 つに対応するリソース・タイプです。

RESTYPE は、表 14に示す任意の値を取ることができます。

表 *14. QUERY SECURITY RESTYPE* 値

RESTYPE 値	**X***name* パラメーター
ATOMSERVICE	XRES
BUNDLE	XRES
DB2ENTRY	XDB2
DOCTEMPLATE	XRES
EPADAPTER	XRES
EPADAPTERSET	XRES
EVENTBINDING	XRES
FILE	XFCT
JOURNALNAME	XJCT
JOURNALNUM 1	XJCT
JVMSERVER	XRES
PROGRAM	XPPT
PSB	XPSB
SPCOMMAND 2	XCMD
TDQUEUE	XDCT
TRANSACTION	XPCT
TRANSATTACH	XTRAN
TSQUEUE	XTST
TSQNAME	XTST
XMLTRANSFORM	XRES

1. 以前のリリースとの互換性のためにサポートされています。
2. SPCOMMAND は、コマンドに対して RESID を指定する場合に使用可能なリソース・タイプです。

XHFS システム初期設定パラメーターは、z/OS UNIX ファイルに対するリソース・セキュリティーを制御します。 **QUERY SECURITY** コマンドには、このパラメーター

に対応する RESTYPE 値がありません。 z/OS UNIX ファイルに対するアクセス制御は、z/OS UNIX システム・サービスで使用される許可システムに従うため、それぞれの動作が異なります。

RESID values

すべてのケースにおいて (SPCOMMAND リソース・タイプを除いて)、リソース ID (RESID の値) はインストール時に定義されます。

RESID 値を定義するときは、リソース ID にブランク (X'40') を使用する場合の影響に注意してください。例えば、次の場合、

```
QUERY SECURITY RESTYPE('PSB') RESID('A B')
```

ブランクによって RESID が区切られるため、RACF はリソース名 A を使用します。

SPCOMMAND の場合、ID は CICS によってあらかじめ決められています。SPCOMMAND で有効な RESID 値のリストは、次のとおりです。

- ASSOCIATION
- ATOMSERVICE
- AUTINSTMODEL
- AUTOINSTALL
- BRFACILITY
- BUNDLE
- BUNDLEPART
- CAPDATAPRED
- CAPINFOSRCE
- CAPOPTPRED
- CAPTURESPEC
- CFDTPOOL
- CONNECTION
- CSD
- DB2CONN
- DB2ENTRY
- DB2TRAN
- DISPATCHER
- DOCTEMPLATE
- DSNAME
- DUMP
- DUMPDS
- ENQUEUE
- EPADAPTER
- EPADAPTERSET
- EPADAPTINSET

- EVENTBINDING
- EVENTPROCESS
- EXCI
- EXITPROGRAM
- FEPIRESOURCE
- FILE
- HOST
- IPCONN
- IRC
- JOURNALMODEL
- JOURNALNAME
- JVMSERVER
- LIBRARY
- MODENAME
- MONITOR
- MQCONN
- MQMON
- MVSTCB
- NODEJSAPP
- OSGIBUNDLE
- OSGISERVICE
- PARTNER
- PIPELINE
- PROCESS
- PROFILE
- PROGRAM
- REQID
- REQUEST
- RESETTIME
- RRMS
- SECURITY
- SHUTDOWN
- STATISTICS
- STORAGE
- SUBPOOL
- SYSDUMPCODE
- SYSTEM
- TASK
- TCLASS
- TCPIP

- TCPIPSERVICE
- TDQUEUE
- TEMPSTORAGE
- TERMINAL
- TRACEDEST
- TRACEFLAG
- TRACETYPE
- TRANDUMPCODE
- TRANSACTION
- TSQUEUE
- TSMODEL
- TSPOOL
- TYPETERM
- UOW
- UOWDSNFAIL
- UOWENQ
- UOWLINK
- URIMAP
- VOLUME
- VTAM
- WEB
- WEBSERVICE
- XMLTRANSFORM

注: VTAM は、現在は z/OS Communications Server と呼ばれています。

QUERY SECURITY RESTYPE を使用すると、アプリケーション・プログラムは、トランザクションが実行されている環境で指定のリソースに対して端末ユーザーが持つアクセス・レベルを RACF から要求できます。

RACF を呼び出す前に、CICS はリソースがインストールされているかを検査します。リソースが存在しない場合、CICS は RACF を呼び出さず、NOTFND 状態を返します。ただし、この検査は PSB に対しては行われない ことに注意してください。

RESTYPE が TRANSATTACH であり、RESID パラメーターに指定されたトランザクションがローカル領域で不明である場合は、NOTFND 状態が返されます。ただし、動的トランザクション・ルーティングが使用されている場合は、トランザクションを端末専有領域にインストールする必要はありません。不明なトランザクション ID が入力された場合は、DTRTRAN システム初期設定パラメーターで指定されたトランザクションが接続されます。

トランザクションは動的にルーティングされる場合があるため、アプリケーション・プログラマーは、NOTFND 状態であっても必ずしも端末ユーザーがトランザクション ID を入力できないわけではないことに注意してください。

QUERY SECURITY RESTYPE によって返される値の例

このセクションでは、システム初期設定パラメーターで指定された内容に応じて、**QUERY SECURITY RESTYPE** によって返される値のいくつかの例を示します。

SEC=NO

SEC=NO を指定して、以下を発行します。

```
QUERY SECURITY RESTYPE('FILE') RESID('PAYFILE') ALTER(alter_cvda)
```

これによって以下が返されます。

```
alter_cvda = DFHVALUE(ALTERABLE)
```

SEC=NO が、CICS 領域全体に対するセキュリティー検査を実行しないことを示すためです。

SEC=YES および **XFCT=NO**

SEC=YES および XFCT=NO を指定して、以下を発行します。

```
QUERY SECURITY RESTYPE('FILE') RESID('PAYFILE') ALTER(alter_cvda)
```

これによって以下が返されます。

```
alter_cvda = DFHVALUE(ALTERABLE)
```

XFCT=NO が、ファイルに対するセキュリティー検査を実行しないことを示すためです。

SEC=YES、**XDCT=YES**、および **SECPRFX=NO**

SEC=YES、XDCT=YES、および SECPRFX=NO を指定して、以下を発行します。

```
QUERY SECURITY RESTYPE('TDQUEUE') RESID('TDQ1') READ(read_cvda)
```

これによって以下が返されます。

```
read_cvda = DFHVALUE(READABLE)
```

これは、ユーザーが DCICSDCT クラスまたは ECICSDCT グループ・クラス内の「TDQ1」に対して READ 以上の権限を持っている場合です。

SEC=YES、**XTRAN=YES**、および **SECPRFX=YES**

SEC=YES、XTRAN=YES、および SECPRFX=YES を指定して、以下を発行します。

```
QUERY SECURITY RESTYPE('TRANSATTACH') RESID('TRN1') READ(read_cvda)
```

これによって以下が返されます。

```
read_cvda = DFHVALUE(NOTREADABLE)
```

これは、ユーザーが TCICSTRN クラスまたは GCICSTRN グループ・クラス内の `cics_region_userid.TRN1` に対して READ 以上の権限を持っていない場合です。

SEC=YES、**XTRAN=YES**、および **SECPRFX=YES**

SEC=YES、XTRAN=YES、および SECPRFX=YES を指定して、以下を発行します。

```
QUERY SECURITY RESTYPE('TRANSATTACH') RESID('TRN1') READ(read_cvda)
```

これによって以下が返されます。

```
read_cvda = DFHVALUE(NOTREADABLE)
```

これは、ユーザーが TCICSTRN クラスまたは GCICSTRN グループ・クラス内の cics_region_userid.TRN1 に対して READ 以上の権限を持っていない場合です。

SEC=YES、**XCMD=$USRCMD**、および **SECPRFX=***prefix*

SEC=YES、XCMD=$USRCMD、および SECPRFX=*prefix* を指定して、以下を発行します。

```
QUERY SECURITY RESTYPE('TRANSATTACH') RESID('TRN1') READ(read_cvda)
```

これによって以下が返されます。

```
read_cvda = DFHVALUE(NOTREADABLE)
```

これは、ユーザーが TCICSTRN クラスまたは GCICSTRN グループ・クラス内の prefix.TRN1 に対して READ 以上の権限を持っていない場合です。

RESCLASS オプション

非 CICS リソースのアクセス・レベルを照会したい場合は、QUERY SECURITY コマンドで RESCLASS オプションを指定します。

RESCLASS は、有効な RACF 一般リソース・クラスの名前です (例えば、TERMINAL、FACILITY、または同等のインストール定義リソース・クラスなど)。28 ページの『システム・リソースを保護するための RACF クラス』を参照してください。RESCLASS によって識別されるクラス名は、変換されずにそのまま使用されます。

注: RACF クラスの DATASET、GROUP、および USER はクラス記述子テーブル (CDT) に出現しないため、これらのクラスに対して照会することはできません。

SECPRFX システム初期設定パラメーターに指定された接頭部の付加は、QUERY SECURITY RESCLASS には適用されません。つまり、CICS は、RACF を呼び出す前に、CICS 領域ユーザー ID もユーザー指定の接頭部も RESID の接頭部として付けません。

システム初期設定パラメーターに SEC=NO が指定されている場合、QUERY SECURITY RESCLASS は常に READABLE、UPDATABLE、CTRLABLE、および ALTERABLE を返します。

QUERY SECURITY RESCLASS に対して、RESID と RESIDLENGTH オプションの両方を指定する必要があります。RACF クラス内のリソース (RESID) の最大長は、クラス記述子テーブル (CDT) に指定されます。RESID 値を定義するときは、RESID にブランク (X'40') を含める場合の影響に注意してください。例えば、次の場合、

```
QUERY SECURITY RESCLASS('MYCLASS') RESID('MY PROFILE') RESIDLENGTH(10)
```

ブランクがあることで INVREQ 状態になります。これは、RACF ではプロファイル名にブランクを埋め込むことが許可されないためです。

注: *Xname* システム初期設定パラメーターによってリソース・クラスが決定する場合、CICS リソースへのアクセスを判別するには、通常は RESTYPE を使用します。ただし、特別な理由により特定の CICS リソース・クラスについて照会したい場合、クラス名はグループ・クラスではなくメンバー・クラスでなければならない (つまり、VCICSCMD ではなく CCICSCMD でなければならない) ことに注意してください。メンバー・クラスが RACLIST によってアクティブ化された場合、グループ・クラス内のプロファイルは自動的に検査されます。例えば、SEC=YES および XCMD=YES が指定されていて、CICS 領域で CCICSCMDと VCICSCMD の両方が RACLIST によってアクティブ化されている場合、QUERY SECURITY RESCLASS('CCICSCMD') は CCICSCMD と VCICSCMD の両方のプロファイルを検査します。

CICS は、関連 X*name* クラスがアクティブな場合のみ (例えば、XCMD=YES または XCMD=$USRCMD の場合)、グループを RACLIST します。

XDB2 システム初期設定パラメーターで CICS に指定したユーザー定義リソース・クラスに定義されている DB2ENTRY リソースへのアクセスを照会する場合も、RESCLASS オプションを使用できます。RACLIST コマンドによるクラスのアクティブ化についてのルールも、XDB2 システム初期設定パラメーターで指定された DB2ENTRY リソース・クラスに適用されます。ユーザー定義の DB2ENTRY リソース・クラスについて詳しくは、32 ページの『DB2ENTRY リソースのリソース・クラス』を参照してください。

QUERY SECURITY RESCLASS('TERMINAL') を発行すると、TERMINAL クラスが以下のコマンドで RACLIST によってシステム・レベルでアクティブ化された場合に限り、TERMINAL と GTERMINL (端末グループ・クラス) の両方のプロファイルが検査されます。

```
SETROPTS RACLIST(TERMINAL)
```

非 **CICS** リソース・クラスの場合は、SETROPTS RACLIST(classname) コマンドを発行すると、グローバル RACLIST を実行できます。詳しくは、『ユーザー定義リソースを RACF に指定する』を参照してください。

ユーザー定義リソースを RACF に指定する

QUERY SECURITY コマンドに RESCLASS オプションを付けて使用する場合は、ユーザー定義クラスの中に、照会しようとする非 CICS リソースを表すユーザー定義リソースを作成する必要があるかもしれません。

以下はその方法です。

1. RACF クラス記述子テーブル (CDT) および RACF ルーター・テーブルにエントリーを追加します。
2. 新しいクラスを活動化し、その新しいクラスにリソースを定義し、そのリソースへのアクセス権限をユーザーに付与します。

注: **QUERY SECURITY** のパフォーマンスを上げるために、新しいリソース・プロファイルを仮想記憶にロードすることを検討してください。

クラス記述子テーブルへの新規リソース・クラスの追加

RACF クラス記述子テーブルには、システム定義の部分と、ICHRRCDE という名前のインストール定義の部分があります。

手順

1. ICHRRCDE マクロをコーディングして、新しいリソース・クラスを ICHRRCDE に追加します。

   MAXLNTH でリソース名の長さを指定します。以下を収めるのに十分な長さを指定します。
   - 8 文字の接頭部
   - ピリオド (.)
   - リソース・タイプの名前

   要確認: RDO リソースでは、リソース・タイプの名前は最大で 12 文字の長さにできます (リソースの名前自体がそれより短い場合でも)。例えば、『PARTITIONSET』 には 12 文字が含まれていますが、PARTITIONSET リソースの名前は 8 文字の長さに制限されているとします。したがって、最大 12 文字のリソース・タイプ名を持つ RDO リソースでは、MAXLNTH=23 を指定します。

   例えば、新規クラス $FILEREC および対応する (オプションの) グループ・クラス $GILEREC を CDT に追加するには、以下のマクロを ICHRRCDE に追加します。

```
$FILEREC ICHERCDE CLASS=$FILEREC,      Entity or Member class          *
                GROUP=$GILEREC,                                        *
                ID=192,                                                *
                MAXLNTH=17,                                            *
                RACLIST=ALLOWED,                                       *
                FIRST=ALPHANUM,                                        *
                OTHER=ANY,                                             *
                POSIT=42,                                              *
                OPER=NO,                                               *
                DFTUACC=NONE

$GILEREC ICHERCDE CLASS=$GILEREC,      Group class                     *
                MEMBER=$FILEREC,                                       *
                ID=191,                                                *
                MAXLNTH=17,                                            *
                FIRST=ALPHANUM,                                        *
                OTHER=ANY,                                             *
                POSIT=42,                                              *
                OPER=NO,                                               *
                DFTUACC=NONE
```

2. ICHRFRTB マクロのコーディングによって、同じクラスを RACF ルーター・テーブル ICHRFR01 に追加します。

```
ICHRFRTB CLASS=$FILEREC,ACTION=RACF
ICHRFRTB CLASS=$GILEREC,ACTION=RACF
```

   ICHERCDE マクロと ICHRFRTB マクロのどちらも、z/OS Security Server RACF マクロおよびインターフェースで説明されています。

3. 2 つのモジュール ICHRRCDE および ICHRFR01 を再作成した場合は、MVS システムを再 IPL して、それらを使用できるようにします。

ユーザー定義リソース・クラスのアクティブ化

新規クラスをシステムにインストールしたら、それらを使用する前に RACF でアクティブ化する必要があります。

これは、システムの SPECIAL 権限を持つユーザーが実行する必要があります。そのユーザーは TSO の下で以下のコマンドを入力します。

```
SETROPTS CLASSACT($FILEREC)
SETROPTS GENERIC($FILEREC)
```

QUERY SECURITY のパフォーマンスを向上させるために、RACLIST オプションを使用して新規リソース・プロファイルを仮想記憶域にロードする必要があります。グループ・クラスを使用している場合、RACLIST オプションは必須です。グループ・クラスとエンティティー・クラスとの間の接続が RACLIST によって解決されるからです。

```
SETROPTS RACLIST($FILEREC)
```

エンティティー・クラス $FILEREC に対して SETROPTS コマンドを発行する必要があります。グループ・クラス $GILEREC に同じ POSIT 番号があるからです。

新規クラス内のリソースの定義

新規クラス内のリソースは、システム SPECIAL 権限を持つユーザー、または新規クラスの CLAUTH 権限を持つユーザーが定義する必要があります。

CLAUTH 権限は、以下の TSO コマンドを発行することで付与されます。

```
ALTUSER userid CLAUTH($FILEREC)
```

必要な権限を持っている場合は、以下の TSO コマンドを発行して、新規リソースを作成できます

```
RDEFINE $FILEREC PAYFILE.SALARY   UACC(NONE)
RDEFINE $FILEREC PAYFILE.TAXBAND  UACC(NONE)
RDEFINE $GILEREC PERSONAL.DETAILS ADDMEM( PERSONAL.DEPT,     +
                                          PERSONAL.MANAGER,  +
                                          PERSONAL.PHONE)    +
                                  UACC(READ)
```

これで、ユーザーに新規リソースの使用を許可する準備ができました。 PAYROLL は、従業員レコードのすべての給与フィールドおよび個人詳細フィールドを更新することが許可されているユーザー・グループの名前であるとします。以下の TSO コマンドは、グループ内のすべてのユーザーに対して UPDATE 権限を付与します。

```
PERMIT PAYFILE.SALARY   CLASS($FILEREC) ID(PAYROLL) ACCESS(UPDATE)
PERMIT PAYFILE.TAXBAND  CLASS($FILEREC) ID(PAYROLL) ACCESS(UPDATE)
PERMIT PERSONAL.DETAILS CLASS($FILEREC) ID(PAYROLL) ACCESS(UPDATE)
```

RACLIST オプションを使用してプロファイルを既にロード済みである場合は、以下のコマンドを発行して仮想記憶域内のプロファイルをリフレッシュします。

```
SETROPTS RACLIST($FILEREC) REFRESH
```

ユーザーの代理権限の照会

ユーザーの代理権限を照会するには、QUERY SECURITY コマンドで RESCLASS('SURROGAT') オプションを指定します。

RESID オプションと RESIDLENGTH オプションも指定する必要があります。指定するべき RESID 値については、 177 ページの『RESID values』で説明します。ただし、このコマンドは XUSER システム初期設定パラメーターによって制御されないため、XUSER=NO が指定されている場合は、NOTREADABLE という予期しない応答を得る可能性があります。例えば、XUSER=YES が指定されている場合に、現行ユーザーが NEWUSER という新規ユーザー ID によるトランザクションの開始を許可されているかどうかを調べるには、以下のコマンドを発行します。

```
QUERY SECURITY RESCLASS('SURROGAT') RESID('NEWUSER.DFHSTART')
RESIDLENGTH(16)  READ(read cvda)
```

QUERY SECURITY のロギング

QUERY SECURITY コマンドのロギングを制御できます。ロギングが有効であり、指定されたリソースへの要求されたアクセス権限を端末ユーザーが持っていない場合は、メッセージ DFHXS1111 が CICS セキュリティー一時データ宛先 CSCS に対して発行されます。該当する場合、RACF メッセージ ICH408I も発行されます。

そのリソースに対して指定されている監査オプションおよびロギング・オプションに応じて、SMF レコードも記録できます。詳しくは、「 z/OS Security Server RACF 監査担当者のガイド」を参照してください。

CVDA に関するプログラミング情報については、CICS-value data areas (CVDAs) を参照してください。

QUERY SECURITY ロギングのオプション

ロギングを制御するには、以下のいずれかのオプションを指定します。

- LOG (デフォルト)
- NOLOG
- LOGMESSAGE(*cvda*)。ここで *cvda* の値は、LOG の場合は 54、NOLOG の場合は 55 です。

例: リソース・セキュリティー検査のための QUERY SECURITY コマンドの使用

アプリケーション内でリソース・セキュリティー検査をカスタマイズするにはさまざまな方法がありますが、その中で、2 つの形式の **QUERY SECURITY** コマンドを使用できます。通常、**QUERY SECURITY** を使用すると、トランザクション・コードをメニューに表示する前に、ユーザーが特定のトランザクションの使用を許可されているかどうかを検査できます。 **QUERY SECURITY** を使用すると、レコード・レベルまたはフィールド・レベルでデータへのアクセスを制御することもできます。

例: セキュリティー検査のレベルの変更

QUERY SECURITY を使用して、RESSEC(YES) または CMDSEC(YES) を指定するアプリケーション・プログラムに対して CICS が実行するものとは異なるレベルのセキュリティー検査を実行できます。

例えば、トランザクションに RESSEC(YES) があり、いくつかの **EXEC CICS READ FILE** コマンドと、いくつかの **EXEC CICS WRITE FILE** コマンドが含まれているとします。各コマンドに対して、CICS はセキュリティー検査を実行し、ユーザーに関連

ファイルへのアクセス権限があることを確認します。これは、毎回同じファイルがアクセスされる可能性があるとしても実行されます。これに代わる方法として、トランザクション定義で RESSEC(NO) を指定して、トランザクション・レベルでセキュリティー検査をオフに切り替えて、アプリケーションの開始時に以下のようなコマンドを実行します。

```
EXEC CICS QUERY SECURITY RESTYPE('FILE') RESID(file_name) UPDATE(cvda)
```

このコマンドにより、トランザクションは RACF をさらに呼び出さなくても続行できます。

注: RESSEC(NO) を使用してリソース・セキュリティー検査をオフに切り替えると、この例のようなファイルだけでなく、すべてのリソース検査がバイパスされます。

例: ユーザーに提供するトランザクションの確認

QUERY SECURITY コマンドを使用して、トランザクション・コードを導入メニューの一部として表示する前に、ユーザーが特定のトランザクションを使用することを許可されているかどうかを検査できます。

この目的でコマンドを使用すると、特定のトランザクションの使用を許可されていないユーザーについての検査のロギングを回避することが必要になる場合があります。そのためには、NOLOG オプションを使用します。

例: レコード・レベルまたはフィールド・レベルでのデータへのアクセスの制御

ファイルの通常の CICS リソース・セキュリティー検査は、ファイル・レベルでのみ動作します。 **QUERY SECURITY** を使用すると、アプリケーションでのデータへのアクセスをレコード・レベルまたはフィールド・レベルで制御できます。

これを実行するには、リソース名 (特定のファイル内のレコードまたはフィールドを表す) を、制御するレコードまたはフィールドに対する適切なアクセス権限を指定して定義します。インストール定義の RACF 一般リソース・クラス内にこれらのリソースを定義し、フィールドを表示または更新する前に、**QUERY SECURITY RESCLASS** コマンドを使用して、ファイル内の特定のフィールドへのユーザーのアクセスを検査することができます (どのフィールドかは、アプリケーション・ロジックにより決定されます)。以下に例を示します。

```
QUERY SECURITY RESCLASS('$FILEREC') RESID('PAYFILE.SALARY')
RESIDLENGTH(14) READ(read_cvda) NOLOG
```

ここで '$FILEREC' は、インストール定義の RACF 一般リソース・クラスです。

CICS トランザクションのセキュリティー

CICS で提供されるトランザクションの RACF プロファイル定義を、3 つのカテゴリーで説明します。各トランザクションは、CICS での使用法を記述するカテゴリー内で識別されます。各カテゴリーは、CICS トランザクション定義と、対応する RACF プロファイルの両方の観点から、推奨セキュリティー仕様を指定します。この 3 つのカテゴリーには必要なすべての CICS トランザクションが含まれており、これらのトランザクションは、CICS システム定義データ・セット (CSD) の初期設

定時に、指定されたグループに生成されます。この初期設定には、CICS サンプル・トランザクション (DFH$ で始まるグループ内のトランザクション) は含まれません。一部のカテゴリー 1 トランザクションは、CSD にありません。それらは、インストール時に CICS によって定義されます。

CICSPlex SM に関連したトランザクションの詳細については、 211 ページの『CICSPlex SM セキュリティーの実装』を参照してください。

3 つのカテゴリーは、以下のとおりです。

カテゴリー **1** トランザクション
: CICS の内部使用専用であり、ユーザー端末から実行できないトランザクション。詳しくは、『カテゴリー 1 トランザクション』を参照してください。

カテゴリー **2** トランザクション
: 特定のサインオン・ユーザーに制限する必要があるトランザクション。例えば、CICS リソースを定義およびインストールするトランザクションへのアクセスは制限した方がよいでしょう。詳しくは、 188 ページの『カテゴリー 2 トランザクション』を参照してください。

カテゴリー **3** トランザクション
: サインオンしているかどうかにかかわらず、すべてのユーザーが使用できるトランザクション。これらのトランザクションは、セキュリティー検査の対象にはなりません。これらのトランザクションに対しては、どのようなセキュリティー定義も冗長です。詳しくは、 193 ページの『カテゴリー 3 トランザクション』を参照してください。

デフォルトでは、すべての CICS トランザクション (カテゴリー 3 トランザクションを除く。 62 ページの『CICS 環境での JES スプールの保護』を参照) が、RACF 保護の対象となります。ただし、トランザクション・セキュリティーをオフに切り替えた状態で CICS 領域を実行する場合は別です。トランザクション・セキュリティーをオフに切り替えるには、以下のいずれかを行います。

- システム初期設定パラメーター SEC=NO を指定して、すべてのセキュリティー検査をオフに切り替えます。
- システム初期設定パラメーター XTRAN=NO を指定して、トランザクション接続セキュリティー検査のみをオフに切り替えます。

トランザクション・セキュリティーを使用して実行する場合 (SEC=YES および XTRAN=YES)、CICS は、カテゴリー 3 トランザクション以外のトランザクション接続ごとにセキュリティー検査を実行します。この検査により、ユーザーがそのトランザクションの実行を許可されているかどうか判別します。

カテゴリー 1 トランザクション

カテゴリー 1 トランザクションが端末と関連付けられることはありません。これらは CICS の内部使用専用であり、ユーザー端末から開始することはできません。これらのトランザクションの CSD 定義は変更しないでください。カテゴリー 1 トランザクションのリストについては、CICS トランザクションのリストを参照してください。

CICS は、領域ユーザー ID がこれらのトランザクションを接続する権限があるかどうかを検査します。領域ユーザー ID が、カテゴリー 1 トランザクションへのアクセスを許可されていない場合、CICS は、無許可のカテゴリー 1 トランザクションごとにメッセージ DFHXS1113 を発行し、初期設定に失敗します。

サンプル CLIST DFH$CAT1 をカスタマイズして実行し、RACF 定義を作成します。プロファイルの所有者、トランザクションの CLASSNAME、および CICS 領域ユーザー ID のリストを指定する必要があります。

サンプル CLIST は、ライブラリー CICSTS55.CICS.SDFHSAMP にあります。

カテゴリー 2 トランザクション

カテゴリー 2 トランザクションは、CICS ユーザーによって開始されるか、CICS ユーザーに関連付けられます。その中には、サインオンしたユーザーに関連付けられるものがあります。アダプターに関連付けられたタスク・ユーザー ID に関連付けられるものもあります。これらのトランザクションの開始権限を、特定の RACF グループに属するユーザー ID に制限します。カテゴリー 2 トランザクションのリストについては、CICS トランザクションのリストを参照してください。

提供されているトランザクションは、推奨される RESSEC および CMDSEC オプションを使用して定義されます。トランザクションの使用を許可されるユーザー・グループが、それらのトランザクションによって使用される CICS リソースおよびコマンドへのアクセスも許可されるようにしてください。これらの定義は変更可能です。変更する場合は、それらの定義を別のグループにコピーします。

ほとんどのカテゴリー 2 トランザクションでは、以下のようにしてそれらを RACF に指定します。

- トランザクション・プロファイル内の UACC(NONE) および AUDIT(FAILURES)。AUDIT(FAILURES) はデフォルトなので、指定する必要はありません。
- アクセス・リスト。

あるリソース・グループ・プロファイルを使用してリソースを保護する場合、別のプロファイルを使用して同じリソースを保護することになっていて、ユーザーが両方のプロファイルへの別々のアクセス権限を持てるようになっている場合には注意してください。プロファイルが異なる (例えば、異なるアクセス・リストを持つ) 場合、RACF は許可検査時に使用されるプロファイルをマージします。このマージ処理は、検査コストを増加させるおそれがあり、特定のユーザーに適用されるアクセス権限を正確に判別するのが困難になる可能性があります。詳しくは、z/OS Security Server RACF セキュリティー管理者のガイドを参照してください。

ユーザーがこのカテゴリーのすべてのトランザクションにアクセスする必要があるとは考えにくいため、複数のサブカテゴリーでトランザクションを定義することを検討してください。インストール済み環境のニーズに最も適した方法で、CICS トランザクションをグループ化できます。サンプル CLIST DFH$CAT2 (ライブラリー CICSTS55.CICS.SDFHSAMP 内) は、カテゴリー 2 トランザクションをグループ化する 1 つの方法を示しています。別のセットアップを使用するには、CLIST を調整するか、独自のものを用意します。

表 *15.* カテゴリー 2 トランザクションの推奨サブカテゴリー。 : 各サブカテゴリーに属するトランザクションを確認するには、CICS トランザクションのリストまたは DFH$CAT2 CLIST を参照してください。

サブカテゴリー	含まれる内容	注
すべてのユーザー用:		
ALLUSER	すべてのユーザーが使用するトランザクション。	「good morning」トランザクション (**GMTRAN** システム初期設定パラメーターで定義) と「good night」トランザクション (**GNTRAN** システム初期設定パラメーターで定義) をこのグループのものとしてトランザクションのリストに追加します。
オペレーター用:		
SYSADM	システムへのフルアクセスを必要とするシステム・プログラマーが使用するトランザクション。	
INQUIRE	リソースの照会のみを必要とする、オペレーターや他のユーザーが使用するトランザクション。	
OPERATOR	オペレーターが使用するトランザクション。	
DBCTL	IMS に対する DBCTL インターフェースのオペレーターが使用するトランザクション。	
CMCIUSER	CICS Explorer を使用するオペレーターと CMCI の他のユーザーが使用するトランザクション。	
アプリケーション開発者用:		
DEVELOPER	非実稼働システムのアプリケーション開発者が使用するトランザクション。	
アプリケーション・ユーザー用:		

表 15. カテゴリー 2 トランザクションの推奨サブカテゴリー。 (続き): 各サブカテゴリーに属するトランザクションを確認するには、CICS トランザクションのリストまたは DFH$CAT2 CLIST を参照してください。

サブカテゴリー	含まれる内容	注
JVMUSER	Liberty アプリケーションのユーザーが使用するトランザクション。	セキュリティー構成によっては、トランザクション CJSA および CJSU を使用するのに、CICS のデフォルト・ユーザーにアクセス権が必要な場合があります。詳しくは、Java アプリケーションに関するセキュリティー、z/OS Connect EE の構成、およびz/OS Connect サービスおよび API の許可の構成を参照してください。 セキュリティー上のベスト・プラクティスとして、Liberty セキュリティーをオンにし、アプリケーション・タスクの実行に CICS デフォルト・ユーザーを使用しないでください。 z/OS Connect サービスの場合、サービスの作業を CICS の特定のトランザクション ID および初期ユーザー ID と関連付けるために CICS が使用する URIMAP リソースをインストールすることができます。 CJSA は、一致する URI がない Web 要求の、デフォルト・トランザクション ID です。任意にアプリケーションが実行されることがないように、CJSA へのアクセスを制限することを検討してください。

表 *15.* カテゴリー *2* トランザクションの推奨サブカテゴリー。 *(*続き*)*: 各サブカテゴリーに属するトランザクションを確認するには、CICS トランザクションのリストまたは DFH$CAT2 CLIST を参照してください。

サブカテゴリー	含まれる内容	注
INTERCOM	複数の領域でトランザクションを実行するアプリケーション・ユーザーが使用するトランザクション。	機能シップを使用する場合、ミラー・トランザクションは機能シップ環境のリモート・ユーザーにとって使用可能でなければなりません。データベースまたはファイルが別の CICS 領域上にある場合、CICS はデータへのアクセス要求を機能シップします。要求は、CICS 提供のいずれかのミラー・トランザクションのもとで実行されます。この状況では、以下の条件が適用されます。 • アプリケーションを実行している端末ユーザーは、ミラー・トランザクションの使用を許可されていなければなりません。 94 ページの『トランザクション・セキュリティー』を参照してください。 • 端末ユーザーは、ミラー・トランザクションがアクセスするデータの使用も許可されていなければなりません。 100 ページの『リソース・セキュリティー』を参照してください。ミラー・トランザクションは RESSEC(YES) が定義された状態で提供されます。したがって、ユーザーのトランザクションが RESSEC(NO) を指定していても、ユーザーがデータへのアクセスを許可されていないと、ミラー・トランザクションは失敗します。 リソース・セキュリティー検査を使用しない場合は、ミラー・トランザクション定義を変更して、RESSEC(NO) を指定します。ミラー・トランザクションは IBM により保護されるリソースであるため、最初にこれらの定義を自分のグループにコピーし、それを変更してください。

表 15. カテゴリー 2 トランザクションの推奨サブカテゴリー。 (続き): 各サブカテゴリーに属するトランザクションを確認するには、CICS トランザクションのリストまたは DFH$CAT2 CLIST を参照してください。

サブカテゴリー	含まれる内容	注
WEBUSER	CICS Web インターフェースを使用してトランザクションを実行するアプリケーション・ユーザーが使用するトランザクション。	CICS デフォルト・ユーザーは、その後にアナライザー・プログラムを使用してタスクに別のユーザー ID が割り当てられる場合でも、最初は CWBA トランザクションへのアクセス権限が必要です。**DFLTUSER** システム初期設定パラメーターに指定された CICS デフォルト・ユーザーがこのトランザクションへのアクセス権限を持っていることを確認してください。提供される CLIST DFH$CAT2 を使用して WEBUSER RACF プロファイルを作成する場合、デフォルト・ユーザーはこのプロファイルへのアクセス権限を持っていなければなりません。
RPCUSER	ONC RPC を使用してトランザクションを実行するアプリケーション・ユーザーが使用するトランザクション。	
PIPEUSER	Web サービス・トランザクションを実行するアプリケーション・ユーザーが使用するトランザクション。	
EVENTUSER	EP アダプターが使用するトランザクション。	これらのトランザクションの RESSEC オプションと CMDSEC オプションが必要としているものと異なる場合は、イベント・バインディング・エディターのアダプター・タブの「拡張オプション」セクションに独自のトランザクション ID を指定することができます。詳しくは、『EP アダプターおよびディスパッチャー情報の指定』を参照してください。
IBM MQ のユーザー用:		
MQADMIN	CKAM CKCN CKDL CKRS CKSD CKSQ	

表 15. カテゴリー 2 トランザクションの推奨サブカテゴリー。 (続き): 各サブカテゴリーに属するトランザクションを確認するには、CICS トランザクションのリストまたは DFH$CAT2 CLIST を参照してください。

サブカテゴリー	含まれる内容	注
MQBRIDGE	CKBC CKBP CKBR	
MQMONITOR	CKTI	
MQSTATUS	CKQC CKBM CKRT CKDP	

カテゴリー 3 トランザクション

カテゴリー 3 トランザクションは、端末ユーザーにより開始されるか、端末に関連付けられます。カテゴリー 3 トランザクションのリストについては、CICS トランザクションのリストを参照してください。

すべての CICS 端末ユーザーは、サインオンしているかどうかに関係なく、このカテゴリーのトランザクションへのアクセス権限を必要とします。このため、カテゴリー 3 トランザクションはセキュリティー検査を免除され、CICS はどの端末ユーザーに対してもこれらのトランザクションの開始を許可します。

これらのトランザクションは、RACF に定義することができます。この定義はタスク接続時の処理には影響しませんが、QUERY SECURITY コマンドをサポートするために必要です。

内部読み取りプログラムに JCL ジョブを実行依頼する場合のセキュリティー

JCL を実行依頼できるユーザーを CICS で保護するには、いくつかの構成定義とセキュリティー定義が必要です。

JCL ジョブは、次の 2 とおりの方法で JCL を作成することによって実行依頼できます。

- 内部読み取りプログラムに定義された外部 TDQ に対して **WRITEQ TD** コマンドを使用する
- **SPOOLOPEN** コマンドで USERID("INTRDR") が指定されたときに **SPOOLWRITE** コマンドを使用する

ジョブの実行依頼に使用されるユーザー ID は、以下の 3 つです。

- 領域ユーザー ID。
- サインオン・ユーザー ID。これは、タスクの実行に使用されているユーザー ID です (ログオンしていない場合は、デフォルトのユーザー ID)。

- ジョブのユーザー ID。これは、JOB カードの USER パラメーターによって指定されます。 JOB ステートメントに USER パラメーターが含まれていない場合、デフォルトはセキュリティー設定によって異なります。これが領域ユーザー ID でない場合、CICS によって USER=*job_userid* が JOB カードに追加されます。CICS TS 5.5 より前のリリースでは、デフォルトは常に領域ユーザー ID です。

TDQ を介して実行依頼された JCL ジョブの保護は、TDQ のリソース・セキュリティーによって行います。 JOB カードに USER パラメーターが指定されている場合は、代理ユーザー検査により保護が強化されます。詳しくは、『TDQ を使用して JCL を実行依頼する場合のセキュリティー』を参照してください。

スプール・コマンドを使用して実行依頼された JCL ジョブの保護は、代理ユーザー検査によって行います。詳しくは、『SPOOLWRITE コマンドを使用して JCL を実行依頼する場合のセキュリティー』を参照してください。

さらに、ジョブ・ユーザー ID のジョブを実行依頼するための権限を領域ユーザー ID に付与するには、領域ユーザー ID のプロファイルが JESSPOOL クラスに含まれている必要があります。代理検査が失敗した場合の RACF 定義およびエラー・メッセージについて詳しくは、 195 ページの『z/OS 代理ユーザー検査』を参照してください

TDQ を使用して JCL を実行依頼する場合のセキュリティー

保護は、TDQ のリソース・セキュリティー検査によって提供されます。

TDQ に書き込まれた JOB カードに USER パラメーターがない場合は、以下のオプションが定義されている場合に実行される代理検査によって追加の保護が提供されます。

- CICS 代理ユーザー検査は、システム初期設定パラメーター **XUSER=YES** によって使用可能になります
- 以下のようにすると、スプール・コマンドの代理ユーザー検査の機能トグルが使用可能になります。

  ```
  com.ibm.cics.spool.surrogate.check=true
  ```

代理検査が失敗すると、**WRITEQ TD** コマンドから NOTAUTH 応答が返されます。代理検査が失敗した場合の RACF 定義およびエラー・メッセージについて詳しくは、 196 ページの『CICS 代理ユーザー検査』を参照してください。

TDQ に書き込まれた JOB カードに USER パラメーターがない場合、ジョブ・ユーザー ID は、TDQ 定義の JOBUSERID オプションで指定された値に設定されます。 CICS による追加の代理検査は行われません。

SPOOLWRITE コマンドを使用して JCL を実行依頼する場合のセキュリティー

スプール・コマンドを使用するには、システム初期設定パラメーター **SPOOL=YES** を使用して CICS を開始する必要があります。

SPOOLWRITE コマンドを使用して作成された JOB カードに USER パラメーターがある場合は、以下のオプションが定義されている場合に実行される代理検査によって保護が提供されます。

- CICS 代理ユーザー検査は、システム初期設定パラメーター **XUSER=YES** によって使用可能になります
- 以下のようにすると、スプール・コマンドの代理ユーザー検査の機能トグルが使用可能になります。

  ```
  com.ibm.cics.spool.surrogate.check=true
  ```

代理検査が失敗すると、**SPOOLWRITE** コマンドから NOTAUTH 応答が返されます。代理検査が失敗した場合の RACF 定義およびエラー・メッセージについて詳しくは、196 ページの『CICS 代理ユーザー検査』を参照してください。

SPOOLWRITE コマンドを使用して作成された JOB カードに USER パラメーターがない場合、ジョブ・ユーザー ID はデフォルトで領域ユーザー ID になります。これは、代理検査の対象になります。次の機能トグルを設定すると、デフォルトのジョブ・ユーザー ID をサインオン・ユーザー ID に変更できます。

```
com.ibm.cics.spool.defaultjobuser=TASK
```

JOB カードでの領域ユーザー ID の指定

JCL を動的に変更せずに、ジョブ・ユーザー ID を使用してジョブを実行できるようにする場合は、JOB カードで USER=&SYSUID を指定します。これにより、ジョブの実行依頼元となった領域ユーザー ID が何であれ、そのユーザー ID を使用してジョブが実行されます。これは、**SPOOLWRITE** コマンドまたは TDQ を使用して作成された JCL ジョブに適用されます。代理セキュリティー検査 (該当する場合) は、領域ユーザー ID を表す &SYSUID を使用してジョブが実行依頼されたときに行われます。

z/OS 代理ユーザー検査

SPOOLWRITE コマンドと **WRITEQ TD** コマンドはサインオン・ユーザー ID によって発行されますが、ジョブは領域ユーザー ID によって実行依頼されます。したがって、ジョブ・ユーザー ID が領域ユーザー ID ではなく、パスワードが指定されていない場合は、ジョブ・ユーザー ID の代わりにジョブを実行依頼する代理権限を領域ユーザー ID に付与する必要があります。これは、CICS 代理ユーザー検査がアクティブであるかどうかに関係なく必須です。

```
RDEFINE SURROGAT job_userid.SUBMIT UACC(NONE) OWNER(sysadmin)
PERMIT job_userid.SUBMIT CLASS(SURROGAT) ID(region_userid) ACCESS(READ)
```

任意のユーザーの代わりにジョブを実行依頼できるように、領域ユーザー ID を構成できます。これを行う場合は、その領域ユーザー ID を CICS の実行以外の目的に使用できないようにすることをお勧めします。

この検査は、ジョブの実行依頼後に実行されます。検査が不合格になるとジョブは失敗し、コンソールおよびジョブ・ログに ICH408I メッセージが出されます。

```
ICH408I USER(job_userid) GROUP(group) NAME(username) SUBMITTER(region_userid)
LOGON/JOB INITIATION - SUBMITTER IS NOT AUTHORIZED BY USER
```

CICS 代理ユーザー検査

以下のオプションが定義されている場合、CICS 代理ユーザー検査が行われます。

- CICS 代理ユーザー検査は、システム初期設定パラメーター **XUSER=YES** によって使用可能になります
- 以下のようにすると、スプール・コマンドの代理ユーザー検査の機能トグルが使用可能になります。

  ```
  com.ibm.cics.spool.surrogate.check=true
  ```

ジョブ・ユーザー ID がサインオン・ユーザー ID ではなく、パスワードが指定されていない場合は、ジョブ・ユーザー ID の代わりにジョブを実行依頼する代理権限をサインオン・ユーザー ID に付与する必要があります。

```
RDEFINE SURROGAT job_userid.SUBMIT UACC(NONE) OWNER(sysadmin)
PERMIT job_userid.SUBMIT CLASS(SURROGAT) ID(signed_on_userid) ACCESS(READ)
```

代理検査を適用可能な場合には、**SPOOLWRITE** または **WRITEQ TD** コマンドで JOB カードが作成されるときに代理検査が実行されます。検査が不合格になるとコマンドは失敗し、NOTAUTH 応答が出されます。さらに、コンソールとジョブ・ログに ICH408I メッセージが出されます。

```
ICH408I USER(job_userid) GROUP(group) NAME(username) SUBMITTER(signed_on_userid)
LOGON/JOB INITIATION - SUBMITTER IS NOT AUTHORIZED BY USER
```

詳細情報

CICS 代理ユーザー検査を使用する方式に移行する場合は、セキュリティーのアップグレードの説明に従ってください。

CICS が CICS 領域のユーザー ID を特定する状況と方法

CICS は、稼働する領域のユーザー ID をさまざまな目的で使用します。

CICS は、領域のユーザー ID を以下の状況で使用します。

- SECPRFX=YES が指定されている場合に、リソース名に接頭部を付けるため。SECPRFX システム初期設定パラメーターの詳細については、 63 ページの『セキュリティー関連システム初期設定パラメーター』 を参照してください。
- カテゴリー 1 トランザクションの検査対象のユーザーとして。
- システム初期設定パラメーターで PLTPIUSR が指定されていない場合に、PLTPI 非端末セキュリティーのデフォルトの PLTPI ユーザーとして。
- SURROGAT 検査のため (例えば、PLTPI およびデフォルトのユーザー ID を使用する権限など)。
- MRO バインド・セキュリティーのため。詳しくは、 335 ページの『MRO セキュリティーの実装』を参照してください。

CICS は領域ユーザー ID を、外部セキュリティー・マネージャーを呼び出すことによって取得します。このセキュリティー・マネージャーは、ユーザー ID を、そのジョブに関連する RACF 制御ブロックから抽出します。セキュリティー・ドメインと MRO バインド・セキュリティーはそれぞれ、RACROUTE REQUEST=EXTRACT マクロを発行することによって領域ユーザー ID を取得しま

す。このマクロからの応答および CICS 領域のセキュリティー・コード識別機構をカスタマイズするには、MVS セキュリティー・ルーター出口 ICHRTX00 を使用します。

第 3 章 ID 伝搬および分散セキュリティー

ID 伝搬によって提供されるメカニズムにより、どこで ID 情報が作成されたかに関係なく、外部セキュリティー・レルムに由来するユーザー ID を保持できます。これにより、分散環境全体におけるアカウンタビリティーが強化されます。

外部コンピューティング環境 (WebSphere® Application Server など) では、ユーザーの ID はその環境に適用されるユーザー識別を使用して認証されます。WebSphere Application Server のようなアプリケーションでは大抵の場合、CICS システムと通信する際に、個別の共用される外部セキュリティー・マネージャーのユーザー ID が使用されます。ユーザーの元の ID は CICS に渡されないため、外部セキュリティー・マネージャーに渡すことができません。これにより、最初のユーザー ID を判別することが困難となり、要求の監査証跡に影響を及ぼします。

ID 伝搬のトピックでは、RACF は外部セキュリティー・マネージャーを表します。CICS で RACF を使用しない場合は、システム管理者に連絡して、ご使用の外部セキュリティー・マネージャー用に設定を構成してください。

分散 *ID* という用語は、リモート・システムに由来するユーザー ID 情報 (例えば、X.500 識別名や関連する LDAP レルムなど) を表します。あるシステムで分散 ID が作成されると、それがネットワークを介して他の 1 つ以上のシステムに渡されます。分散 ID はすべて CICS 以外の場所に由来するものです。CICS が分散 ID のソースになることはありませんが、分散 ID をさらに伝搬させることはできます。

以下は、ID 伝搬 CICS ライブラリーの使用には適用されません。 Liberty での ID 伝搬マッピングについては、分散 ID マッピングを使用した Liberty JVM サーバーのセキュリティーの構成を参照してください。

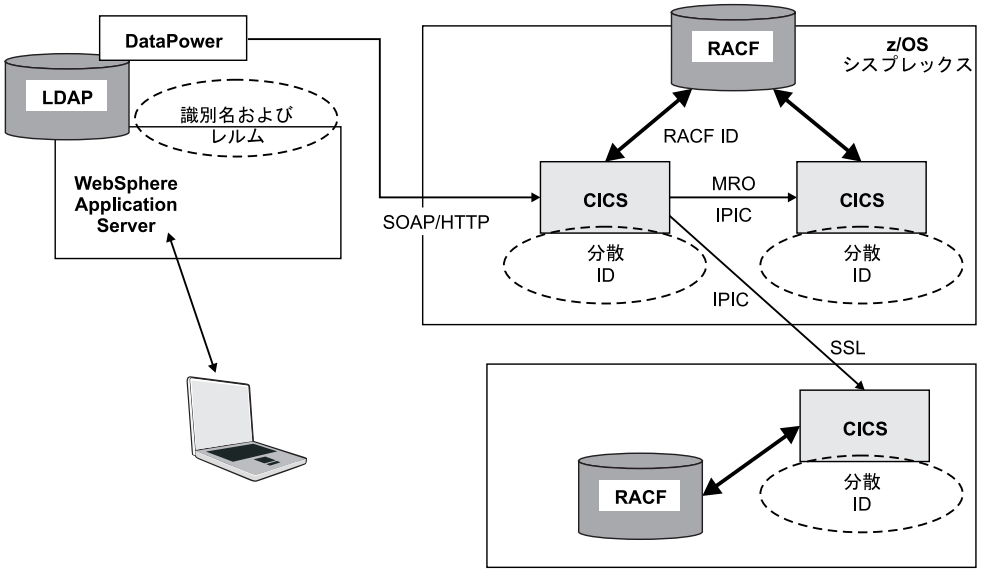
分散 ID は、MRO および IPIC 接続を介してシスプレックスに入ると、接続設定に関係なくシスプレックス内で自動的に伝搬されます。 CICS セキュリティーは、分散 ID をユーザー ID に関連する追加情報として扱います。分散 ID はユーザー ID とセットでなければ存在できません。

シスプレックスの外部では、サポートに応じて分散 ID を伝搬できます。別のものが分散 ID を受け取るかどうかは、それが ID 伝搬に関与しているかどうかによって決まります。 200 ページの『ID 伝搬のサポートおよび要件』を参照してください。

ID 伝搬を使用することにより、どこで ID 情報が作成されたかに関係なく、許可や監査で使用するために分散 ID を保存できます。分散 ID は、外部クライアント・アプリケーションからサーバーに送られる要求と一緒に送信される ID コンテキストの一部であり、許可プロセスの一部としてサーバーのアクセス制御に取り込まれます (例えば、WebSphere Application Server の代わりに CICS Transaction Gateway でその処理を行います)。

© Copyright IBM Corp. 1989, 2018

この図は、外部的にユーザーを識別する X.500 識別名および関連する LDAP レルムが、WebSphere Application Server からの要求で、どのように CICS システムに渡されるかを示しています。識別名およびレルム (ネットワークを介した送信時に分散 ID として CICS で認識される) は、z/OS セキュリティー・コンテキストに伝搬され、RACF ユーザー ID に関連付けられます。z/OS RACF コマンド **RACMAP** を使用すると、マッピング・フィルターを使用して、識別名およびレルムを RACF ユーザー ID へ相互に関連付け、分散資格情報を保持し、ガバナンスおよび監査要件を満すことができます。 RACF は識別名およびレルムに関する情報を CICS に提供します。これにより、CICS 内で最初のユーザーの ID の取得を行うことができます。



202 ページの『ID 伝搬を使用するためのネットワーク・トポロジーの例』 では、この図についてさらに詳しく説明されており、別のネットワークおよび製品との組み合わせで、どのように分散 ID がサポートされるかを示しています。

ID 伝搬のサポートおよび要件

CICS は、さまざまな製品間での ID 伝搬をサポートします。 ID 伝搬への参加を許可するには、以下の要件を満たしていることを確認します。

ID 伝搬の参加の要約

ID 伝搬は宣言 ID の 1 つの形式であり、結果として参加には以下のいくつかの要素が必要になります。

- すべての関係パーティーは、分散 ID を処理できる必要があります。
- トラステッド接続は、すべての外部パーティーに接続する必要があります。

システムが ID 伝搬に参加していない場合、分散 ID 情報は無視され、ユーザー ID 情報が以前と同じように使用されます。

伝搬の場合、アウトバウンド要求は参加タスク (関連付けられた分散 ID がユーザー ID にあるタスク) から実行する必要があります。

ID 伝搬のサポート

CICS は、以下の場合に ID 伝搬のサポートを提供します。

- CICS ECI リソース・アダプターを使用し、IPIC 接続を介する、WebSphere Application Server から CICS へのインバウンド要求。
- Web サービス要求で WS-Security ヘッダー・エレメントを使用する。ルーティングされるインバウンド Web サービス要求は、ID 伝搬をサポートしません。
- CICS システムの間で IPIC 接続および MRO 接続を使用する。分散 ID は、参加タスクから MRO 接続または IPIC 接続に渡された場合にのみ、CICS によって使用されます。
- ローカルまたは機能シップされた **START** コマンドを発行するトランザクション。以下の状態はこのサポートの例外であり、この場合には分散 ID は伝搬されません。
  - **START** コマンドが USERID または TERMID を指定している場合。
  - **START** コマンドが LU61 接続または LU62 接続を介してリモート領域にシップされた場合。
  - 動的にルーティングされた **START** コマンドが遅延した場合。
- Liberty を使用する場合。 Liberty での ID 伝搬マッピングについては、分散 ID マッピングを使用した Liberty JVM サーバーのセキュリティーの構成を参照してください。

ID 伝搬を使用するためのネットワーク・トポロジーの例は、サポートされる要求および接続を使用してユーザー・セキュリティー情報を受け渡す方法を説明する図および例を提供します。

ID 伝搬のためのソフトウェア要件

ID 伝搬に参加している各ホスト・システムには、以下の最小ソフトウェアが必要です。

- z/OS バージョン 1 リリース 11 以降。

ID 伝搬のための RACF 要件

ID 伝搬のためのクライアントおよび CICS 構成定義を更新する前に、RACMAP コマンドおよび SETR RACLIST(IDIDMAP) コマンドの RACF 設定を構成する必要があります。

ID 伝搬のための CICS 要件

分散 ID をフローできるようにするには、CICS にはいくつかの要件があります。

- SEC=YES システム初期設定パラメーターを指定して、セキュリティーを有効にする必要があります。
- 外部セキュリティー・マネージャー (例えば、RACF) は、分散 ID を受け入れるように構成する必要があります。
- すべてのパートナー・システムは、分散 ID を処理できる必要があります。
- CICS は、最大 246 バイトの長さの識別名、および最大 252 バイトの長さのレルム名をサポートします。

- IPIC 接続は、サポートされる ID コンテキスト (ICRX ID トークン) に制限され、その合計サイズは 2000 バイトを超えてはなりません。

ID 伝搬を使用するためのネットワーク・トポロジーの例

ID 伝搬は、CICS への IPIC 接続または Web サービス要求のいずれかを使用するネットワーク・トポロジーでサポートされます。以下のサンプル・トポロジーを使用すると、構成を計画するのに役立ちます。

IPIC 接続および CICS Transaction Gateway を使用するサンプル構成

CICS への IPIC 接続を使用している場合、CICS Transaction Gateway を WebSphere Application Server と CICS ECI リソース・アダプターとの間のインターフェースとして使用できます。 CICS Transaction Gateway およびいずれの通信 CICS システムも同じシスプレックス上にない場合は、SSL も必要です。以下のサンプル・トポロジーは、X.500 識別名および関連する LDAP レルムが要求とともに、IPIC 接続を介して WebSphere Application Server から CICS Transaction Gateway を経て、同じシスプレックス上の CICS システムにどのように受け渡されるかを示しています。次に識別名およびレルム (それらがネットワークを介して送信される場合、CICS では分散 ID として認識される) は、シスプレックス内の MRO または IPIC のいずれかを使用して CICS システム間で、または SSL を介した IPIC を使用して異なるシスプレックスにある CICS システム間で受け渡されます。このシナリオに表示されている複数の CICS システムは、シスプレックスの内外でどのように CICS システムを接続できるかを示しています。ただし、複数の CICS システムが、ID 伝搬を可能にするための構成の必須部分というわけではありません。分散 ID は、z/OS セキュリティー・コンテキスト (アクセサー環境エレメント (ACEE) とも呼ばれる) に伝搬され、RACF **RACMAP** コマンドに指定されているマッピング・ルールを使用して RACF ユーザー ID に関連付けられます。RACF は、識別名およびレルムに関する情報と RACF ユーザー ID を CICS に提供します。これにより、CICS 内で最初のユーザーの ID を取得できます。

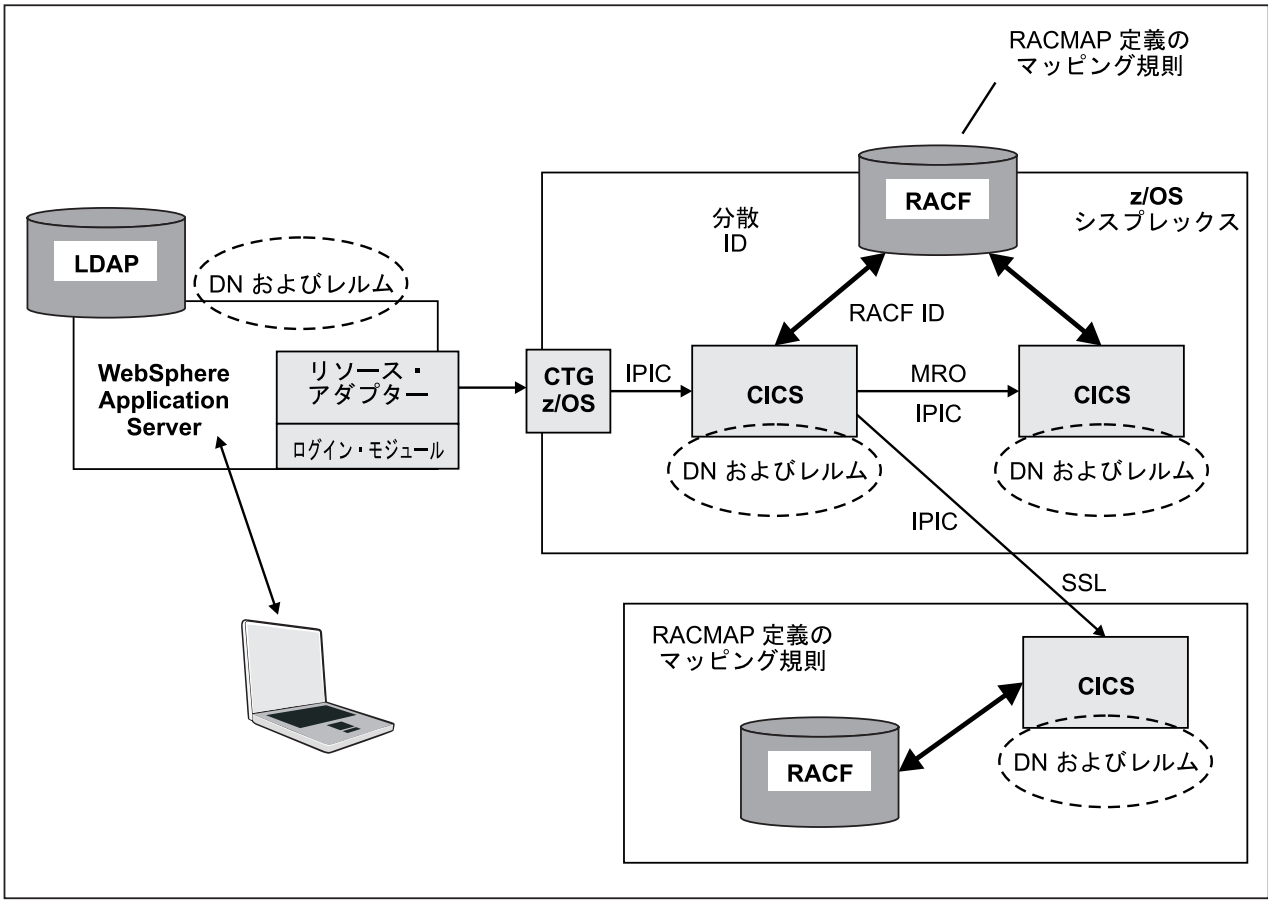


図 3. *IPIC* 接続を介して *CICS* と通信する、*WebSphere Application Server* および *CICS Transaction Gateway* を使用した *ID* 伝搬の例。

CICS Transaction Gateway および ID 伝搬の詳細については、CICS Transaction Gateway for z/OSまたはCICS Transaction Gateway for Multiplatformsを参照してください。

Web サービス要求および WebSphere DataPower® を使用するサンプル構成

Web サービス要求を CICS に送信するときは、WebSphere DataPower を WebSphere Application Server と CICS との間のインターフェースとして使用できます。 WebSphere DataPower アプライアンスを CICS Web サービス WS-Security サポートと一緒に使用して、XML デジタル署名を処理し、事前定義された RACF ユーザー ID へのマッピングを実行できます。

以下のサンプル・トポロジーは、X.500 識別名および関連する LDAP レルムが要求とともに、WebSphere Application Server から WebSphere DataPower を経てどのように受け渡されるかを示しています。識別名とレルムは、Web サービス要求の拡張 ID コンテキスト参照 WS-Security ヘッダー・エレメントに含まれて、CICS システムに送信されます。 ICRX ID トークンについて詳しくは、z/OS Security Server RACF Data Areas を参照してください。次に識別名およびレルム (それらがネットワークを介して送信される場合、CICS では分散 ID として認識される) は、シスプレックス内の MRO または IPIC のいずれかを使用して CICS システム

間で、または SSL を介した IPIC を使用して異なるシスプレックスにある CICS システム間で受け渡されます。 このシナリオに表示されている複数の CICS システムは、シスプレックスの内外でどのように CICS システムを接続できるかを示しています。ただし、複数の CICS システムが、ID 伝搬を可能にするための構成の必須部分というわけではありません。分散 ID は、z/OS セキュリティー・コンテキスト (アクセサー環境エレメント (ACEE) とも呼ばれる) に伝搬され、RACF **RACMAP** コマンドに指定されているマッピング・ルールを使用して RACF ユーザー ID に関連付けられます。 RACF は識別名およびレルムに関する情報と RACF ユーザー ID を CICS に提供します。これにより、CICS 内で最初のユーザーの ID を取得できます。

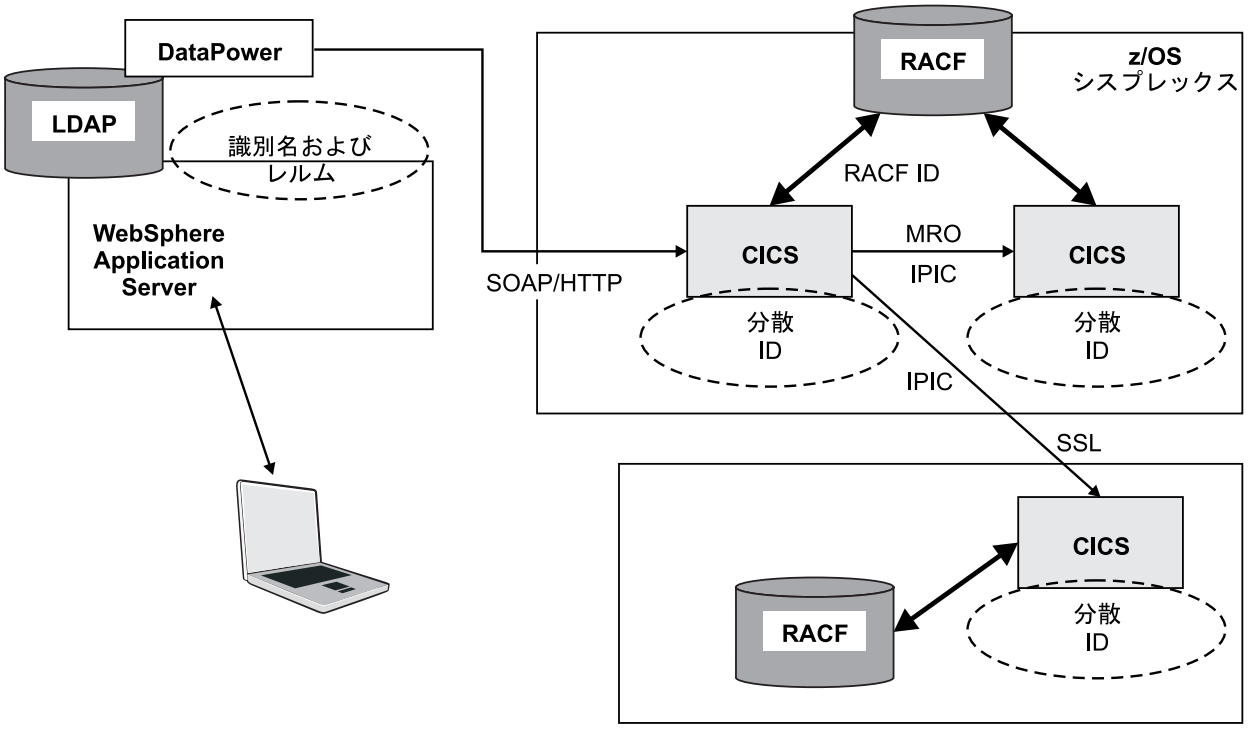


図 *4. Web* サービス要求を使用して *CICS* と通信する、*WebSphere Application Server* および *WebSphere DataPower* を使用した *ID* 伝搬の例。

ID 伝搬の構成

RACF は分散 ID をマップできるように構成する必要があります。 CICS システムは、IPIC 接続または Web サービス要求からの分散 ID 情報を受け入れるように構成できます。 CICS Liberty サーバーは、分散識別マッピングを使用することもできます。

MRO 接続を使用している場合は、CICS システム間の分散 ID 情報のみを受け入れることができます。 ATTACHSEC(IDENTIFY) が指定されていることを確認する必要がありますが、その他の構成変更は必要ありません。

ID 伝搬のための RACF の構成

分散 ID フィルターを作成、削除、およびリストするには、RACF **RACMAP** コマンドを使用します。分散 ID フィルターを含む IDIDMAP リソース・プロファイルをリフレッシュするには、RACF **SETR RACLIST(IDIDMAP)** コマンドを使用します。クライアントおよび CICS 構成定義を更新する前に、ID 伝搬の RACF 設定を構成する必要があります。

始める前に

z/OS システムの RACF プロファイルを更新するために必要な権限を持っていることを確認します。

このタスクについて

RACF では、分散 *ID フィルター* という語を使用して、RACF ユーザー ID と 1 つ以上の分散 ID の間のマッピングの関連付けを記述します。ユーザーが **RACMAP** コマンドを使用してフィルターを定義すると、分散 ID が RACF ユーザー ID に関連付け (マップ) されます。分散 ID が **RACMAP** コマンドでマップされない場合、デフォルトの CICS ユーザー ID は使用されず、セキュリティー・エラーが発行されます。それぞれの分散 ID フィルターは、IDIDMAP クラスの一般リソース・プロファイルに保管されています。

IDIDMAP プロファイルの所有者は、**RACMAP MAP** コマンド発行者のユーザー ID です。

SETR RACLIST(IDIDMAP) コマンドが発行されると、IDIDMAP クラスがアクティブになります。分散ユーザーが認証され、サポートされるトランザクションが z/OS システムに送信されると、RACF はそのユーザーの識別名とレルムを UTF-8 データの文字ストリングとして受け取ります。 RACF には、UTF-8 データ・エンコードの結果として多数の制限があります。例えば、識別名が 246 バイトを超えているか、またはレルムが 252 バイトを超えている場合、**RACMAP MAP** コマンドは失敗します。「z/OS Security Server RACF セキュリティー管理者のガイド」で、UTF-8 データ値の制限に関するトピックを参照してください。

RACF は、識別名とレルムを使用して、IDIDMAP プロファイルで、データに最もよく一致する名前値を含む分散 ID フィルターを検索します。最もよく一致するフィルターが検出されると、RACF は RACF ユーザー ID を割り当てます。

手順

1. RACF の FACILITY クラス内に `IRR.IDIDMAP.*` という総称プロファイルを作成します。
2. このプロファイルへの一般アクセスを可能にするために、UPDATE 権限をユーザー・グループに付与します。 以下に示す例では、ユーザー・グループの名前として `dev` を使用しています。

   ```
   RDEFINE FACILITY IRR.IDIDMAP.** OWNER(cpssing)
   PERMIT IRR.IDIDMAP.** CLASS(FACILITY) ID(dev) ACCESS(UPDATE)
   SETROPTS RACLIST(FACILITY) REFRESH
   ```

3. 特定のユーザー権限を付与して IDIDMAP クラスを管理するには、CLAUTH アクセス権限を付与します。 以下に示す例では、アクセス権限が付与されたユーザーとして、`usera`、`userb`、および `userc` を使用しています。

```
ALTUSER (usera,userb,userc) CLAUTH(IDIDMAP)
```

4. 次のコマンドを発行して、変更内容をアクティブにします。

```
SETROPTS CLASSACT(IDIDMAP) RACLIST(IDIDMAP)
```

5. 実行中のシステムの RACMAP 定義を変更すると、キャッシュ・コピーが、新しい RACMAP 定義と同じユーザー ID にマップされなくなる場合があります。マッピングをリセットするには、キャッシュ・コピーが削除されるまで待機する必要があります。識別名とレルムのペア (分散ユーザー ID) が、USRDELAY SIT パラメーターで指定された時間の長さの間に使用されない場合、コピーは削除されます。 パフォーマンス上の理由から、CICS は識別名とレルムをキャッシュに入れて、識別名とレルムからの最初の要求のみが、RACMAP 定義を使用して ACEE を作成することを RACF に要求するようにします。

タスクの結果

分散ユーザー情報は RACF 内でマップされます。 RACF は、分散 ID を RACF ユーザー ID にマップするように構成されます。

次のタスク

これで IPIC 接続、Web サービス接続、または CICS Liberty サーバーを CICS に対して構成できます。 RACF **RACMAP** コマンドおよび **SETR RACLIST(IDIDMAP)** コマンドの詳細については、「z/OS Security Server RACF セキュリティー管理者のガイド」の分散 ID フィルターに関する情報を参照してください。

第 4 章 外部セキュリティー・マネージャーの呼び出し

CICS には外部セキュリティー・マネージャー (ESM) へのインターフェースが用意されています。ESM は、リソース・アクセス管理機能 (RACF)、ベンダー製品、またはユーザー作成のいずれかでも構いません。CICS セキュリティーは、MVS System Authorization Facility (SAF) インターフェースを使用して、許可要求を外部セキュリティー・マネージャー (ESM) に送信します。

通常は、RACF が存在する場合、MVS ルーターは RACF に制御を渡します。ただし、ルーター出口を呼び出すことにより、MVS ルーターのアクションを変更できます。

ルーター出口を使用して、例えば、ユーザー作成またはベンダー提供の ESM に制御を渡すことができます。独自のセキュリティー・マネージャーを使用する場合は、MVS ルーター出口ルーチンを提供する必要があります。

注: この情報は、主に RACF 以外のユーザーを対象としています。RACF を使用したセキュリティー処理に関する確実な情報については、RACF ファシリティーを参照してください。

MVS ルーターの概要

System Authorization Facility (SAF) は、*MVS*ルーター と呼ばれるシステム・サービスを使用したセキュリティー処理の集中制御をインストール済み環境に実現します。 MVS ルーターは、リソース制御を提供および要求するすべての製品に共通のシステム・インターフェースを提供します。

リソース管理コンポーネントおよびサブシステム (CICS など) は、アクセス管理検査や許可関連検査などの処理における特定の意思決定機能の一部として、MVS ルーターを呼び出します。これらの機能は制御点 と呼ばれます。この単一 SAF インターフェースにより、製品間やシステム間で共用される共通制御機能の使用が促進されます。

システムで RACF が使用可能な場合、MVS ルーターは RACF ルーターに制御を渡し、RACF ルーターで適切な RACF 機能が呼び出されます。パラメーター情報と RACF ルーター・テーブル (ルーター呼び出しを RACF 機能に関連付ける) によって、適切な機能が決まります。ただし RACF ルーターを呼び出す前に、MVS ルーターは、インストール済み環境によって提供されるセキュリティー処理出口 (オプション) がインストールされている場合はそれを呼び出します。詳しくは、ユーザー提供の ESM に制御を渡すを参照してください。

RACF がインストールされていない場合でも、System Authorization Facility (SAF) と SAF ルーターはすべての MVS システムに存在します。SAF ルーターは RACF の一部ではありませんが、CICS など、多くのシステム・コンポーネントとプログラムが RACROUTE マクロおよび SAF を介して RACF を呼び出します。したがってインストール済み環境では、RACF パラメーター・リストを変更し、カスタマイズされたセキュリティー処理を SAF ルーター内で行うことができます。

© Copyright IBM Corp. 1989, 2018

SAF ルーター出口のコーディング方法について詳しくは、「z/OS Security Server RACF メッセージおよびコード」を参照してください。

MVS ルーターについて詳しくは、『z/OS Security Server RACROUTE Macro Reference』の『System Authorization Facility (SAF)』および 『z/OS MVS Installation Exits』の『ICHRTX00 - MVS ルーター出口』を参照してください。

ESM 出口プログラムによる CICS 関連情報へのアクセス方法

CICS は ESM を起動するときに、ESM 出口プログラムが使用できるよう、現在の CICS 環境に関する情報をインストール・データ・パラメーター・リストの形で受け渡します。出口プログラムがインストール・データ・パラメーター・リストにアクセスする方法は、ご使用の ESM が RACF か否かによって異なります。

RACROUTE REQUEST=AUTH を発行する ESM 出口プログラムを使用しており、その出口が CICS からのセキュリティー要求により実行されている場合は、参照されるすべての CLASS が RACLIST にリストされるようにしてください。

RACF ユーザー出口パラメーター・リスト

RACF ユーザー出口を作成する場合は、RACF ユーザー出口パラメーター・リストから直接、インストール・データ・パラメーター・リストのアドレスを見つけることができます。

ユーザー出口パラメーター・リスト内のこの関連フィールドの名前は、RACROUTE REQUEST タイプと、呼び出される RACF ユーザー出口によって異なります。REQUEST タイプ、出口名、フィールド名の関係を表 16 に示します。

表 *16.* インストール・データ・パラメーター・リストのアドレスの取得

RACROUTE REQUEST タイプ	**RACF** 出口	出口リスト・マッピング・マクロ	パラメーター・リストのフィールド名 **(**注 **1** および **2** を参照。**)**
VERIFY	ICHRIX01	ICHRIXP	RIXINSTL
	ICHRIX02	ICHRIXP	RIXINSTL
AUTH	ICHRCX01	ICHRCXP	RCXINSTL
	ICHRCX02	ICHRCXP	RCXINSTL
FASTAUTH	ICHRFX01	ICHRFXP	RFXANSTL
	ICHRFX02	ICHRFXP	RFXANSTL
LIST	ICHRLX01	ICHRLX1P	RLX1INST
	ICHRLX02	ICHRLX2P	RLX2PRPA

注**:**

1. xxxINSTL フィールドがインストール・パラメーター・リストを指すのは、CICS システム初期設定パラメーターに ESMEXITS=INSTLN をコーディングした場合のみです。このパラメーターのデフォルト値は NOINSTLN であり、これはインストール・データが渡されないことを意味します。(ESMEXITS を SIT オーバーライドとしてコーディングできないことに注意してください。)

2. RLX2PRPA には、ICHRLX01 ユーザー出口パラメーター・リスト (RLX1P) のアドレスが格納されます。RLX1P のフィールド RLX1INST によってインストール・データ・パラメーター・リストが指し示されます。
3. REQUEST=EXTRACT の場合は RACF ユーザー出口がないため、インストール・パラメーター・データは渡されません。 MVS ルーター出口 ICHRTX00 を使用して、カスタマイズを行う必要があります。

RACF 出口とその機能の簡単な説明については、z/OS Security Server RACF セキュリティー管理者のガイドを参照してください。 RACF 出口パラメーター・リストの詳細については、z/OS Security Server RACF システム・プログラマーのガイドを参照してください。

CICS RACROUTE 呼び出しでの RACF 出口の使用に関する指針

CICS は、多くの RACROUTE 要求を実行します。特に、RO TCB での SIGNON、CHANGE PASSWORD、CHANGE PHRASE コマンドなど、RACF アクセス制御環境エレメント (ACEE) を作成または削除する要求を多く実行します。要求は、RO TCB で逐次実行されます。したがって、長い経過時間を伴うものを出口で実行するべきではありません。さもないと、他のセキュリティー要求や TCB の使用がブロックされます。

インストール・データ・パラメーター・リスト

インストール・データ・パラメーター・リストにより、外部セキュリティー・マネージャー (ESM) 出口プログラムは、処理中の CICS セキュリティー・イベントと現在の CICS 環境に関する詳細情報にアクセスできるようになります。

現在の CICS 環境に関する以下の詳細が得られます。

- CICS 領域の APPLID
- 共通作業域
- 呼び出されるトランザクション
- 実行中のプログラム
- CICS 端末 ID
- SNA LU 名
- 端末ユーザー域
- 8 バイトの通信域

第 5 章 CICSPlex SM のセキュリティー

このパートでは、CICSPlex SM のセキュリティーを実装する方法を説明します。

CICSPlex SM セキュリティーの実装

RACF 使用の CICSPlex SM セキュリティーを実装するには、各種の CICSPlex SM 機能へのアクセスが必要なユーザーを決める必要があります。また、いくつかのタスクを実行して、CICSPlex SM クラス名とリソース名を定義し、セキュリティーのアクティブ化や RACF プロファイルのリフレッシュも行う必要があります。

このタスクについて

RACF 以外の SAF 準拠の外部セキュリティー・マネージャー (ESM) を使用する場合は、 260 ページの『ユーザー提供の外部セキュリティー・マネージャーの呼び出し』を参照してください。以下の手順では、CICSPlex SM のセキュリティーのセットアップ方法を要約します。各手順は、以降のトピックで詳細に説明があります。

手順

1. CICSPlex SM へのアクセスを必要とするユーザーを決める。
2. CICSPlex SM の一般セキュリティー要件を検討する。
3. CICSPlex SM データ・セットの RACF プロファイルを作成する
4. CICSPlex SM 開始タスクを RACF に定義します。
5. CICS トランザクション・セキュリティーが CMAS でアクティブである場合は、 RACF への CICSPlex SM トランザクションを定義する。
6. CICS トランザクション・セキュリティーが MAS でアクティブである場合は、RACF への CICSPlex SM トランザクションを定義する。
7. CICSPlex SM ビューの RACF プロファイルを作成する
8. CICSPlex SM Web ユーザー・インターフェース・リソースの RACF プロファイルを作成する 詳細については、『Web ユーザー・インターフェース・リソースへのアクセス制御』を参照してください。
9. オプション: CICSSYS ビュー、CPLEXDEF ビュー、または MAS ビューを使用して、シミュレートしたセキュリティー検査をアクティブにする。
10. CICSPlex SM および CICS セキュリティー関連のシステム初期設定パラメーターを使用して、CMAS および MAS のセキュリティーをアクティブ化する。

© Copyright IBM Corp. 1989, 2018

だれが CICSPlex SM リソースへのアクセスを必要とするかの判別

だれが CICSPlex SM リソースへのアクセスを必要とするかを判別するため、質問に回答し、マトリックスを完成させてください。次にその結果を使用することにより、RACF でリソースへのアクセスを制御するために必要な PERMIT ステートメントを作成することができます。

このタスクについて

CICSPlex SM リソースへのアクセスは、次の 2 とおりの方法で制御できます。

- CICSPlex SM ビューによって管理されるオブジェクトにアクセスを制限することにより。この制限は、ビュー自体へのアクセスには影響しませんが、データが表示されないようにします。
- Web ユーザー・インターフェースのビュー・セット、メニュー、およびビュー・エディターへのアクセスを制限することにより。この制限は管理対象のオブジェクトへのアクセスには影響しませんが、ビュー・セット、メニュー、およびビュー・エディター自体へのアクセスができないようにします。

手順

1. だれが CICSPlex SM リソースへのアクセスを必要とするかを判別するため、以下の質問に回答してください。

   CICSPlex SM を使用するユーザー・グループはどれですか**?**

   企業において、RACF に対していくつかのユーザー・グループが既に定義されていることでしょう。 CICSPlex SM へのアクセスを必要とする典型的なグループには、システム・プログラミング、操作、ヘルプ・デスク、アプリケーション・プログラミング、およびパフォーマンス・モニターが含まれます。それらのグループは、セキュリティー・マトリックスの列見出しとして使用されます。それらに対応する RACF グループ ID を提供することができます。 (必要なら、企業の状況に合わせてマトリックスのグループを無視したり置換したり追加したりできます。)

   各グループで、どの **CICSPlex SM** ビューを使用する必要がありますか**?**

   CICSPlex SM では、ビューを使用することにより CICS リソースを管理します。ビューは、構成、トポロジー、ワークロード管理、リアルタイム分析、操作、モニタリング、ビジネス・アプリケーション・サービス、および CICSplex 管理というように機能別に分類されます。ビュー・グループによっては、すべてのユーザーには当てはまらないものがあります。一部のユーザー・グループでは、ビューのうちのあるサブセットのみ使用します。例えば、システム・プログラミング・グループではすべてのビューの作業をすることが必要であるのに対して、ヘルプ・デスク・グループで使用することが必要なのは 1 つか 2 つのみです。ビュー・グループのリストは、マトリックスの左側に縦方向に示され、その CICSPlex SM リソース名の高位修飾子も示されます。

   各 **RACF** グループでどのタイプのアクセスが必要ですか**?**

   だれが何を使用することが必要かを決定した後、すべてのビューによって管理されるすべてのオブジェクトへのユニバーサルなアクセスを停止します。次に、特定のビュー・グループに対する読み取り、更新、また

は変更アクセス権限を選択的に許可することができます。マトリックスを完成するには、一群のビューへのアクセスを必要とする各 RACF グループに対して、READ、UPDATE、または ALTER のアクセス権限を指定します。

- ユーザーがリソースに対する照会を実行することを許可するには、READ アクセスを指定します。
- ユーザーが、SET または UPDATE コマンドを使用して、あるいはアクションを実行することにより値を変更することを許可するには、UPDATE アクセス権限を指定します。またユーザーは、BAS リソース・オブジェクトなどの定義を作成したり削除したりすることもできます。
- インストールされているリソースをユーザーが CICS から破棄したり、ユーザーが BAS リソース・オブジェクトをインストールしたりすることを許可するには、ALTER アクセス権限を指定します。

ヒント: アプリケーション・プログラマーの場合、だれが CICS 変換プログラムで CPSM オプションの使用を許可されるかを制御する必要がある場合、RACF を使用することにより、変換時に DFHSMTAB 表をロードすることを許可されるのがだれかを制御することができます。RACF プログラム制御について詳しくは、z/OS Security Server RACF セキュリティー管理者のガイドを参照してください。 DFHSMTAB は、CICSPlex SM API のコマンドを定義する言語定義表です。これはオンデマンドでのみロードされます。

各グループでは、**CICSPlex SM Web** ユーザー・インターフェースのどのビュー、メニューにアクセスする必要がありますか**?**

Web ユーザー・インターフェースのビューとメニューは、通常はユーザー定義ですが、Web ユーザー・インターフェースと同じように、多くの場合、ビューは機能別に分類されます。ビュー・セットとメニューによっては、すべてのユーザーに適するわけではない場合があります。特定のユーザー・グループでは、ビューのサブセットへのアクセスが必要です。例えば、システム・プログラミング・グループではすべてのビューとビュー・エディターへのアクセスが必要かもしれませんが、ヘルプ・デスク・グループではビュー・エディターや CICSPlex SM リソースの定義を管理するビューを使用する必要はないかもしれません。

2. 質問に回答したらセキュリティー・マトリックスに情報を記入してください。

表 *17.* セキュリティー・マトリックス

RACF グループ → **CICSPlex SM** ビュー・グループ ↓	システム プログラミング **ID( )**	操作 **ID( )**	ヘルプ・ デスク **ID( )**	アプリケーション プログラミング **ID( )**	パフォーマンス **ID( )**
構成 CONFIG					
トポロジー TOPOLOGY					
ワークロード管理 WORKLOAD					

表 *17.* セキュリティー・マトリックス (続き)

RACF グループ → **CICSPlex SM** ビュー・グループ ↓	システム プログラミング **ID( )**	操作 **ID( )**	ヘルプ・ デスク **ID( )**	アプリケーション プログラミング **ID( )**	パフォーマンス **ID( )**
リアルタイム分析 ANALYSIS					
操作 OPERATE					
モニター MONITOR					
ビジネス・アプリケーション・サービス BAS					

表 18 は、実動 CICSplex のための完成したセキュリティー・マトリックスのサンプルです。

表 *18.* サンプル・セキュリティー・マトリックス

RACF グループ → **CICSPlex SM** ビュー・グループ ↓	システム プログラミング **ID(SYSPGRP)**	操作 **ID(OPSGRP)**	ヘルプ・ デスク **ID(HELPGRP)**	アプリケーション プログラミング **ID(APPLGRP)**	パフォーマンス **ID(PERFGRP)**
構成 CONFIG	UPDATE				
トポロジー TOPOLOGY	UPDATE	UPDATE	READ		
ワークロード管理 WORKLOAD	UPDATE			READ	
リアルタイム分析 ANALYSIS	UPDATE	UPDATE	READ		READ
操作 OPERATE	ALTER	UPDATE	READ	READ	READ
モニター MONITOR	UPDATE	READ			READ
ビジネス・アプリケーション・サービス BAS	ALTER	ALTER		UPDATE	

3. CPSMOBJ クラスがアクティブであること、そして総称プロファイルを定義可能であることを確認します。

```
SETROPTS CLASSACT(CPSMOBJ)
SETROPTS GENERIC(CPSMOBJ)
SETROPTS GENCMD(CPSMOBJ)
```

4. CICSPlex SM のすべての機能に対するビューとアクション・コマンドのすべてを保護するための RACF プロファイルを作成します。

```
RDEF CPSMOBJ ** UACC(NONE) OWNER(admin_group) NOTIFY(admin_user)
```

CPSMOBJ は、CICSPlex SM メンバー・クラスです。二重アスタリスクは、この RDEF ステートメントに CICSPlex SM ビューのすべてが含まれていることを示しています。

5. サンプル・マトリックスの情報を使用することにより、特定のビュー・グループへのアクセスを許可することができます。 例えば、システム・プログラミング・グループでは、全ビュー・グループへの更新アクセスと BAS ビューへの ALTER アクセスが必要です。それは、以下の 3 つの PERMIT ステートメントのみにより定義できます。

```
PERMIT ** CLASS(CPSMOBJ)  ID(SYSPGRP) ACCESS(UPDATE)
PERMIT BAS.** CLASS(CPSMOBJ) ID(SYSPGRP) ACCESS(ALTER)
```

二重アスタリスクは、この PERMIT ステートメントが CICSPlex SM ビューのすべてに影響することを示しています。

以下の PERMIT ステートメントは、操作およびヘルプ・デスクのグループのためにトポロジー・ビューのすべてに対する適切なアクセスを付与します。

```
PERMIT TOPOLOGY.** CLASS(CPSMOBJ) ID(OPSGRP)  ACCESS(UPDATE)
             PERMIT TOPOLOGY.** CLASS(CPSMOBJ) ID(HELPGRP) ACCESS(READ)
```

ワークロード管理ビューの場合、

```
PERMIT WORKLOAD.** CLASS(CPSMOBJ) ID(APPLGRP) ACCESS(READ)
```

リアルタイム分析ビューの場合、

```
PERMIT ANALYSIS.** CLASS(CPSMOBJ) ID(OPSGRP)  ACCESS(UPDATE)
PERMIT ANALYSIS.** CLASS(CPSMOBJ) ID(HELPGRP) ACCESS(READ)
PERMIT ANALYSIS.** CLASS(CPSMOBJ) ID(PERFGRP) ACCESS(READ)
```

操作ビューの場合、

```
PERMIT OPERATE.** CLASS(CPSMOBJ) ID(OPSGRP)  ACCESS(UPDATE)
PERMIT OPERATE.** CLASS(CPSMOBJ) ID(HELPGRP) ACCESS(READ)
PERMIT OPERATE.** CLASS(CPSMOBJ) ID(APPLGRP) ACCESS(READ)
PERMIT OPERATE.** CLASS(CPSMOBJ) ID(PERFGRP) ACCESS(READ)
```

モニター・ビューの場合、

```
PERMIT MONITOR.** CLASS(CPSMOBJ) ID(APPLGRP) ACCESS(READ)
PERMIT MONITOR.** CLASS(CPSMOBJ) ID(PERFGRP) ACCESS(READ)
```

ビジネス・アプリケーション・サービス・ビューの場合、

```
PERMIT BAS.** CLASS(CPSMOBJ) ID(OPSGRP) ACCESS(ALTER)
PERMIT BAS.** CLASS(CPSMOBJ) ID(APPLGRP) ACCESS(UPDATE)
```

タスクの結果

簡単のため、これらの PERMIT ステートメントでは、リソース名に二重アスタリスクを使用することにより、幅広いビュー・グループへのアクセスを付与しています。しかし、必要なら、PERMIT ステートメントにさらに具体性の高いリソース名を使用することができます。詳しくは、 227 ページの『プロファイルでの CICSPlex SM リソース名の指定』を参照してください。

次のタスク

独自のセキュリティー・マトリックスを完成させ、それとこのセクションの残りの部分の情報を使用することにより、自分の企業に必要なだけのプロファイルを作成

することができます。 262 ページの『サンプル・タスク: セキュリティー』 に、詳細なプロファイルのサンプルが示されています。

CICSPlex SM セキュリティーに関する一般要件

ご使用のシステムが最小要件を満たしていることを確認するために、RACF 構成を見直します。

- CICSPlex SM を使用することが予期されるすべてのユーザーの ID は、CMAS がある各 MVS システム内で RACF に対して定義する必要があります。個別の各ユーザーに対して、各 MVS システムの ID は同じでなければなりません。
- CICSPlex SM 定義および CICS コマンドとリソースに対するユーザーのアクセス権限は、CICSPlex SM が使用するすべての MVS システム内で、一貫した方法で RACF に対して定義する必要があります。

さらに、CMAS アドレス・スペースでは、MAS と関連付けられた **DFLTUSER** システム初期設定パラメーターで指定されたユーザーに対して、セキュリティー環境が作成されます。

CICSPlex SM データ・セットのプロファイルの作成

RACF データ・セット保護を使用して、CICSPlex SM データ・セットへのアクセスを制限する必要があります。

手順

1. UACC(NONE) を指定して汎用アクセスを禁止します。
2. 以下のそれぞれに割り当てられている RACF USERID に対して、データ・セットへの最小限のアクセス権限を必ず許可します。
   - すべての CMAS ジョブまたは開始タスク。
   - すべての MAS。
   - CICSPlex SM WUI および API から CICSPlex SM を使用することが許可されているすべての個人 (システム管理者とユーザーの両方)。

   表 19 に、CICSPlex SM データ・セット、および各タイプのユーザー ID に付与する必要のある最小のアクセス権限がリストされています。

表 *19.* ユーザー *ID* による *CICSPlex SM* データ・セットに対するアクセス

Data set name (データ・セット名**)**	**CMAS**	**MAS**	システム管理者	個々のユーザー
SYS1.CICSTS55.CPSM.SEYULPA	NONE	READ	UPDATE	NONE
SYS1.CICSTS55.CPSM.SEYULINK	READ	NONE	UPDATE	NONE
CICSTS55.CPSM.SEYUAUTH	READ	READ	UPDATE	READ
CICSTS55.CPSM.SEYULOAD	READ	READ	UPDATE	NONE
CICSTS55.CPSM.SEYUPARM	READ	READ	UPDATE	NONE
CICSTS55.CPSM.SEYUCMOD	NONE	NONE	UPDATE	NONE
CICSTS55.CPSM.SEYUCOB	NONE	NONE	UPDATE	READ
CICSTS55.CPSM.SEYUC370	NONE	NONE	UPDATE	READ
CICSTS55.CPSM.SEYUDEF	READ	READ	UPDATE	READ
CICSTS55.CPSM.SEYUCLIB	NONE	NONE	UPDATE	READ
CICSTS55.CPSM.SEYUMLIB	NONE	NONE	UPDATE	READ
CICSTS55.CPSM.SEYUPLIB	NONE	NONE	UPDATE	READ
CICSTS55.CPSM.SEYUTLIB	NONE	NONE	UPDATE	READ

表 19. ユーザー ID による CICSPlex SM データ・セットに対するアクセス (続き)

Data set name (データ・セット名)	CMAS	MAS	システム管理者	個々のユーザー
CICSTS55.CICS.SDFHINST	NONE	NONE	UPDATE	NONE
CICSTS55.CPSM.SEYUMAC	NONE	NONE	UPDATE	READ
CICSTS55.CPSM.SEYUOS2	NONE	NONE	UPDATE	NONE
CICSTS55.CPSM.SEYUPL1	NONE	NONE	UPDATE	READ
CICSTS55.CPSM.SEYUPROC	NONE	NONE	UPDATE	READ
CICSTS55.CPSM.SEYUSAMP	NONE	NONE	UPDATE	READ
CICSTS55.CPSM.EYUSDEF	NONE	NONE	UPDATE	UPDATE
CICSTS55.CPSM.EYUDREP	UPDATE	NONE	UPDATE	NONE
CICSTS55.CPSM.EYUIPRM	NONE	NONE	UPDATE	NONE

次のタスク

RACF データ・セット保護の詳細情報が必要な場合は、「z/OS Security Server RACF セキュリティー管理者のガイド」を参照してください。

MAS エージェント・ユーザー ID の判別

RACF を使用して CICSPlex SM セキュリティーを実装する場合は、MAS で使用される CICSPlex SM トランザクションの許可や CICS 代理セキュリティー検査などの一部の状況で、MAS エージェント・ユーザー ID がセキュリティー検査に関係することに注意してください。また、それに応じて RACF 定義を計画する必要があります。次の手順に従って、MAS エージェント・ユーザー ID を判別してください。

このタスクについて

MAS エージェント・ユーザー ID は、以下の要因によって決まります。

- CICS のリリース。ただし、CICSPlex SM のリリースとは関係ありません。
- MAS エージェントが初期化された方法
- システム初期設定テーブルで PLTPIUSR が指定されているかどうか (CICS 領域の初期設定時に MAS エージェントが初期化された場合)

手順

次のようにして MAS エージェント・ユーザー ID を判別します。

表 20. MAS エージェント・ユーザー ID

CICSPlex SM リリース	CICS リリース	MAS エージェントが初期化された方法	システム初期設定テーブルに PLTPIUSR があるか?	MAS エージェント・ユーザー ID
提供されているすべてのリリース	V5.4 以降	CICS 領域の初期設定	いいえ/はい	CICS 領域ユーザー ID
提供されているすべてのリリース	V5.4 以降	COLM トランザクションの使用	適用外	CICS 領域ユーザー ID

表 20. MAS エージェント・ユーザー ID (続き)

CICSPlex SM リリース	CICS リリース	MAS エージェントが初期化された方法	システム初期設定テーブルに PLTPIUSR があるか?	MAS エージェント・ユーザー ID
提供されているすべてのリリース	V5.3 以前	CICS 領域の初期設定	いいえ	CICS 領域ユーザー ID
提供されているすべてのリリース	V5.3 以前	CICS 領域の初期設定	はい	PLTPIUSR
提供されているすべてのリリース	V5.3 以前	COLM トランザクションの使用	適用外	COLM を開始したサインオン・ユーザー ID

CICSPlex SM 開始タスクの定義

開始タスクとして CMAS を実行する場合は、適切なプロシージャー名を、適切に許可された USERID に関連付ける必要があります。

このタスクについて

これは通常、STARTED 一般リソース・クラスまたは RACF ICHRIN03 テーブルを使用して実施します。関連付ける USERID の名前は、プロシージャーの名前と一致している必要はありません。各 USERID には、カタログ式プロシージャー内で参照されるすべてのデータ・セットへの適切なアクセス・レベルが必要です。

STARTED クラスについての追加情報は、z/OS Security Server RACF セキュリティー管理者のガイドを参照してください。 ICHRIN03 について詳しくは、z/OS Security Server RACF システム・プログラマーのガイドを参照してください。

注: 割り当てる USERID およびグループ名が RACF に対して定義されていない場合、開始タスクは、未定義ユーザーの制限された権限のみで実行されます。この場合、アドレス・スペースが保護リソースにアクセスできるのは、リソースの汎用アクセス権限 (UACC) が、要求された操作を許可するのに十分である場合のみです。

CMAS における CICSPlex SM トランザクションの定義

トランザクション接続セキュリティーが CMAS でアクティブである (すなわち、システム初期設定パラメーターで SEC=YES および XTRAN=YES|*classname* が指定されている) 場合は、RACF に対して、CMAS で実行する CICSPlex SM トランザクションを定義する必要があります。

手順

1. 以下の CICSPlex SM トランザクションを RACF に対して定義する必要があります。

BMLT	LPLT	PRLT	WMWC
LCPP	LPRT	PRPR	WMWD
LCMU	LPSC	PSLT	WMWT
LECI	LPSM	TICT	WSCL
LECR	LRLT	TIRT	WSLW
LECS	LSGT	TIST	XDBM
LEEI	LSRT	TSMH	XDNC
LEER	LWTM	TSPD	XDND
LEMI	MCCM	TSSC	XDNE
LEMS	MCTK	TSSJ	XDNR
LENS	MMEI	WMCC	XDNS
LMIR	MMIS	WMGR	XDSR
LNCI	MMST	WMLA	XLEV
LNCS	PEAD	WMQB	XLNX
LNMI	PELT	WMQM	XLST
LNMS	PMLT	WMQS	XMLT
LPDG	PNLT	WMSC	XQST
LPLK	PPLT		XZLT

上記のトランザクションのリストは、SEYUSAMP サンプル・ライブラリーの EYU$CDEF メンバーにも収められています。

a. これらのトランザクションのセキュリティーを以下のように定義します。
   - CMAS が PLT プログラムによって開始され、PLTPIUSR システム初期設定パラメーターにユーザー ID が指定されている場合は、PLTPIUSR ユーザー ID にこれらのトランザクションを接続する権限があることを確認します。
   - CMAS が PLT プログラムによって開始され、PLTPIUSR システム初期設定パラメーターにユーザー ID が指定されていない場合は、CICS 領域ユーザー ID にこれらのトランザクションを接続する権限があることを確認します。
   - CMAS が SIT パラメーター CPSMCONN の使用によって開始され、PLTPIUSR システム初期設定パラメーターにユーザー ID が指定されている場合は、PLTPIUSR ユーザー ID にこれらのトランザクションを接続する権限があることを確認します。
   - CMAS が SIT パラメーター CPSMCONN の使用によって開始され、PLTPIUSR システム初期設定パラメーターにユーザー ID が指定されていない場合は、CICS 領域ユーザー ID にこれらのトランザクションを接続する権限があることを確認します。

b. いずれかの CMTCMDEF または CMTPMDEF に指定されたセキュリティー属性に応じて、接続された CMAS からフローする可能性があるユーザー ID に、これらのトランザクションを接続する権限があることを確認します。

2. CMAS として実行されている CICS リリースに関係なく、トランザクション・セキュリティーがアクティブである場合、RACF に対して以下のデバッグ・トランザクションを定義します。
   - CODB
   - COD0

- COD1
- COD2
- COLU

これらのトランザクションは端末に関連付けられており、IBM サポート担当員の手引きにより、デバッグ目的で提供されます。これらのトランザクションを開始する権限は、CICSPlex SM の問題を解決するために IBM との連携作業に関わる可能性があるユーザーのみに制限する必要があります。

3. CESD シャットダウン支援トランザクションへのアクセス権限をユーザーに与えます。 CICSPlex SM トランザクションを接続したり、デバッグ・トランザクションを定義したりできるユーザーは、CMAS 障害が発生した場合に、CESD にアクセスする必要があります。
4. COSD トランザクションにアクセスするために CMAS をシャットダウンすることが必要になる可能性があるユーザーのみを許可します。 COSD トランザクションは、端末ユーザーによる CMAS のシャットダウンを許可します。

管理対象 CICS 領域でのトランザクションの定義

外部セキュリティー・マネージャーで実行できる CICS 領域の場合は、ESM に対して CICSPlex SM によって管理される CICS 領域で実行されるトランザクションを定義することが必要になる場合があります。

このタスクについて

トランザクション接続セキュリティーが CICS 領域でアクティブである場合 (つまり、SEC=YES および XTRAN=YES|*classname* がシステム初期設定パラメーターに指定されている場合)、適切なクラスで RACF に対して以下のトランザクションを定義する必要があります。

COHT
COIE
COIR
COI0
CONA
COND
CONH
CONL
CONM
COWC

領域ユーザー ID には、そのトランザクションに対する READ 権限が付与されている必要があります。これらのトランザクションは、CICS 提供トランザクションのカテゴリー 1 内にあり、必要なアクセスはこのカテゴリー内のすべてのトランザクションと一緒に定義できます。 CICS 提供トランザクションのカテゴリー 1 の詳細については、Security for CICS-supplied transactionsを参照してください。

CICSPlex SM 状況プローブ (STATDEF) が MAS で実行される場合、領域ユーザー ID には、トランザクション CORT に対する READ アクセス権限を付与する必要があります。

デバッグ・トランザクションを呼び出す可能性があるユーザーには、以下のトランザクションに対する READ アクセス権限が付与されている必要があります。

CODB
COD0
COD1
COD2
COLU

CMAS へのリンクに定義される CONNECTION/SESSION ペアのセキュリティー属性は、これらのトランザクションを実行することが許可されるユーザーを定義します。相互通信セキュリティーの詳細については、271 ページの『相互通信セキュリティーの概要』 を参照してください。

COSH トランザクションにより、端末ユーザーはエージェント・コードの実行を停止できます。このトランザクションへのアクセスは、この方法でエージェント・コードを停止する必要があるユーザーのみに制限してください。

CICSPlex SM Web ユーザー・インターフェース・セキュリティーのセットアップ

CICS セキュリティー、Secure Sockets Layer (SSL) サポート、および MVS データ・セットへのアクセスに関する Web ユーザー・インターフェースのセキュリティー要件を設定できます。

ユーザーのセキュリティー・アクセス権の要約

表 21 に、Web ユーザー・インターフェースのユーザーに必要なセキュリティー・アクセス権を要約します。

表 *21. Web* ユーザー・インターフェースのユーザーに必要なセキュリティー・アクセス権

ユーザーの役割	**CICS Web** サポート	管理者	ユーザー	ビュー・エディター
トランザクション	COVP COVE COVU	COVG COVC	COVA	COVA
CICS 代理ユーザー・セキュリティー		はい		
ビュー・エディター・プロファイル				はい
CICSPlex SM および CICS セキュリティー			個々のユーザーに応じたアクセス権	個々のユーザーに応じたアクセス権

Web ユーザー・インターフェース・サーバー領域における CICS セキュリティー

Web ユーザー・インターフェース・サーバー領域が CICS セキュリティーをアクティブにして稼働している場合は、CICS Web サポートのために、管理者、ビュー・エディターのユーザーにより求められるセキュリティー・アクセス権を定義する必要があります。

CICS トランザクション・セキュリティーを使用すると、COVC トランザクションを使用して Web ユーザー・インターフェース・サーバーを制御することを許可されるユーザーを制限できます。

CICS Web インターフェースのセキュリティー・アクセス権

CICS トランザクション・セキュリティー機能が使用されている場合、CICS DFLTUSER に COVP、COVU、および COVE トランザクションへのアクセス権を与える必要があります。

管理者のセキュリティー・アクセス権

Web ユーザー・インターフェースを始動するユーザー ID (PLTPI を介して自動始動した場合は、COVC または PLTPIUSR の端末ユーザー) は、COVC および COVG トランザクションへのアクセス権を持っている必要があります。 CICS 代理ユーザー・セキュリティー検査が Web ユーザー・インターフェース・サーバー領域でアクティブになっている場合は、その Web ユーザー・インターフェースを始動したユーザー ID (PLTPI を介して自動始動した場合は、COVC または PLTPIUSR の端末ユーザー) が、すべての Web ユーザー・インターフェース・ユーザーを対象にした SURROGAT クラスの wui-userid.DFHSTART への READ アクセス権を持っている必要があります。

ビュー・エディターのユーザーのセキュリティー・アクセス権

Web ユーザー・インターフェースのユーザーは、COVA トランザクションおよび CICSPlex SM へのアクセス権が必要です。ビュー・エディターのユーザーは、COVA トランザクション、CICSPlex SM およびビュー・エディター・プロファイルへのアクセス権が必要です。

Web ユーザー・インターフェースに正常にサインオンするすべてのユーザーは、Web ユーザー・インターフェースによってカスタマイズ可能ビューとメニュー・ヘルプのサービスが提供されている場合、それらすべてのカスタマイズ可能ビュー・ページとメニュー・ヘルプ・ページへのアクセス権を持っています。

Secure Sockets Layer のサポート

接続の暗号化を実現する Secure Sockets Layer (SSL) サポートを使用してセキュア接続を提供できます。 また、SSL サポートのために指定する必要がある Web ユーザー・インターフェース・サーバーの初期設定パラメーター **TCPIPSSL** および **TCPIPSSLCERT** については、Web ユーザー・インターフェース・サーバーの初期設定パラメーターを参照してください。SSL について詳しくは、Configuring CICS to use SSLを参照してください。

Web ユーザー・インターフェースの SSL サポートでは、サーバー認証のみを使用します。 ユーザー認証は、外部セキュリティー・マネージャー (ESM) のユーザー ID とパスワードによって行われます。

WUI の CICSPlex SM トランザクションの定義

RACF などの外部セキュリティー・マネージャー (ESM) で実行できる Web ユーザー・インターフェース (WUI) 領域の場合は、その領域で実行される CICSPlex SM トランザクションを ESM に対して定義する必要があります。

始める前に

これは、SEC=YES および XTRAN=YES システム初期設定パラメーターが指定されている WUI 領域で、トランザクション接続セキュリティーがアクティブである場合に必要です。

手順

1. MAS 領域用のものと同じ定義を作成します。 220 ページの『管理対象 CICS 領域でのトランザクションの定義』を参照してください。
2. 以下のユーザー ID の COVG トランザクションおよび COVCトランザクションに対する READ 権限を定義します。
   - 領域ユーザー ID
   - 領域の **PLTPIUSR** システム初期設定パラメーターに指定されているすべてのユーザー ID
   - すべての WUI システム管理者
3. WUI のデフォルト・ユーザー ID 用の COVE、COVP、および COVU の各トランザクションに対する READ 権限を定義します。
4. すべての WUI ユーザー用の COVA トランザクションに対する READ 権限を定義します。

例

上記のトランザクションのリストは、CSD グループ EYU$WDEF にも収められています。

MVS データ・セットへのアクセスを許可する

標準の CICS および CICSPlex SM の要件に加えて、CICS 領域のユーザー ID には、表に記載されている DD 名に関連付けられたデータ・セットにアクセスする権限が必要です。

表 *22. MVS* データ・セットに対して必要なセキュリティー・アクセス権

DDnames	必要なアクセス権限
EYUWUI	READ
DFHHTML	READ
EYUCOVI (および複製)	READ
EYUWREP	UPDATE
EYULOG	UPDATE
EYUCOVE (および複製)	UPDATE

セキュリティーをアクティブにしての Web ユーザー・インターフェース・サーバーの実行

Web ユーザー・インターフェース・サーバーが、CICS システム初期設定パラメーター **SEC** を YES に設定して実行している場合、ユーザーに対して、誰が Web ユーザー・インターフェースにアクセスできるか、どのリソースが表示されるか、どのアクションが実行可能か、およびビュー・エディターの使用を制御できます。

CICSPlex SM セキュリティーを CICSPlex SM API で使用するように既にセットアップした場合、ユーザーは、Web ユーザー・インターフェースを使用するときには API を使用するときと同じアクセス・レベルを持つことになります。

Web ユーザー・インターフェース・サーバーへの接続を試行すると、 CICSPlex SM Web ユーザー・インターフェースの「Signon (サインオン)」パネルが表示されます。 このパネルに入力されるユーザー ID およびパスワードは、 SSL サポートを使用している場合を除き、TCP/IP 接続を経由してプレーン・テキストで Web ユーザー・インターフェース・サーバーに渡されてから、外部セキュリティー・マネージャーによって検査されます。 外部セキュリティー・マネージャーが大/小文字混合のパスワードをサポートする場合、この機能がアクティブであれば、サインオンのときにパスワード・フィールドの隣にアイコンが表示されます。

Web ユーザー・インターフェースに正常にサインオンするすべてのユーザーは、Web ユーザー・インターフェースによってカスタマイズ可能ビューとメニュー・ヘルプのサービスが提供されている場合、それらすべてのカスタマイズ可能ビュー・ページとメニュー・ヘルプ・ページへのアクセス権を持っています。

セキュリティーがアクティブの場合、検査システムのプログラミング・インターフェース・コマンドによって作成されるメッセージには、WUI にログオンするのに使用するユーザー ID が含まれます。 SPI commands that can be auditedを参照してください。

Web ユーザー・インターフェース・サーバーへのサインオンを許可されるユーザーを制御するために、 Web ユーザー・インターフェースの CICS アプリケーション ID を、RACF APPL 検査を使用して保護できます。 システム・リソースを保護するための RACF クラスを参照してください。

管理対象リソースへのアクセスには、CPSMOBJ クラス内のプロファイルを使用する、標準の CICSPlex SM セキュリティーが使用されます。 例えば、CICS 領域ビューを表示するには、CPSMOBJ クラス・プロファイル OPERATE.REGION.context.scope を介して、Web ユーザー・インターフェースのユーザーに READ 権限が必要です。

CICS リソースへのアクセス、およびビュー内でのリソースに対するアクションは、CICSPlex SM のシミュレートされた CICS セキュリティー検査を使用します。これは通常のCICS RACF リソースおよびコマンド・セキュリティー・プロファイルを使用します。 例えば、CICS 領域に対してシャットダウン・アクションを発行するためには、コマンド・セキュリティーがターゲット CICS 領域でアクティブな場合、 Web ユーザー・インターフェースのユーザーは CCICSCMD クラスでの SHUTDOWN コマンドに対する UPDATE 権限を必要とします。

Web ユーザー・インターフェース・リソースへのアクセス制御

外部セキュリティー・マネージャーを使用して、ビュー、メニュー、エディター、ヘルプ情報へのユーザー・アクセスを制御できます。また COVC を使用して、ビュー、メニュー、マップ、ユーザー・グループ、およびユーザー・オブジェクトのインポートとエクスポートを制御できます。

ナビゲーション・フレームは、セキュリティー検査を受けません。 ユーザー・アクセスを制御するには、適切なプロファイルを FACILITY クラスに作成する必要があります。 以下の ESM FACILITY プロファイルを使用できます。

EYUWUI.wui_server_applid.VIEW.viewsetname
- ビュー・セットの保護に使用されます。

EYUWUI.wui_server_applid.MENU.menuname
- メニューの保護に使用されます。

EYUWUI.wui_server_applid.MAP.mapname
- マップの保護に使用されます。

EYUWUI.wui_server_applid.HELP.helpmembername
- ヘルプ・ページの保護に使用されます。

EYUWUI.wui_server_applid.EDITOR
- ビュー・エディターの保護に使用されます。

EYUWUI.wui_server_applid.USER
- ユーザーおよびユーザー・グループと共にユーザー・エディターの保護に使用されます。

wui_server_applid は、サーバーの CICS アプリケーション ID です。

ユーザーには、WUI リソースに対する読み取りまたは更新アクセス権限を与えることができます。

- 読み取り -- メイン・インターフェースのビュー、メニュー、マップ、およびヘルプ情報を使用したり、COVC を使用してビュー、メニュー、マップ、ユーザー・グループ、またはユーザー・オブジェクトをエクスポートしたりします。
- 更新 -- エディターにアクセスして項目を作成、更新、削除したり、COVC を使用してビュー、メニュー、マップ、ユーザー・グループ、またはユーザー・オブジェクトをインポートしたりします。

使用している ESM がプロファイルへのアクセスを付与も拒否もしない場合 (例えば、RACF プロファイルが定義されていない場合など)、Web ユーザー・インターフェースに正常にサインオンしたすべてのユーザーは、リソースへのアクセス権限を持つことになります。 総称プロファイルをセットアップすることによって、not authorized をデフォルトにすることができます。

注: このセキュリティーの保護対象として意図されているのはビューおよびメニュー自体であって、ビューおよびメニューが管理するオブジェクトではありません。それらは、通常の CICSPlex SM セキュリティーで保護されます。

ビュー・エディターの中で編集または削除するビュー・セット、マップまたはメニューを選択するときには、自分が更新アクセス権限を持っている項目しかリストされません。ただし、コピーする項目を選択するときは、自分が読み取りアクセス権

限を持っているすべての項目が表示されます。 このようにすることによって、自分が読み取り専用のアクセス権限を持っているオブジェクトを、更新可能なネーム・スペースの専用コピーとしてコピーすることができます。

アクセス可能なビューをブラウズするときは、セキュリティー例外はログに記録されません。 ユーザーに対して表示されるのは、アクセス不能なビューをフィルターに掛けて除去した後のリストです。 ただし、ユーザーが未許可のアクション (例えば、拒否されたネーム・スペースにビューを作成するなど) をしようとすると、EYULOG セキュリティー例外メッセージ EYUVS1100E が発行されます。

WUI セキュリティー・プロファイルの例

以下の例では、RACF TSO コマンド構文を使用します。デフォルトの CICS RACF クラスが使用されること、およびセキュリティー・プレフィックシングは使用されないことが想定されています。

これは適切なプロファイルを定義するための唯一の方法ではありません。 これらの例を調整して、インストール先環境の要件および規格に適合させることができます。

これらの例で、小文字のストリングは適切な使用 ID またはリソースに置き換える必要があります。

- 例 1

  Web ユーザー・インターフェースのユーザー・グループを作成します。

  ```
  ADDGROUP (WUISERV,WUIADM,WUIUSER,WUIEDIT)
  ```

- 例 2

  Web ユーザー・インターフェースのトランザクションを保護するためにプロファイルを定義します。

  ```
  RDEFINE GCICSTRN WUISYS UACC(NONE) ADDMEM(COVP,COVU,COVE)
  RDEFINE GCICSTRN WUIADMIN UACC(NONE) ADDMEM(COVG,COVC)
  RDEFINE TCICSTRN COVA UACC(NONE)
  ```

- 例 3

  ユーザー・グループに適切なプロファイルへの権限を与えます。

  ```
  PERMIT WUISYS CLASS(GCICSTRN) ID(WUISERV) ACCESS(READ)
  PERMIT WUIADMIN CLASS(GCICSTRN) ID(WUIADM) ACCESS(READ)
  PERMIT COVA CLASS(TCICSTRN) ID(WUIUSER,WUIEDIT) ACCESS(READ)
  ```

- 例 4

  トランザクション・セキュリティー・プロファイルをリフレッシュします。

  ```
  SETROPTS RACLIST(TCICSTRN) REFRESH
  ```

- 例 5

  ビュー・エディター・プロファイルを定義して、ユーザー・グループに適切なアクセス権限を与えます。

  ```
  RDEFINE FACILITY EYUWUI.wui_server_applid.EDITOR UACC(NONE)
  PERMIT EYUWUI.wui_server_applid.EDITOR CLASS(FACILITY) ID(WUIEDIT) ACCESS(UPDATE)
  ```

- 例 6

ビュー・セット・プロファイルを定義して、ユーザー・グループに適切なアクセス権限を与えます。

```
RDEFINE FACILITY EYUWUI.wui_server_applid.VIEW.viewsetname UACC(NONE)
PERMIT EYUWUI.wui_server_applid.VIEW.viewsetname CLASS(FACILITY) ID(WUIUSER)
ACCESS(READ)
```

- 例 7

  ユーザーを適切な Web ユーザー・インターフェース・グループに接続します。

  ```
  CONNECT wui_server_dfltuser GROUP(WUISERV)
  CONNECT (wui_server_pltpiusr,wui_administrator) GROUP(WUIADM)
  CONNECT (wui_administrator,wui_view_designer) GROUP(WUIEDIT)
  CONNECT (wui_operator1,wui_operator2...) GROUP(WUIUSER)
  ```

- 例 8

  Web ユーザー・インターフェース領域で CICS 代理ユーザー・セキュリティーがアクティブな場合、以下のものに類似した定義が必要になります。

  ```
  DEFINE SURROGAT wui_administrator.DFHSTART UACC(NONE)
  PERMIT wui_administrator.DFHSTART CLASS(SURROGAT) ID(WUIADM) ACCESS(READ)
  DEFINE SURROGAT wui_view_designer.DFHSTART UACC(NONE)
  PERMIT wui_view_designer.DFHSTART CLASS(SURROGAT) ID(WUIADM) ACCESS(READ)
  DEFINE SURROGAT wui_operator1.DFHSTART UACC(NONE)
  PERMIT wui_operator1.DFHSTART CLASS(SURROGAT) ID(WUIADM) ACCESS(READ)
  DEFINE SURROGAT wui_operator2.DFHSTART UACC(NONE)
  PERMIT wui_operator2.DFHSTART CLASS(SURROGAT) ID(WUIADM) ACCESS(READ)

  SETROPTS RACLIST(SURROGAT) REFRESH
  ```

セキュリティーがアクティブでない Web ユーザー・インターフェース・サーバーの実行

Web ユーザー・インターフェース・サーバーが CICS システム初期設定パラメーター **SEC=NO** を指定して実行している場合、Web ユーザー・インターフェースのユーザーは、COVC トランザクションで「セッション」の識別に使用されたユーザー ID を入力する必要があります。 ユーザーは、パスワードを入力する必要はありません。

ユーザー ID は、外部セキュリティー・マネージャーに定義されている必要はありません。 ビューおよび CICS リソースへのアクセスのアクセス検査は、Web ユーザー・インターフェース・サーバーの CICS 領域の **DFLTUSER** に基づいて行われます。Web ユーザー・インターフェースのすべてのユーザーは、ビュー・エディターとすべてのメニュー、ビュー・セット、およびヘルプ・メンバーにアクセスできます。

セキュリティーがアクティブでない場合、検査システムのプログラミング・インターフェース・コマンドによって作成されるメッセージには、CMAS か MAS のいずれかのデフォルトのユーザー ID が含まれます。アップグレード資料のSPI の変更点を参照してください。

プロファイルでの CICSPlex SM リソース名の指定

RACF プロファイルは、特定の CICS システム、CICS システムのグループ、または CICSplex を構成するすべてのシステムの CICSPlex SM ビューに対して作成できます。

このタスクについて

このリストには、RACF プロファイルで使用される CICSPlex SM ビューのリソース名が含まれています。

CICSPlex SM ビューは、実行する機能を反映するグループに分割されています。各機能グループ内で、ビューはそのタイプ別に分割されています。機能グループは、コンテキストを追加したり、グループによってはスコープやプラットフォームを追加したりして、さらに限定できます。特定のビューのセット (およびその関連するアクション・コマンド) へのアクセスは、以下のいずれかのリソース名形式を使用して、プロファイル内でセットを特定することで制御できます。

function.type.context

function.type.context.scope

ここで、

function
影響を受ける、以下の CICSPlex SM 関数の名前です。

ANALYSIS
リアルタイム分析

BAS ビジネス・アプリケーション・サービス

CLOUD
プラットフォームおよびアプリケーション

CONFIG
CMAS 構成

CSD CICS システム定義

MONITOR
リソース・モニター

OPERATE
操作

TOPOLOGY
CICSplex 構成

WORKLOAD
ワークロード管理

type 影響を受ける CICSPlex SM 関数を限定する、領域の固有名または総称名です。固有名は以下のとおりです。

AIMODEL
CICS AIMODEL

APPLCTN
プラットフォームで実行するアプリケーションのインスタンス

APPLICATION
プラットフォームにデプロイされるアプリケーション

BRFACIL
Link3270 ブリッジ機能

CONNECT
CICS 接続

DB2DBCTL
Db2/DBCTL リソースおよびサブシステム

DEF CICSPlex SM の定義

DOCTEMP
文書テンプレート
ENQMODEL
CICS グローバル・エンキュー・モデル
ENTJAVA
CorbaServer およびデプロイ済み DJAR
EXIT CICS 出口
FEPI CICS FEPI リソース
FILE CICS ファイル
IPCONN
IPIC 接続
JOURNAL
ジャーナル・モデル
PARTNER
CICS パートナー
PLATFORM
プラットフォーム
PROCTYPE
CICS BTS プロセス・タイプ
PROFILE
CICS プロファイル
PROGRAM
CICS プログラム
REGION
CICS 領域データ
RQMODEL
CICS 要求モデル
TASK
CICS アクティブ・タスク
TCPIPS
TCP/IP サービス
TDQUEUE
CICS 一時データ・キュー
TERMINAL
CICS 端末
TRAN
CICS トランザクション
TSQUEUE
CICS 一時記憶域キュー
UOW CICS 作業単位

type は、指定された関数に対して有効でなければなりません。 231 ページの表 23 に、有効な `function.type` の組み合わせがリストされています。

context

context は、指定された `function` および `type` によって影響を受ける、CMAS または CICSplex の固有名または総称名です。 `function` が

CONFIG である場合、context は CMAS または CICSplex にすることができます。他のすべての関数の場合、context は CICSplex でなければなりません。

scope scope は、context として識別される CICSplex 内の CICS システムの固有名または総称名です。 context が CMAS である場合、type が DEF である場合 (CSD リソースを処理している場合を除く)、または function が CLOUD である場合は、scope を指定しないでください。

注:

1. scope という語が CICS システムを意味しているのは、このセクションの文脈でのみです。それは、CICSplex 環境の一部として定義したスコープ (CICS システム・グループ) を意味しておらず、BAS 論理スコープのことでもありません。
2. 名前が総称システム名と一致しない場合に、CICS システム・グループを構成するすべてのシステムを組み込むには、各システムのプロファイルを設定する必要があります。

リソース名でのアスタリスクの使用

定義する必要があるプロファイルの数を減らすためには、* (1 つのアスタリスク) および ** (2 つの連続するアスタリスク) を使用して、1 つ以上の項目を表すことができます。アスタリスクの使用はオプションです。

注: プロファイル定義でアスタリスクを使用する前に、関連クラスの総称がアクティブになっていることを確認します。

```
SETROPTS GENERIC(CPSMOBJ,CPSMXMP)
```

以下の例は、アスタリスクの使用方法を示しています。

OPERATE.*.EYUPLX01.EYUPLX01
OPERATE 関数内で有効な任意のタイプに関連付けられているすべてのビュー・コマンドとアクション・コマンドは、コンテキストとスコープが EYUPLX01 である場合に認識されることを示します。

OPERATE.PROGRAM.**
OPERATE 関数内で PROGRAM タイプに関連付けられているすべてのビュー・コマンドとアクション・コマンドは、現行のコンテキストとスコープに関係なく認識されることを示します。

OPERATE.**
OPERATE 関数内で有効な任意のタイプに関連付けられているすべてのビュー・コマンドとアクション・コマンドは、現行のコンテキストとスコープに関係なく認識されることを示します。

**** 任意の 関数内で有効な任意の タイプに関連付けられているすべての ビュー・コマンドとアクション・コマンドは、現行のコンテキストとスコープに関係なく認識されることを示します。

有効なリソース名の組み合わせ

CICSPlex SM API または WUI を使用してリソース・テーブルへのアクセスを制御するプロファイルを作成するときは、関数とタイプの組み合わせを使用できます。

表 23 に、有効な関数とタイプの組み合わせと、それぞれの組み合わせに関連付けられているリソース名がリストされています。 WUI を使用する場合、リソース名は対応するビュー・セットのビューの下部に示されていることに注意してください。

表 *23. CICSPlex SM* を使用してアクセス可能なリソースに対する関数とタイプの組み合わせ

関数.タイプ	リソース・テーブル名と使用法
ANALYSIS.DEF	**ACTION** アクション定義を作成、表示、および維持します。 **APACTV** 分析点仕様に関連付けられた分析定義を表示します。 **APSPEC** 分析点仕様を作成、表示、および維持します。 **CMDMPAPS** 1 次 CMAS の役割を識別します。 **CMDMSAPS** 2 次 CMAS の役割を識別します。 **EVALDEF** 評価定義を作成、表示、および維持します。 **EVENT** CICSplex の状況の変化を表示します。 **EVENTDTL** イベントの原因となった分析定義に関連付けられた評価定義を表示します。 **LNKSRSCG** CICS システム・グループと分析仕様との間のリンクを記述します。 **LNKSRSCS** CICS システムと分析仕様との間のリンクを記述します。 **RTAACTV** CICS システムの分析定義と状況定義を表示します。 **RTADEF** 分析定義を作成、表示、および維持します。 **RTAGROUP** 分析グループを作成、表示、および維持します。 **RTAINAPS** 分析点仕様内の分析グループを表示します。 **RTAINGRP** 分析グループ内の分析定義と状況定義を表示します。 **RTAINSPC** 分析仕様内の分析グループを表示します。 **RTASPEC** 分析仕様を作成、表示、および維持します。 **STATDEF** 状況定義を作成、表示、および維持します。 **STAINGRP** RTAGROUP 内の状況定義のメンバーシップ関係を識別します。
BAS.APPLICTN	**BUNDDEF** バンドルとしてデプロイされたアプリケーションをインストールします。
BAS.CONNECT	**CONNDEF** ISC/MRO 接続定義をインストールします。 **IPCONDEF** IPIC 接続定義をインストールします。 **SESSDEF** セッション定義をインストールします。
BAS.DB2DBCTL	**DB2CDEF** Db2 接続定義をインストールします。 **DB2EDEF** Db2 エントリー定義をインストールします。 **DB2TDEF** Db2 トランザクション定義をインストールします。

表 23. *CICSPlex SM* を使用してアクセス可能なリソースに対する関数とタイプの組み合わせ (続き)

関数.タイプ	リソース・テーブル名と使用法
BAS.DEF	**ATOMDEF** Atom サービス定義を作成、表示、および維持します。 **BUNDDEF** バンドル定義を作成、表示、および維持します。 **CONNDEF** MRO および SNA を介した ISC 接続定義を作成、表示、および維持します。 **DB2CDEF** Db2 接続定義を作成、表示、および維持します。 **DB2EDEF** Db2 エントリー定義を作成、表示、および維持します。 **DB2TDEF** Db2 トランザクション定義を作成、表示、および維持します。 **DOCDEF** 文書テンプレート定義を作成、表示、および維持します。 **ENQMDEF** エンキュー・モデル定義を作成、表示、および維持します。 **FENODDEF** FEPI ノード定義を作成、表示、および維持します。 **FEPOODEF** FEPI プール定義を作成、表示、および維持します。 **FEPRODEF** FEPI プロパティー・セット定義を作成、表示、および維持します。 **FETRGDEF** FEPI ターゲット定義を作成、表示、および維持します。 **FILEDEF** ファイル定義を作成、表示、および維持します。 **IPCONDEF** IPIC 接続定義を作成、表示、および維持します。 **JRNMDEF** ジャーナル・モデル定義を作成、表示、および維持します。 **JVMSVDEF** JVM サーバー定義を作成、表示、および維持します。 **LIBDEF** LIBRARY 定義を作成、表示、および維持します。 **LSRDEF** LSR プール定義を作成、表示、および維持します。 **MAPDEF** マップ・セット定義を作成、表示、および維持します。 **MQCONDEF** IBM MQ 接続定義を作成、表示、および維持します。 **PARTDEF** パートナー定義を作成、表示、および維持します。 **PIPEDEF** パイプライン定義を作成、表示、および維持します。 **PROCDEF** プロセス・タイプ定義を作成、表示、および維持します。

表 23. *CICSPlex SM* を使用してアクセス可能なリソースに対する関数とタイプの組み合わせ (続き)

関数.タイプ	リソース・テーブル名と使用法
BAS.DEF (続き)	**PROFDEF** プロファイル定義を作成、表示、および維持します。 **PROGDEF** プログラム定義を作成、表示、および維持します。 **PRTNDEF** 区画セット定義を作成、表示、および維持します。 **RASGNDEF** リソース割り当てを作成、表示、および維持します。 **RASINDSC** 記述内のリソース割り当てを表示します。 **RASPROC** リソース割り当てプロセスを表示します。 **RDSCPROC** リソース記述プロセスを表示します。 **RESDESC** リソース記述を作成、表示、維持、およびインストールします。 **RESGROUP** リソース・グループを作成、表示、維持、およびインストールします。 **RESINDSC** 記述内のリソース・グループを表示します。 **RESINGRP** グループ内のリソース定義を表示します。 **RQMDEF** 要求モデル定義を作成、表示、および維持します。 **SESSDEF** セッション定義を作成、表示、および維持します。 **SYSRES** CICS システム・リソースを表示します。 **TCPDEF** TCP/IP サービス定義を作成、表示、および維持します。 **TDQDEF** 一時データ・キュー定義を作成、表示、および維持します。 **TERMDEF** 端末定義を作成、表示、および維持します。 **TRANDEF** トランザクション定義を作成、表示、および維持します。 **TRNCLDEF** トランザクション・クラス定義を作成、表示、および維持します。 **TYPTMDEF** 入力条件定義を作成、表示、および維持します。 **URIMPDEF** URI マップ定義を作成、表示、および維持します。 **WEBSVDEF** Web サービス定義を作成、表示、および維持します。

表 23. *CICSPlex SM* を使用してアクセス可能なリソースに対する関数とタイプの組み合わせ (続き)

関数.タイプ	リソース・テーブル名と使用法
BAS.DEF (続き)	**ATMINGRP** リソース・グループ内の Atom サービス定義のメンバーシップを記述します。 **BUNINGRP** リソース・グループ内のバンドル定義のメンバーシップを記述します。 **CONINGRP** リソース・グループ内の MRO または SNA を介した ISC 接続定義のメンバーシップを記述します。 **DOCINGRP** リソース・グループ内の文書テンプレート定義のメンバーシップを記述します。 **D2CINGRP** リソース・グループ内の Db2 接続定義のメンバーシップを記述します。 **D2EINGRP** リソース・グループ内の Db2 エントリー定義のメンバーシップを記述します。 **D2TINGRP** リソース・グループ内の Db2 トランザクション定義のメンバーシップを記述します。 **ENQINGRP** リソース・グループ内の ENQ/DEQ モデル定義のメンバーシップを記述します。 **FILINGRP** リソース・グループ内のファイル定義のメンバーシップを記述します。 **FNOINGRP** リソース・グループ内の FEPI ノード定義のメンバーシップを記述します。 **FPOINGRP** リソース・グループ内の FEPI プール定義のメンバーシップを記述します。 **FPRINGRP** リソース・グループ内の FEPI プロパティー・セット定義のメンバーシップを記述します。 **FSGINGRP** リソース・グループ内のファイル・キー・セグメント定義のメンバーシップを記述します。 **FTRINGRP** リソース・グループ内の FEPI ターゲット定義のメンバーシップを記述します。 **IPCINGRP** リソース・グループ内の IPIC 接続定義のメンバーシップを記述します。 **JMSINGRP** リソース・グループ内の JVM サーバー定義のメンバーシップを記述します。 **JRMINGRP** リソース・グループ内のジャーナル・モデル定義のメンバーシップを記述します。 **LIBINGRP** リソース・グループ内の LIBRARY 定義のメンバーシップを記述します。 **LSRINGRP** リソース・グループ内の LSR プール定義のメンバーシップを記述します。 **MAPINGRP** リソース・グループ内のマップ・セット定義のメンバーシップを記述します。 **MQCINGRP** グループ内の IBM MQ 接続定義を表示します。

表 23. *CICSPlex SM* を使用してアクセス可能なリソースに対する関数とタイプの組み合わせ (続き)

関数.タイプ	リソース・テーブル名と使用法
BAS.DEF (続き)	**PARINGRP** リソース・グループ内のパートナー定義のメンバーシップを記述します。 **PGMINGRP** リソース・グループ内のプログラム定義のメンバーシップを記述します。 **PIPINGRP** リソース・グループ内のパイプライン定義のメンバーシップを記述します。 **PRCINGRP** リソース・グループ内のプロセス・タイプ定義のメンバーシップを記述します。 **PRNINGRP** リソース・グループ内の区画セット定義のメンバーシップを記述します。 **PROINGRP** リソース・グループ内のプロファイル定義のメンバーシップを記述します。 **SESINGRP** リソース・グループ内のセッション定義のメンバーシップを記述します。 **TCLINGRP** リソース・グループ内のトランザクション・クラス定義のメンバーシップを記述します。 **TCPINGRP** リソース・グループ内の TCP/IP サービス定義のメンバーシップを記述します。 **TDQINGRP** リソース・グループ内の一時データ・キュー定義のメンバーシップを記述します。 **TRMINGRP** リソース・グループ内の端末定義のメンバーシップを記述します。 **TRNINGRP** リソース・グループ内のトランザクション定義のメンバーシップを記述します。 **TSMINGRP** リソース・グループ内の一時記憶域モデル定義のメンバーシップを記述します。 **TYPINGRP** リソース・グループ内の入力条件定義のメンバーシップを記述します。 **URIINGRP** リソース・グループ内の URI マップ定義のメンバーシップを記述します。 **WEBINGRP** リソース・グループ内の Web サービス定義のメンバーシップを記述します。
BAS.DOCTEMP	**DOCTEMP** 文書テンプレート定義をインストールします。
BAS.ENQMODEL	**ENQMDEF** エンキュー・モデル定義をインストールします
BAS.ENTJAVA	**JVMSVDEF** JVM サーバー定義をインストールします。
BAS.FILE	**FILEDEF** ファイル定義をインストールします。
BAS.JOURNAL	**JRNMDEF** ジャーナル・モデル定義をインストールします。
BAS.PARTNER	**PARTDEF** パートナー定義をインストールします。
BAS.PROCTYPE	**PROCDEF** BTS プロセス・タイプ定義をインストールします。
BAS.PROFILE	**PROFDEF** プロファイル定義をインストールします。
BAS.PROGRAM	**LIBDEF** LIBRARY 定義をインストールします。 **MAPDEF** マップ・セット定義をインストールします **PROGDEF** プログラム定義をインストールします。 **PRTNDEF** 区画セット定義をインストールします

表 23. *CICSPlex SM* を使用してアクセス可能なリソースに対する関数とタイプの組み合わせ (続き)

関数.タイプ	リソース・テーブル名と使用法
BAS.REGION	**LSRDEF** LSR プール定義をインストールします。
	MQCONDEF
	WebSphere MQ 接続定義をインストールします。
	TRNCLDEF
	トランザクション・クラス定義をインストールします。
BAS.TCPIPS	**ATOMDEF**
	Atom サービス定義をインストールします。
	PIPEDEF
	パイプライン定義をインストールします。
	TCPDEF TCP/IP サービス定義をインストールします。
	URIMPDEF
	URI マップ定義をインストールします。
	WEBSVDEF
	Web サービス定義をインストールします。
BAS.TDQUEUE	**TDQDEF**
	一時データ・キュー定義をインストールします
BAS.TERMINAL	**TERMDEF**
	端末定義をインストールします。
	TYPTMDEF
	入力条件定義をインストールします。
BAS.TRAN	**TRANDEF**
	トランザクション定義をインストールします。
CLOUD.APPLICATION	**APPLDEF**
	プラットフォームにアプリケーション定義をインストールします。
	APPLCTN
	プラットフォームにデプロイされたアプリケーションを表示および管理します。
CLOUD.DEF	**APPLDEF**
	プラットフォームのアプリケーション定義を作成、表示、および維持します。
	PLATDEF
	プラットフォーム定義を作成、表示、および維持します。
CLOUD.PLATFORM	**PLATDEF**
	プラットフォーム定義をインストールします。
	PLATFORM
	プラットフォームを表示および管理します。
	MGMTPART
	アプリケーションおよびプラットフォームの管理パーツを作成、表示、および維持します。
CONFIG.DEF	**CICSPLEX**
	CICSplex 内の CMAS を表示および管理します。
	CMAS アクティブな CMAS を表示および管理します。
	CMASLIST
	CMAS およびその特性への接続を記述します。
	CMASPLEX
	CMAS の CICSplex を表示します。
	CMTCMDEF
	CMAS リンクを作成、表示、および維持します。
	CMTCMLNK
	アクティブな CMAS リンクを表示します。
	CMTPMLNK
	アクティブな CMAS から MAS へのリンクを表示します。
	CPLEXDEF
	CICSplex の定義を作成、表示、および維持します。
	CPLXCMAS
	CICSplex への CMAS を表示します。

表 23. *CICSPlex SM* を使用してアクセス可能なリソースに対する関数とタイプの組み合わせ (続き)

関数.タイプ	リソース・テーブル名と使用法
CSD.DEF	**ATOMDEF** Atom サービス定義を作成、インストール、表示、および維持します。 **BUNDDEF** バンドル定義を作成、インストール、表示、および維持します。 **CONNDEF** MRO および SNA を介した ISC 接続定義を作成、インストール、表示、および維持します。 **CSDGROUP** CSD リソース・グループ定義を作成、インストール、表示、および維持します。 **CSDINGRP** CSD リソース・グループのメンバーシップを記述します。 **CSDINLST** CSD リソース・グループ・リストのメンバーシップを記述します。 **CSDLIST** CSD リソース・グループ・リスト定義を作成、インストール、表示、および維持します。 **DB2CDEF** Db2 接続定義を作成、インストール、表示、および維持します。 **DB2EDEF** Db2 エントリー定義を作成、インストール、表示、および維持します。 **DB2TDEF** Db2 トランザクション定義を作成、インストール、表示、および維持します。 **DOCDEF** 文書テンプレート定義を作成、インストール、表示、および維持します。 **ENQMDEF** エンキュー・モデル定義を作成、インストール、表示、および維持します。 **FILEDEF** ファイル定義を作成、インストール、表示、および維持します。 **IPCONDEF** IPIC 接続定義を作成、インストール、表示、および維持します。 **JRNMDEF** ジャーナル・モデル定義を作成、インストール、表示、および維持します。 **JVMSVDEF** JVM サーバー定義を作成、インストール、表示、および維持します。 **LIBDEF** LIBRARY 定義を作成、インストール、表示、および維持します。 **LSRDEF** LSR プール定義を作成、インストール、表示、および維持します。 **MAPDEF** マップ・セット定義を作成、インストール、表示、および維持します。 **MQCONDEF** IBM MQ 接続定義を作成、インストール、表示、および維持します。

表 23. *CICSPlex SM* を使用してアクセス可能なリソースに対する関数とタイプの組み合わせ (続き)

関数.タイプ	リソース・テーブル名と使用法
CSD.DEF (続き)	**PARTDEF** パートナー定義を作成、インストール、表示、および維持します。
	PIPEDEF パイプライン定義を作成、インストール、表示、および維持します。
	PROCDEF プロセス・タイプ定義を作成、インストール、表示、および維持します。
	PROFDEF プロファイル定義を作成、インストール、表示、および維持します。
	PROGDEF プログラム定義を作成、インストール、表示、および維持します。
	PRTNDEF 区画セット定義を作成、インストール、表示、および維持します。
	SESSDEF セッション定義を作成、表示、および維持します。
	TCPDEF TCP/IP サービス定義を作成、インストール、表示、および維持します。
	TDQDEF 一時データ・キュー定義を作成、インストール、表示、および維持します。
	TERMDEF 端末定義を作成、インストール、表示、および維持します。
	TRANDEF トランザクション定義を作成、インストール、表示、および維持します。
	TRNCLDEF トランザクション・クラス定義を作成、インストール、表示、および維持します。
	TSMINGRP リソース・グループ内の一時記憶域モデル定義のメンバーシップを記述します。
	TYPTMDEF 入力条件定義を作成、インストール、表示、および維持します。
	URIMPDEF URI マップ定義を作成、インストール、表示、および維持します。
	WEBSVDEF Web サービス定義を作成、インストール、表示、および維持します。
MONITOR.CONNECT	**MCONNECT** ISC と MRO の接続
	MMODENAME LU 6.2 モード名
MONITOR.DB2DBCTL	**MDB2THRD** Db2 スレッド
MONITOR.DEF	**LNKSMSCG** CICS システム・グループとモニター仕様との間のリンクを記述します。
	LNKSMSCS CICS システムとモニター仕様との間のリンクを記述します。
	MONDEF モニター定義を作成、表示、および維持します。
	MONGROUP モニター・グループを作成、表示、および維持します。
	MONINGRP モニター・グループ内のモニター定義を作成、表示、および維持します。
	MONINSPC モニター仕様内のモニター・グループを作成、表示、および維持します。
	MONSPEC モニター仕様を作成、表示、および維持します。
	POLMON 特定の CICS システム内のモニター定義を記述します。
MONITOR.FEPI	**MFEPICON** FEPI 接続

表 23. *CICSPlex SM* を使用してアクセス可能なリソースに対する関数とタイプの組み合わせ (続き)

関数.タイプ	リソース・テーブル名と使用法
MONITOR.FILE	**MCMDT** データ・テーブル **MLOCFILE** ローカル・ファイル **MREMFILE** リモート・ファイル
MONITOR.JOURNAL	**MJOURNL** ジャーナル **MJRNLNAM** ジャーナル名
MONITOR.PROGRAM	**MPROGRAM** Programs (プログラム)
MONITOR.REGION	**MCICSDSA** 動的ストレージ域 **MCICSRGN** CICS システム **MLSRPBUF** LSRPOOL バッファー・プール **MLSRPOOL** LSRPOOL **MTRANCLS** トランザクション・クラス
MONITOR.TDQUEUE	**MINDTDQ** 間接一時データ・キュー **MNTRATDQ** 区画内一時データ・キュー **MREMTDQ** リモート一時データ・キュー **MTDQGBL** グローバル区画内一時データ・キュー **MXTRATDQ** 区画外一時データ・キュー
MONITOR.TERMINAL	**MTERMNL** 端末
MONITOR.TRAN	**MLOCTRAN** ローカル・トランザクション **MREMTRAN** リモート・トランザクション
MONITOR.TSQUEUE	**MTSQGBL** グローバル一時記憶域キュー
OPERATE.AIMODEL	**AIMODEL** 自動インストール・モデル **CRESAIMD** CICS システム内の自動インストールされた端末モデルのインスタンスを記述します。

表 23. *CICSPlex SM* を使用してアクセス可能なリソースに対する関数とタイプの組み合わせ (続き)

関数.タイプ	リソース・テーブル名と使用法
OPERATE.APPLICTN	**BUNDLE** バンドルに関する情報を提供する CICS リソース・オブジェクト。 **BUNDPART** バンドル・パーツに関する情報を提供する CICS リソース・オブジェクト。 **CRESBUND** CICS システム内のバンドルのインスタンスを記述します。 **CRESXMLT** CICS システム内の XMLTRANSFORM リソースのインスタンスを記述します。 **EPADAPT** EP アダプターに関する情報を提供する CICS リソース・オブジェクト。 **EPADSET** EP アダプター・セットに関する情報を提供する CICS リソース・オブジェクト。 **EPAINSET** イベント処理アダプター・セット内のイベント処理アダプターの名前を提供する CICS リソース・オブジェクト。 **EVCSDATA** アプリケーション・データ述部に関する情報を提供する CICS リソース・オブジェクト。 **EVCSINFO** 情報源に関する情報を提供する CICS リソース・オブジェクト。 **EVCSOPT** アプリケーション・オプション述部に関する情報を提供する CICS リソース・オブジェクト。 **EVCSPEC** キャプチャー仕様に関する情報を提供する CICS リソース・オブジェクト。 **EVNTBIND** イベント・バインディングに関する情報を提供する CICS リソース・オブジェクト。 **EVNTGBL** イベント処理に関する情報を提供する CICS リソース・オブジェクト。 **OSGIBUND** OSGi バンドルに関する情報を提供する CICS リソース・オブジェクト。 **OSGISERV** OSGi サービスに関する情報を提供する CICS リソース・オブジェクト。 **XMLTRANS** XMLTRANSFORM リソース。
OPERATE.BRFACIL	**BRFACIL** LINK3270 ブリッジ機能
OPERATE.CONNECT	**CONNECT** MRO 接続と SNA を介した ISC 接続 **CRESCONN** CICS システム内の MRO または SNA を介した ISC 接続のインスタンスを記述します。 **CRESIPCN** CICS システム内の IPIC 接続のインスタンスを記述します。 **CRESMODE** CICS システム内の LU6.2 モードネームのインスタンスを記述します。 **IPCONN** IPIC 接続 **MODENAME** LU 6.2 モード名

表 23. *CICSPlex SM* を使用してアクセス可能なリソースに対する関数とタイプの組み合わせ (続き)

関数.タイプ	リソース・テーブル名と使用法
OPERATE.DB2DBCTL	**CRESDB2C** CICS システム内の Db2 接続のインスタンスを記述します。 **CRESDB2E** CICS システム内の Db2 エントリーのインスタンスを記述します。 **CRESDB2P** CICS 領域内の Db2 パッケージ・セットのインスタンスを記述します。 **CRESDB2T** CICS システム内の Db2 トランザクションのインスタンスを記述します。 **DB2CONN** Db2 接続 **DB2ENTRY** Db2 エントリー **DB2PKGST** Db2 パッケージ・セット **DB2TRN** Db2 トランザクション **DBCTLSS** DBCTL サブシステム **DB2SS** Db2 サブシステム **DB2THRD** Db2 スレッド **DB2TRAN** Db2 トランザクション
OPERATE.DOCTEMP	**CRESDOCT** CICS システム内の文書テンプレートのインスタンスを記述します。 **DOCTEMP** 文書テンプレート
OPERATE.ENQMODEL	**CRESENQM** CICS システム内の ENQ/DEQ モデルのインスタンスを記述します。 **ENQMODEL** エンキュー・モデル
OPERATE.ENTJAVA	**CLCACHE** 共用クラス・キャッシュ **CRESJVMS** CICS 領域内の JVM サーバーのインスタンスを記述します。 **JVM** Java™ 仮想マシン **JVMPOOL** JVM プール **JVMPROF** JVM プロファイル **JVMSERV** JVM サーバー
OPERATE.EXIT	**CRESGLUE** CICS システム内のグローバル・ユーザー出口のインスタンスを記述します。 **CRESTRUE** CICS システム内のタスク関連ユーザー出口のインスタンスを記述します。 **EXITGLUE** グローバル・ユーザー出口 **EXITTRUE** タスク関連のユーザー出口 **EXTGLORD** 順序が付けられたグローバル・ユーザー出口

表 23. *CICSPlex SM* を使用してアクセス可能なリソースに対する関数とタイプの組み合わせ (続き)

関数.タイプ	リソース・テーブル名と使用法
OPERATE.FEPI	**CRESFECO** CICS システム内の FEPI 接続のインスタンスを記述します。 **CRESFEND** CICS システム内の FEPI ノードのインスタンスを記述します。 **CRESFEPO** CICS システム内の FEPI プールのインスタンスを記述します。 **CRESFETR** CICS システム内の FEPI ターゲットのインスタンスを記述します。 **FEPICONN** FEPI 接続 **FEPINODE** FEPI ノード **FEPIPOOL** FEPI プール **FEPIPROP** FEPI プロパティー・セット **FEPITRGT** FEPI ターゲット
OPERATE.FILE	**CFDTPOOL** カップリング・ファシリティー・データ・テーブル・プール **CMDT** データ・テーブル **CRESDSNM** CICS システム内のデータ・セットのインスタンスを記述します。 **CRESFILE** CICS システム内のファイルのインスタンスを記述します。 **DSNAME** データ・セット **LOCFILE** ローカル・ファイル **REMFILE** リモート・ファイル
OPERATE.JOURNAL	**CRESJRNL** CICS システム内のジャーナルのインスタンスを記述します。 **CRESJRNM** CICS システム内のジャーナル名のインスタンスを記述します。 **JRNLMODL** ジャーナル・モデル **JRNLNAME** システム・ログおよび一般ログ **STREAMNM** MVS ログ・ストリーム
OPERATE.PARTNER	**CRESPART** CICS システム内のパートナー・テーブルのインスタンスを記述します。 **PARTNER** CICS パートナー
OPERATE.PROCTYPE	**CRESPRTY** CICS システム内のプロセス・タイプのインスタンスを記述します。 **PROCTYP** プロセス・タイプ
OPERATE.PROFILE	**CRESPROF** CICS システム内のプロファイルのインスタンスを記述します。 **PROFILE** CICS プロファイル

表 23. *CICSPlex SM* を使用してアクセス可能なリソースに対する関数とタイプの組み合わせ (続き)

関数.タイプ	リソース・テーブル名と使用法
OPERATE.PROGRAM	**CRESPRGM** CICS システム内のプログラムのインスタンスを記述します。 **PROGRAM** Programs (プログラム) **LIBDSN** LIBRARY データ・セット名 **LIBRARY** LIBRARY データ・セット **RPLLIST** DFHRPL データ・セット
OPERATE.REGION	**CICSDSA** 動的ストレージ域 **CICSPAGP** CICS ページ・プール **CICSRGN** CICS システム **CICSSTOR** すべての CICS 動的ストレージ域 **CRESSDMP** CICS システム内のシステム・ダンプ・コードのインスタンスを記述します。 **CRESTDMP** CICS システム内のトランザクション・ダンプ・コードのインスタンスを記述します。 **DOMSPOOL** CICS ストレージ・ドメイン・サブプール **DSPGBL** グローバル CICS ディスパッチャー **DSPMODE** CICS ディスパッチャー TCB モード **DSPPOOL** CICS ディスパッチャー TCB プール **ENQUEUE** CICS エンキュー **FEATURE** 機能トグル **LOADACT** 動的ストレージ域による CICS ローダー・アクティビティー **LOADER** CICS ローダー・アクティビティー **LSRPBUF** LSR プールのバッファー使用 **LSRPOOL** LSR プール **MASHIST** MAS ヒストリー **MONITOR** CICS モニターおよび統計 **MQCON** IBM MQ 接続 **MQCONN** IBM MQ 接続統計 **MQINI** IBM MQ 開始キュー **MVSESTG** MVS ストレージ・エレメント **MVSTCB** MVS TCB **MVSTCBGL** グローバル MVS TCB **MVSWLM** z/OS ワークロード・マネージャー

表 23. *CICSPlex SM* を使用してアクセス可能なリソースに対する関数とタイプの組み合わせ (続き)

関数.タイプ	リソース・テーブル名と使用法
OPERATE.REGION (続き)	**RECOVERY** CICS リカバリー・マネージャー **REQID** 指定時刻に出される要求 **SYSDUMP** システム・ダンプ・コード **SYSPARM** CICSPlex SM によって管理されるアクティブ CICS システムでの、システム初期設定パラメーターまたはシステム初期設定パラメーター・オーバーライドを記述する CICS リソース。 **TRANCLAS** トランザクション・クラス **TRANDUMP** トランザクション・ダンプ・コード
OPERATE.RQMODEL	**CRESRQMD** CICS システム内の要求のインスタンスを記述します。
OPERATE.TASK	**EXCI** **HTASK** 完了したタスク **TASK** アクティブ・タスク **TASKESTG** タスク・ストレージ・エレメント **TASKFILE** タスクのファイル使用 **TASKRMI** 個々のタスクによる RMI 使用 **TASKTSQ** 個々のタスクによる TSQ 使用 **TSKSPOLS** すべてのタスク・サブプール **TSKSPOOL** タスク・ストレージ・サブプール
OPERATE.TCPIPS	**ATOMSERV** Atom サービス **CRESATOM** CICS システム内の Atom サービスのインスタンスを記述します。 **CRESTCPS** CICS システム内の TCP/IP サービスのインスタンスを記述します。 **HOST** URI ホスト **IPFACIL** IPIC 接続セッション **PIPELINE** パイプライン **TCPIPGBL** TCP/IP グローバル統計 **TCPIPS** TCP/IP サービス **URIMAP** URI マップ **URIMPGBL** URI マップ・グローバル統計 **WEBSERV** Web サービス

表 23. *CICSPlex SM* を使用してアクセス可能なリソースに対する関数とタイプの組み合わせ (続き)

関数.タイプ	リソース・テーブル名と使用法
OPERATE.TDQUEUE	**CRESTDQ** CICS システム内の一時データ・キューのインスタンスを記述します。 **EXTRATDQ** 区画外一時データ・キュー **INDTDQ** 間接一時データ・キュー **INTRATDQ** 区画内一時データ・キュー **REMTDQ** リモート一時データ・キュー **TDQGBL** 区画内一時データ・キュー使用量
OPERATE.TERMINAL	**CRESTERM** CICS システム内の端末のインスタンスを記述します。 **TERMNL** 端末
OPERATE.TRAN	**CRESTRAN** CICS システム内のトランザクションのインスタンスを記述します。 **LOCTRAN** ローカル・トランザクション **REMTRAN** リモート・トランザクション **TASKASSC** タスク関連データ
OPERATE.TSQUEUE	**CRESTSMD** CICS システム内の一時記憶域キューのインスタンスを記述します。 **TSQUEUE** 一時記憶域キュー **TSMODEL** 一時記憶域モデル **TSPOOL** 一時記憶域プール **TSQGBL** グローバル一時記憶域キュー **TSQSHR** 共用一時記憶域キュー **TSQNAME** 長い一時記憶域キュー
OPERATE.UOW	**UOWDSNF** 中断された作業単位の表示 **UOWENQ** 実行作業単位のエンキューの表示 **UOWLINK** 作業単位のリンクの表示 **UOW** 実行作業単位の表示

表 23. *CICSPlex SM* を使用してアクセス可能なリソースに対する関数とタイプの組み合わせ (続き)

関数.タイプ	リソース・テーブル名と使用法
TOPOLOGY.DEF	**CSYSGRP** CICS システム・グループを作成、表示、および維持します。 **CSYSDEF** CICS システムを作成、表示、および維持します。 **CSGLCGCG** CICS システム・グループから外部システム・グループへのリンクを記述します。 **CSGLCGCS** CICS システムからシステム・グループへのリンクを記述します。 **MAS** CICSplex 内の CICS システムを表示します。 **MASSTAT** CMAS による CICSplex 内の CICS システムを表示します。 **PERIODEF** 期間定義を表示します。 **SYSLINK** CICS システム間に存在するリンクに関する情報を表示します。
WORKLOAD.DEF	**DTRINGRP** トランザクション・グループ内のトランザクションを表示します。 **LNKSWSCG** CICS システム・グループとワークロード仕様との間のリンクを記述します。 **LNKSWSCS** CICS システムとワークロード仕様との間のリンクを記述します。 **TRANGRP** トランザクション・グループを作成、表示、および維持します。 **WLMATAFF** アクティブ・アフィニティーを表示および破棄します。 **WLMATGRP** アクティブ・トランザクション・グループを表示および破棄します。 **WLMATRAN** トランザクション・ディレクトリーを表示および破棄します。 **WLMAWAOR** ワークロード内のアクティブな AOR を表示します。 **WLMAWDEF** アクティブ・ワークロード定義を表示および破棄します。 **WLMAWORK** アクティブ・ワークロードを表示します。 **WLMAWTOR** ワークロード内のアクティブな TOR を表示します。 **WLMDEF** ワークロード定義を作成、表示、および維持します。 **WLMGROUP** ワークロード・グループを作成、表示、および維持します。 **WLMINGRP** グループ内のワークロード定義を表示します。 **WLMINSPC** ワークロード仕様内のワークロード・グループを表示します。 **WLMSPEC** ワークロード仕様を作成、表示、および維持します。

プラットフォームおよびアプリケーションのセキュリティー

プラットフォーム上にデプロイされているアプリケーションのリソースを保護するには、CICSplex でのプラットフォームとアプリケーションに対応する CICSPlex SM 用の RACF セキュリティー・プロファイルを作成します。

プラットフォームおよびアプリケーションのセキュリティーは、その他の CICSPlex SM コンポーネントのセキュリティーと同様の方法でセットアップします。セキュ

リティー・プロファイル内でビュー (およびそれらに関連するアクション・コマンド) からなる特定のセットを識別することにより、そのセットへのアクセスを制御します。これらのセキュリティー・プロファイルを使用すると、プラットフォームやアプリケーションのインストール、有効化/無効化、使用可能化/使用不可化、照会、または破棄を行う権限をユーザーに与えることができます。これにより、無許可ユーザーはこれらのリソースを作成したり管理したりできなくなります。

あるユーザーにプラットフォームまたはアプリケーションに対してアクションを実行する権限を付与すると、そのプラットフォームまたはアプリケーションに対して動的に生成されるリソースについても、そのユーザーが同じアクションを実行できる権限を付与することになります。例えば、あるアプリケーションを有効にする権限を持つユーザーは、CICSplex 内のすべてのプラットフォームの CICS 領域にインストールされたアプリケーション用 CICS バンドルを有効にする権限も持ちます。アプリケーションまたはプラットフォームを介した CICS バンドルの操作時には、CICS コマンドとリソース・セキュリティー検査、および CICSPlex SM でシミュレートされる CICS セキュリティー検査は行われません。

プラットフォームとそこにデプロイされたアプリケーションを保護するには、以下に挙げる機能とタイプを組み合わせてセキュリティー・プロファイルを設定することができます。

CLOUD.DEF.*context*

このセキュリティー・プロファイルは、プラットフォームとアプリケーションの定義を含んでいるリソース・テーブル PLATDEF および APPLDEF を対象とします。 *context* は、このセキュリティー・プロファイルの対象となる CICSplex の固有名または総称名です。

このセキュリティー・プロファイルに対する UPDATE アクセス権限を持つユーザーは、CICSPlex SM データ・リポジトリー内でプラットフォームとアプリケーションの定義の作成、更新、および除去を行うことができます。READ アクセス権限を持つユーザーは、CICSPlex SM データ・リポジトリー内でこれらの定義を表示できます。

CLOUD.PLATFORM.*context*

このセキュリティー・プロファイルは、PLATDEF リソースのインストールおよび PLATFORM リソースに対する操作を対象とします。また、ユーザーに対して管理パーツ (MGMTPART リソース) の表示も許可します。 *context* は、このセキュリティー・プロファイルの対象となる CICSplex の固有名または総称名です。

このセキュリティー・プロファイルに対する ALTER アクセス権限を持つユーザーは、CICSplex でのプラットフォームのインストールおよび破棄を行うことができます。 (ユーザーがプラットフォームをインストールするには、PLATDEF リソースを対象とする CLOUD.DEF プロファイルに対する READ アクセス権限も必要です。) UPDATE アクセス権限を持つユーザーはプラットフォームを有効/無効にすることができます。また、UPDATE アクセス権限を持つユーザーは、CICS 領域をプラットフォームの領域タイプに追加したり、CICS 領域をプラットフォームの領域タイプから除去したりすることもできます。 READ アクセス権限を持つユーザーは PLATFORM リソースと MGMTPART リソースを表示できます。これらの権限は、CICSplex に存在するすべてのプラットフォームに適用されます。

CLOUD.APPLICATION.*context*

このセキュリティー・プロファイルは、APPLDEF リソースのインストールおよび APPLCTN リソースに対する操作を扱います。 *context* は、このセキュリティー・プロファイルの対象となる CICSplex の固有名または総称名です。

このセキュリティー・プロファイルに対する ALTER アクセス権限を持つユーザーは、CICSplex でのアプリケーションのインストールおよび破棄を行うことができます。 (ユーザーがアプリケーションをインストールするには、APPLDEF リソースを対象とする CLOUD.DEF プロファイルに対する READ アクセス権限も必要です。) UPDATE アクセス権限を持つユーザーは、アプリケーションを有効/無効にしたり、使用可能/使用不可にしたりできます。 READ アクセス権限を持つユーザーは APPLCTN リソースを表示できます。これらの権限は、CICSplex に存在するすべてのプラットフォーム内のすべてのアプリケーションに適用されます。特定のアプリケーションに関して異なるセキュリティー権限が必要な場合は、別の CICSplex を使用してアプリケーションのデプロイ場所のプラットフォームをホストしてください。

注**:** これらのセキュリティー・プロファイルは、保守ポイント CMAS でのみ検査されます。セキュリティー検査は、保守ポイント CMAS の EYULOG 内のメッセージ EYUCR0009I によって報告されます。違反についてメッセージ EYUCR0009I を受け取るには、CICSPlex SM システム・パラメーター (EYUPARM) **SECLOGMSG** を YES に設定する必要があります。 **SECLOGMSG** についての詳細は、CICSPlex SM システム・パラメーターを参照してください。

CLOUD セキュリティー・プロファイルはプラットフォームやアプリケーション用に動的に生成されるリソースへのアクションを対象としますが、それらがインストールされた CICSplex および CICS 領域にある個別リソースに対してユーザーは以下のような限定的なアクションを直接実行できます。

- プラットフォームの一部である CICS 領域の CICSPlex SM トポロジー定義 (CSYSDEF、つまり CICS システム定義) を変更できます。領域タイプ・レベルで指定されている属性値はロックされていて変更できませんが、他の属性値は変更できます。
- プラットフォームまたはアプリケーションのインストール時に動的に作成された BUNDLE リソースを使用可能/使用不可にしたり、有効/無効にしたり、照会したりできます。プラットフォームまたはアプリケーションのインストール時に作成されたバンドルを直接、個別に破棄することはできません。
- CICS バンドル内部で定義され、プラットフォームまたはアプリケーションのインストール時に動的に作成されたリソース (例えば PROGRAM リソース) を照会できます。これらのリソースがプラットフォームまたはアプリケーションのインストール時に作成された場合は、リソースを直接的に有効化、無効化、および破棄することはできません。

プラットフォームまたはアプリケーションのインストール時に作成された、プラットフォームの一部である CICS 領域、または個々の CICS バンドル、または CICS バンドル内に定義されたリソースに対してアクションを直接実行するときに、 CICS コマンドとリソース・セキュリティー検査、および CICSPlex SM でシミュレートされる CICS セキュリティー検査が適用されます。

- TOPOLOGY.DEF.*context* セキュリティー・プロファイルは、プラットフォームの一部である個々の CICS 領域の CICSPlex SM トポロジー定義で行われるアクションを対象とします。 *context* は、このセキュリティー・プロファイルの対象となる CICSplex の固有名または総称名です。 UPDATE アクセス権限を持つユーザーは、プラットフォームの一部である CICS 領域の CSYSDEF を変更できますが、プラットフォームそのものでロックされている属性値は変更できません。
- プラットフォームまたはアプリケーションのインストール時に作成される CICS バンドルには、$ 文字で始まる固有の名前が生成されます。この方法で動的に作成された個々の CICS バンドルに対するアクションのセキュリティーを設定するには、BUNDLE リソース・タイプおよびリソース名 `$*` を指定してセキュリティー・プロファイルをセットアップします。 `BUNDLE.$*` に対する UPDATE アクセス権限を持つユーザーは、プラットフォームやアプリケーションのために作成された BUNDLE リソースを使用可能/使用不可にしたり、有効/無効にしたりできます。また、READ アクセス権限を持つユーザーは、それらの BUNDLE リソースを照会することができます。

さまざまなプラットフォームにデプロイされるアプリケーションについて、セキュリティー対策を個々の PROGRAM リソースに適用する場合、アプリケーション・エントリー・ポイントとして宣言されているプログラムは保護しますが、そのアプリケーション内の他のプログラムは保護しないようにしてください。あるプラットフォーム上にデプロイされるアプリケーションの一部を構成するプログラムについて指定するセキュリティー設定値は、パブリックなプログラムとプライベートなプログラムの両方に適用されます。アプリケーションのバージョンは考慮されません。アプリケーション・エントリー・ポイントとして宣言されるプログラムには、使用環境内で固有の PROGRAM リソース名が必要です。しかし、そのアプリケーションの下位レベルで実行されるプログラムを保護すると、別のアプリケーションで同じ名前のプログラムが実行されている可能性があり、その場合、予期しない結果になる可能性があります。この場合、アプリケーション・エントリー・ポイントとして宣言されているプログラムについてはアクセス権限がユーザーに付与されている一方、そのプログラム名の別のインスタンスによるセキュリティー設定が有効になっているために、アプリケーションの下位レベルで実行されるプログラムにはアクセスする許可がないということがあり得ます。アプリケーション・エントリー・ポイント・プログラムとして宣言されているプログラムに適用するセキュリティー対策としては、アプリケーション全体に適用されるものを考慮してください。

以前の CICS リリースで CICS バンドルを使用していた場合は、これらのバンドルに関してユーザーに与えたセキュリティー権限を確認してください。 CICS バンドルのセキュリティーのセットアップ方法によっては、個々の CICS バンドルに対するアクションの実行権限を持つユーザーが、バンドル・インストールの過程で動的に作成されるリソースに対してアクションを実行できるようになっている可能性があります。 BUNDLE リソースに対する権限のレベルが引き続き適切であることを確認してください。

250 ページの表 24に、以下の対象に対して実行されるアクションに適用されるセキュリティー検査を要約します。すなわち、1) プラットフォーム、2) アプリケーション、3) プラットフォームまたはアプリケーションのインストール時に動的に作成された個々の CICS バンドル、あるいは 4) プラットフォームまたはアプリケーション用の CICS バンドル内で定義されたリソースです。

表 24. プラットフォーム、アプリケーション、および生成される *CICS* バンドルに対する操作のセキュリティー検査

操作	プラットフォーム **(CICS** バンドルを含む**)**	アプリケーション **(**バンドルを含む**)**	動的に作成された **CICS** バンドル	動的に作成された **CICS** バンドルで定義されるリソース
定義	CLOUD.DEF プロファイル (UPDATE、または定義を表示するための READ)。また、TOPOLOGY.DEF プロファイル (プラットフォームのインストール後に個々の CICS 領域 CSYSDEF を変更するための UPDATE)	CLOUD.DEF プロファイル (UPDATE、または定義を表示するための READ)	リソース定義を個別に管理できません	リソース定義を個別に管理できません
インストール	CLOUD.PLATFORM プロファイル (ALTER) および CLOUD.DEF プロファイル (READ)	CLOUD.APPLICATION プロファイル (ALTER) および CLOUD.DEF プロファイル (READ)	個別にインストールすることはできません	個別にインストールすることはできません
有効化または無効化	CLOUD.PLATFORM プロファイル (UPDATE)	CLOUD.APPLICATION プロファイル (UPDATE)	CICS コマンドとリソースのセキュリティー、および CICSPlex SM でシミュレートされる CICS セキュリティーの検査 (BUNDLE.$* プロファイルを使用)	個別に有効化/無効化することはできません
使用可能または使用不可にする	適用外	CLOUD.APPLICATION プロファイル (UPDATE)	CICS コマンドとリソースのセキュリティー、および CICSPlex SM でシミュレートされる CICS セキュリティーの検査 (BUNDLE.$* プロファイルを使用)	個別に使用可能または使用不可にすることはできません。
Inquire (問い合わせ)	CLOUD.PLATFORM プロファイル (READ) - 管理パーツの表示も許可します	CLOUD.APPLICATION プロファイル (READ)	CICS コマンドとリソースのセキュリティー、および CICSPlex SM でシミュレートされる CICS セキュリティーの検査 (BUNDLE.$* プロファイルを使用)	CICS コマンドとリソースのセキュリティー、および CICSPlex SM でシミュレートされる CICS セキュリティーの検査
破棄	CLOUD.PLATFORM プロファイル (ALTER)	CLOUD.APPLICATION プロファイル (ALTER)	個別に破棄できません	個別に破棄することはできません

CICSPlex SM 用のセキュリティーのセットアップ、およびセキュリティー・プロファイルの作成について詳しくは、Implementing CICSPlex SM securityを参照してください。

CICS シミュレーション・セキュリティーのアクティブ化

CICSPlex SM リソース・クラスを使用して、操作ビューおよびモニター・ビューへのアクセスを許可するための RACF プロファイルを作成すると、ユーザーがどのビューにアクセスできるかを CICSPlex SM が決定します。ただし、CICSPlex SM は、そのユーザーがビュー内に表示されている CICS リソースへのアクセスを許可されているかどうかは判別できません。

このタスクについて

CICS シミュレーション・セキュリティー検査 をアクティブにすることで、CICSPlex SM プロファイルが提供するセキュリティーを強化することができます。シミュレーション・セキュリティーは、既存の RACF プロファイルを使用して、CICS リソース、CICS コマンド、またはその両方へのアクセスを制御します。これは、操作ビューとモニター・ビューでのみ使用可能です。この組み合わせのプロファイルを使用すると、CICSPlex SM プロファイルは、どのビューのセットにアクセスできるかを決定し、ユーザーの CICS リソース・プロファイルは、ビュー内のどのリソースにアクセスできるかを決定します。例えば、ユーザーにファイル・ビュー・コマンドおよび関連アクション・コマンドの発行を許可する CICSPlex SM プロファイルを作成して、それから CICS シミュレーション・セキュリティーに、ユーザーがアクセスを許可されるファイルを判別させることができます。

注**:**

1. CICSPlex SM および CICS セキュリティー・パラメーターがシミュレーション・セキュリティーにどのように影響を与える可能性があるかについての重要な情報は、 253 ページの『CICSPlex SM のセキュリティーのアクティブ化』 を参照してください。
2. シミュレーション・セキュリティーは、CICSPlex SM プロファイルのみを使用するよりもかなり多くの処理オーバーヘッドを必要とするため、パフォーマンスに悪影響があります。
3. CICSPlex SM CICS シミュレーション・セキュリティーには、CICS 代理セキュリティーのシミュレーションは含まれません。CICS 代理セキュリティー検査が必要である場合、CICS 領域を正しく保護する方法の手引きについては、252 ページの『CICS 代理セキュリティー検査に関する考慮事項』を参照してください。

手順

- シミュレーション・セキュリティー検査をアクティブ化または非アクティブ化するには、CSYSDEF ビュー (単一の CICS システムの場合) または CPLEXDEF ビュー (複数システムの場合) を使用します。 CICS リソース検査、または CICS コマンド検査、あるいはその両方を実行するかどうかを指示できます。 CICS リソース検査は、ビュー内に表示するリソースを制御します。 CICS コマンド検査は、ビュー内で使用できるコマンドを制御します。
- アクティブな CICS システムに対するシミュレーション・セキュリティー検査を一時的にアクティブ化または非アクティブ化するには、MAS ビューを使用します。

セキュリティー検査からのユーザーおよびリソースの免除

セキュリティー検査を必要としない特定の個人が存在する場合があります。CICSPlex SM プロファイルによって十分に保護されているため、セキュリティー検査に関わる必要がない特定の CICS リソースが存在する場合もあります。

これらの個人およびリソースを、CICSPlex SM CPSMXMP リソース・クラスを使用して、CICS シミュレーション・セキュリティー検査から免除することができます。免除は、CICS シミュレーション・セキュリティー検査のみをバイパスし、基本的な CICSPlex SM リソース検査はバイパスしません。

例えば、あるユーザーに、CICS コマンド CEMT INQ FILE を発行する RACF 権限がない場合、同等の CICSPlex SM コマンド **LOCFILE** の発行をユーザーに許可するプロファイルを免除クラス内に作成することで、そのユーザーが同じ結果を得られるようにすることができます。

免除プロファイルを作成するには、以下の手順を実行します。

1. 免除するリソースを決定して、それを **PERMIT** コマンドで指定します。 227 ページの『プロファイルでの CICSPlex SM リソース名の指定』で説明されているリソース名形式を使用します。
2. クラス名 CPSMXMP を指定します。この RACF クラスは、シミュレーション・セキュリティー検査の免除を制御します。
3. 必要なアクセスのタイプを指定します。
   - セキュリティー検査をバイパスしない場合は、ACCESS(NONE) を指定します。
   - INQUIRE レベルのコマンドのセキュリティー検査をバイパスする場合は、ACCESS(READ) を指定します。
   - INQUIRE レベル、SET レベル、および PERFORM レベルのコマンドのセキュリティー検査をバイパスする場合は、ACCESS(UPDATE) を指定します。
   - DISCARD レベルのコマンドを含むすべてのコマンドのセキュリティー検査をバイパスする場合は、ACCESS(ALTER) を指定します。
4. 免除を適用するユーザーまたはグループを指定します。

以下の例は、コンテキストが EYUPLX01 であり、スコープが EYUMAS1A であるときに、グループ EYUGRP2 を構成する個人が、MONITOR 関数内の TERMINAL タイプに関連付けられたすべてのビュー・コマンドとアクション・コマンドのセキュリティー検査をバイパスできるように免除プロファイルを定義する方法を示しています。

```
PERMIT  MONITOR.TERMINAL.EYUPLX01.EYUMAS1A  /* Resource name    */+
        CLASS(CPSMXMP)                      /* Class name       */+
        ACCESS(UPDATE)                      /* Access           */+
        ID(EYUGRP2)                         /* User or group    */+
                                            /*  granted access  */
```

CICS 代理セキュリティー検査に関する考慮事項

CICS 代理セキュリティー検査が必要な場合、CICS 代理セキュリティー検査の実行対象になるユーザー ID を決定して、RACF 定義内でその計画を立てる必要があります。

状況によっては、CICS は、ユーザーに対して代理セキュリティー検査を発行し、別のユーザーに代わって実行することが必要なアクションを要求します。この要求を処理できるようにするには、アクションを要求するユーザーが、他のユーザー ID の代理権限を持つ必要があります。

ただし、CICSPlex SM CICS シミュレーション・セキュリティーには、CICS 代理セキュリティーのシミュレーションは含まれません。

CICS 代理セキュリティーでは、要求を発行したユーザーではなく、MAS エージェント・ユーザー ID に対して検査が実行されます。MAS エージェント・ユーザー ID を判別するには、 217 ページの『MAS エージェント・ユーザー ID の判別』の手順に従ってください。

CICSPlex SM のセキュリティーのアクティブ化

CMAS および MAS のセキュリティーをアクティブ化するには、CICSPlex SM および CICS セキュリティー関連システム初期設定パラメーターを設定する必要があります。

手順

1. CMAS および MAS を開始するために使用する JCL に定義された EYUPARM データ・セットまたはメンバーに、CICSPlex SM パラメーター SEC を指定します。
2. MAS を開始するために使用する CICS システム初期設定パラメーターに、CICS パラメーター SEC= を指定します。

タスクの結果

これらのパラメーターをともに使用すると、どのセキュリティー検査を実行するかが決まります。可能なパラメーターの組み合わせは、表 25 で説明します。

表 *25.* セキュリティー検査を制御するパラメーター

CMAS (CICSPlex SM パラメーター**)**	**MAS (CICS** システム初期設定パラメーター**)**	説明
SEC(YES)	SEC=YES	CICSSYS ビューと CPLEXDEF ビューの設定に応じて、ビュー選択検査とシミュレーション・セキュリティー検査の両方が行われる場合があります。つまり、ユーザーが特定のビューを表示できるかどうかが CICSPlex SM によって判別された後、そのビューにどの情報を提供できるかがシミュレーション・セキュリティーによって判別されるということです。
SEC(YES)	SEC=NO	ビュー選択が発生します。シミュレーション・セキュリティー検査は CICSSYS ビューまたは CPLEXDEF ビューから要求された場合でも発生しません。つまり、ユーザーが指定されたビューを表示できるかどうかが CICSPlex SM によって判別された後、そのビューにどの情報を提供できるかを判別するためにシミュレーション・セキュリティー検査が実行されることはないということです。
SEC(NO)	SEC=YES	CICSPlex SM は、MAS が CMAS に接続することを許可しません。このようにすると、セキュリティーを要求する MAS が、セキュリティーを提供できない CMAS に接続することが妨げられます。

注: CICSPlex SM は、システム初期設定パラメーター
XCMD、XDB2、XDCT、XFCT、XHFS、XJCT、XPCT、XPPT、および XRES をどれも尊重します。すなわち、CICSPlex SM は、指定されたコマンドとリソースをセキュリティー検査に含めたり、そこから除外したりします。これらのシステム初期設定パラメーターには、MAS ごとに YES、NO、または CLASS NAME を指定できます。ただし CMAS の場合は、これらの各システム初期設定パラメーターに NO を指定しなければなりません。

CICSPlex SM の RACF プロファイルのリフレッシュ

CICSPlex SM は、RACF データベースでの不要な入出力を減らすために、RACF 情報のキャッシュ・コピーを使用します。RACF 定義を変更したら、場合によっては、キャッシュされた情報を CMAS がリフレッシュするように強制することが必要になります。

このタスクについて

これには、キャッシュ内の一般リソース・プロファイルおよびユーザー・プロファイルのリフレッシュが含まれています。

手順

キャッシュ内の一般リソース・プロファイルのリフレッシュ
CMAS は、RACF に、初期設定時にその一般リソース・プロファイルのキャッシュ・コピーを作成することを要求します。

- CMAS の初期設定中に、CPSMOBJ (GCPSMOBJ を含む) および CPSMXMP 内に RACF プロファイルのグローバル・コピーが作成されます。
- MAS の初期設定中に、MAS は、
  XCMD、XDB2、XDCT、XFCT、XJCT、XPCT、および XPPT の各システム初期設定パラメーターに指定された情報を使用して、使用中の CICS セキュリティー・クラスを判別します。この情報は、関連する RACF プロファイルのグローバル・コピーを作成するために使用されている CMAS に渡されます。

キャッシュ内の一般リソース・プロファイルをリフレッシュするには、以下の手順を実行します。

1. グローバル・コピーの対象となるプロファイルを識別するには、RACF コマンド **SETROPTS LIST** を発行します。 出力の「GLOBAL=YES RACLIST ONLY」セクションには、どの一般リソース・クラスがグローバルにコピーされたかが示されます。
2. これらのいずれかのクラスに (例えば、**RALTER**、**RDEFINE**、**RDELETE**、または **PERMIT** コマンドを使用して) 変更が加えられたときはいつでも、キャッシュをリフレッシュする必要があります。 以下の RACF コマンドを使用します。

   ```
   SETR RACLIST(classname) GENERIC(classname) REFRESH
   ```

   マルチ CMAS 環境では、このコマンドは、同じ RACF データベースを共用する MVS イメージ上でのみキャッシュをリフレッシュします。
3. 他の CMAS が、同じ RACF データベースを共用しない別の MVS イメージ上で稼働している場合、その別のイメージ上で **SETR** コマンドを繰り返します。

キャッシュ内のユーザー・プロファイルのリフレッシュ

CMAS は、セキュリティー検査の実行時に、ユーザー ID のセキュリティー情報をキャッシュ内に保管します。デフォルトでは、この情報がキャッシュ内に保持されるのは、**SECTIMEOUT** CICSPlex SM パラメーターで指定された期間、ユーザーが CMAS で非アクティブ状態であり、CMAS がユーザー ID タイムアウト・チェックを実行するまでです。

ユーザー ID に (例えば **CONNECT**、**REMOVE**、**ALTUSER**、または **DELUSER** コマンドを使用して) 変更が加えられた場合は、必ず、CMAS がそのユーザーのセキュリティー情報をキャッシュから破棄したときにその変更は可視になります。

4. タイムアウト処理が実行される前に CMAS がキャッシュからセキュリティー情報を破棄するように強制するには、以下のいずれかのアクションを使用します。
   - 単一の CMAS の場合、CMAS オブジェクトに対して **RESET USERID** アクションを使用します。
   - 複数の CMAS の場合、CMASLIST オブジェクトに対して **RESET USERID** アクションを使用します。

   どちらのアクションも、WUI および API から使用可能です。

   ユーザーが以前に CMAS を使用したことがあり、通常のタイムアウト処理が行われるのを待機したくない場合は、このアクションを実行します。
5. タイムアウトの対象となるユーザー ID に対して CMAS が即時検査を実行するように強制するには、以下のいずれかのアクションを使用します。
   - 単一の CMAS の場合、CMAS オブジェクトに対して **PURGE** アクションを使用します。
   - 複数の CMAS の場合、CMASLIST オブジェクトに対して **PURGE** アクションを使用します。

   どちらのアクションも、WUI および API から使用可能です。

CICSPlex SM セキュリティー検査シーケンス

ユーザーは、CICSplex を構成する 1 つ以上の CICS システムに関するデータを収集したり、そのシステムに対してアクションを実行したりする、単一の CICSPlex SM コマンドを発行できます。これらの CICS システムは、さまざまな MVS イメージ内に置くことができます

ユーザーが要求を発行すると、要求は、ターゲット CICS システムを管理する CMAS に送信されます。 258 ページの図 5 および 259 ページの図 6 は、ユーザーからの要求のセキュリティー要件を評価するために CICSPlex SM がたどるプロシージャーを示すフローチャートです。このプロシージャーの説明は、以下のとおりです。

1. CICSPlex SM は、CICSPlex SM ルールが要求の処理を許可するかどうかを判別します。
   - 許可されていない場合、CICSPlex SM は要求を終了し、エラー・メッセージを発行します。
2. CICSPlex SM は、CICS シミュレーション・セキュリティー検査を実行するかどうかを決定します。
   - 実行しない場合、処理は9 (256 ページ) から続行されます。

3. CICSPlex SM は、CICS シミュレーション・セキュリティー検査からユーザーを免除するかどうかを決定します。
   - 免除する場合、処理は9 から続行されます。
4. CICSPlex SM は、CICS シミュレーション・コマンド検査を実行するかどうかを決定します。
   - 実行しない場合、処理は6 から続行されます。
5. CICSPlex SM は、ユーザーにコマンドの処理を許可するかどうかを決定します。
   - 許可されていない場合、CICSPlex SM は要求を終了し、エラー・メッセージを発行します。
6. CICSPlex SM は、要求がアクションであるかどうか (情報の要求ではないかどうか) を判別します。
   - アクションでない場合、処理は9 から続行されます。
7. CICSPlex SM は、CICS シミュレーション・リソース検査を実行するかどうかを決定します。
   - 実行しない場合、処理は9 から続行されます。
8. CICSPlex SM は、ユーザーがリソースに関する情報へのアクセスを許可されているかどうかを判別します。
   - 許可されていない場合、CICSPlex SM は要求を終了し、エラー・メッセージを発行します。
9. CICSPlex SM は、アクションを実行するか、または情報を取得します。
10. CICSPlex SM は、要求がアクションであるかどうか (情報の要求ではないかどうか) を判別します。
    - アクションである場合、CICSPlex SM はアクションの結果を返します。
11. CICSPlex SM は、CICS シミュレーション・セキュリティー検査を実行するかどうかを決定します。
    - 実行しない場合、CICSPlex SM は要求された情報を適切なビューで返します。
12. CICSPlex SM は、CICS シミュレーション・セキュリティー検査からユーザーを免除するかどうかを決定します。
    - 免除する場合、CICSPlex SM は要求された情報を適切なビューで返します。
13. CICSPlex SM は、CICS シミュレーション・リソース検査を実行するかどうかを決定します。
    - 実行しない場合、CICSPlex SM は要求された情報を適切なビューで返します。
14. CICSPlex SM は、ユーザーがリソースに関する情報へのアクセスを許可されているかどうかを判別します。
    - 許可されていない場合、CICSPlex SM は要求された情報を適切なビューから除外します。
15. CICSPlex SM は、別のリソースの情報が要求されているかどうかを判別します。
    - 要求されている場合、処理は14 から続行されます。

- 要求されていない場合、CICSPlex SM は要求された情報を適切なビューで返します。

それ以上のセキュリティー検査は不要です。

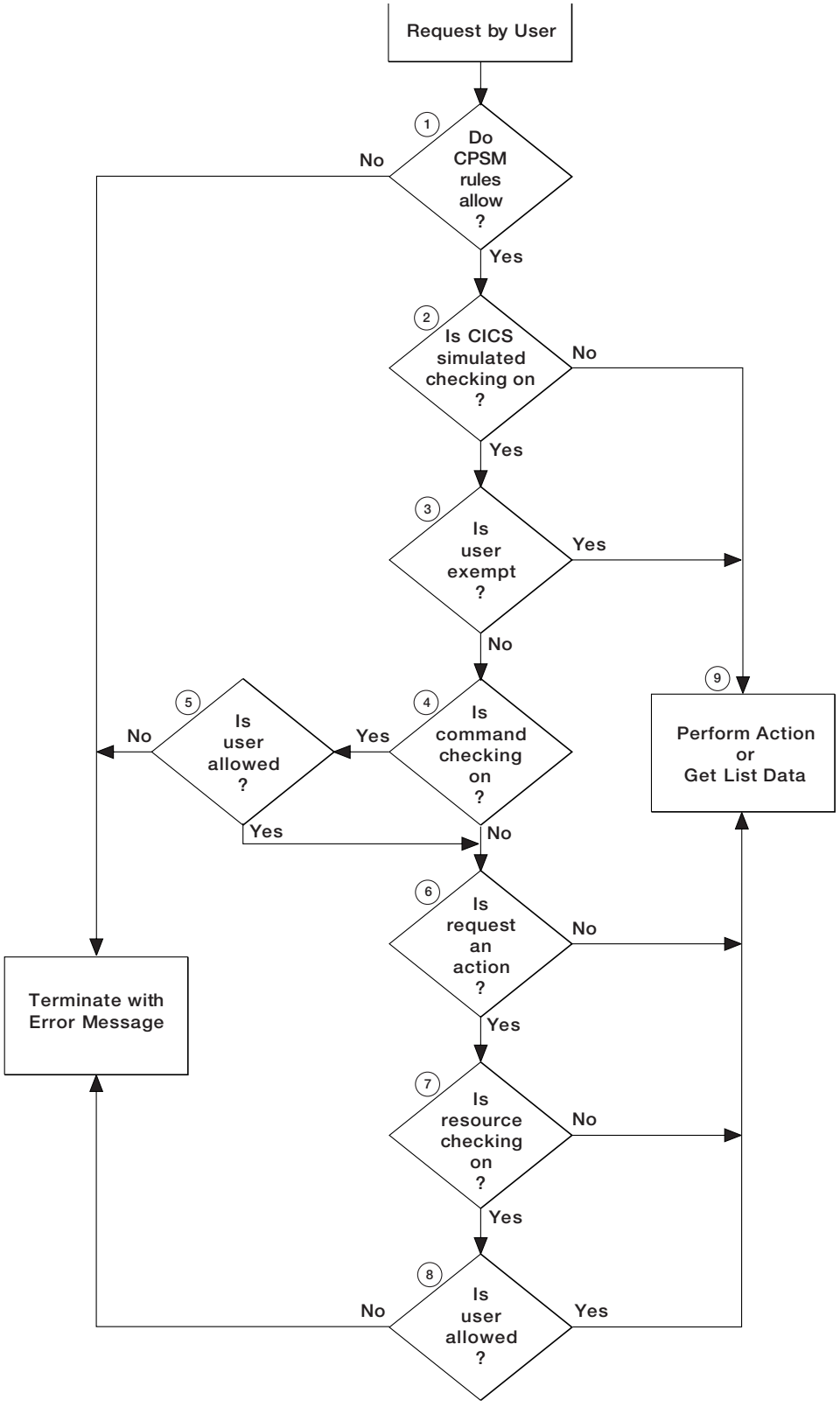


図 *5. CICSPlex SM* セキュリティー検査シーケンスのフローチャート - 第 *1* 部

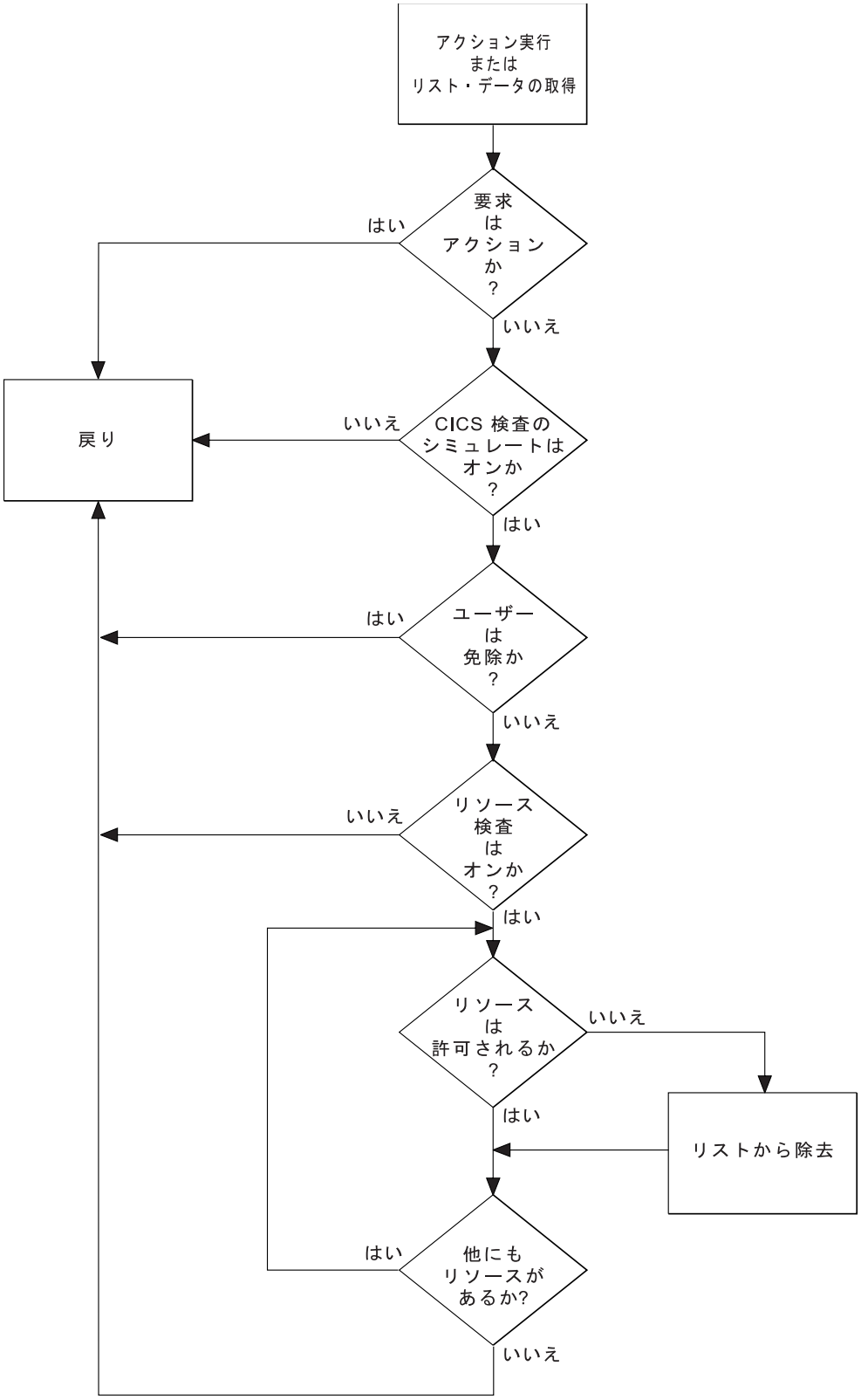


図 *6. CICSPlex SM* セキュリティー検査シーケンスのフローチャート - 第 *2* 部

ユーザー提供の外部セキュリティー・マネージャーの呼び出し

CICSPlex SM は、外部セキュリティー・マネージャー (ESM) へのインターフェースを提供します。外部セキュリティー・マネージャーとして、ユーザー提供のものや、リソース・アクセス管理機能 (RACF) を使用できます。

すべてのユーザー提供プログラムからの戻り時に、CICSPlex SM は必ず、1 次スペース変換モードで、すべてのアクセス・レジスターの元の内容が復元され、すべての汎用レジスター (戻りコードおよびリンケージ情報を提供するものを除く) が復元された状態で、制御を受け取る必要があります。変換モードについては、z/Architecture 解説書を参照してください。

注: この情報は主に、非 RACF ユーザーを対象としています。 RACF を使用したセキュリティー処理については、 211 ページの『CICSPlex SM セキュリティーの実装』を参照してください。

CICSPlex SM ESM インターフェースの概要

CICSPlex SM セキュリティーは、RACROUTE マクロを使用して MVS System Authorization Facility (SAF) インターフェースを取得し、許可要求を ESM に送付します。

通常、RACF が存在する場合は、それに対して MVS ルーターが制御を渡します。ただし、MVS ルーターのアクションは、ルーター出口を呼び出すことで変更できます。ルーター出口は、例えば、ユーザー提供またはベンダー提供の ESM に制御を渡すために使用できます。独自のセキュリティー・マネージャーを使用する場合は、MVS ルーター出口ルーチンを提供する必要があります。

CICSPlex SM が許可要求を送付するための RACROUTE マクロを発行する制御点については、 261 ページの『CICSPlex SM セキュリティーの制御点』で説明されています。

MVS ルーターの概要

System Authorization Facility (SAF) は、*MVS*ルーター と呼ばれるシステム・サービスを使用したセキュリティー処理の集中制御をインストール済み環境に実現します。 MVS ルーターは、リソース制御を提供および要求するすべての製品に共通のシステム・インターフェースを提供します。

リソース管理コンポーネントおよびサブシステム (CICS など) は、アクセス管理検査や許可関連検査などの処理における特定の意思決定機能の一部として、MVS ルーターを呼び出します。これらの機能は制御点 と呼ばれます。この単一 SAF インターフェースにより、製品間やシステム間で共用される共通制御機能の使用が促進されます。

システムで RACF が使用可能な場合、MVS ルーターは RACF ルーターに制御を渡し、RACF ルーターで適切な RACF 機能が呼び出されます。パラメーター情報と RACF ルーター・テーブル (ルーター呼び出しを RACF 機能に関連付ける) によって、適切な機能が決まります。ただし RACF ルーターを呼び出す前に、MVS ルーターは、インストール済み環境によって提供されるセキュリティー処理出口

(オプション) がインストールされている場合はそれを呼び出します。詳しくは、ユーザー提供の ESM に制御を渡すを参照してください。

RACF がインストールされていない場合でも、System Authorization Facility (SAF) と SAF ルーターはすべての MVS システムに存在します。SAF ルーターは RACF の一部ではありませんが、CICS など、多くのシステム・コンポーネントとプログラムが RACROUTE マクロおよび SAF を介して RACF を呼び出します。したがってインストール済み環境では、RACF パラメーター・リストを変更し、カスタマイズされたセキュリティー処理を SAF ルーター内で行うことができます。SAF ルーター出口のコーディング方法について詳しくは、「z/OS Security Server RACF メッセージおよびコード」を参照してください。

MVS ルーターについて詳しくは、『z/OS Security Server RACROUTE Macro Reference』の『System Authorization Facility (SAF)』および 『z/OS MVS Installation Exits』の『ICHRTX00 - MVS ルーター出口』を参照してください。

CICSPlex SM セキュリティーの制御点

すべての RACROUTE マクロは、CMAS から発行されます。 CICS シミュレーション・セキュリティー検査をサポートするために必要なマクロは、ターゲット MAS の接続先の CMAS から発行されます。

以下のリストは、ESM を呼び出すために CICSPlex SM が使用する RACROUTE マクロと、それらが発行される制御点を要約しています。

RACROUTE
説明されているマクロの「フロントエンド」であり、MVS ルーターを呼び出します。 RACF がシステム上に存在していない場合、RACROUTE は MVS ルーター出口経由で代替 ESM にルーティングできます。

RACROUTE REQUEST=VERIFY
ユーザー・サインオン時 (パラメーター ENVIR=CREATE を指定)、およびユーザー・サインオフ時 (パラメーター ENVIR=DELETE を指定) に、CMAS に対して発行されます。 ISPF エンド・ユーザー・インターフェース要求の場合、サインオン呼び出しは、指定されたコンテキストをサポートする CMAS でウィンドウを開くときに行われます。サインオフ呼び出しは、ウィンドウを閉じるときに行われます。このマクロは、アクセス制御環境エレメント (ACEE) を作成または破棄します。それは、以下の CICSPlex SM CMAS 制御点で発行されます。

CMAS への ISPF エンド・ユーザー・インターフェース・ユーザーの接続
API CONNECT スレッドの作成
単一システム・イメージ・コマンドのルーティング
CMAS からの ISPF エンド・ユーザー・インターフェース・ユーザーの切断
API DISCONNECT スレッドの終了

RACROUTE REQUEST=FASTAUTH
ACEE によって識別されたユーザーの代わりに、リソース検査時に発行されます。これはハイパフォーマンス形式の REQUEST=AUTH であり、ストレージ内リソース・プロファイルを使用し、以下の CICSPlex SM CMAS 制御点で発行されます。

CICS シミュレーション・セキュリティー検査

ビュー選択/API セキュリティー

RACROUTE REQUEST=AUTH
これは、より高度なパス長形式のリソース検査であり、PLEXMGR セキュリティー検査時に発行されます。これは、REQUEST=FASTAUTH の後にロギングおよび監査を実行するために呼び出すこともできます。

RACROUTE REQUEST=LIST
REQUEST=FASTAUTH が必要とするストレージ内プロファイル・リストを作成および削除するために発行されます (各リソース・クラスに 1 つの REQUEST=LIST マクロが必要です)。それは、以下の CICSPlex SM CMAS 制御点で発行されます。
MAS に対して CICSPlex SM セキュリティーが初期設定されているとき。
CMAS または CMASD セキュリティー・アクション・コマンド (SEC) が発行されたとき。

これらのマクロについて詳しくは、「z/OS Security Server RACF マクロおよびインターフェース」を参照してください。

サンプル・タスク: セキュリティー

以下の情報は、標準的なセキュリティー・セットアップ・タスクの例を示しています。これは、独自のモデルとして使用できます。

以下に示すのは、すべての RACF の例に適用されるいくつかの一般的なポイントです。

- これらのタスク例に示されている各 RACF コマンドは、CICSPlex SM 構成内のすべての RACF データベースに対して 1 回発行する必要があります。そのため、2 つの未接続の RACF データベースがある場合 (1 つは MVS1 上で、もう 1 つは MVS2 上)、各 RACF コマンドは 2 回 (各システム上で 1 回ずつ) 発行する必要があります。
- すべての RACF コマンド例で、小文字のストリングは、ユーザーの企業に適した値に置き換える必要があります。例えば、ストリング `admin_user` は、関連する CICSPlex SM リソースのセキュリティーを担当する管理者のユーザー ID に置き換える必要があります。
- すべての RACF タスク例は、RACF の拡張総称命名 (**) を使用します。企業でこの規則を使用しない場合、同等のプロファイルの作成については、RACF の資料を参照してください。
- これらの例では、運用や管理といった RACF グループが使用されています。そのようなグループを作成することをお勧めします。

最初の例は、すべての CICSPlex SM 機能およびリソースを保護するいくつかの RACF プロファイルの作成について説明しています。次のステップは、特定のリソースの特定のユーザーに対してアクセス権限を選択的に許可することです。

例: すべての CICSPlex SM リソースの保護

すべての CICSPlex SM リソースを保護するための RACF プロファイルを作成するには、以下を実行します。

1. CPSMOBJ クラスがアクティブであること、そして総称プロファイルを定義可能であることを確認します。

```
SETROPTS CLASSACT(CPSMOBJ) GENERIC(CPSMOBJ)
```

2. RACF プロファイルを作成して、CICSPlex SM のすべての関数のすべてのビュー・コマンドとアクション・コマンドを保護します。

```
RDEF CPSMOBJ ** UACC(NONE) OWNER(admin_group) NOTIFY(admin_user)
```

   このコマンドは、RACF がすべての CPSMOBJ リソース・エンティティー名と一致すると見なすプロファイル (**) を定義していますが、実質的にはすべての CICSPlex SM リソースを保護することになります。また、何らかの違反があれば admin_user に通知されることも指定します。

3. 次のステップは、ステップ 2 と非常によく似ており、構成内の各 CICSplex に対して 1 つの RACF プロファイルを定義します。各プロファイルは、その CICSplex のすべての CICSPlex SM 関数およびリソースを保護します。そのようにする目的は、CICSplex 固有のリソースにアクセス権限を付与する場合の柔軟性をさらに向上させることにあります。この例では、2 つの CICSplex があるので、次の 2 つの RACF プロファイルを作成します。

```
RDEF CPSMOBJ *.*.PLXPROD1.* UACC(NONE) OWNER(admin_group) +
     NOTIFY(admin_user)
RDEF CPSMOBJ *.*.PLXPROD2.* UACC(NONE) OWNER(admin_group) +
     NOTIFY(admin_user)
```

   ステップ 2 を複数の CICSplex 固有プロファイルで置き換えることはできません。そのようなプロファイルは後から作成する CICSplex を必ずしも保護するものとはならず、コンテキストが CICSplex ではなく CMAS である CICSPlex SM 関数を保護することもできません。例えば、ステップ 2 を実行しなかった場合も、CONFIG ビューは無保護になります。

4. ステップ 3 では、CICSPlex SM のすべての関数とリソースを CICSplex レベルで保護します。このステップでは、CICSPlex SM の CONFIG 定義関数と TOPOLOGY 定義関数へのアクセスを制御するプロファイルを定義します。これにより、管理者などの「特殊」ユーザーには、必要なアクセス権限を選択的に許可できます。(これらの 2 つの関数に対する更新権限を持つユーザーは誰でも CICSplex 構成を変更できるので、アクセス権限を制限する必要があります。)

```
RDEF CPSMOBJ CONFIG.DEF.** UACC(NONE) OWNER(admin_group)
RDEF CPSMOBJ TOPOLOGY.DEF.** UACC(NONE) OWNER(admin_group)
```

これで CICSPlex SM の関数とリソースへのアクセスの制御が完了したため、特定のユーザーまたはユーザー・グループへのアクセス権限の付与を開始できます。

例: CICSPlex SM オペレーターへの適切な権限の付与

CICSPlex SM オペレーターは、少なくともすべての操作ビューに対するアクセス権限が必要です。この例では、CICSPlex SM オペレーターに、すべての操作ビューに対する更新権限と、モニタリング・ビューに対する読み取り権限を付与する方法を示します。これによりオペレーターには、モニター・データの表示は許可されますが、モニター定義の作成および変更は許可されません。

1. CICSPlex SM オペレーターに、OPERATE ビューに対する更新権限を付与します。

```
RDEF CPSMOBJ OPERATE.** OWNER(admin_group) UACC(NONE)
PE OPERATE.** CLASS(CPSMOBJ) ID(ops_group) A(UPDATE)
```

2. CICSPlex SM オペレーターに、MONITOR ビューに対する読み取り権限を付与します。

```
RDEF CPSMOBJ MONITOR.** UACC(NONE) OWNER(admin_group)
PE MONITOR.** CLASS(CPSMOBJ) ID(ops_group) A(READ)
```

どちらのステップでも、まずリソースを保護するための RACF プロファイルを作成し、次にグループ ops_group 内のユーザーにアクセス権限を付与することがわかります。

例: ユーザーへの MVS システム A 上のすべてのトランザクションに対する読み取り権限の付与

この例では、ユーザー PAYUSR1 に、MVS システム A 上の CICS システムで実行するすべてのトランザクションに対する読み取り権限を付与する方法を示します。

この例では、すべて CICSplex PLXPROD1 に属する 3 つの CICS システム (CICSAA01、CICSAA02、および CICSAA03) があります。

1. 適切な RACF プロファイルを次のように定義します。

```
RDEF CPSMOBJ OPERATE.TRAN.PLXPROD1.CICSAA0* UACC(NONE) +
     OWNER(admin_group)
```

2. ユーザー PAYUSR1 に、MVS システム A 上のすべてのトランザクションに対する読み取り権限を付与します。

```
PE OPERATE.TRAN.PLXPROD1.CICSAA0* CLASS(CPSMOBJ) I(PAYUSR1) A(READ)
```

第 6 章 プラットフォームおよびアプリケーションのセキュリティー

プラットフォーム上にデプロイされているアプリケーションのリソースを保護するには、CICSplex でのプラットフォームとアプリケーションに対応する CICSPlex SM 用の RACF セキュリティー・プロファイルを作成します。

プラットフォームおよびアプリケーションのセキュリティーは、その他の CICSPlex SM コンポーネントのセキュリティーと同様の方法でセットアップします。セキュリティー・プロファイル内でビュー (およびそれらに関連するアクション・コマンド) からなる特定のセットを識別することにより、そのセットへのアクセスを制御します。これらのセキュリティー・プロファイルを使用すると、プラットフォームやアプリケーションのインストール、有効化/無効化、使用可能化/使用不可化、照会、または破棄を行う権限をユーザーに与えることができます。これにより、無許可ユーザーはこれらのリソースを作成したり管理したりできなくなります。

あるユーザーにプラットフォームまたはアプリケーションに対してアクションを実行する権限を付与すると、そのプラットフォームまたはアプリケーションに対して動的に生成されるリソースについても、そのユーザーが同じアクションを実行できる権限を付与することになります。例えば、あるアプリケーションを有効にする権限を持つユーザーは、CICSplex 内のすべてのプラットフォームの CICS 領域にインストールされたアプリケーション用 CICS バンドルを有効にする権限も持ちます。アプリケーションまたはプラットフォームを介した CICS バンドルの操作時には、CICS コマンドとリソース・セキュリティー検査、および CICSPlex SM でシミュレートされる CICS セキュリティー検査は行われません。

プラットフォームとそこにデプロイされたアプリケーションを保護するには、以下に挙げる機能とタイプを組み合わせてセキュリティー・プロファイルを設定することができます。

CLOUD.DEF.*context*

このセキュリティー・プロファイルは、プラットフォームとアプリケーションの定義を含んでいるリソース・テーブル PLATDEF および APPLDEF を対象とします。 *context* は、このセキュリティー・プロファイルの対象となる CICSplex の固有名または総称名です。

このセキュリティー・プロファイルに対する UPDATE アクセス権限を持つユーザーは、CICSPlex SM データ・リポジトリー内でプラットフォームとアプリケーションの定義の作成、更新、および除去を行うことができます。READ アクセス権限を持つユーザーは、CICSPlex SM データ・リポジトリー内でこれらの定義を表示できます。

CLOUD.PLATFORM.*context*

このセキュリティー・プロファイルは、PLATDEF リソースのインストールおよび PLATFORM リソースに対する操作を対象とします。また、ユーザ

© Copyright IBM Corp. 1989, 2018

ーに対して管理パーツ (MGMTPART リソース) の表示も許可します。 *context* は、このセキュリティー・プロファイルの対象となる CICSplex の固有名または総称名です。

このセキュリティー・プロファイルに対する ALTER アクセス権限を持つユーザーは、CICSplex でのプラットフォームのインストールおよび破棄を行うことができます。 (ユーザーがプラットフォームをインストールするには、PLATDEF リソースを対象とする CLOUD.DEF プロファイルに対する READ アクセス権限も必要です。) UPDATE アクセス権限を持つユーザーはプラットフォームを有効/無効にすることができます。また、UPDATE アクセス権限を持つユーザーは、CICS 領域をプラットフォームの領域タイプに追加したり、CICS 領域をプラットフォームの領域タイプから除去したりすることもできます。 READ アクセス権限を持つユーザーは PLATFORM リソースと MGMTPART リソースを表示できます。これらの権限は、CICSplex に存在するすべてのプラットフォームに適用されます。

CLOUD.APPLICATION.*context*

このセキュリティー・プロファイルは、APPLDEF リソースのインストールおよび APPLCTN リソースに対する操作を扱います。 *context* は、このセキュリティー・プロファイルの対象となる CICSplex の固有名または総称名です。

このセキュリティー・プロファイルに対する ALTER アクセス権限を持つユーザーは、CICSplex でのアプリケーションのインストールおよび破棄を行うことができます。 (ユーザーがアプリケーションをインストールするには、APPLDEF リソースを対象とする CLOUD.DEF プロファイルに対する READ アクセス権限も必要です。) UPDATE アクセス権限を持つユーザーは、アプリケーションを有効/無効にしたり、使用可能/使用不可にしたりできます。 READ アクセス権限を持つユーザーは APPLCTN リソースを表示できます。これらの権限は、CICSplex に存在するすべてのプラットフォーム内のすべてのアプリケーションに適用されます。特定のアプリケーションに関して異なるセキュリティー権限が必要な場合は、別の CICSplex を使用してアプリケーションのデプロイ場所のプラットフォームをホストしてください。

注: これらのセキュリティー・プロファイルは、保守ポイント CMAS でのみ検査されます。セキュリティー検査は、保守ポイント CMAS の EYULOG 内のメッセージ EYUCR0009I によって報告されます。違反についてメッセージ EYUCR0009I を受け取るには、CICSPlex SM システム・パラメーター (EYUPARM) **SECLOGMSG** を YES に設定する必要があります。 **SECLOGMSG** についての詳細は、CICSPlex SM システム・パラメーターを参照してください。

CLOUD セキュリティー・プロファイルはプラットフォームやアプリケーション用に動的に生成されるリソースへのアクションを対象としますが、それらがインストールされた CICSplex および CICS 領域にある個別リソースに対してユーザーは以下のような限定的なアクションを直接実行できます。

- プラットフォームの一部である CICS 領域の CICSPlex SM トポロジー定義 (CSYSDEF、つまり CICS システム定義) を変更できます。領域タイプ・レベルで指定されている属性値はロックされていて変更できませんが、他の属性値は変更できます。

- プラットフォームまたはアプリケーションのインストール時に動的に作成された BUNDLE リソースを使用可能/使用不可にしたり、有効/無効にしたり、照会したりできます。プラットフォームまたはアプリケーションのインストール時に作成されたバンドルを直接、個別に破棄することはできません。
- CICS バンドル内部で定義され、プラットフォームまたはアプリケーションのインストール時に動的に作成されたリソース (例えば PROGRAM リソース) を照会できます。これらのリソースがプラットフォームまたはアプリケーションのインストール時に作成された場合は、リソースを直接的に有効化、無効化、および破棄することはできません。

プラットフォームまたはアプリケーションのインストール時に作成された、プラットフォームの一部である CICS 領域、または個々の CICS バンドル、または CICS バンドル内に定義されたリソースに対してアクションを直接実行するときに、 CICS コマンドとリソース・セキュリティー検査、および CICSPlex SM でシミュレートされる CICS セキュリティー検査が適用されます。

- TOPOLOGY.DEF.*context* セキュリティー・プロファイルは、プラットフォームの一部である個々の CICS 領域の CICSPlex SM トポロジー定義で行われるアクションを対象とします。 *context* は、このセキュリティー・プロファイルの対象となる CICSplex の固有名または総称名です。 UPDATE アクセス権限を持つユーザーは、プラットフォームの一部である CICS 領域の CSYSDEF を変更できますが、プラットフォームそのものでロックされている属性値は変更できません。
- プラットフォームまたはアプリケーションのインストール時に作成される CICS バンドルには、$ 文字で始まる固有の名前が生成されます。この方法で動的に作成された個々の CICS バンドルに対するアクションのセキュリティーを設定するには、BUNDLE リソース・タイプおよびリソース名 `$*` を指定してセキュリティー・プロファイルをセットアップします。 `BUNDLE.$*` に対する UPDATE アクセス権限を持つユーザーは、プラットフォームやアプリケーションのために作成された BUNDLE リソースを使用可能/使用不可にしたり、有効/無効にしたりできます。また、READ アクセス権限を持つユーザーは、それらの BUNDLE リソースを照会することができます。

さまざまなプラットフォームにデプロイされるアプリケーションについて、セキュリティー対策を個々の PROGRAM リソースに適用する場合、アプリケーション・エントリー・ポイントとして宣言されているプログラムは保護しますが、そのアプリケーション内の他のプログラムは保護しないようにしてください。あるプラットフォーム上にデプロイされるアプリケーションの一部を構成するプログラムについて指定するセキュリティー設定値は、パブリックなプログラムとプライベートなプログラムの両方に適用されます。アプリケーションのバージョンは考慮されません。アプリケーション・エントリー・ポイントとして宣言されるプログラムには、使用環境内で固有の PROGRAM リソース名が必要です。しかし、そのアプリケーションの下位レベルで実行されるプログラムを保護すると、別のアプリケーションで同じ名前のプログラムが実行されている可能性があり、その場合、予期しない結果になる可能性があります。この場合、アプリケーション・エントリー・ポイントとして宣言されているプログラムについてはアクセス権限がユーザーに付与されている一方、そのプログラム名の別のインスタンスによるセキュリティー設定が有効になっているために、アプリケーションの下位レベルで実行されるプログラムにはアクセスする許可がないということがあり得ます。アプリケーション・エントリ

ー・ポイント・プログラムとして宣言されているプログラムに適用するセキュリティー対策としては、アプリケーション全体に適用されるものを考慮してください。

以前の CICS リリースで CICS バンドルを使用していた場合は、これらのバンドルに関してユーザーに与えたセキュリティー権限を確認してください。 CICS バンドルのセキュリティーのセットアップ方法によっては、個々の CICS バンドルに対するアクションの実行権限を持つユーザーが、バンドル・インストールの過程で動的に作成されるリソースに対してアクションを実行できるようになっている可能性があります。 BUNDLE リソースに対する権限のレベルが引き続き適切であることを確認してください。

250 ページの表 24に、以下の対象に対して実行されるアクションに適用されるセキュリティー検査を要約します。すなわち、1) プラットフォーム、2) アプリケーション、3) プラットフォームまたはアプリケーションのインストール時に動的に作成された個々の CICS バンドル、あるいは 4) プラットフォームまたはアプリケーション用の CICS バンドル内で定義されたリソースです。

表 *26.* プラットフォーム、アプリケーション、および生成される *CICS* バンドルに対する操作のセキュリティー検査

操作	プラットフォーム **(CICS** バンドルを含む**)**	アプリケーション **(**バンドルを含む**)**	動的に作成された **CICS** バンドル	動的に作成された **CICS** バンドルで定義されるリソース
定義	CLOUD.DEF プロファイル (UPDATE、または定義を表示するための READ)。また、TOPOLOGY.DEF プロファイル (プラットフォームのインストール後に個々の CICS 領域 CSYSDEF を変更するための UPDATE)	CLOUD.DEF プロファイル (UPDATE、または定義を表示するための READ)	リソース定義を個別に管理できません	リソース定義を個別に管理できません
インストール	CLOUD.PLATFORM プロファイル (ALTER) および CLOUD.DEF プロファイル (READ)	CLOUD.APPLICATION プロファイル (ALTER) および CLOUD.DEF プロファイル (READ)	個別にインストールすることはできません	個別にインストールすることはできません
有効化または無効化	CLOUD.PLATFORM プロファイル (UPDATE)	CLOUD.APPLICATION プロファイル (UPDATE)	CICS コマンドとリソースのセキュリティー、および CICSPlex SM でシミュレートされる CICS セキュリティーの検査 (BUNDLE.$* プロファイルを使用)	個別に有効化/無効化することはできません
使用可能または使用不可にする	適用外	CLOUD.APPLICATION プロファイル (UPDATE)	CICS コマンドとリソースのセキュリティー、および CICSPlex SM でシミュレートされる CICS セキュリティーの検査 (BUNDLE.$* プロファイルを使用)	個別に使用可能または使用不可にすることはできません。

表 26. プラットフォーム、アプリケーション、および生成される *CICS* バンドルに対する操作のセキュリティー検査 (続き)

操作	プラットフォーム **(CICS** バンドルを含む**)**	アプリケーション **(**バンドルを含む**)**	動的に作成された **CICS** バンドル	動的に作成された **CICS** バンドルで定義されるリソース
Inquire (問い合わせ)	CLOUD.PLATFORM プロファイル (READ) - 管理パーツの表示も許可します	CLOUD.APPLICATION プロファイル (READ)	CICS コマンドとリソースのセキュリティー、および CICSPlex SM でシミュレートされる CICS セキュリティーの検査 (BUNDLE.$* プロファイルを使用)	CICS コマンドとリソースのセキュリティー、および CICSPlex SM でシミュレートされる CICS セキュリティーの検査
破棄	CLOUD.PLATFORM プロファイル (ALTER)	CLOUD.APPLICATION プロファイル (ALTER)	個別に破棄できません	個別に破棄することはできません

CICSPlex SM 用のセキュリティーのセットアップ、およびセキュリティー・プロファイルの作成について詳しくは、Implementing CICSPlex SM securityを参照してください。

第 7 章 相互通信のセキュリティー

以下のトピックでは、LU6.2 または LU6.1 を使用したシステム間連絡 (ISC) 環境、複数領域操作 (MRO) 環境、および IP 相互接続 (IPIC) 環境で、セキュリティーの計画と実装を行う方法を説明します。

相互通信セキュリティーの概要

このトピックでは、CICS システムが相互接続している場合、または他の互換システムと接続している場合に、セキュリティーがどのように機能するかについて概説します。

相互通信セキュリティーの概要

単一の CICS システムでは、セキュリティーを利用して、システムのうち、端末ユーザーが作業に必要な部分にしかアクセスできないようにします。

相互接続されたシステムでは、同じ基本原則が適用されますが、接続、セッション、およびパートナーの定義も含まれるようになりました。さらに、ある CICS システムのユーザーがトランザクションを開始すると、別の CICS システム内のリソースにアクセスできるという事実も考慮に入れる必要があります。

このトピックでは、読者が単一の CICS システムに対するセキュリティーのセットアップに精通していることを前提としています。

特に、次の概念について理解している必要があります。

- ユーザー・サインオン。 75 ページの『サインオン・プロセス』を参照してください。
- ユーザー・セキュリティーとトランザクション・セキュリティーとの関係により、特定のユーザーに呼び出しが許可されるトランザクションが決定される方法。(72 ページの『CICS ユーザーの検証』および 94 ページの『トランザクション・セキュリティー』を参照してください。)
- リソース・セキュリティーにより、ユーザーにアクセスが許可される他のリソースが決定される方法。 100 ページの『リソース・セキュリティー』を参照してください。

CICS システムの相互接続グループは、ユーザー・プロファイルまたはグループ・プロファイルを複数回定義することが必要な場合がある点で、単一の CICS システムとは異なります。(これらのプロファイルの定義については、 15 ページの『RACF ユーザー・プロファイル』および 23 ページの『RACF グループ・プロファイル』を参照してください。) つまり、別々の RACF データベースを使用している各 CICS システム内と、ユーザーがトランザクションを接続したりリソースにアクセスしたりする可能性がある CICS システム内で、これらのプロファイルを定義することが必要な場合があります。これらのプロファイルを計画するときは、機能シップ、トランザクション・ルーティング、非同期処理、分散プログラム・リンク、分散トランザクション処理、または外部呼び出しインターフェース (EXCI) をユーザ

© Copyright IBM Corp. 1989, 2018

ーが開始する可能性がある、あらゆるケースを検討する必要があります。相互通信の方法については、分散トランザクション処理の概要を参照してください。

相互通信セキュリティーの計画

CICS システムの相互通信セキュリティーは、他のシステムに送信される要求よりも、CICS リソースへのアクセスのための着信要求に関係があります。

着信要求のセキュリティー問題は、特定のリモート・システムで特定のユーザーが CICS システムのリソースにアクセスしようとしているときに発生します。そのアクセスは許可されていますか、または拒否するべきでしょうか。

以下のセクションでは、セキュリティー検査を適用できる着信要求の処理のポイントを説明しています。

相互通信バインド時のセキュリティー

最初の要件は、2 つのシステム間で確立されるセッションに対するものです。これはもちろん、どの要求でも生じるというわけではありません。セッションはいったん確立されると、通常は長時間継続します。さらに、セッションを確立する接続要求は、環境に応じて、リモート・システムまたは CICS システムのいずれかが発行できます。ただし、セッションの確立により、システムには機密漏れの可能性が存在し始めるようになります。

セキュリティーの懸案は、無許可のリモート・システムが CICS システムに接続しないようにすることです。つまり、リモート・システムが本当にその主張どおりのシステムであることを確認することです。このレベルのセキュリティーは、バインド時セキュリティーと呼ばれます。(SNA を介した ISC 接続の場合、それはシステム・ネットワーク体系 **(SNA)** セッション・セキュリティーとも呼ばれます。) それは、セッションを確立するための要求がリモート・システムとの間で送受信されるときに適用できます。

注**:** バインドという用語は、以下のすべてを指すために使用します。

- システム間で SNA セッションを確立するために使用される **SNA BIND** コマンド
- システム間の IPIC 接続を確立するために使用される **CICS** 接続要求
- CICS 領域間通信の MRO セッションを確立するために使用される CICS 接続要求

APPC (LU6.2)、IPIC、複数領域操作 (MRO) の各リンクに対してバインド時セキュリティーを指定できますが、LU6.1 リンクの場合は指定できません。システムでのバインド時セキュリティーの定義については、ご使用の環境に応じて、 275 ページの『LU6.2 でのバインド時のセキュリティー』、 326 ページの『IPIC バインド時のセキュリティー』、または 336 ページの『MRO でのバインド時のセキュリティー』を参照してください。

相互通信リンク・セキュリティー

システム間の各リンクには、ユーザー ID によって定義された権限が付与されます。

重要な点として、それ自体がどのトランザクションおよびどのリソースへのアクセスも許可されていないリンクを介して、ユーザーはそれらにアクセスできないことに注意してください。つまり、各ユーザーの許可は、リンクの権限全体のサブセットです。

トランザクションおよびリソースへのリモート・システムのアクセスを制限するには、リンク・セキュリティーを使用します。リンク・セキュリティーは、リモート・システム全体に割り当てる単一のユーザー・プロファイルと関係があります。単一システム環境内のユーザー・セキュリティーと同じように、リンク・セキュリティーは以下のものを制御します。

トランザクション・セキュリティー
: これは、特定のトランザクションを接続するためのリンクの権限を制御します。

リソース・セキュリティー
: これは、特定のリソースにアクセスするためのリンクの権限を制御します。これは、トランザクション定義に RESSEC(YES) が指定されている、リモート・システムからのいずれかのセッションで実行するトランザクションに適用されます。

コマンド・セキュリティー
: これは、接続されたトランザクションが発行するコマンドに対するリンクの権限を制御します。これは、トランザクション定義に CMDSEC(YES) が指定されている、リモート・システムからのいずれかのセッションで実行するトランザクションに適用されます。

代理ユーザー・セキュリティー
: これは、新規ユーザー ID を使用してトランザクションの START を実行したり、関連するユーザー ID を使用してリソースをインストールしたりするためのリンクの権限を制御します。

詳しくは、 274 ページの『相互通信のためのトランザクション、リソース、コマンド、および代理ユーザーのセキュリティー』を参照してください。

相互通信のユーザー・セキュリティー

リンクに対してセットアップするセキュリティー・プロファイルに加え、システム内のトランザクション、コマンド、およびリソースへの各リモート・ユーザーのアクセスをさらに制限することができます。

SNA を介した ISC リンクおよび MRO リンクの場合は、CONNECTION 定義で ATTACHSEC パラメーターを指定します。 IPIC リンクの場合、IPCONN リソース定義で USERAUTH パラメーターを指定します。

ユーザー・セキュリティーは、リンク・セキュリティーと同様に、トランザクション、リソース、コマンド、および代理セキュリティーを区別します。ユーザー・セキュリティーによって、ユーザーの権限をリンクの権限を超えて引き上げることはできません。詳しくは、 274 ページの『相互通信のためのトランザクション、リソース、コマンド、および代理ユーザーのセキュリティー』を参照してください。

システムでのユーザー・セキュリティーの定義については、ご使用の環境に応じて、 282 ページの『LU6.2 でのユーザー・セキュリティー』、 330 ページの

『IPIC ユーザー・セキュリティー』、または 341 ページの『MRO でのユーザー・セキュリティー』を参照してください。

LU6.1 リンクにはユーザー・セキュリティーを指定できません。 LU6.1 の場合、ユーザー・セキュリティーはリンク・セキュリティーと同じであると見なされます。

相互通信のためのトランザクション、リソース、コマンド、および代理ユーザーのセキュリティー

システムのセキュリティーを定義する最後のステップは、アクセス・パラメーターが、リンクおよび個々のリモート・ユーザーのために、トランザクション、リソース、コマンド、および代理ユーザーに定義したプロファイルと一致していることを確認することです。

単一システム環境でのこれらのセキュリティー・レベルの定義については、トランザクション・セキュリティー、リソース定義のセキュリティー、およびCICS command securityを参照してください。

リソースおよびコマンドは、トランザクション定義でセキュリティー保護を明示的に要求しない限り、非セキュアです。

相互通信セキュリティー・レベルの要約

表 27 は、バインド時、トランザクション、リソース、およびコマンドの各セキュリティーを示しており、さらに CICS が LU6.1、LU6.2、IPIC、および MRO の各プロトコルの下でこれらのセキュリティー・レベルを強制する方法を示しています。それは、2 つのレベルの許可 (ユーザーおよびリンク) が、3 つのセキュリティー・レベルにおいてどのような関係があるのかを示しています。

これらの環境の選択の手引きについては、相互通信方式を参照してください。

表 27 は、相互通信セキュリティーの要約を示しています。

表 27. システム間セキュリティーおよび領域間セキュリティーの要約

セキュリティー・レベル	セキュリティー検査	**LU タイプ 6.1**	**LU タイプ 6.2**	**IPIC**	**MRO**
バインド時のセキュリティー (BIND の受信時)	BIND 要求が受け入れる必要があるか。	検査なし	RACF からのセッション鍵	パートナーからの SSL クライアント証明書	RACF FACILITY クラスの DFHAPPL プロファイル
バインド時のセキュリティー (BIND の送信時)	リモート・システムが正しいものか。	検査なし	RACF からのセッション鍵	パートナーへの SSL クライアント証明書	RACF FACILITY クラスの DFHAPPL プロファイル

表 27. システム間セキュリティーおよび領域間セキュリティーの要約 (続き)

セキュリティー・レベル	セキュリティー検査	LU タイプ 6.1	LU タイプ 6.2	IPIC	MRO
リンク・セキュリティー	リンクに、トランザクションを接続する権限があるか。	リンク権限は、セッションがバインドされた直後に、CONNECTION 定義の SECURITYNAME 属性または SESSIONS 定義の USERID 属性に指定されたユーザー ID のサインオンによって設定されます。		リンク権限は、接続の確立の直後に、IPCONN LINKAUTH 属性に従って決定されたユーザー ID のサインオンによって設定されます。	リンク権限は、セッションがバインドされた直後に、SESSIONS 定義の USERID 属性に指定されたユーザー ID のサインオンによって設定されます。
トランザクション・セキュリティー	リモート・ユーザーに、このシステムにアクセスする権限があるか。	検査なし	リモート・ユーザーの権限は、サインオン時に設定されます。		
トランザクション・セキュリティー	リモート・ユーザーに、トランザクションを接続する権限があるか。	リンク権限	リモート・ユーザーの権限は、サインオンによって、この接続要求時 (場合によっては、同じユーザーからの前の接続要求時) に設定されます。		
リソース、コマンド、および代理セキュリティー	セッションに、トランザクションが使用する他のリソースにアクセスする権限があるか。	リンク権限は、セッションがバインドされた直後に、CONNECTION 定義の SECURITYNAME 属性または SESSIONS 定義の USERID 属性に指定されたユーザー ID のサインオンによって設定されます。		リンク権限は、接続の確立の直後に、IPCONN LINKAUTH 属性に従って決定されたユーザー ID のサインオンによって設定されます。	リンク権限は、セッションがバインドされた直後に、SESSIONS 定義の USERID 属性に指定されたユーザー ID のサインオンによって設定されます。
リソース、コマンド、および代理セキュリティー	リモート・ユーザーに、トランザクションが使用する他のリソースにアクセスする権限があるか。	リンク権限	リモート・ユーザーの権限は、サインオンによって、この接続要求時 (場合によっては、同じユーザーからの前の接続要求時) に設定されます。		

注: 単一システムの場合は、RACF ファシリティーの説明に従って、RACF にリソースおよびユーザーのプロファイルを必ず定義してください。

LU6.2 セキュリティーの実装

このトピックでは、LU6.2 のセキュリティーを実装する方法を説明します。

LU6.2 でのバインド時のセキュリティー

APPC セッションを確立するための要求をリモート・システムとの間で送受信する場合 (つまりセッションのバインド時) には、セキュリティー検査を適用できます。これは、バインド時のセキュリティー (または、SNA 用語ではセッション・セキュリティー) と呼ばれ、LU6.2 アーキテクチャーの CICS 実装の一部です。

これは、無許可システムがいずれかの CICS システムにセッションをバインドするのを防ぐことを目的としています。

バインド時のセキュリティーは、LU6.2 アーキテクチャーではオプションです。リモート・システムでバインド時のセキュリティーがサポートされていない場合は、それを指定しないでください。 SNA は、セッション・セキュリティーを適用する方法を定義し、CICS TS はこのアーキテクチャーに準拠します。別のシステムに接続する場合は、そのシステムもこのアーキテクチャーとの互換性があることを確認します。

リモート・システムへの LU6.2 接続を定義する場合は、すべてのインバウンド・バインド要求がそのリモート・システムで発信され、すべてのアウトバウンド・バインド要求が同じシステムにルーティングされると想定します。ただし、伝送回線が切り替えられたか侵入された可能性がある場合は、接続の両端でセッション・セキュリティーを指定することによって、無許可のセッション・バインドから保護します。

バインド要求を成功させるためには、両端が RACF に定義されている同じセッション鍵を保持している必要があります。セッションがバインドされる場合、CICS が取るアクションは以下によって決まります。

- **SEC** および **XAPPC** システム初期設定パラメーターの指定方法
- CONNECTION 定義の BINDSECURITY 属性の指定方法
- リンクに対して APPCLU セキュリティー・プロファイルが定義済みかどうか

SIT 内で SEC=YES および XAPPC=YES を指定し、CSD 接続定義内で BINDSECURITY(YES) を指定し、パートナー・システムに対して BINDSECURITY(YES) も指定する場合は、バインド・セキュリティー検証が試行されます。

BINDSECURITY(NO) を指定した場合、SIT の指定は重要ではありません。

表 28 では、どのような動作になるかを要約しています。

表 *28.* リソース定義に対する *APPC* バインド時のセキュリティー・リレーションシップ

SEC 値	**XAPPC** 値	**BINDSECURITY** 値	**RACF APPCLU** プロファイル	結果の **CICS** アクション
YES	YES	YES	定義済み (注 1 を参照)	CICS は、バインド時に RACF から APPCLU プロファイルを抽出して、リモート・システムを検証します。
YES	YES	YES	未定義	CICS は RACF から APPCLU プロファイルを抽出できないので、バインドを拒否します。
YES	YES	NO	任意の値	CICS はバインドを検証できず、それを拒否します。

表 28. リソース定義に対する *APPC* バインド時のセキュリティー・リレーションシップ (続き)

SEC 値	**XAPPC** 値	**BINDSECURITY** 値	**RACF APPCLU** プロファイル	結果の **CICS** アクション
YES	NO	任意の値	任意の値	CICS はバインドを検証できず、それを拒否します。
NO	任意の値	任意の値	任意の値	CICS はバインドを検証できず、それを拒否します。

注:

1. RACF APPCLU プロファイルが定義されているが、セッション・セグメントがロックされているか有効期限が切れている場合は、SESSKEY には値は指定されず、バインド要求は必ず拒否されます。
2. この表は、パートナーが BINDSECURITY(YES) を指定したときの応答を示しています。

APPCLU 一般リソース・クラスでのプロファイルの定義

LU6.2 でバインド時のセキュリティーを使用する場合は、APPCLU 一般リソース・クラス内にプロファイルを定義する必要があります。APPCLU リソース・クラスは、z/OS Communications Server セッションの確立時に、APPC パートナー論理装置 (LU タイプ 6.2) の ID を検査するために使用されます。

これを実行するには、以下の手順を実行します。

1. z/OS Communications Server システム・プログラマーに問い合わせて、各セッション・パートナーの以下の情報を入手します。
   - ネットワーク ID および LU の ID。
2. セッション・パートナーの各ペアに対して、APPCLU 一般リソース・クラス内に 2 つのプロファイルを作成します。

   一方のシステムで、次の RDEFINE コマンドを入力します。

```
RDEFINE APPCLU netid1.luid1.luid2 UACC(NONE)
        SESSION(SESSKEY(password))
```

   もう一方のシステムで、次の RDEFINE コマンドを入力します。

```
RDEFINE APPCLU netid2.luid2.luid1 UACC(NONE)
         SESSION(SESSKEY(password))
```

   説明:

   netid1

   netid2

   これらは、パートナーのネットワーク ID (NETID) です。これらの ID は、SYS1.VTAMLST の ATCSTRxx メンバーである、z/OS Communications Server の開始オプション NETID に指定されます。

   luid1

luid2
これらは、パートナーの LU 名です。いずれの場合でも、最初に指定する LU 名はローカル LU 名、2 番目の LU 名はリモート LU 名です。

session-key
これは、16 進数字または 8 文字のパスワードであり、リモート・システムのセッション鍵と一致します。 16 進数字は、例えば SESSKEY (X'0123456789ABCDEF') のように、引用符で囲みます。

両方のシステムで同じセッション鍵を指定する必要があります。セッション鍵が一致しない場合は、セッションを確立できません。

RACF ではセッション鍵を指定する必要はありませんが、セッション鍵が指定されていない場合、CICS はバインドを拒否します。

3. 各 LU ペアのパートナー間でセッションの属性を定義します。これを実行するには、RDEFINE コマンドおよび RALTER コマンドの SESSION オプションを使用して、各 APPCLU プロファイルの SESSION セグメントを定義します。各 SESSION セグメントには、以下の情報を指定できます。

CONVSEC
LU ペアのパートナー間で行われる会話ごとに実行されるセキュリティー検査のレベルを指定します。 CICS はこの情報を使用しません。代わりに、CONNECTION リソースの ATTACHSEC オペランドに指定された情報を使用します。詳細については、CONNECTION リソースを参照してください。

INTERVAL
セッション鍵の変更が必要になるまでの、セッション鍵の最大有効日数を指定します。

これがリンクのリモート・エンドのユーザーに与える可能性がある影響について認識しておく必要があります。いずれかのパスワードの有効期限が切れた場合、リンクは確立できません。プロファイル・レコードの監査によっては、ICH415I メッセージが書き出される場合もあれば書き出されない場合もあります。 279 ページの『LU6.2 のバインド時のセキュリティーの指定』を参照してください。(CICS は、パスワードの有効期限が切れた場合は、メッセージ DFHZC4942 を CSNE 宛先に発行します。) パスワード・インターバルがいつ頃期限切れになるかを把握しておき、その理由でリンクが失敗するということがないようにしてください。 CICS はパスワードの有効期限が間もなく切れるというときにメッセージを表示することはありませんが、レコードを SMF ログに書き込みます。

LOCK
プロファイルにロック済みとしてマーク付けします。プロファイルがロックされている場合、セッションはバインドされず、CICS はメッセージ DFHZC4941 を発行します。

NOLOCK
プロファイルにロック解除としてマーク付けします。

LU6.2 バインドの制御について詳しくは、z/OS Security Server RACF セキュリティー管理者のガイドを参照してください。

LU6.2 のバインド時のセキュリティーの指定

バインド時のセキュリティーは、CONNECTION リソース定義内で定義しますが、適切なシステム初期設定パラメーターを選択する必要もあります。

図 7 は、BINDSECURITY オプションを指定する必要がある、APPC 外部セッション・セキュリティーの定義方法を示しています。

```
CEDA DEFINE CONNECTION(name)
  GROUP(groupname)
  ACCESSMETHOD(VTAM)
  SECURITYNAME(name)
  PROTOCOL(APPC)
  NETNAME(name)
  BINDSECURITY(YES)
```

図 7. バインド時のセキュリティー

APPC 端末が TERMINAL-TYPETERM のペアとして定義される場合、BINDSECURITY オペランドは TERMINAL 定義上にあります。

注: VTAM は、z/OS Communications Server (for SNA または IP) です。

バインド時のセキュリティーの監査

セキュリティーがアクティブである (システム初期設定パラメーターで SEC=YES が指定されている) 場合、CICS はバインド・セキュリティー監査を実行します。

以下の条件はバインド失敗と見なされ、RACF は SMF レコードを書き込み、次のメッセージを発行します。

- セッション鍵がパートナーの鍵と一致しない。
- セッション・セグメントがロックされている。
- セッション・セグメントの有効期限が切れている。
- セッション鍵がヌルである。
- セッション・セグメントが存在しない。
- セッション・セグメントの取得が失敗した。
- セッション・バインドが失敗した。

以下の条件はバインド成功と見なされ、RACF は SMF レコードを書き込みますが、メッセージは発行しません。

- セッションが正常にバインドされた。
- セッション鍵は、6 日が経過する前に有効期限が切れる。

以下のいずれかの条件が当てはまる場合は、SMF レコードが書き込まれます。

- プロファイルの監査オプションが、(AUDIT(ALL(READ))) に設定されている。
- SETROPTS LOGOPTIONS(ALWAYS(APPCLU)) が設定されている。

SMF 監査レコードが書き込まれると、次の 2 つのことが実行されます。

- メッセージ ICH700051 が、プロファイルの通知オプションで指定されたユーザー ID に送信されます。 APPCLU クラスを担当している RACF 管理者の TSO ユーザー ID を指定することが推奨されます。

- セキュリティー・コンソール (宛先コードが 9 の任意の MVS コンソール) は、メッセージ ICH415I を受け取ります。これには、メッセージ ICH70005I に似たテキストが含まれています。

これらの監査レコードは、SMF から抽出され、次のサンプル RACF 報告書作成プログラム制御ステートメントを使用してリストできます。

```
//RACFRW EXEC PGM=IKJEFT01
//SORTWKxx DD your sort files
//SYSPRINT DD SYSOUT=*
//SYSTPRT DD SYSOUT=*
//RSMFIN DD DSN=your smf dumped data, DISP=SHR
//SYSTIN DD *

RACFRW TITLE('Bind Security Reports') GENSUM
SELECT PROCESS
EVENT APPCLU
LIST SORT(DATE,TIME)
END

//
```

RACF 報告書作成プログラムについては、 z/OS Security Server RACF 監査担当者のガイドを参照してください。

使用中の RACF プロファイルの変更 - 注意

APPC 接続の使用中の RACF プロファイルを変更するときは、注意してください。

CICS は、SETROPTS RACLIST(APPCLU) REFRESH コマンドが発行された後に、プロファイル内の変更を認識します。バインド時のセキュリティー処理は、接続の各セッションが獲得されたときに行われます。接続の一部のセッションしか獲得されず、APPC プロファイルが無効になった場合、いずれかの未獲得セッションを確立しようとすると、バインド・セキュリティーは失敗します。これにより、それら未使用のいずれかのセッションの割り振りを試行するトランザクションが、無期限に中断状態になる可能性があります。

無効なプロファイルの理由: APPC プロファイルは、いくつかの理由で無効になる場合があります。例えば、次のような理由があります。

- セッション鍵の有効期限が切れる。
- セッション鍵が変更され、一方のシステムで SETROPS REFRESH が実行されるが、他方のシステムでは対応する変更やリフレッシュが行われない。
- REFRESH が実行されている間、プロファイルがロックされる。

既に獲得済みのセッションは、別のセッションでバインド・セキュリティーが失敗しても、正常に機能し続けます。有効期限切れセッション鍵を使用している場合でも、接続上のいずれかのセッションが有効期限日付前に獲得されており、かつ獲得された状態を維持している場合は、接続は有効期限を過ぎても引き続き使用できます。したがって、有効期限切れセッション鍵の効果は、接続 (またはセッション) を獲得するときにのみ分かります。

セッション鍵の有効期限が切れそうであることを知らせる警告メッセージは作成されません。ただし、まもなく有効期限が切れる鍵を使用すると、SMF レコードが書き込まれる場合があります。したがって、RACF 報告書作成プログラムを定期的に

使用することで、どの鍵の保守が必要であるかを把握できます。そのようにしない場合は、有効期限が切れそうなセッション鍵を使用するときに、鍵の有効期限がいつ切れるかを覚えておく必要があります。さらに、セッション鍵の有効期限が切れて接続ができなくなることで発生する可能性がある中断を最小限に抑えるための、適切な処置を取る必要もあります。例えば、セッション・キーの変更、セキュリティーの再作成 (両方の CICS システムに対して)、および接続再獲得の必要の可能性について計画する必要があります。

接続の使用中に APPC プロファイルが無効になるという問題は、SESSIONS 定義に AUTOCONNECT(YES) または AUTOCONNECT(ALL) を指定することによって回避できます。これにより、接続の獲得時には、すべてのセッションが確立 (獲得) されます。

Link security with LU6.2

リンク・セキュリティーは、ユーザーがアクセスできるリソースを、アクセス元のリモート・システムに応じてより詳細に制限します。リンク・セキュリティーの実際的な効果として、リモート・ユーザーによるトランザクションへの接続や、リンク・ユーザー ID が権限を持たないリソースへのアクセスを抑止します。

リンク・セキュリティーが使用中であるときは、各セッションにはリンク・ユーザー *ID* によって定義された権限が付与されます。 LU6.2 の場合は、接続内のすべてのセッションで同じリンク・ユーザー ID を使用できます。または、接続内の異なるセッション・グループで異なるリンク・ユーザー ID を使用できます。さらに、あるセッション・グループではリンク・セキュリティーを使用し、他のグループでは使用しないように指定することもできます。

CICS へのトランザクション・ルーティングおよび機能シップを実行するには、少なくとも 1 つのセキュリティー検査が必要です。ただし、リンク・ユーザー ID がローカル領域ユーザー ID と一致する場合、セキュリティー検査は最小限に抑えられます。

- ユーザー ID が一致すると、1 つのセキュリティー検査だけが実行されます。これは、デフォルト・ユーザー (ATTACHSEC=LOCAL の場合)、または受信した FMH-5 接続要求内のユーザー ID (ATTACHSEC=non-LOCAL の場合) のいずれかに対して実行されます。
- ユーザー ID が一致せず ATTACHSEC=LOCAL の場合、リソース検査はリンク・ユーザー ID に対してのみ行われます。ATTACHSEC=non-LOCAL の場合、2 つのリソース検査が必ず行われます。 1 つはリンク・ユーザー ID に対して、2 番目の検査は接続要求でリモート・ユーザーから受信したユーザー ID に対して行われます。

リンク・セキュリティーを確立する際に障害が生じると、ローカル領域のデフォルト・ユーザーのセキュリティーがリンクに提供されます。これは例えば、事前設定セッションのユーザー ID が取り消された場合に起きる可能性があります。

LU6.2 接続のリンク・セキュリティーの指定

このタスクについて

CONNECTION リソース定義または SESSIONS リソース定義でリンク・セキュリティーを指定します。これらのリソースのいずれかについての詳細情報が必要な場合は、CONNECTION リソースおよびSESSIONS リソースを参照してください。

手順

- 接続のすべてのセッションが同じリンク・ユーザー ID を持つように指定するには、CONNECTION リソース定義の SECURITYNAME 属性を指定します。この属性の値を指定しない場合、CICS はデフォルト・ユーザー ID を使用します。
- 接続内のセッションの個々のグループに別々のリンク・ユーザー ID を指定するには、SESSION リソース定義の USERID 属性を指定します。セッションの各グループで、指定された値は CONNECTION 定義の SECURITYNAME 属性をオーバーライドします。

LU6.2 でのユーザー・セキュリティー

ユーザー・セキュリティーによって、 281 ページの『Link security with LU6.2』で説明されているリンク・セキュリティーに加えて、端末にサインオンしたユーザーに対する 2 番目の検査を行えます。以下の説明を読んだ後に、ユーザー・セキュリティーが提供する特別なレベルのセキュリティー検査を必要とするかどうかを検討してください。

CONNECTION 定義の ATTACHSEC パラメーターを使用して、以下のレベルのユーザー・セキュリティーを指定できます。

- *LOCAL*。これは、ユーザー ID またはパスワードの送信を要求することでユーザーをさらに検査することをしたくない場合に指定します。ご使用のシステムにとってリンクの権限が十分であると判断したためにユーザー・セキュリティーが不要である場合は、LOCAL を指定します。これを行う方法については、 284 ページの『リンク定義でのユーザー・セキュリティーの指定』を参照してください
- *Non-LOCAL*。これは、ユーザー ID、またはユーザー ID とパスワードの送信を要求することでユーザーをさらに検査したい場合に指定します。 Non-LOCAL には、以下のタイプの検査が含まれます。
  - IDENTIFY

    ユーザー ID は送信する必要がありますが、パスワードは要求されません。
  - VERIFY

    パスワードも必ず送信する必要があります。
  - PERSISTENT VERIFICATION

    パスワードはユーザーの最初の接続要求で送信されます。
  - MIXIDPE

    識別または持続検査のいずれかを実行します。

注: 『非ローカル・ユーザー・セキュリティー検査』では、これらのタイプのユーザー検査についてさらに詳しく説明しています。それらの指定については、 284 ページの『リンク定義でのユーザー・セキュリティーの指定』を参照してください。

非ローカル・ユーザー・セキュリティー検査

CICS 間システム接続で、端末専有領域 (TOR)、アプリケーション専有領域 (AOR)、およびデータ専有領域 (DOR) がある場合、端末オペレーターは TOR にサインオンし、AOR でトランザクションを接続し、DOR でリソースにアクセスします。これら 3 つのシステムすべてが非ローカル・ユーザー・セキュリティーを実装する場合、それぞれで同じオペレーターがユーザーとして登録されます。

通常の手順では、オペレーターがパスワードを使用して TOR にサインオンします。 CICS はそのパスワードがシスプレックス全体で有効であると見なし、さらに検査するために AOR と DOR に渡す必要はないと判断します。 AOR と DOR で必要とされるのはユーザーを識別することのみであり、それからユーザーはパスワードなしでサインオンします。したがって、パスワードは AOR への接続要求では送信されません。これは、ネットワーク上でパスワードの受け渡しが行われないため、より安全であると考えられます。

リモート・システムがリンクを使用させる前にユーザーを (ある種のサインオン・メカニズムで) 検査すると CICS が確信できることが分かっている場合は、IDENTIFY を指定します。 CICS-CICS 間通信にユーザー・セキュリティーが必要な場合は、IDENTIFY を使用します。CICS は CICS-CICS 間接続でのパスワード・フローをサポートしません。

フロントエンドにセキュリティー・マネージャーがない場合は、接続要求が AOR に到達する前に、ユーザー ID とパスワードを使用してユーザーを検証できない可能性があります。次に AOR は、パスワードを使用したサインオンによってユーザー自体を検証できるように、ユーザー ID とパスワードの両方をフロントエンドから受け取る必要があります。

リモート・システムのユーザーをリモート・システム自体で検査済みであっても独自のシステムでそれらを検査したい理由がある場合、またはリモート・システムにセキュリティー・マネージャーがなく、その固有のユーザーを検査できないという場合は、VERIFY を指定します。

プログラマブル・ワークステーションが AOR でトランザクション接続要求を繰り返し行うと、検査数が多くなることが原因でパフォーマンスが低下します。これらのセキュリティー手順を定義した LU6.2 アーキテクチャーにより、持続検査が可能であるため、ソフトウェア・オーバーヘッドを削減できます。このプロトコルを使用する場合、最初の接続要求にはユーザー ID とパスワードを含めますが、ユーザーがサインオンすれば、後続のすべての接続要求では必要とされるのはユーザー ID のみです。

リモート・ユーザーが繰り返し接続要求を送信する場合は、検査のオーバーヘッドを減らすために PERSISTENT を指定します。ただし、リモート・システムは、現在サインオンしているユーザーのリストを保持することで、持続検査の管理に協力できる必要があります。

一部のリモート APPC システムでは、会話ごとに異なるサインオン要件 (例えば、CPI コミュニケーション会話) が混在しています。この場合、CICS は、着信した ID 要求または持続要求を受け入れる必要があります。

これらのどのタイプのユーザー検査を使用するかを決定するには、リモート・システムが独自のセキュリティーをどの程度管理できるのか、および管理できない場合は、定義する CICS システムにどの程度依存する必要があるのかを知っておく必要があります。

- ユーザー ID を把握する必要がありますか。そうでない場合は、LOCAL を使用します。
- リモート・システムはそれ自体のユーザーを検査できますか。そうである場合は、IDENTIFY を使用します。そうでない場合は、ユーザー ID とパスワードを接続要求で送信できますか。そうである場合は、VERIFY を使用します。
- リモート・システムは、ユーザー ID とパスワードを追跡することで、持続検査をサポートしますか。そうである場合、PERSISTENT を指定します。ただし CICS-CICS 間通信を使用しない場合は、MIXIDPE を指定します。

CONNECTION 定義の ATTACHSEC オペランドを使用して、接続ごとにこれらのレベルの検査を指定します。これについては、『リンク定義でのユーザー・セキュリティーの指定』で説明されています。

リンク定義でのユーザー・セキュリティーの指定

リモート・システムにおいて必要となるユーザー・セキュリティーのレベルは、CONNECTION 定義の ATTACHSEC オペランドで指定します (図 8 を参照)。

このトピックでは、CONNECTION 定義の ATTACHSEC オペランドのパラメーターを、CICS がどのように解釈するかについて説明します。ただし、CICS トランザクション・ルーティングには特別の規則が適用されます (291 ページの『LU6.2 でのトランザクション・ルーティング・セキュリティー』 を参照)。 CEDA を使用して ATTACHSEC を定義する例を、図 8 で紹介します。

```
CEDA DEFINE CONNECTION(name)
  GROUP(groupname)

  ATTACHSEC(LOCAL|IDENTIFY|VERIFY|PERSISTENT|MIXIDPE)
```

図 *8.* ユーザー・セキュリティーのサインオン・レベルの定義

注**:** APPC 端末が TERMINAL-TYPETERM のペアとして定義される場合、ATTACHSEC オペランドは TERMINAL 定義上にあります。

ATTACHSEC オペランドによって、着信トランザクション接続要求のサインオン要件を指定します。お使いのシステムからリモート・システムへ発行される接続要求には、影響を与えません。そうした要求は、リモート・システムで処理されます。

APPC セッションがバインドされると、各サイドは他方のサイドに対して、その着信要求について行われる接続セキュリティーのユーザー検査のレベルを伝えます。これについてのネゴシエーションは行われません。

ATTACHSEC オペランドの意味**:** 以下に示すのは、ATTACHSEC の指定可能なオペランドです。

LOCAL
ユーザー ID がリモート・システムによって提供されないことを示します。受け取った場合、接続は失敗します。 CICS は、リンク・セキュリティー・プロファイルと同等のユーザー・セキュリティー・プロファイルを作成します。リモート・ユーザーに対して RACF プロファイルを指定する必要はありません。 LOCAL がデフォルト値です。

IDENTIFY
すべての接続要求でユーザー ID が予期されることを示します。システムのすべてのリモート・ユーザーは RACF で識別される必要があります。

ユーザー ID とパスワードの両方がある接続要求が、ATTACHSEC(IDENTIFY) が指定されているリンク上で受け取られると、CICS はその接続要求を拒否しません。 CICS は接続要求を、ATTACHSEC(VERIFY) で定義されているのと同じように処理します。

ヌル (X'00') のユーザー ID または不明のユーザー ID を受け取った場合、CICS は接続要求を拒否します。

VERIFY
ユーザー ID に加え、ユーザー・パスワードもローカル RACF データベースに照らした検査に必要であることを指定します。システムのすべてのリモート・ユーザーは RACF で識別される必要があります。

ATTACHSEC(IDENTIFY) のユーザー ID の検査に適用されるルールは、ATTACHSEC(VERIFY) にも適用されます。有効なユーザー ID を受け取ったが、パスワード検査に失敗すると、CICS は接続要求を拒否します。通信システムが CICS for AIX の場合、ATTACHSEC=IDENTIFY を使用する必要があります。

注**:** CICS 以外の製品は、LU6.2 リンクを介して CICS Transaction Server for z/OS AOR に接続できます。次に、それらは SNA LU6.2 FMH-5 ATTACH メカニズムを使用して、CICS AOR 上でトランザクションを開始します。このメカニズムを非セキュア・システムから使用する場合は、接続定義で ATTACHSEC=VERIFY オプションを使用して、AOR 上のトランザクションを保護する必要があります。 (288 ページの『SNA プロファイルおよび接続時セキュリティー』を参照してください。)

PERSISTENT
新規ユーザーの最初の接続要求では、ユーザー ID とユーザー・パスワードが要求されることを示します。ただし同じユーザーの以降のすべての接続要求では、ユーザー ID のみを入力する必要があります。 (システムのすべてのリモート・ユーザーは RACF で識別される必要があります)。ユーザーがトランザクションの接続を許可されていないため接続要求が最終的に失敗するとしても、ユーザーは最初の接続試行でサインオンします。

注**:** PERSISTENT は CICS-CICS 間通信には使用できません。

MIXIDPE
リモート・ユーザーのサインオン・レベルが、接続要求で送信されたパラメーターによって決定されることを示します。 PERSISTENT または IDENTIFY が指定可能です。

LU6.2 でのリモート・ユーザー・サインオン状況**:**

ATTACHSEC パラメーターの IDENTIFY、MIXIDPE、PERSISTENT、および VERIFY を指定すると、最初の接続要求に関連した会話が完了した後に、リモート・ユーザーはサインオンしたままになります。

これ以降 CICS は、以下のいずれかのイベントが発生するまで、同じユーザーからの接続要求を新規サインオンなしで受け入れます。

- このユーザーの接続要求に関連した最後のトランザクションの完了後に、USRDELAY システム初期設定パラメーター システム初期設定パラメーターに指定された期間が経過します。
- CICS に通知されます。詳細については、 20 ページの『リモート・ユーザーの RACF プロファイルの変更』を参照してください。
- CICS 領域がシャットダウンします。

ユーザーを取り消すなどしてサインオン・リモート・ユーザーの RACF プロファイルを変更した場合、以下のいずれかの状態となるまで、CICS は最初の接続要求時に設定された許可を引き続き使用します。

- トランザクションが同期点を実行する。
- 接続要求が終了する。
- RACF が CICS にユーザー・プロファイルの変更を通知し、そのサインオン・ユーザー ID に関連付けられている接続要求が、LOCAL を除く ATTACHSEC のすべてのオペランドに対して完了したためにサインオフとなる。
- RACF が CICS にユーザー・プロファイルの変更を通知し、新規接続要求が行われ、**USRDELAY** システム初期設定パラメーターの値が有効期限切れになっていないためにサインオフとなる。このサインオフの後には、サインオンが続きます。

パスワード検査**:**

ATTACHSEC(PERSISTENT) (または ATTACHSEC(PERSISTENT) として処理される ATTACHSEC(MIXIDPE)) を使用している場合、CICS は各リモート・システムに対して持続検査 **(PV)** サインオン元リストと呼ばれるテーブルを保持します。

これは、パスワードが検査済みで、エントリーがリストに残っている限りそれ以上のパスワード検査を必要としないユーザーのリストです。エントリーは、以下の時点までリストに残ります。

- ユーザーのサインオン・エントリーが最後に使用されてから、システム初期設定パラメーター PVDELAY で指定された期間が経過するまで。

  PVDELAY は、リモート・システムの PV サインオン元リストにエントリーが残ることができる（つまり、接続要求ごとにそのパスワードを再検証する必要がない) 期間を定義します。 PVDELAY の値の指定については、PVDELAY シス

テム初期設定パラメーターを参照してください。調整については、CICS システムのパフォーマンスの向上を参照してください。

- このシステムとの接続は、CICS が再始動される、接続が失われる、または CICS がユーザーから無効な接続要求を受け取るなどの理由で終了します。

持続検査がリモート・ユーザーに対して稼働しているときに、そのユーザーが PV サインオン元リストから削除されると、CICS はリモート・システムに、そのユーザーのサインオフ要求を発行して通知し、リモート・システム内の PV サインオン先リストからエントリーを削除します。

ATTACHSEC(VERIFY) を指定した場合、リモート・ユーザーのパスワードはすべての接続要求 について検査されます。これは、ユーザーがこのシステムにアクセスする権限を持っていることを確認し、このパスワードが正しいことを確認し、ユーザーのセキュリティー権限を確立するためです。

リモート・ユーザーに関する情報

ユーザーに関する情報は、リモート・システムから接続要求を使用して送信できます。

つまり、どのリモート・システムが要求を出しているかだけでなく、リモート・システムでどのユーザーが要求を出しているかにも基づいてリソースを保護できます。

このトピックでは、リモート・ユーザー・セキュリティーに関連するいくつかの概念と、CICS がユーザー情報を送受信する方法を説明しています。

ユーザーを RACF に対して定義する必要があります。リモート・ユーザーが RACF に対して定義されていない場合、そのリモート・ユーザーからのすべての接続要求は拒否されます。

LU6.2 リンクの CICS リモート・ユーザー・セキュリティーは、LU6.2 アーキテクチャーを実装します。 LU6.2 アーキテクチャーは、ユーザー ID、ユーザー・パスワード、およびユーザー・プロファイルを、トランザクションの接続要求を使用して送信できます。

ユーザー・プロファイルは、ユーザー ID の代わりに、またはユーザー ID に追加して送信できます。プロファイル名は (提供されている場合)、グループ ID として扱われます。

ユーザーが、グループ ID が明示的に指定されて (例えば EXEC CICS SIGNON で、または CESN パネルで GROUPID パラメーターを入力することで) フロントエンド・システムに追加されている場合、これは CICS によって、アウトバウンド付加 FMH で、CONNECTION 定義に ATTACHSEC(IDENTIFY) が指定されている LU6.2 リンクに対して伝搬されます。ユーザーが最初にフロントエンド・システムに追加された時点でグループ ID をデフォルトに設定することが許可されていた場合、アウトバウンド FMH5 にプロファイル・フィールドは含まれません。グループ ID がバックエンド・システムに渡された場合、グループ ID はバックエンドで ADD_USER 処理の一部として使用されます。つまり、ユーザー ID は、ADD_USER が正常に実行されるように、バックエンドの ESM で渡されたグループのメンバーとして定義する必要があります。

PLTPI 処理によって開始されたタスクがリモート・リソースにアクセスする可能性がある場合には、PLTPIUSR システム初期設定パラメーターを使用することをお勧めします。これにより、ユーザー ID がローカル領域内のユーザーと同じグループにないという、リモート領域での問題が回避されます。これは、ローカル領域の PLTPI ユーザーが明示的なグループ ID を使用して追加されておらず、その結果として、グループ ID がリモート領域に伝搬されないためです。

CICS は ATTACHSEC(IDENTIFY) の会話でユーザー ID を送信します。 表 29 は、CICS が送信するユーザー ID を決定する方法を示しています。

表 *29.* 接続時ユーザー *ID — LU6.2*

ローカル・タスクの特性	**CICS** によってリモート・システムに送信されるユーザー **ID**
関連する端末があるタスク - ユーザー ID	端末ユーザー ID
関連する端末があるタスク - ユーザーのサインオンなし、および端末定義に USERID の指定なし	ローカル・システムのデフォルト・ユーザー ID
インターバル制御 START コマンドによって開始された、関連する端末および USERID がないタスク (機能シップまたは分散トランザクション処理 (DTP) を使用する場合)	START コマンドを発行したタスクのユーザー ID
USERID オプションを指定して開始されたタスク	START コマンドに指定されたユーザー ID
CICS 内部システム・タスク	CICS 領域のユーザー ID
一時データ・トリガーによって開始された、関連する端末がないタスク	キューを定義する一時データ宛先定義に指定されたユーザー ID
一時データ・トリガーによって開始された、関連する端末があるタスク	端末ユーザー ID
PLTPI から開始されたタスク	PLTPIUSR

リモート・ユーザーへのサインオンには、次の 2 つの目的があります。

- リモート・ユーザーが CICS システムへのアクセスを許可されていることを確認する
- サインオンが成功した場合、リモート・ユーザーの権限を確立する

CICS は、 286 ページの『LU6.2 でのリモート・ユーザー・サインオン状況』で説明されている状況下では、リモート・ユーザーをサインオフします。

SNA プロファイルおよび接続時セキュリティー

CICSTS55.CICS での LU6.2 接続時セキュリティーの実装は、アーキテクチャーに厳密に従います。

特に、以下の点に注意してください。

- SNA プロファイル・サポートの導入と、SNA 接続時セキュリティー処理への準拠は、アップグレード問題の原因となる場合があります。
- プロファイルをサポートすることで、誤ってコーディングされたプロファイルが付加 FMH-5 で送信された場合、結果として特定の接続要求が拒否されます。

- FMH-5 のアクセス・セキュリティー・サブフィールドでの問題を回避するための検査は、以下のとおりです。
  - 認識されないサブフィールドの検査
  - 無効な長さのサブフィールドの検査
  - 同じタイプの複数のサブフィールドの検査
- 完全な 10 文字のユーザー ID とパスワードが受け入れられます。末尾ブランク ((X'40') は、セキュリティー・マネージャーに渡される前に削除されます。セキュリティー・マネージャーは接続要求を拒否するか、または処理を続行する前にユーザー ID とパスワードを 8 文字の形式に変換します。
- 接続要求に FMH-5 のセキュリティー・パラメーターが含まれていない場合、CONNECTION リソース定義に USEDFLTUSER(YES) が指定されていない限り、その接続要求は拒否されます。この場合、デフォルト・ユーザーのセキュリティー機能が適用されます。
- 受け取った有効な SNA プロファイルは、FMH-5 のユーザー ID がサインオンした後に FMH-5 のユーザー ID が関連付けられる ESM グループ ID として扱われます。
- SNA プロファイルが受信され、接続に ATTACHSEC=PERSISTENT があった場合は、アーキテクチャーへの適合が確認されます。それは、サインオン元リスト内のユーザーにさらに資格を付与するためには使用されません。これは、ATTACHSEC=MIXIDPE が指定されている接続上で受け取る、永続的なサインオン・フローにも適用されます。

LU6.2 でのトランザクション・セキュリティー、リソース・セキュリティー、およびコマンド・セキュリティー

単一システム環境の場合と同様に、ユーザーには以下を行うための許可が必要です。

- トランザクションを接続する (トランザクション・セキュリティー)。
- トランザクションが使用するようにプログラムされているすべてのリソースにアクセスする。これらのレベルは、リソース・セキュリティー、代理ユーザー・セキュリティー、およびコマンド・セキュリティーと呼ばれます。

トランザクション・セキュリティー

単一システム環境の場合、トランザクションのセキュリティー要件はトランザクションを定義する際に示されます (トランザクション・セキュリティー を参照)。

LU6.2 環境では、トランザクションを開始する前に、以下の 2 つの基本セキュリティー要件を満たしている必要があります。

- リンクにはトランザクションを開始するための十分な権限が必要です。
- ATTACHSEC(LOCAL) 以外のものが指定されている場合、ユーザー・セキュリティーが有効になります。したがって、要求を行っているユーザーは、システムにアクセスしてトランザクションを開始するための十分な権限を持っている必要があります。

注: トランザクション・セキュリティーは、ミラー・トランザクションにも適用されます。 293 ページの『LU6.2 での機能シップ・セキュリティー』を参照してください。

CICS ルーティング・トランザクション CRTE

接続されたリモート・システム上にあるトランザクションをローカル・システムでリモートとして定義する代わりに、CICS ルーティング・トランザクション (CRTE) を IPIC、LU6.2、または MRO リンクと共に使用してそれらのトランザクションを実行できます。

CRTE は、あまり頻繁に使用しないトランザクションや、すべてのシステム上にある CEMT などのトランザクションに対して特に役立ちます。

CRTE を開始した端末が、リモート・システムで定義されているか、またはローカル・システムでシップ可能と定義されていることを確認してください。リモート・システムが保護されている場合、端末オペレーターは RACF 権限を必要とします。

CRTE で実行されるトランザクションに対する AOR でのセキュリティー検査は、ATTACHSEC (MRO リンクと LU6.2 リンクの場合) または USERAUTH (IPIC リンクの場合) で指定されている内容に依存していません。TOR にサインオンしているユーザー ID にも依存していません。その代わり、セキュリティー検査は、ユーザーが CRTE の使用時にサインオンするかどうかに依存しています。

- ユーザーがサインオンしない 場合、作成される代理端末は、AOR デフォルト・ユーザーに関連付けられます。トランザクションが実行されると、このデフォルトのユーザーに対してセキュリティー検査が実施されます。リンク・ユーザー ID に対しても検査が行われ、ルーティング・アプリケーション自体がリソースにアクセスする権限を持っているかどうかが確認されます。
- ユーザーがサインオンする 場合、CRTE の実行時に CESN トランザクションを使用して、代理がサインオン・ユーザーのユーザー ID を指します。リソースへのアクセスを試行するトランザクションの場合、セキュリティー検査は、代理のサインオン・ユーザーのユーザー ID と、リンク・ユーザー ID に対して行われます。

リソース・セキュリティーおよびコマンド・セキュリティー

相互通信環境でのリソース・セキュリティーおよびコマンド・セキュリティーは、単一システム環境とほぼ同じ方法で処理されます。

リソースおよびコマンドのセキュリティー検査は、インストールされているトランザクション定義でそれらの検査が必須であると (例えば、図 9 に示すように CEDA DEFINE TRANSACTION コマンドなどで) 指定されている場合にのみ実行されます。

```
CEDA DEFINE TRANSACTION
  .
  RESSEC(YES)
  CMDSEC(YES)
  .
```

図 9. トランザクションのリソースおよびコマンド・セキュリティーの指定

トランザクション定義でリソース・セキュリティー検査が RESSEC(YES) を使用して指定されている場合、リンクとユーザーの両方ともに、接続されたトランザクションがアクセスするリソースに対する十分な権限が備えられている必要があります。

トランザクション定義でコマンド・セキュリティー検査が CMDSEC(YES) を使用して指定されている場合、リンクとユーザーの両方ともに、接続されたトランザクションが発行するシステム・プログラミング・コマンド (162 ページの表 11 を参照) に対する十分な権限が備えられている必要があります。

リソースおよびコマンド・セキュリティーの指定についての詳しい説明は、リソース定義のセキュリティー および CICS command security を参照してください。

NOTAUTH 例外状態**:**

トランザクションがリソースへのアクセスを試行したが、リソース・セキュリティー検査に失敗した場合、NOTAUTH 条件が発生します。

トランザクションが CICS ミラー・トランザクションである場合、NOTAUTH 条件は、それを通常の方法で処理できる要求元トランザクションに返されます。

LU6.2 でのトランザクション・ルーティング・セキュリティー

トランザクション・ルーティングにおいて、トランザクションにアクセスするためのユーザーの権限は、TOR および AOR の両方で検査できます。

TOR では、リモートとして定義されたトランザクションへのアクセス権限を、ローカル・トランザクションの場合と同様にユーザーが持っていることを確認するために、検査が行われます。この検査によって、中継プログラムの実行がユーザーに許可されるかどうかが決まります。

トランザクションを呼び出す端末は、リモート・システム上で定義される (または、ローカル・システムで「シップ可能」と定義される) 必要があります。リモート・システムが保護されている場合は、端末オペレーターに RACF 権限が必要となります。リモート・システム上の端末の定義方法は、ユーザー・セキュリティーの適用方法に影響します。

- リモート端末の定義で USERID パラメーターが指定されない場合は、以下のようになります。
  - ATTACHSEC(IDENTIFY) で定義されるリンクの場合、ユーザーのトランザクション・セキュリティーおよびリソース・セキュリティーは、リモート・ユーザーのサインオン時に確立されます。ユーザーがサインオンで使用するユーザー ID は、(DFLTUSER システム初期設定パラメーター内で) 明示的または暗示的のいずれで表される場合であっても、このセキュリティー機能を有します (リモート・システムに割り当てられます)。
  - ATTACHSEC(LOCAL) で定義されたリンクの場合、トランザクション・セキュリティー、コマンド・セキュリティー、およびリソース・セキュリティーは、リンクの権限により制限されます。

いずれの場合も、 281 ページの『Link security with LU6.2』に説明されている方法で、リンク・セキュリティーに対する検査が行われます。

注**:** トランザクション・ルーティングの際に、3 文字のオペレーター識別子が TOR から AOR の代理端末エントリーへ送信されます。代理端末がシップインされた場合、この識別子はセキュリティー目的には使用されませんが、メッセージで参照されることがあります。

トランザクション・ルーティング PSB が要求を出す際には、次の条件が両方とも満たされている必要があります。

- 接続定義の ATTACHSEC は LOCAL であってはならない (つまり、IDENTIFY、PERSISTENT、MIXIDPE、または VERIFY のいずれかになる)。
- リモート・システムのシステム初期設定パラメーターとして、PSBCHK=YES が指定されている。

事前設定セキュリティー端末およびトランザクション・ルーティング

TERMINAL 定義の USERID パラメーターに値が指定されている場合、端末には事前設定セキュリティーがあります。事前設定されているセキュリティー端末からトランザクション・ルーティングのセキュリティー面を考慮する場合は、事前設定セキュリティーは、トランザクション・ルーティング要求を開始したユーザーの属性ではなく、端末の属性であることを覚えておいてください。

トランザクション・ルーティング中に、トランザクション・ルーティング要求が発行された端末を表す代理端末が AOR に作成されます。代理端末に事前設定セキュリティーがあるかどうかは、次のようないくつかの要素に依存しています。

- TOR の端末のリモート端末定義が AOR に存在し、USERID パラメーターを指定している場合は、代理端末はこのユーザー ID を使用して事前設定されます。リモート端末定義で USERID パラメーターが指定されていない場合、代理端末には事前設定セキュリティーはありません。
- リモート端末定義が AOR に存在しない場合、代理端末の事前設定セキュリティー特性は、TOR からシップされた端末定義から決定されます。シップされた端末定義に事前設定セキュリティーがある場合、AOR への接続が ATTACHSEC=LOCAL で定義されていない限り、代理端末にも事前設定セキュリティーがあります。この場合、AOR にシップされた事前設定セキュリティー情報はすべて無視されます。

CICS ルーティング・トランザクション CRTE

接続されたリモート・システム上にあるトランザクションをローカル・システムでリモートとして定義する代わりに、CICS ルーティング・トランザクション (CRTE) を IPIC、LU6.2、または MRO リンクと共に使用してそれらのトランザクションを実行できます。

CRTE は、あまり頻繁に使用しないトランザクションや、すべてのシステム上にある CEMT などのトランザクションに対して特に役立ちます。

CRTE を開始した端末が、リモート・システムで定義されているか、またはローカル・システムでシップ可能と定義されていることを確認してください。リモート・システムが保護されている場合、端末オペレーターは RACF 権限を必要とします。

CRTE で実行されるトランザクションに対する AOR でのセキュリティー検査は、ATTACHSEC (MRO リンクと LU6.2 リンクの場合) または USERAUTH (IPIC リンクの場合) で指定されている内容に依存していません。TOR にサインオンしているユーザー ID にも依存していません。その代わり、セキュリティー検査は、ユーザーが CRTE の使用時にサインオンするかどうかに依存しています。

- ユーザーがサインオンしない 場合、作成される代理端末は、AOR デフォルト・ユーザーに関連付けられます。トランザクションが実行されると、このデフォル

トのユーザーに対してセキュリティー検査が実施されます。リンク・ユーザー ID に対しても検査が行われ、ルーティング・アプリケーション自体がリソースにアクセスする権限を持っているかどうかが確認されます。

- ユーザーがサインオンする 場合、CRTE の実行時に CESN トランザクションを使用して、代理がサインオン・ユーザーのユーザー ID を指します。リソースへのアクセスを試行するトランザクションの場合、セキュリティー検査は、代理のサインオン・ユーザーのユーザー ID と、リンク・ユーザー ID に対して行われます。

LU6.2 での機能シップ・セキュリティー

CICS が機能シップ要求を受け取るときに、呼び出されるトランザクションはミラー・トランザクションです。

ミラー・トランザクションの CICS 提供の定義はすべて、リソース・セキュリティー検査を指定しますが、コマンド・セキュリティー検査は指定しません。これは、リンクまたは他のシステムのユーザー ID プロファイルのいずれかが必要な権限を持っていない場合、リモート・リソースにアクセスできないことを意味します。

CICS 提供のミラー・トランザクションの定義がセキュリティー戦略のニーズに適していない場合は、グループ DFHISC 内の定義を独自のグループにコピーし、変更し、再インストールすることで変更できます。詳しくは、Security for CICS-supplied transactionsを参照してください。

リソース定義にリモート・リソースを含める場合、ローカル・リソースの場合と同じように、ローカルでセキュリティー検査が実行されるように調整できます。さらに、リソースを所有するシステムは、ユーザー ID を受け取ることができる場合、独立検査を適用するようにすることができます。したがって、セキュリティー制限の適用は、両方の側または一方の側に行うか、あるいはどちらにも行わないという選択ができます。

注**:** 機能シップされた要求に対して SYSID オプションを指定する場合、セキュリティー検査は、リモート・システムでは実行されますが、ローカル・システムではバイパスされることに注意してください。 294 ページの図 10 では、どのような動作になるかを要約しています。

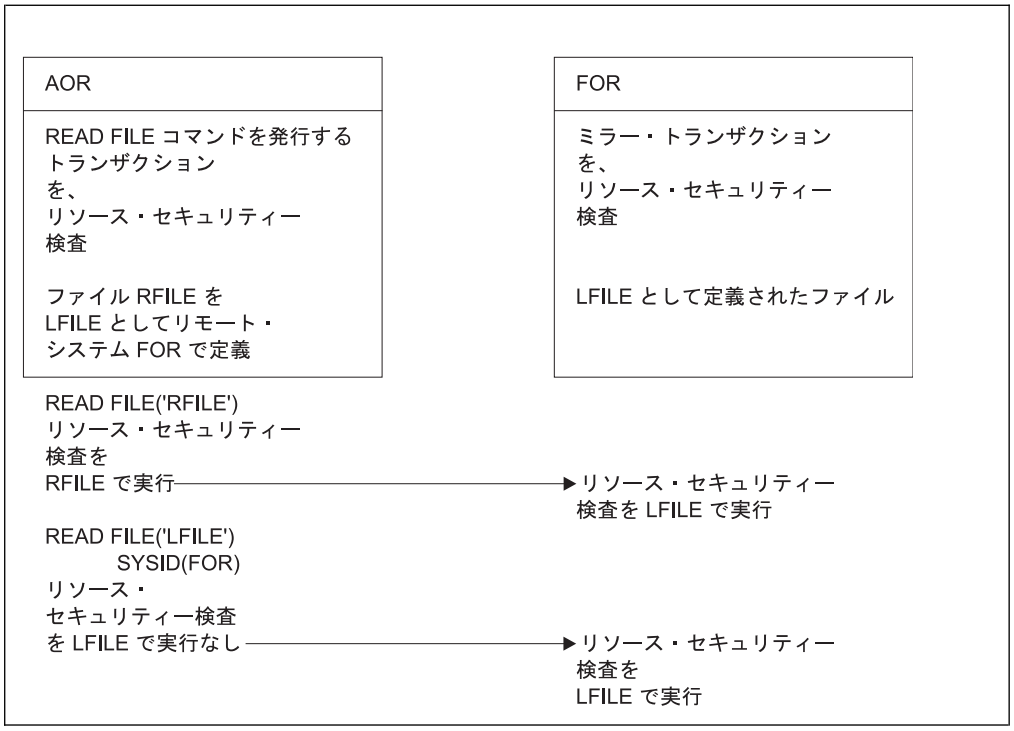


図 *10. SYSID* の指定ありまたは指定なしで実行されるセキュリティー検査： この例は、アプリケーション専有領域内のトランザクションがリモート・ファイルに対してファイル制御要求を発行するときに、どのようなセキュリティー検査が実行されるかを示しています。

- アプリケーション専有領域 (AOR) 内で、ファイル RFILE は、ファイル専有領域 (FOR) 内の LFILE という名前で、リモートとして定義されます。ファイル制御要求を発行するトランザクションに対して、リソース・セキュリティー検査がアクティブです。
- FOR 内では、ミラー・トランザクションに対してリソース・セキュリティー検査がアクティブです。

これには 2 つのケースがあります。最初のケースは次のとおりです。

1. AOR 内のトランザクションが EXEC CICS READ FILE('RFILE') を発行します。リソース・セキュリティー検査がファイル RFILE に対して実行されます。
2. 要求は FOR に送信され、そこではリソース・セキュリティー検査がファイル LFILE に対して実行されます。

2 番目のケースは次のとおりです。

AOR 内のトランザクションが、EXEC CICS READ FILE('RFILE') を *SYSID* オプションを指定して 発行します。リソース・セキュリティー検査はファイル RFILE に対して実行されません。

要求は FOR に送信され、そこではリソース・セキュリティー検査がファイル LFILE に対して実行されます。

LU6.2 での分散プログラム・リンク・セキュリティー

CICS 分散プログラム・リンク (DPL) 機能を使用すると、CICS プログラム (クライアント・プログラム) が、リモートの CICS 領域にある別の CICS プログラム (サーバー・プログラム) を呼び出すことができます。 DPL は、LINK API コマン

ドの SYSID オプションまたは PROGRAM リソース定義の REMOTESYSTEM オプションがリモート CICS 領域を指定するときに使用されます。

LINK コマンドの SYSID オプションがリモート CICS システムを指定する場合、クライアント領域はリソース・セキュリティー検査を実行しませんが、サーバー領域でのリソース検査はそのまま実行されます。

リモート領域のサーバー・プログラムは、機能シップされた他の CICS 要求に対するのと同じように、ミラー・トランザクションによって実行されます。ただし、ミラーに関連付けられるトランザクション名は、クライアント領域で LINK コマンドがどのように呼び出されるかに依存しています。通常の接続セキュリティーがミラー・トランザクションに適用されるため、ユーザーはトランザクション名について認識しておく必要があります。

- TRANSID オプションが DPL コマンドで指定されている場合、指定されたトランザクション名がミラーに使用されます。
- TRANSID オプションが DPL コマンドで省略されているが、クライアント領域のプログラム・リソース定義では使用されている場合、ミラーの名前はプログラムの TRANSID 仕様から取られます。

それ以外の場合、ミラー・トランザクションのデフォルト名が使用されます。これは、会話の起点および LU6.2 同期レベルに応じて決まります。

- 同期レベル 1 を使用している場合、ミラーのデフォルトのトランザクション名は **CVMI** です。このトランザクション名は、以下の場合に使用されます。
  - クライアント領域で SYNCONRETURN オプションが DPL コマンドに指定されている場合
  - LU6.2 CONNECTION 定義が SINGLESESS(YES) を指定している場合
  - 接続が LU6.2 端末 (DEVICE(APPC) が指定された TYPETERM を使用するリソース定義がある端末) を使用する場合
- 同期レベル 2 を使用している場合、デフォルトのトランザクション名は **CSMI** です。 CSMI は前述のどの条件も満たされない場合に使用されます。

ミラーを実行しているトランザクション名へのアクセスをユーザーに許可します。ユーザー ID が許可されるどうかは、LOCAL または非 LOCAL の接続セキュリティーが使用されているかどうかによって決まります。これについては、 296 ページの『LU6.2 での AOR で実行されるセキュリティー検査』で説明されています。サーバー領域内でミラー・トランザクションが RESSEC(YES) で定義されている場合、これらのユーザー ID は、ミラーによってリンクされているサーバー・プログラムへのアクセスも許可されている必要があります。サーバー・プログラムが任意の CICS リソースにアクセスする場合、同じユーザー ID はそれらへのアクセスを許可される必要があります。サーバー・プログラムが任意のシステム・プログラミング・コマンドを呼び出し、サーバー領域内でミラー・トランザクションが CMDSEC(YES) で定義されている場合、同じユーザー ID はそれらのコマンドへのアクセスを許可される必要があります。

セキュリティー上の理由でミラー・トランザクションが接続できない場合、NOTAUTH 条件は発生しませんが、TERMERR 条件がクライアント領域内の発行元アプリケーションに返されます。ミラー・トランザクションが正常に接続されたが、サーバー領域内の分散プログラムへのリンクを許可されない場合、NOTAUTH

条件が発生します。セキュリティー上の理由からサーバー・プログラムがいずれかの CICS リソースへのアクセスに失敗した場合にも、NOTAUTH 条件が発生します。

サーバー・プログラムは、サーバー領域で実行されるときは、CICS API コマンドの DPL サブセットに制限されます。サポートされないコマンドは、セキュリティー関連情報を返すコマンドなどです。制限されるコマンドに関するプログラミング情報については、CICS コマンド・サマリーを参照してください。 DPL の詳細については、CICS 分散プログラム・リンクを参照してください。

LU6.2 での AOR で実行されるセキュリティー検査

AOR および TOR で SECURITYNAME がどのように指定されているかに応じて、セキュリティー検査は異なります。

表 30 および 297 ページの表 31 に示されているリンク・ユーザー ID は、CONNECTION リソース定義の SECURITYNAME、または SESSION リソース定義の USERID で指定されているものです。

ユーザー ID が SESSIONS 定義で指定され、リンク検査が行われる場合、使用されるユーザー ID は SESSIONS 定義のユーザー ID です。

SECURITYNAME にユーザー ID が指定されない場合は、代わりに AOR のデフォルト・ユーザー ID が使用されます。ただし、SECURITYNAME ユーザー ID が AOR の領域ユーザー ID と同じ場合、リンクは AOR と同じセキュリティーを持っていると見なされ、リンク・セキュリティーは完全に省略されます。省略されたリンク・セキュリティーの影響は、リンクに対して LOCAL 接続セキュリティーまたは非 LOCAL 接続セキュリティーが指定されているかどうかによって異なります。

- LOCAL 接続セキュリティーの場合、SESSIONS 定義の USERID に指定されたセキュリティーが使用されます。これも省略される場合は、AOR のデフォルト・ユーザー ID が使用されます。
- 非 LOCAL 接続セキュリティーの場合、SESSIONS 定義の USERID に指定されたセキュリティーは使用されません。 TOR から受け取ったユーザー ID のみがセキュリティーの判別に使用されます。

注**:** TOR の領域ユーザー ID、および AOR に対する TOR の CONNECTION 定義の SECURITYNAME はどちらも、AOR のセキュリティー検査には関係しません。

表 30 は、ATTACHSEC(LOCAL) を指定した場合の検査方法を示しています。

表 *30. LU6.2* および *ATTACHSEC(LOCAL)*

AOR の領域ユーザー **ID**	接続定義の **SECURITYNAME**	**SESSION** 定義の **USERID**	**AOR** での検査
USERIDA	指定されていません	指定されていません	AOR DFLTUSER に対する検査
USERIDA	指定されていません	USERIDA	AOR DFLTUSER に対する検査

表 30. LU6.2 および ATTACHSEC(LOCAL) (続き)

AOR の領域ユーザー **ID**	接続定義の **SECURITYNAME**	**SESSION** 定義の **USERID**	**AOR** での検査
USERIDA	指定されていません	USERIDB	USERIDB に対する検査
USERIDA	USERIDA	指定されていません	AOR DFLTUSER に対する検査
USERIDA	USERIDB	指定されていません	USERIDB に対する検査
USERIDA	USERIDA	USERIDA	AOR DFLTUSER に対する検査
USERIDA	USERIDA	USERIDB	USERIDB に対する検査
USERIDA	USERIDB	USERIDA	DFLTUSER に対する検査
USERIDA	USERIDB	USERIDB	USERIDB に対する検査
USERIDA	USERIDB	USERIDC	USERIDC に対する検査

表 31は、ATTACHSEC パラメーター IDENTIFY (または PERSISTENT、MIXIDPE) を指定した場合の検査方法を示しています。

表 31. LU6.2 と ATTACHSEC(IDENTIFY)、ATTACHSEC(PERSISTENT)、および ATTACHSEC(MIXIDPE)

AOR の領域ユーザー **ID**	接続定義の **SECURITYNAME**	**SESSION** 定義の **USERID**	**AOR** での検査
USERIDA	指定されていません	指定されていません	送信済みユーザー ID および AOR DFLTUSER
USERIDA	指定されていません	USERIDA	送信済みユーザー ID のみ
USERIDA	指定されていません	USERIDB	送信済みユーザー ID および USERIDB
USERIDA	USERIDA	指定されていません	送信済みユーザー ID のみ
USERIDA	USERIDA	USERIDA	送信済みユーザー ID のみ
USERIDA	USERIDA	USERIDB	送信済みユーザー ID および USERIDB
USERIDA	USERIDB	指定されていません	送信済みユーザー ID および USERIDB
USERIDA	USERIDB	USERIDC	送信済みユーザー ID および USERIDC

LU6.1 セキュリティーの実装

このトピックでは、LU6.1 のリンク・セキュリティーを実装する方法を説明します。

LU6.1 リンクの場合、CICS は要求元システムの ID を検査できず、セキュリティーを理由にバインド要求が拒否されることはありません。可能な場合は、LU6.2 リンクによって提供されるシステム間セキュリティーを使用することをお勧めします。バインド時のセキュリティーおよびユーザー・セキュリティーは、LU6.1 リンクに適用できないことに注意してください。

LU6.1 でのリンク・セキュリティー

リンク・セキュリティーは、ユーザーがアクセスできるリソースを、アクセス元のリモート・システムに応じて制限します。

リンク・セキュリティーの実際的な効果として、リモート・ユーザーによるトランザクションへの接続や、リンク・ユーザー ID が権限を持たないリソースへのアクセスを抑止します。

システム間の各リンクには、リンク・ユーザー ID によって定義されたアクセス権限が付与されます。 LU6.1 のリンク・ユーザー ID は、この接続のためのセッション定義で定義されたユーザー ID です。そこで定義されない場合、リンク・ユーザー ID は、接続定義に指定された SECURITYNAME ユーザー ID であると見なされます。 SECURITYNAME がない場合、リンク・ユーザー ID はローカル領域のデフォルト・ユーザー ID です。

セキュリティー検査を行わずに CICS に機能シップすることはできません。ただし、リンク・ユーザー ID がローカル領域のユーザー ID に一致する場合、セキュリティー検査は最小化されます。

- ユーザー ID が一致する場合、ローカル領域のデフォルト・ユーザーに対してリソース検査が実行されます。
- ユーザー ID が一致しない場合、リンク・ユーザー ID に対してリソース検査が実行されます。

リンク・セキュリティーを確立する際に障害が生じると、ローカル領域のデフォルト・ユーザーのセキュリティーがリンクに提供されます。これは例えば、事前設定セッションのユーザー ID が取り消された場合に起きる可能性があります。

LU6.1 接続のリンク・セキュリティーの指定

このタスクについて

CONNECTION リソース定義または SESSIONS リソース定義でリンク・セキュリティーを指定します。これらのリソースのいずれかについての詳細情報が必要な場合は、CONNECTION リソースおよびSESSIONS リソースを参照してください。

手順

- 接続のすべてのセッションが同じリンク・ユーザー ID を持つように指定するには、CONNECTION リソース定義の SECURITYNAME 属性を指定します。この属性の値を指定しない場合、CICS はデフォルト・ユーザー ID を使用します。
- 接続内のセッションの個々のグループに別々のリンク・ユーザー ID を指定するには、SESSIONS リソース定義の USERID 属性を指定します。セッションの各グループで、指定された値は CONNECTION 定義の SECURITYNAME 属性をオーバーライドします。

LU6.1 での ATTACHSEC の指定

LU6.1 リンクの使用時には、リモート・ユーザーに関する情報はセキュリティーのために入手できません。この場合、ユーザーの権限はリンク自体の権限であると見なされ、リソースの保護はリンク・セキュリティーのみに依存しなければならなくなります。

LU6.1 の使用時には、CONNECTION リソース定義に指定できるのは ATTACHSEC(LOCAL) のみです。 図 11 は、CEDA を使用してこれを実行する例を示しています。

```
CEDA DEFINE CONNECTION(name)
  GROUP(groupname)
  .
  ATTACHSEC(LOCAL)
```

図 *11. LU6.1* でのユーザー・セキュリティーのサインオン・レベルの定義

LOCAL がデフォルト値です。これは、リモート・システムからのユーザー ID が不要であり、それを受け取った場合でも無視することを示します。ここで CICS は、リンク・セキュリティー・プロファイルと同等のユーザー・セキュリティー・プロファイルを作成します。リモート・ユーザーに対して RACF プロファイルを指定する必要はありません。

LU6.1 でのトランザクション・セキュリティー、リソース・セキュリティー、およびコマンド・セキュリティー

単一システム環境の場合と同様に、リンクには以下を行うための許可が必要です。

- トランザクションを接続する。
- トランザクションが使用するようにプログラムされているすべてのリソースにアクセスする。

これにより、トランザクション・セキュリティー、リソース・セキュリティー、およびコマンド・セキュリティーと呼ばれるセキュリティー・レベルが実現します。

トランザクション・セキュリティー

単一システム環境の場合、トランザクションのセキュリティー要件はトランザクションを定義する際に示されます (トランザクション・セキュリティー を参照)。

LU6.1 環境では、トランザクションはリンクに十分な権限がある場合にのみ開始できます。

リソースおよびコマンド・セキュリティー

相互通信環境でのリソース・セキュリティーおよびコマンド・セキュリティーは、単一システム環境とほぼ同じ方法で処理されます。

CICS がリソースおよびコマンドのセキュリティー検査を実行するのは、インストールされているトランザクション定義でその検査が必須であると (例えば、図 12 に示すように CEDA DEFINE TRANSACTION コマンドなどで) 指定されている場合のみです。

```
CEDA DEFINE TRANSACTION
  .
  RESSEC(YES)
  CMDSEC(YES)
  .
```

図 *12.* トランザクションのリソースおよびコマンド・セキュリティーの指定

トランザクション定義でリソース・セキュリティー検査が RESSEC(YES) を使用して指定されている場合、リンクには、接続されたトランザクションがアクセスするリソースに対する十分な権限が備えられている必要があります。

トランザクション定義でコマンド・セキュリティー検査が CMDSEC(YES) を使用して指定されている場合、リンクには、接続されたトランザクションが発行するコマンド (COLLECT、DISCARD、INQUIRE、PERFORM、および SET) に対する十分な権限が備えられている必要があります。

リソースおよびコマンド・セキュリティーの指定についての詳しい説明は、リソース定義のセキュリティー および リソース定義のセキュリティー を参照してください。

NOTAUTH 例外状態**:**

トランザクションがリソースへのアクセスを試行したが、リソース・セキュリティー検査に失敗した場合、NOTAUTH 条件が発生します。

トランザクションが CICS ミラー・トランザクションである場合、NOTAUTH 条件は、それを通常の方法で処理できる要求元トランザクションに返されます。

LU6.1 での機能シップ・セキュリティー

CICS が機能シップ要求を受け取るときに、呼び出されるトランザクションはミラー・トランザクションです。

ミラー・トランザクションの CICS 提供の定義はすべて、リソース・セキュリティー検査を指定しますが、コマンド・セキュリティー検査は行いません。これは、リンクが必要な権限を持っていない場合に、リモート・リソースにアクセスできないことを意味します。

LU6.1 リンクを経由したトランザクション・ルーティングはサポートされないことに注意してください。

CICS 提供のミラー・トランザクションの定義がセキュリティー戦略のニーズに適していない場合は、グループ DFHISC 内の定義を独自のグループにコピーし、変更し、再インストールすることで変更できます。詳しくは、Security for CICS-supplied transactionsを参照してください。

リソース定義にリモート・リソースを含める場合、ローカル・リソースの場合と同じように、ローカルでセキュリティー検査が実行されるように調整できます。さらに、リソースを所有するシステムは、ユーザー ID を受け取ることができる場合、独立検査を適用するようにすることができます。したがって、セキュリティー制限の適用は、両方の側または一方の側に行うか、あるいはどちらにも行わないという選択ができます。

注: 機能シップされた要求に対して SYSID オプションを指定する場合、セキュリティー検査は、リモート・システムでは実行されますが、ローカル・システムではバイパスされることに注意してください。 302 ページの図 13 では、どのような動作になるかを要約しています。

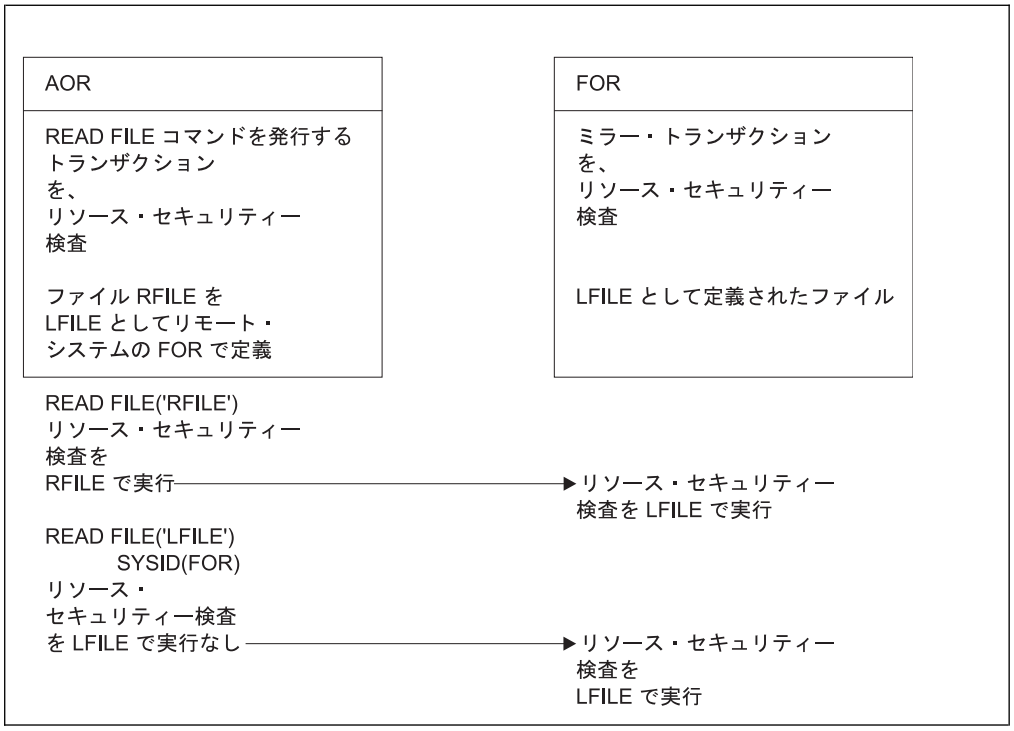


図 *13. SYSID* の指定ありまたは指定なしで実行されるセキュリティー検査： この例は、アプリケーション専有領域内のトランザクションがリモート・ファイルに対してファイル制御要求を発行するときに、どのようなセキュリティー検査が実行されるかを示しています。

- アプリケーション専有領域 (AOR) 内で、ファイル RFILE は、ファイル専有領域 (FOR) 内の LFILE という名前で、リモートとして定義されます。ファイル制御要求を発行するトランザクションに対して、リソース・セキュリティー検査がアクティブです。
- FOR 内では、ミラー・トランザクションに対してリソース・セキュリティー検査がアクティブです。

これには 2 つのケースがあります。最初のケースは次のとおりです。

1. AOR 内のトランザクションが EXEC CICS READ FILE('RFILE') を発行します。リソース・セキュリティー検査がファイル RFILE に対して実行されます。
2. 要求は FOR に送信され、そこではリソース・セキュリティー検査がファイル LFILE に対して実行されます。

2 番目のケースは次のとおりです。

AOR 内のトランザクションが、EXEC CICS READ FILE('RFILE') を *SYSID* オプションを指定して 発行します。リソース・セキュリティー検査はファイル RFILE に対して実行されません。

要求は FOR に送信され、そこではリソース・セキュリティー検査がファイル LFILE に対して実行されます。

LU6.1 での AOR で実行されるセキュリティー検査

このセクションでは、LU6.1 環境の AOR および TOR で SECURITYNAME がどのように指定されているかに応じた、AOR でのセキュリティー検査の実行方法について概要を示します。

表 32 に示されているリンク・ユーザー ID は、CONNECTION 定義の SECURITYNAME、または SESSIONS 定義の USERID で指定されているものです。

ユーザー ID が SESSIONS 定義で指定され、リンク検査が行われる場合、使用されるユーザー ID は SESSIONS 定義のユーザー ID です。

表 32 は、ATTACHSEC(LOCAL) を指定した場合の検査方法を示しています。

TOR の領域ユーザー **ID**、および **AOR** に対する **TOR** の **CONNECTION** 定義の **SECURITYNAME** はどちらも、**AOR** のセキュリティー検査には関係しません。

表 *32. AOR* で実行されるセキュリティー検査

AOR の領域ユーザー **ID**	**CONNECTION** 定義の **SECURITYNAME**	**SESSION** 定義の **USERID**	**AOR** での検査
USERIDA	指定されていません	指定されていません	AOR DFLTUSER に対する検査
USERIDA	指定されていません	USERIDA	AOR DFLTUSER に対する検査
USERIDA	指定されていません	USERIDB	USERIDB に対する検査
USERIDA	USERIDA	指定されていません	AOR DFLTUSER に対する検査
USERIDA	USERIDB	指定されていません	USERIDB に対する検査
USERIDA	USERIDA	USERIDA	AOR DFLTUSER に対する検査
USERIDA	USERIDA	USERIDB	USERIDB に対する検査
USERIDA	USERIDB	USERIDA	AOR DFLTUSER に対する検査
USERIDA	USERIDB	USERIDB	USERIDB に対する検査
USERIDA	USERIDB	USERIDC	USERIDC に対する検査

APPC パスワード有効期限管理

このトピックには、拡張プログラム間通信機能 (APPC) のパスワード有効期限管理 (PEM) に関する情報が記載されています。

APPC パスワード有効期限管理の概要

CICS での APPC パスワード有効期限管理 (PEM) は、APPC 体系化サインオン・トランザクションの受信サポートを提供します。

サンプル・プログラムをコピーして変更すると便利です。ガイダンス用に、付属のサンプル・プログラムがライブラリー CICSTS55.CICS.SDFHSAMP に含まれています。プログラムは DFH$SNPW で、Windows NT 用の PEM サンプル・プログラムです。

PEM リクエスターの例、およびプログラムによって作成される CICS PEM サーバー・ユーザー・データの例については、PEM クライアントおよび CICS PEM サーバーのユーザー・データの例を参照してください。

注**:** APPC PEM に関する情報では、サインオン という用語は APPC アーキテクチャーで定義された意味で使用されており、この情報の他の場所で使用されている意味とは異なっています。

APPC PEM の実行内容

CICS での APPC PEM は、APPC 体系化サインオン・トランザクションの受信サポートを提供します。このトランザクションは、ユーザー ID とパスワードのペアを検証し、以下によってパスワード変更要求を処理します。

- ユーザーを識別し、そのユーザーの ID を認証する
- 認証処理中の特定のユーザーにパスワードの有効期限が切れたことを通知する
- パスワードの有効期限が切れたとき (または切れる前) にユーザーにパスワードを変更させる
- 現在のパスワードが有効である期間をユーザーに通知する
- 特定のユーザー ID を使用したシステムへの無許可のアクセス試行に関する情報を提供する

APPC PEM の利点

APPC PEM には、以下の利点があります。

- ユーザーは APPC リンクのパスワードを更新できます。

  これは期限切れパスワードがある場合に特に便利です。 APPC PEM をサポートしない APPC リンク上では、リモート・システムでユーザーのパスワードの有効期限が切れた場合、その固有のシステムからパスワードを更新することはできません。非 APPC PEM システム上での唯一の代替手段は、非 APPC リンク (LU2 3270 エミュレーション・セッションなど) を使用してリモート・システムに直接ログオンして、パスワードを更新することです。
- これにより APPC ユーザーには、サインオン状況に関する追加情報 (最後にサインオンした日時など) が提供されます。
- ユーザーが正しいパスワードまたはパスチケットを入力すると、ユーザー ID が取り消されているかどうか、またはパスワードの有効期限が切れているかどうかがユーザーに通知されます。

APPC PEM を使用するために必要なこと

APPC PEM を使用するには、以下が必要です。

PEM クライアント
: PEM クライアントは、体系化されたサインオン・トランザクションとの会話を開始できる、任意の APPC 論理装置 (LU) またはノードにできます。

ただし、APPC PEM を使用することの利点は、独自の ESM を持たない LU またはノード (例えばプログラマブル・ワークステーションなど) を使用する場合に大きくなります。 APPC PEM により、そのような LU またはノードのユーザーは、CICS が使用する ESM 内で自分のパスワード値を管理できます。 PEM クライアントは *PEM* サーバー にリンクされます。

PEM サーバー
PEM サーバーは、APPC PEM をサポートする任意の APPC LU とすることができます。この情報では、CICS Transaction Server for z/OS, バージョン 5 リリース 5 が提供する PEM サーバーについて説明します。これ以降は、CICS PEM サーバーと呼びます。

外部セキュリティー・マネージャー
RACF や同等の ESM などの外部セキュリティー・マネージャーは、PEM サーバーで使用できるようにする必要があります。

PEM クライアント (リクエスター) と PEM サーバーは、APPC セッションによってリンクされます。 この構成を、図 14に示します。

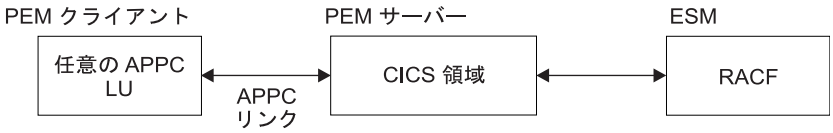


図 *14. APPC PEM* 構成

外部セキュリティー・インターフェース

パスワード有効期限管理は拡張され、外部セキュリティー・インターフェース (ESI) が組み込まれました。

ESI は CICS Transaction Server for z/OS の一部ではありませんが、それによって非 CICS アプリケーションは、拡張 APPC PEM で提供されるサービスを呼び出すことができます。 ESI は、非 CICS アプリケーションによるパスワードの変更や検査をしやすくするための追加機能を提供します。

ESI が提供する 2 つの機能は、以下のとおりです。

- **CICS_VerifyPassWord**。これによりクライアント・アプリケーションは、指定されたユーザー ID のパスワードが、RACF または同等の外部セキュリティー・マネージャーによって記録されたパスワードと一致していることを確認できます。
- **CICS_ChangePassWord**。これによりクライアント・アプリケーションは、RACF によって記録された、指定されたユーザー ID のパスワードを変更できます。

これらの機能により、PEM 要求と応答の形式についての知識、およびローカル SNA サーバーへのインターフェースについての知識を暗に示す APPC 会話をアプリケーション・プログラマーが管理しなくても、非 CICS アプリケーション・プログラムは PEM 要求側として機能することができます。

ESI パスワード管理機能の詳細については、「*Client/Server Application Programming*」を参照してください。

PEM クライアントと CICS PEM サーバーの役割

CICS Transaction Server for z/OS, バージョン 5 リリース 5 は APPC 会話ではパスワードを送信しません。つまり、サインオン・トランザクションを接続できますが、開始はできず、必ず PEM サーバーとして機能する必要があります。したがって、構成内には常に、PEM クライアントとして機能するサインオン・トランザクションを開始できる LU が含まれます。

PEM クライアントはサインオン情報を収集し、それを SNA サービス・トランザクション・プログラムであるサインオン・トランザクション・プログラムによって CICS PEM サーバーに送信します。

PEM サインオンを CICS サインオンと混同しないように注意してください。 CICS では、PEM サインオンにより、APPC LU はユーザー ID とパスワードを検査および管理できます。検査または更新の後には、ユーザー ID またはパスワードは FMH5 付加ヘッダーの ASIS 部分に組み込まれることになります。この FMH5 が APPC リンクを介して CICS に送信されると、ATTACHSEC が非ローカルである場合、このユーザー ID で CICS にサインオンします。したがって、PEM サインオンによって ESM の最終接続情報と最終アクセス情報が更新されるという結果にはなりません。

CICS PEM サーバーは、サインオン要求を処理し、ユーザーのパスワードを更新し (必要な場合)、応答とその他のデータ (パスワード失効や、無許可サインオン試行に関する情報など) を含む応答を PEM クライアントに返します。次に PEM クライアントは、必要に応じてデータを処理します。

注**:** CICS Transaction Server for z/OS バージョン 2 リリース 1 でパスワードの検査が成功しても、ターゲット CICS システムにはサインオンしていません。

CICS 接続で持続検査が指定されている場合は、パスワードの検査が成功すると、ユーザーは LUIT テーブルに追加されます。他の接続を受け取っていない場合は、PVDELAY インターバルの後に CLS3 トランザクション・フローを受け取ります。

APPC PEM でのサインオンの例

この例は、ユーザーが有効期限切れのパスワードを使用して PEM クライアントにサインオンする際の一連のイベントを示しています。

308 ページの図 15 には、一連の流れが示されています。

1. ユーザーは PEM クライアントにサインオンしようとし、パスワードを入力します。
2. PEM クライアントは、CICS PEM サーバーにユーザー ID とパスワードを送信します。
3. CICS PEM サーバーは、外部セキュリティー・マネージャーにユーザー ID とパスワードを渡します。
4. 外部セキュリティー・マネージャーは、パスワードを検査します。

5. パスワードの有効期限が切れていたため、サインオン試行は失敗します。外部セキュリティー・マネージャーが CICS PEM サーバーに通知し、次に CICS PEM サーバーが PEM クライアントに通知します。
6. PEM クライアントはユーザーに、パスワードの有効期限が切れていたことを通知し、新規パスワードを要求します。
7. ユーザーは、新規パスワードを入力します。
8. PEM クライアントは CICS PEM サーバーに、ユーザー ID、古いパスワード、および新規パスワードを送信します。
9. CICS PEM サーバーは外部セキュリティー・マネージャーに、ユーザー ID、古いパスワード、および新規パスワードを渡します。
10. 外部セキュリティー・マネージャーは、新旧両方のパスワードを検査します。新しいパスワードが提供されているので、サインオン試行は成功します。外部セキュリティー・マネージャーが CICS PEM サーバーに通知し、次に CICS PEM サーバーが PEM クライアントに通知します。
11. PEM クライアントは、サインオン要求が成功したことをユーザーに通知します。

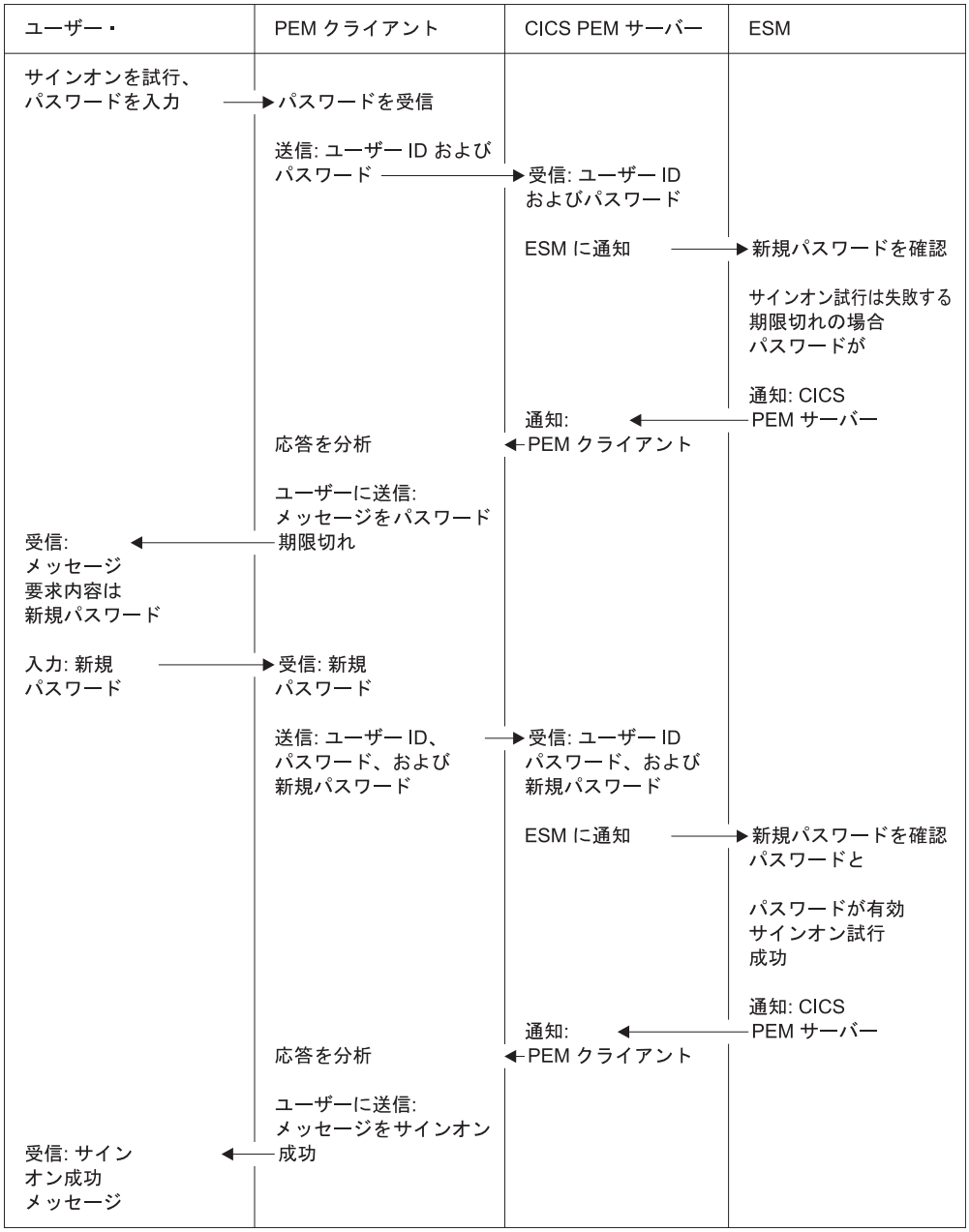


図 *15. APPC PEM* でのサインオンの例

APPC PEM 処理の概要

CICS は PEM サーバーを提供します。これは、サインオン・トランザクション・プログラムに対する ATTACH を PEM クライアントから受け取ったときに開始される CICS トランザクションとしての APPC PEM の受信側です。

CICS は、要求に関連付けられているサインオン・データを取得し、サインオンを実行する ESM を呼び出し、ユーザー ID のサインオン詳細を取得します。サインオン・データに新規パスワード値が含まれている場合、CICS は、サインオンを要求する ESM を呼び出すときに、この値を含めます。

PV が使用されていて、サインオンが正しく実行されている場合、ユーザーは CICS の PV「サインオン元」リストと、PEM クライアントの PV「サインオン先」リストに追加されます。これらの「サインオン」リストは、検証済みユーザー ID を追跡します。

CICS PEM サーバーは応答を作成し、それを PEM クライアントに返し、その後で CICS PEM サーバー・トランザクションは正常に終了します。

PEM クライアント処理

PEM クライアント・サインオン・トランザクション・プログラムは、以下のことを行います。

1. サインオン情報を、例えば次のようにして入手します。
   - サインオン情報 (ユーザー ID、パスワード、必要な場合には新規パスワード) を要求するメッセージをユーザーに表示する。
   - 既にローカルで認証されているユーザーからのサインオン情報にアクセスする。
2. APPC 会話を介して CICS PEM サーバーにサインオン情報を送信します。
3. 同じ APPC 会話で、CICS PEM サーバーから応答を受け取ります。
4. PV が使用されており (ATTACHSEC=PERSISTENT または ATTACHSEC=MIXIDPE のいずれかが CONNECTION 定義で指定されている)、サインオンが成功した場合は、ユーザーの名前を PV サインオン先リストに追加します。
5. CICS PEM サーバーから応答情報を処理します。例えば、次のようにします。
   - ユーザーに情報を表示する。
   - データを処理し、ユーザーのみがアクセスできるファイルに保管する。

CICS PEM サーバー処理

CICS PEM サーバーは、以下の処理を実行します。

1. サインオン PEM クライアントから、ユーザー ID とパスワードをオプションの新規パスワードとともに受け入れます。
2. その ESM を使用して、ユーザーの妥当性検査を試行します。

   ユーザー ID とパスワードが有効で、パスワードの有効期限が切れていない場合は、CICS PEM サーバーはその ESM から以下の情報を抽出します。
   - 最後の正常なサインオンの日時。
   - パスワードの有効期限が切れる日時 (CICS PEM サーバー自体によって ESM から抽出されたデータで計算)
   - 最後の正常なサインオン以降、サインオン試行が失敗した回数。
3. サインオンが成功か失敗か、および失敗の場合はその理由を示す応答を PEM クライアントに返します。

```
Status           = (X'00') OK
Date-Time        = Current date and time
Last-Date-Time   = Date and time of previous successful sign-on
Expiry-Date-Time = Date and time password will expire
Revoke-Count     = Number of unsuccessful sign-on attempts made with
                   this userid since the previous successful sign-on
```

応答について詳しくは、PEM クライアントの入出力データを参照してください。

注: ESM は、無効なサインオン試行を処理する場合はいつでも、取り消しカウントを増分します。サインオン要求は、非 CICS システム (例えば、TSO ユーザー) から出される場合があります。

サインオンが失敗した場合、CICS は PEM クライアントにサインオン完了状況値と、必要に応じてフォーマット設定エラー値を返します。詳細については、PEM クライアントの入出力データを参照してください。

4. PV が使用されており (ATTACHSEC=PERSISTENT または ATTACHSEC=MIXIDPE のいずれかが CONNECTION 定義で指定されている)、サインオンが成功した場合は、ユーザーの名前を PV サインオン元リストに追加します。

PEM クライアントと CICS PEM サーバーの間の予期されるフロー

311 ページの図 16 から 314 ページの図 19 は、正常および失敗のサインオン試行 (PV ありおよび PV なし) の予期されるフローを示しています。

注: PEM クライアント・サインオン・トランザクションの CICS サポートは、サインオン (またはサインオンとパスワード変更) の要求が、単一ユーザーのものと想定しています。シングル・サインオン・トランザクション内での、複数のユーザー ID のサインオン要求のバッチ処理はサポートされません。複数のサインオン要求が入力データに渡される場合、CICS PEM サーバーは最初の要求のみを処理します。

正常なサインオン — 非 **PV** 接続: 311 ページの図 16 は、PV が使用されない場合の正常なサインオン時の、PEM クライアントと CICS PEM サーバーの間の予期されるフローを示しています。

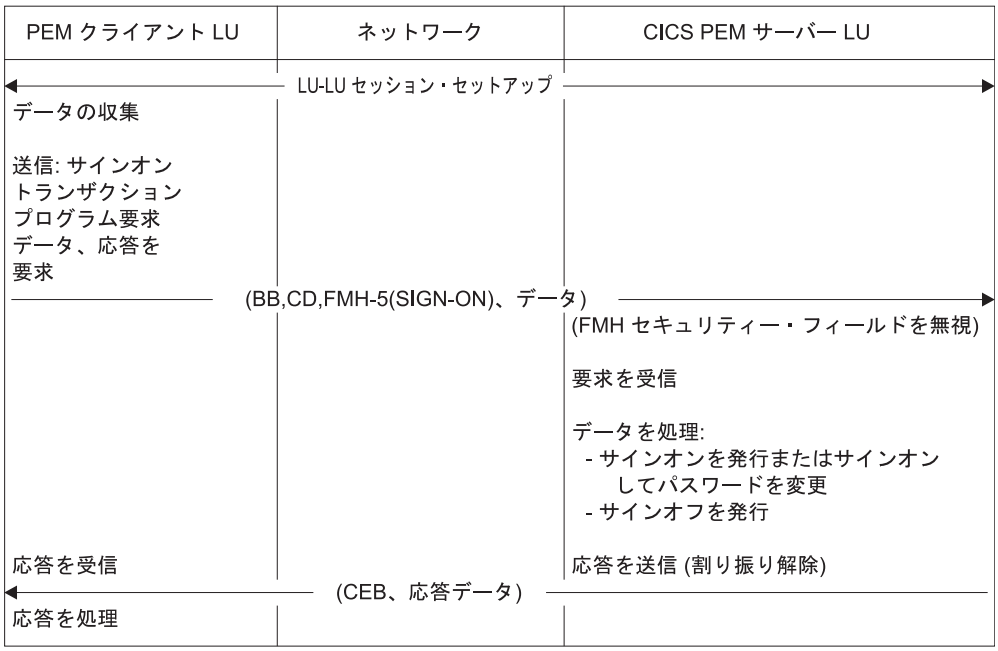


図 *16.* 非 *PV* 接続での正常なサインオン時の予期されるフロー： 予期されるイベントの順序は以下のとおりです。

1. PEM クライアント論理装置と CICS PEM サーバー論理装置の間のセッションが確立されている。
2. PEM クライアントはデータを収集し、サインオン・プログラム要求のデータを応答を要求しているサーバーに送信する。要求フローは BB、CD、FMH5、およびデータを含んでいる。
3. CICS PEM サーバーは FMH5 のセキュリティー・フィールドを無視している。これは、サインオンを発行するデータ (または変更されたパスワードによるサインオン) を処理します。
4. サーバーは応答を送信し、セッションの割り振りを解除します。応答フローは CEB およびデータを含んでいる。
5. クライアントが応答を受信して処理している。

注**:** FMH-5 のすべてのセキュリティー・フィールド (ユーザー ID、パスワード、UP、AV、PV1、および PV2 の各ビット) は、サインオン・トランザクションの接続時に、CICS PEM サーバーによって無視されます。

失敗したサインオン **—** 非 **PV** 接続**:** 312 ページの図 17 は、PV が使用されない場合の失敗したサインオン時の、PEM クライアントと CICS PEM サーバーの間の予期されるフローを示しています。

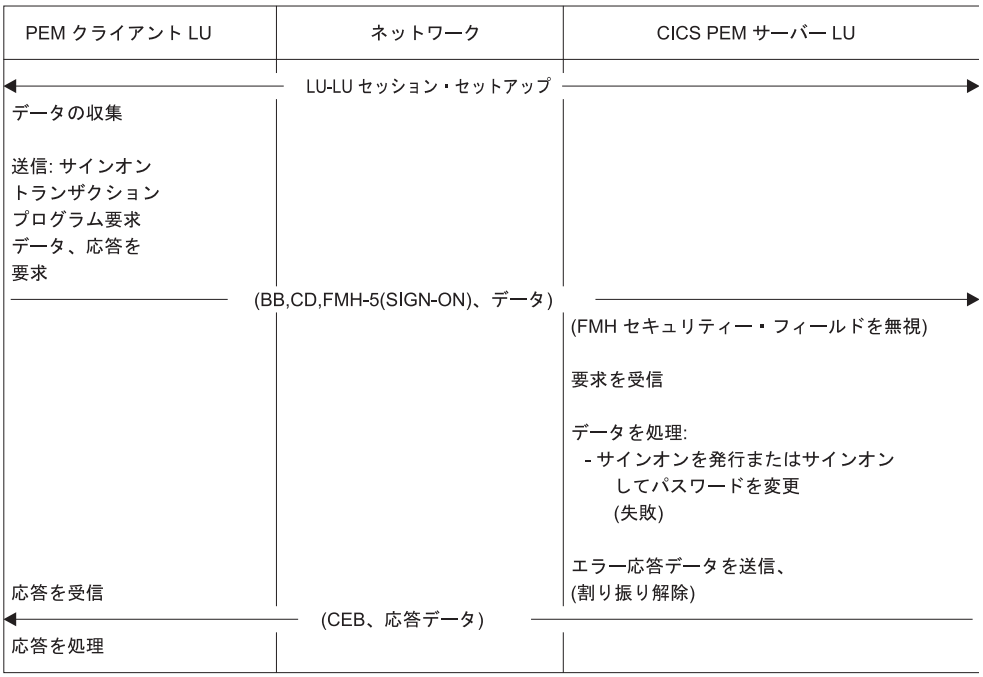


図 *17.* 失敗したサインオン — 非 *PV* 接続： 予期されるイベントの順序は以下のとおりです。

1. PEM クライアント論理装置と CICS PEM サーバー論理装置の間のセッションが確立されている。
2. PEM クライアントはデータを収集し、サインオン・プログラム要求のデータを応答を要求しているサーバーに送信する。要求フローは BB、CD、FMH5、およびデータを含んでいる。
3. CICS PEM サーバーは FMH5 のセキュリティー・フィールドを無視している。これは、サインオンを発行するデータ (または変更されたパスワードによるサインオン) を処理します。要求は失敗します。
4. サーバーはエラー応答を送信し、セッションの割り振りを解除する。応答フローは CEB およびデータを含んでいる。
5. クライアントが応答を受信して処理している。

注**:** ユーザー ID のサインオン要求が失敗した場合、CICS PEM サーバーは PEM クライアントに対してサインオフをスケジュールします。

正常なサインオン **— PV** 接続**:** 313 ページの図 18 は、PV 接続での正常なサインオン時の PEM クライアントと CICS PEM サーバーの間の予期されるフローを示しています。

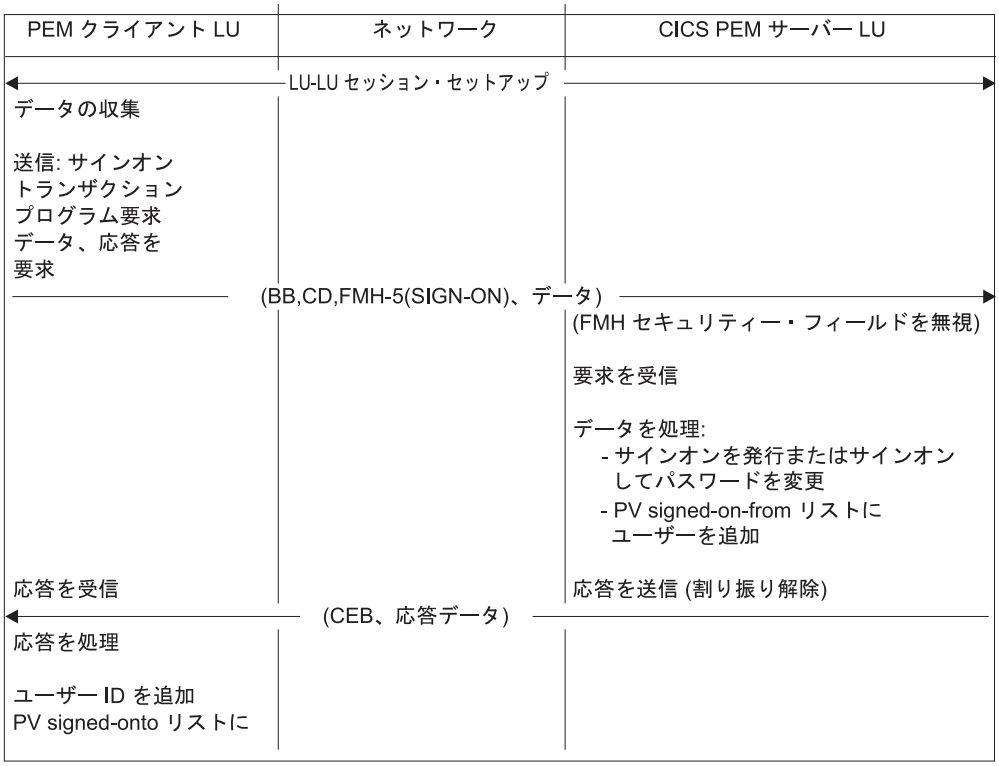


図 *18.* 正常なサインオン — *PV* 接続： 予期されるイベントの順序は以下のとおりです。

1. PEM クライアント論理装置と CICS PEM サーバー論理装置の間のセッションが確立されている。
2. PEM クライアントはデータを収集し、サインオン・プログラム要求のデータを応答を要求しているサーバーに送信する。要求フローは BB、CD、FMH5、およびデータを含んでいる。
3. CICS PEM サーバーは FMH5 のセキュリティー・フィールドを無視している。これは、サインオンを発行するデータ (または変更されたパスワードによるサインオン) を処理します。
4. サーバーは、PV サインオン元リストにユーザーを追加する。
5. サーバーは応答を送信し、セッションの割り振りを解除する。応答フローは CEB およびデータを含んでいる。
6. クライアントが応答を受信して処理している。
7. クライアントは、サインオン先リストにユーザー ID を追加する。

注**:**

1. FMH-5 のすべてのセキュリティー・フィールド (ユーザー ID、パスワード、UP、AV、PV1、および PV2 の各ビット) は、サインオン・トランザクションの接続時に、CICS PEM サーバーによって無視されます。
2. CICS PEM サーバーは、サインオンおよびパスワード変更要求が正常に実行され、CONNECTION 定義に ATTACHSEC=MIXIDPE または ATTACHSEC=PERSISTENT のいずれかが指定されている場合にのみ、ユーザー ID をその PV サインオン元リストに追加します。
3. PEM クライアントは、CICS PEM サーバーから正常なサインオン応答を受け取った場合にのみ、ユーザー ID をその PV サインオン先リストに追加する必要があります。ユーザー ID は CICS PEM サーバーの PV サインオン元リス

トに追加されているため、このユーザー ID からの後続のすべての接続要求は、サインオン済みとしてフローすることができます。

失敗したサインオン **— PV** 接続**:** 図 19 は、PV 接続での失敗したサインオン時の PEM クライアントと CICS PEM サーバーとの間の予期されるフローを示しています。

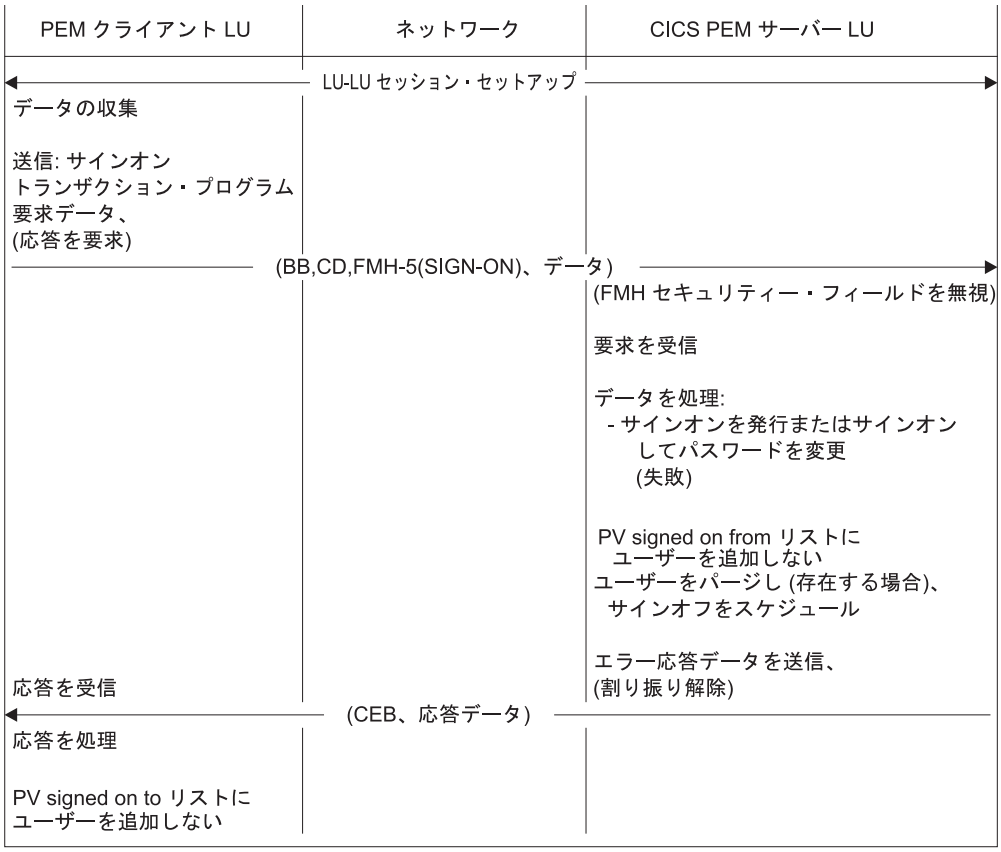


図 *19.* 失敗したサインオン *— PV* 接続： 予期されるイベントの順序は以下のとおりです。

1. PEM クライアント論理装置と CICS PEM サーバー論理装置の間のセッションが確立されている。
2. PEM クライアントはデータを収集し、サインオン・プログラム要求のデータを応答を要求しているサーバーに送信する。要求フローは BB、CD、FMH5、およびデータを含んでいる。
3. CICS PEM サーバーは FMH5 のセキュリティー・フィールドを無視している。これは、サインオンを発行するデータ (または変更されたパスワードによるサインオン) を処理します。要求は失敗します。
4. サーバーは、サインオン元リストにユーザーを追加しない。ユーザーが既にリスト上にある場合、そのユーザーはリストから消去され、サインオフがスケジュールされます。
5. サーバーはエラー応答を送信し、セッションの割り振りを解除する。応答フローは CEB およびデータを含んでいる。
6. クライアントが応答を受信して処理している。
7. ユーザーはサインオン先リストには追加されない。

注**:** ユーザー ID のサインオン要求が失敗し、そのユーザーが PV サインオン元リストにある場合、CICS は PEM クライアントに対してサインオフをスケジュール

します。この場合、CICS はサインオフ・トランザクション・プログラム出力データを PEM クライアントに送信します。そこでそのデータは処理され、ユーザー ID は PV サインオン先リストから削除されます。

PEM クライアントのセットアップ

PEM クライアントをセットアップするときは、次の点に注意してください。

- 基本の (非マップ式としての とも呼ばれる) 会話タイプを使用します。パートナーに受信させるデータを送信することに加え、制御バイト (アセンブラー言語または C) を追加して、データを汎用データ・ストリーム (GDS) と呼ばれる SNA 定義形式に変換する必要があります。
- サインオン・トランザクション・プログラムの SNA サービス・トランザクション・プログラム名は **X'06F3F0F1'** です。これは、CICS トランザクション CLS4 に使用する必要があるトランザクション ID (XTRANID) でもあります。 CICS TRANSACTION 定義に XTRANID を指定します。
- CICS PEM サーバー・サインオン・トランザクションは、同期レベル **0** トランザクションとして実行します。それが 0 以外の同期レベルで開始されると、ISSUE ABEND が送信され、会話が解放されます。
- ユーザー ID とパスワードを EBCDIC に変換します。これらが EBCDIC でない場合、 ESM はこれらを認識できず、エラーを発行します。ユーザー ID とパスワードを EBCDIC に変換する例については、APPC パスワード有効期限管理の概要で説明されている、いずれかのサンプル・プログラムを参照してください。

  ESM が RACF である場合はさらに、ユーザー ID とパスワードは大文字でなければなりません。
- サインオン・トランザクション・プログラムの ATTACH 要求で SECURITY(NONE) を指定します。 CICS は、このトランザクションの ATTACH 機能管理ヘッダー FMH-5 で渡されるすべての ATTACH セキュリティー・フィールドを無視します。
- CICS は、サインオン・トランザクション・プログラムの PROFILE オプションの受け取りをサポートしません。データ ID (ID) X'00' が提供されている場合、CICS は、状況値 X'06' (データ・フォーマットが誤っている) を、フォーマット設定エラー X'0002' (除外された構造が存在している) とともに返します。状況値およびフォーマット設定エラー値については、 317 ページの『PEM クライアントの入出力データ』 を参照してください。
- 新規パスワード ID X'06' は、X'FF01' 要求データ ID とともに使用する場合にのみ許可され、必要になります。新規パスワードが X'FF01' 以外のデータ ID とともに提供されている場合、CICS は、状況値 X'06' (データ・フォーマットが誤っている) を、フォーマット設定エラー X'0002' (除外された構造が存在している) とともに返します。状況値およびフォーマット設定エラー値については、317 ページの『PEM クライアントの入出力データ』 を参照してください。
- CICS は、長さが 8 文字以下のユーザー ID とパスワードのみをサポートします。ユーザー ID またはパスワードの長さが (ブランクとヌルを除去した後に) 8 文字を超えている場合、CICS は状況値 X'06' (データ・フォーマットが誤っている) を、フォーマット設定エラー X'000F' (データ値が範囲外) とともに返しま

す。状況値およびフォーマット設定エラー値については、 317 ページの『PEM クライアントの入出力データ』 を参照してください。

- プログラム初期設定パラメーター (PIP) データは、サインオン・トランザクションの ALLOCATE ではオプションであり、送信しても無視されます。
- サインオン・トランザクションは、GDS ISSUE SIGNAL コマンドを受け取っても、それを無視します。
- CICS PEM サーバーは、GDS ISSUE ERROR コマンドを受け取ると、ERROR で応答して会話を解放します。
- CICS PEM サーバーは、GDS FREE コマンドを受け取ると、会話を解放します。 (これは会話エラーのタイプに関する診断情報を提供しません。)
- CICS PEM サーバー・トランザクションは、最大バッファー・サイズを超えるデータの受け取りをサポートしません。初期 LL 内の連結ビットが設定されている場合、コマンドは無視されます。連結されたデータも無視されます。

ユーザー・データのフォーマット

APPC 基本会話の一般ルールの一部として、ユーザー・データは LL-ID データ・フォーマットでなければならず (LL と ID はそれぞれ 2 バイト長)、付加 FMH-5 ヘッダーの後に続けなければなりません。

CICS DFHCLS4 プログラムでは、ユーザー入力データ・ストリームは、 317 ページの図 20 に示すフォーマットに適合している必要があります。そうでない場合、CICS はそのデータを拒否します。

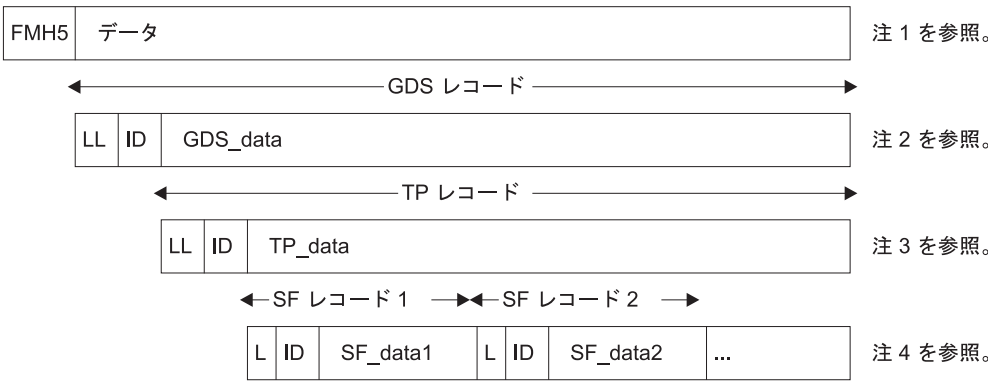


図 *20.* ユーザー・データのフォーマット：

注**:**

1. これは、データがある付加 FMH-5 ヘッダーです。データは、GDS 変数を介して PEM クライアントと CICS PEM サーバーとの間で受け渡されます。 GDS については、z/OS Communications Server: SNA プログラマーズ LU 6.2 ガイドを参照してください。
2. **GDS** レコードには、LL - ID データ・フォーマットの GDS データが含まれます。それぞれの意味は以下のとおりです。
   - `LL` (2 バイト長) は、GDS レコードの長さであり、LL と ID の長さも含まれます。
   - `ID` (2 バイト長) は、データ・レコードが表す内容を示します。例えば、X'1221' (サインオン・データ) です。
3. GDS データ・レコード自体は、LL-ID データ・レコードです。この例では、トランザクション・プログラム・レコード (または **TP** レコード) です。それぞれの意味は以下のとおりです。
   - `LL` (2 バイト長) は、TP レコードの長さであり、LL と ID の長さも含まれます。
   - `ID` (2 バイト長) は、TP が実行する機能を示します。例えば、X'FF00' (サインオン) または X'FF01' (サインオンとパスワード変更) です。
4. TP データ・レコードは、L-ID データ・レコードに分割されます (ここで、L と ID はそれぞれ **1** バイト長です)。これらは、サブフィールド (または **SF** レコード) と呼ばれています。それぞれの意味は以下のとおりです。
   - `L` は SF レコードの長さであり、`L` と `ID` の長さも含まれます。
   - `ID` は、渡されるサブフィールドを示します。例えば、X'01' (ユーザー ID)、X'02' (パスワード)、X'06' (新規パスワード) です。

PEM クライアントの入出力データ

CICS PEM サーバーの機能を実行するために、CICS PEM サーバーは、PEM クライアント・サインオン・トランザクション・プログラムとの間で入力データの受け取りと出力データの送信を行います。

CICS PEM サーバーの機能については、CICS PEM サーバー処理で説明されています。

- PEM クライアントは CICS PEM サーバーにデータを送信します。これについては、PEM クライアントによって送信されるサインオン入力データで説明されています。
- CICS PEM サーバーは PEM クライアントにデータを送信します。これについては、CICS PEM サーバーによって返されるサインオン出力データおよびそのサブトピックで説明されています。

データがPEM クライアントのセットアップで説明されている標準に準拠しており、そのフォーマットがユーザー・データのフォーマットで説明されているとおりであることを確認します。 GDS フローでのサインオン出力データの例については、新規パスワードでのサインオンを参照してください。

基本会話情報およびデータは、ユーザー・データのフォーマットで説明するとおり、付加 FMH に含まれています。サインオン要求は、トランザクション X'06F3F0F1' (サインオン・トランザクション・プログラム用の SNA サービス・トランザクション・プログラム名) を接続します。

PEM クライアントによって送信されるサインオン入力データ

CICS PEM サーバーは、PEM クライアント・サインオン・トランザクション・プログラムからの入力データを必要とします。

GDS フローでのサインオン入力データの例については、 323 ページの『新規パスワードでのサインオン』を参照してください。

表 *33. CICS PEM* サーバーに送信されるサインオン要求およびデータ

長さ (バイト)	値	意味
2	X'nnnn'	GDS データ全体の長さ (この 2 バイト長の値を含む)。
2	X'1221'	サインオン・データのデータ ID。
2	X'nnnn'	この 2 番目の (ネストされた) データ構造 (長さ、データ ID、およびデータ) の長さ (この 2 バイト長の値を含む)。
2	X'FF00' または X'FF01'	サインオン要求データのデータ ID、またはサインオンとパスワード変更要求データのデータ ID (新規パスワード・サブフィールドは X'FF00' には許可されません)。
1	X'nn'	ユーザー ID のサブフィールドの長さ (この 1 バイト長の値を含む)。
1	X'01'	ユーザー ID のサブフィールドのデータ ID。
X'nn'-2	C'xxxxxxxx'	ユーザー ID。
1	X'mm'	パスワードのサブフィールドの長さ (この 1 バイト長の値を含む)。
1	X'02'	パスワードのサブフィールドのデータ ID。
X'mm'-2	C'xxxxxxxx'	パスワード。
1	X'pp'	新規パスワードのサブフィールドの長さ (この 1 バイト長の値を含む)。
1	X'06'	新規パスワードのサブフィールドのデータ ID。
X'pp'-2	C'xxxxxxxx'	新規パスワード。

CICS PEM サーバーによって返されるサインオン出力データ

CICS PEM サーバーは、PEM クライアントにサインオン出力データを返します。

表 34 は、このデータをリストしています。 GDS フローでのサインオン出力データの例については、 324 ページの『正しいサインオン・データに対する応答』および 326 ページの『間違ったデータ・フォーマットに対する応答』を参照してください。

表 *34. PEM* クライアントに返されるサインオン出力データ

長さ (バイト)	値	必須またはオプション	意味
2	X'nnnn'	Required	GDS データ全体の長さ (この 2 バイト長の値を含む)。
2	X'1221'	Required	サインオン・データのサブフィールドのデータ ID。
2	X'nnnn'	Required	この 2 番目の (ネストされた) データ構造 (長さ、データ ID、およびデータ) の長さ (この 2 バイト長の値を含む)。
2	X'FF02'	Required	サインオン応答データのデータ ID。
1	X'03'	Required	サインオン完了状況のサブフィールドの長さ (この 1 バイト長の値を含む)。
1	X'00'	Required	サインオン完了状況のサブフィールドのデータ ID。
1	X'00' から X'06'	Required	サインオン完了状況。 321 ページの表 36 を参照してください。
1	X'04'	オプション	サインオン要求フォーマット設定エラーのサブフィールドの長さ (この 1 バイト長の値を含む)。
1	X'01'	オプション	サインオン要求フォーマット設定エラーのサブフィールドのデータ ID。
2	X'0000' から X'0003'、X'0005' から X'0007'、X'000F'	オプション	サインオン要求フォーマット設定エラー。 322 ページの表 37 を参照してください。
1	X'0A'	オプション	現在の正常なサインオンの日時のサブフィールドの長さ (この 1 バイト長の値を含む)。
1	X'02'	オプション	現在の正常なサインオンの日時のサブフィールドのデータ ID。

表 *34. PEM* クライアントに返されるサインオン出力データ (続き)

長さ (バイト)	値	必須またはオプション	意味
8	フォーマットについては、 321 ページの表 35 を参照してください。	オプション	現在の正常なサインオンの日時。返される日時は、ESM によってユーザー・プロファイルから抽出されます。
1	X'0A'	オプション	最後の正常なサインオンの日時のサブフィールドの長さ (この 1 バイト長の値を含む)。
1	X'03'	オプション	最後の正常なサインオンの日時のサブフィールドのデータ ID。
8	フォーマットについては、 321 ページの表 35 を参照してください。	オプション	最後の正常なサインオンの日時。返される日時は、ESM によってユーザー・プロファイルから抽出されます。
1	X'0A'	オプション	パスワード有効期限切れの日時のサブフィールドの長さ (この 1 バイト長の値を含む)。
1	X'04'	オプション	パスワード有効期限切れの日時のサブフィールドのデータ ID。
8	フォーマットについては、 321 ページの表 35 を参照してください。	オプション	パスワード有効期限切れの日時。 (返される日時は、ESM から取得されたデータから計算されます。)
1	X'04'	オプション	取り消しカウントのサブフィールドの長さ (この 1 バイト長の値を含む)。
1	X'05'	オプション	取り消しカウントのサブフィールドのデータ ID。
2	X'nnnn'	オプション	取り消しカウント。

日時サブフィールドのフォーマット:

このトピックでは、CICS PEM サーバーが PEM クライアントに返す可能性がある、日時サブフィールドのフォーマットをリストします。

321 ページの表 35 に、CICS PEM サーバーが PEM クライアントに返すことができる日時サブフィールドのフォーマットがリストされています。 GDS フローの日時サブフィールドの例については、 324 ページの『正しいサインオン・データに対する応答』を参照してください。

表 35. 24 時間クロックを使用する日時サブフィールドのフォーマット

位置	意味
00	2 バイトの年の値。例えば、1994 は X'07CB'。
02	1 バイトの月の値。1 月は X'01'、12 月は X'0C'。
03	1 バイトの日の値。1 日は X'01'、31 日は X'1F'。
04	1 バイトの時の値。深夜 0 時は X'00'、23 時は X'17'。
05	1 バイトの分の値。正時は X'00'、59 分は X'3B'。
06	1 バイトの秒の値。始点の 0 秒は X'00'、59 秒は X'3B'。
07	1 バイトの 100 分の 1 秒の値。始点は X'00'、最大は X'63'。

注: 特定の 1 日の最大時間値は、23 時間、59 分、59.99 秒 (小数部は 10 進数) です。午前 0 時は、次の日の 0 時 0 分 0 秒です。

PEM クライアントに返されるサインオン完了状況値:

このトピックでは、サインオン応答データの状況完了サブフィールドで CICS PEM サーバーが PEM クライアントに返す可能性がある、サインオン完了状況値について説明します。

表 36 は、サインオン応答データの状況完了サブフィールドで CICS PEM サーバーが PEM クライアントに返すことができる、サインオン完了状況値を説明しています。 GDS フローのサインオン完了状況値の例については、 324 ページの『正しいサインオン・データに対する応答』を参照してください。

表 36. PEM クライアントに返されるサインオン完了状況値

状況値	意味
X'00'	以下のすべての条件が当てはまります。 • ユーザー ID が有効 • パスワードが有効 • パスワードの有効期限が切れていないか、または新規の有効なパスワードが指定されている この状況値が返されるときは、新規パスワード (指定した場合) は設定され、PV 処理 (使用した場合) は完了しています。
X'01'	ユーザー ID が受信側に認識されていません。
X'02'	ユーザー ID は有効ですが、パスワードが正しくありません。
X'03'	ユーザー ID は有効で、パスワードは正しいですが、有効期限が切れています。新規パスワードを設定する必要があります。
X'04'	ユーザー ID は有効で、パスワードは正しいですが、新規パスワードが受信側のセキュリティー・システムで受け入れられません。
X'05'	セキュリティー機能が失敗しました。機能が実行されません。
X'06'	データ・フォーマットが正しくありません。特定のエラーが、 322 ページの表 37 で説明されているサインオン要求フォーマット設定エラー・サブフィールドに含まれています。

注: CICS PEM サーバーは、以下のどのサインオン状況値も PEM クライアントに返しません。

- X'07' — 一般セキュリティー・エラー
- X'08' — パスワード変更は完了していますが、サインオンに失敗しました。

PEM クライアントに返されるサインオン要求フォーマット設定エラー**:**

このトピックでは、CICS PEM サーバーが PEM クライアントに返す可能性がある、サインオン要求フォーマット設定エラーの値をリストします。

表 37 に、CICS PEM サーバーが PEM クライアントに返す可能性がある、サインオン要求フォーマット設定エラーの値がリストされています。それぞれは 2 バイトのバイナリー値です。 GDS フローでのサインオン要求フォーマット設定エラーの例については、 326 ページの『間違ったデータ・フォーマットに対する応答』を参照してください。

表 *37. PEM* クライアントに返されるサインオン要求フォーマット設定エラーの値

エラー値	説明
X'0000'	以下に説明されていない未定義エラーです。
X'0001'	必要な構造がありません。
X'0002'	除外された構造が存在しています。
X'0003'	反復不能構造の複数のオカレンスがあります。
X'0005'	認識できない構造が、それが除外された場所にあります。
X'0006'	長さが、指定された範囲を超えています。この値により、長さの計算が合っており、送信側がその長さで構造を送信しようとしたことが想定されます。
X'0007'	長さ例外。長さの計算が合っていません。
X'000F'	データ値が範囲外です。

アプリケーション設計

他のすべてのトランザクションの前にサインオン・トランザクションを実行するように、アプリケーションを設計します。

これにより、すべてのパスワード検査とパスワード変更を、アプリケーション固有の機能から切り離しておくことができます。マルチタスキング・システムでは、複数のサインオン・トランザクションを並列セッションで開始することができます。アプリケーション・レベルの ALLOCATE 要求を処理するコードが、サインオン・プロセスを完了まで直列化することが重要です。これにより、両方のフローが確実にサインオン済みとなります。

直前の正常なサインオンの日時を記録するために、CICS PEM サーバー・サインオン・プログラムは、サインオンを実行する前に、ESM からパスワード・データを抽出します。システムが共用ユーザー ID を使用しており、2 つのユーザーが同時にサインオンを試行した場合、またはユーザーがマルチタスキングを実行している場合、現在のサインオンに対して PEM クライアントに返される時刻値は、ESM データベースに記録されているタイム・スタンプとは同じでないことがあります。複数のシステムで実行する必要があり、PEM クライアントに返されるサインオン時刻に依存するアプリケーションを作成する場合は、このことを思い出してください。
(この状態は、サインオン・プロセスが推奨どおりに直列化されていれば、単一システムに当てはまることはないはずです。)

PV を使用しており、PVDELAY で指定された間隔を過ぎており、ユーザー ID が PV サインオン元リストから削除されている場合、アプリケーションがサインオン・プロセスを再度直列化することを許可する必要があります。

PEM クライアントおよび CICS PEM サーバーのユーザー・データの例

PEM クライアントと CICS PEM サーバーの間では、GDS 変数を使用してデータが渡されます。

以下の例を使用して、PEM クライアントによって送信されるデータを検査できます。以下の例は、CICSTS55.CICS.SDFHSAMP ライブラリーに示されているサンプル PEM クライアント・プログラムによって作成されます。これについては、 303 ページの『APPC パスワード有効期限管理の概要』で説明されています。このプログラムは、以下の値を使用します。

partner_LU_alias
: hostcics

LU_alias
: ps2lua

mode_name
: lu62ss

独自の PEM クライアント・プログラムを作成する場合は、コミュニケーション・マネージャー構成に定義されている値を使用します。

新規パスワードでのサインオン: 以下に示すのは、新規パスワードを使用したサインオンの成功例です。

```
PEM hostcics ps2lua lu62ss sec2r01 drtnnom hursley
```

ユーザー ID、パスワード、および新規パスワードが正しく入力されています。

PEM クライアントは、以下の 16 進数ユーザー・データを CICS PEM サーバーに送信します。

```
0231221001FFF010901E2C5C3F2D9F0F10902C4D9E3D5D5D6D40906C8E4D9E2D3C5E8
```

これには、 318 ページの表 33 で説明されているように、以下の値が含まれます。

0023 GDS 変数全体の長さ (この 2 バイト長の値を含む)

1221 サインオンのデータ ID

001F この 2 番目の (ネストされた) データ構造 (長さ、データ ID、およびデータ) の長さ (この 2 バイト長の値を含む)

FF01 サインオンおよびパスワード変更要求データのデータ ID

09 ユーザー ID のサブフィールドの長さ (この 1 バイト長の値を含む)

01 ユーザー ID のサブフィールドの ID

E2C5C3F2D9F0F1
: EBCDIC のユーザー ID (SEC2R01)

09 パスワードのサブフィールドの長さ (この 1 バイト長の値を含む)

02 パスワードのサブフィールドの ID

C4D9E3D5D5D6D4
EBCDIC のパスワード (DRTNNOM)

09 新規パスワードのサブフィールドの長さ (この 1 バイト長の値を含む)

06 新規パスワードのサブフィールドの ID

C8E4D9E2D3C5E8
EBCDIC の新規パスワード (HURSLEY)

正しいサインオン・データに対する応答: 図 21 は、入力された正しいサインオン・データに対する応答の例を示しています。

```
PEM_OK
GDS LLID
00 2d 12 21
Sign-on Reply LLID
00 29 ff 02
Sign-on Completion Status Subfield
03 00 00
Date & Time of Current Successful Sign-on Subfield
0a 02 07 ca 01 14 0d 24 31 62
Date & Time of Last Successful Sign-on Subfield
0a 03 07 ca 01 11 16 1b 23 3e
Date & Time Password Will Expire Subfield
0a 04 07 ca 02 03 00 00 00 00
Revoke Count Subfield
04 05 00 00
```

図 *21.* 正しいサインオン・データに対する応答

PEM クライアントに返された最初の 3 行の 16 進数ユーザー・データは、 319 ページの表 34 で説明する、以下の必須 値を示しています。

002d GDS 変数の全長 (この 2 バイト長の値を含む)

1221 サインオン・データのデータ ID

0029 この 2 番目の (ネストされた) データ構造 (長さ、データ ID、およびデータ) の長さ (この 2 バイト長の値を含む)

FF02 サインオン応答データのデータ ID

03 サインオン完了状況のサブフィールドの長さ (この 1 バイト長の値を含む)

00 サインオン完了状況のデータ ID

00 サインオン完了状況。 00 は、ユーザー ID とパスワードが有効であったが、パスワードの有効期限は切れていなかったことを示します。 (サインオン完了状況値のリストについては、 321 ページの表 36 を参照してください。)

図 21 で、PEM クライアントに返された最後の 4 行の 16 進数ユーザー・データは、 319 ページの表 34 で説明する、以下の必須値を示しています。 (319 ページの表 34 に示されているフォーマット設定エラー・サブフィールドは含まれておらず、エラーがないことが示されていることに注意してください。)

0A 現在の正常なサインオンの日時のサブフィールドの長さ (この 1 バイト長の値を含む)

02 現在の正常なサインオンの日時のデータ ID

現在の正常なサインオンの日時 (321 ページの表 35 で説明)

07CA 年 (1994)

01 月 (1 月)

14 日 (20)

0D 時 (13)

24 分 (36)

31 秒 (49)

62 100 分の 1 秒 (98)

0A 直前の正常なサインオンの日時のサブフィールドの長さ

03 直前の正常なサインオンの日時のデータ ID

直前の正常なサインオンの日時 (321 ページの表 35 で説明)

07CA 年 (1994)

01 月 (1 月)

11 日 (17)

16 時 (22)

1B 分 (27)

23 秒 (35)

3E 100 分の 1 秒 (62)

0a パスワード有効期限切れの日時のサブフィールドの長さ (この 1 バイト長の値を含む)

04 パスワード有効期限切れの日時のデータ ID のサブフィールドの長さ

パスワード有効期限切れの日時 (321 ページの表 35 で説明)

07ca 年 (1994)

02 月 (2 月)

03 日 (14)

00 時 (00)

00 分 (00)

00 秒 (00)

00 100 分の 1 秒 (00)

04 取り消しカウントのサブフィールドの長さ (この 1 バイト長の値を含む)

05 取り消しカウントのサブフィールドのデータ ID

0000 取り消しカウント (0000 は、このユーザー ID を使用した最後の正常なサインオン以降に、失敗したサインオン試行がなかったことを意味します)。

間違ったデータ・フォーマットに対する応答: 図 22 は、入力された間違ったデータに対する応答の例を示しています。

```
PEM_OK
GDS LLID
00 0F 12 21
Sign-on Reply LLID
00 0B FF 02
Sign-on Completion Status Subfield
03 00 06
Sign-on Request Formatting Error Subfield
04 01 00 0F
```

図 22. 間違ったデータ・フォーマットに対する応答

PEM クライアントに返された最初の 3 行の 16 進数ユーザー・データは、 319 ページの表 34 で説明する、以下の必須値を示しています。

000F GDS データ全体の長さ (この 2 バイト長の値を含む)

1221 サインオン・データのデータ ID

000B この 2 番目の (ネストされた) データ構造 (長さ、データ ID、およびデータ) の長さ (この 2 バイト長の値を含む)

FF02 サインオン応答データのデータ ID

03 サインオン完了状況のサブフィールドの長さ (この 1 バイト長の値を含む)

00 サインオン完了状況のサブフィールドのデータ ID

06 間違ったデータ・フォーマットを示すサインオン完了状況 06 (サインオン完了状況値のリストについては、 321 ページの表 36 を参照)。

PEM クライアントに返された最終行の 16 進数ユーザー・データは、エラーがある場合にのみ返される、次のオプション値を示しています。 (オプション値については、 319 ページの表 34 で説明しています。)

04 サインオン要求フォーマット設定エラーのサブフィールドの長さ (この 1 バイト長の値を含む)

01 サインオン要求フォーマット設定エラーのサブフィールドのデータ ID

000F 「データ値が範囲外」を示す、サインオン要求フォーマット設定エラー (発生する可能性がある他のフォーマット設定エラーについては、 322 ページの表 37 を参照)。

IPIC セキュリティーの実装

これらのトピックでは、IPIC 接続のセキュリティーを実装する方法を説明します。

IPIC バインド時のセキュリティー

接続を確立するための要求をリモート・システムとの間で送受信する際、セキュリティー検査が行われます。これは、バインド時のセキュリティーと呼ばれます。許可が与えられていないシステムが CICS に接続するのを防ぐのがその目的です。

CICS が IPIC を使用して別の CICS 領域と通信する場合、各 CICS システムには、定義済みの IPCONN リソースと TCPIPSERVICE リソースが必要です。

各 CICS システムは IPCONN を使用して、受信側として機能するパートナー・システム TCPIPSERVICE にデータを送信します。通信を開始する CICS 領域がクライアントで、リモート・システムがサーバーです。

IPIC の場合は、Secure Sockets Layer (SSL) クライアント証明書の交換によってバインド・セキュリティーがサポートされます。 2 つの CICS 領域が正常に接続できるようにし、無許可システムには接続させないようにするには、以下のようにします。

- SEC システム初期設定パラメーター は、両方の領域で 『YES』 でなければなりません。
- ローカル領域とリモート領域の両方で IPCONN 定義を以下のように指定してください。
  - SSL(YES)。
  - CIPHERS(*value*)。 CIPHERS 属性は、以下の 2 とおりの方法のいずれかで指定できます。
    - 最大 28 個の 2 桁の暗号スイート・コードを示すリストとして解釈される 56 桁までの 16 進数字で構成されるストリング。
    - SSL 暗号スイート仕様ファイルの名前。これは、**USSCONFIG** システム初期設定パラメーターにより指定されているディレクトリーの `/security/ciphers` サブディレクトリーにある z/OS UNIX ファイルのことです。詳細については、暗号スイートおよび暗号スイート仕様ファイルを参照してください。

      CEDA を使用してリソースを定義する場合、CICS は、ENCRYPTION システム初期設定パラメーター システム初期設定パラメーターで指定されている暗号化のレベルに応じ、受け入れ可能コードのデフォルト・リストを使用して自動的に属性を初期化します。
  - オプションとして、CERTIFICATE(*X.509_certificate_label*)。IPCONN が獲得される際の SSL ハンドシェーク中、指定の証明書がクライアント証明書として使用されます。CERTIFICATE が指定されないと、CICS 領域ユーザー ID の鍵リングに定義されているデフォルトの証明書が使用されます。

  IPCONN は、アウトバウンド 側の接続を定義します。これらの設定は CICS に SSL ハンドシェークを開始するように指示します。 SSL ハンドシェーク中に、CICS はパートナー領域に、TCPIPSERVICE で指定された証明書を要求します。リモート領域 TCPIPSERVICE が SSL(CLIENTAUTH) を指定している場合、リモート・システムは、ハンドシェークの一部として発信元システムの証明書を要求します。
- ローカル領域とリモート領域の両方の TCPIPSERVICE リソース 定義で、以下を指定します。
  - PROTOCOL(IPIC)。
  - SSL(CLIENTAUTH) または SSL(YES)。
  - CIPHERS(*value*)。

– オプションとして、CERTIFICATE(*X.509_certificate_label*)。指定の証明書がサーバー証明書として使用されます。 CERTIFICATE が指定されないと、CICS 領域ユーザー ID の鍵リングに定義されているデフォルトの証明書が使用されます。

TCPIPSERVICE 定義は、接続のインバウンド 側を定義します。これらの設定値によって、クライアントが IPCONN を獲得することを許可する前に有効な SSL クライアント証明書を受信する必要があることを CICS に指示します。また、これらの設定は、CICS が TCPIPSERVICE CERTIFICATE を送信することを指定します。この設定が指定されない場合は、デフォルトの証明書が送信されます。

TCPIPSERVICE に SSL(YES) が指定されている場合、サーバーは、ハンドシェーク中にクライアント証明書を要求したり受信したりすることはありません。

両方の CICS 領域の TCPIPSERVICE に SSL(YES) が指定されている場合、どちらの CICS 領域も認証されます。

両方の CICS 領域の TCPIPSERVICE に SSL(CLIENTAUTH) が指定されている場合、どちらの CICS 領域も 2 回認証されます。

ほとんどの環境では、両方の領域の TCPIPSERVICE に SSL(YES) を指定するか、または一方の領域に SSL(NO)、他方の領域に SSL(CLIENTAUTH) を指定することで、適切なレベルのセキュリティーを実現できます。

注**:** LINKAUTH が CERTUSER と指定されている場合、IPCONN は SSL(CLIENTAUTH) が定義されている TCPIPSERVICE を参照する必要があります。

両方の領域の TCPIPSERVICE に SSL(NO) が指定されている場合、バインド時セキュリティーは実行できません。

リモート・クライアントが CICS サーバーによって信頼されている場合、バインド時セキュリティーは必要ありませんが、トランザクション接続に渡されるすべてのユーザー ID とパスワードは、サーバー領域の外部セキュリティー・マネージャーで有効なものでなければなりません。

IPIC リンク・セキュリティー

リンク・セキュリティーは、ユーザーがアクセスできるリソースを、アクセス元のリモート・システムに応じて制限します。 リンク・セキュリティーの実際的な効果として、リモート・ユーザーによるトランザクションへの接続や、リンク・ユーザー ID が権限を持たないリソースへのアクセスを抑止します。

リンク・セキュリティーを使用すると、リンク・ユーザー *ID* によって定義されている権限がすべての要求に与えられます。IPCONN の場合、接続に対するすべての要求が同じリンク・ユーザー ID を持ちます。

IPIC リンク・セキュリティーの指定

IPCONN のリンク・ユーザー ID を指定するには、LINKAUTH オプションを使用して、セキュリティーが初期設定された CICS システム (SEC=YES) でのリンク・セキュリティー用のユーザー ID の設定方法を指定します。以下のように指定できます。

CERTUSER

パートナー・システムとの TCP/IP 通信を SSL 用に構成し、SSL ハンドシェーク中にパートナー・システムから証明書を受け取る必要があります。

IPCONN リソースは、SSL(CLIENTAUTH) で定義されている TCPIPSERVICE リソースを参照する必要があります。

受け取った証明書は外部セキュリティー・マネージャーに定義し、リンク・セキュリティーを確立する際に使用するユーザー ID と関連付ける必要があります。

SECUSER

SECURITYNAME 属性に指定されたユーザー ID を使用してリンク・セキュリティーを確立するように指定します。

デフォルト値は LINKAUTH(SECUSER) です。

セキュリティーが初期設定された CICS システム (SEC=YES) では、リンク・ユーザー ID はリモート・システムの権限を設定するために使用されます。リンク・ユーザー ID は、この領域の有効な RACF ユーザー ID である必要があります。この領域の保護リソースに対するアクセスは、RACF ユーザー・プロファイルとそのグループ・メンバーシップに基づいています。

リモート・ユーザーによってトランザクションが接続されると、少なくとも 1 つのセキュリティー検査が必ず実行されます。ただし、指定したリンク・ユーザー ID がローカル領域ユーザー ID と一致する場合は、セキュリティー検査は以下のように最小限に抑えられます。

ユーザー ID が一致すると、1 つのセキュリティー検査だけが実行されます。この検査は、デフォルト・ユーザー (USERAUTH=LOCAL または DEFAULTUSER の場合)、または受信したインバウンド接続要求内にあるユーザー ID (USERAUTH=IDENTIFY または VERIFY の場合) のいずれかに対して行われます。

ユーザー ID が一致せず、USERAUTH=LOCAL の場合は、リソース検査はリンク・ユーザー ID に対してのみ行われます。 USERAUTH=IDENTIFY または VERIFY の場合、2 つのリソース検査が必ず行われます。1 番目の検査はリンク・ユーザー ID に対して、2 番目の検査は接続要求でリモート・ユーザーから受信したユーザー ID に対して行われます。 USERAUTH=DEFAULTUSER の場合、2 つのリソース検査が必ず行われます。 1 番目の検査はリンク・ユーザー ID に対して、2 番目の検査はデフォルト・ユーザーに対して行われます。

330 ページの『IPIC ユーザー・セキュリティー』も参照してください。

リンク・セキュリティーを確立する際に障害が生じると、ローカル領域のデフォルト・ユーザーのセキュリティーがリンクに提供されます。これは例えば、リンク・ユーザー ID が取り消されている場合などに起きる可能性があります。

IPIC ユーザー・セキュリティー

ユーザー・セキュリティーにより、要求側から流れてきて受信するセキュリティー・コンテキスト (その下でトランザクションが動作する) の 2 度目の検査が行われます。 受信するセキュリティー・コンテキストには通常、クライアントを実行しているユーザーのユーザー ID が含まれています。

セキュリティー・コンテキストは、以下のいずれかのタイプのユーザーを識別できます。

- TOR で端末にサインオンしているユーザー
- DPL 要求を発行するトランザクションのユーザー ID
- ECI 要求のクライアント・ユーザー

IPIC 接続の場合、IPCONN リソース定義の USERAUTH 属性は、着信要求のサインオン要件を指定します。ご使用のシステムからリモート・システムに発行される要求には、影響がありません。そうした要求は、リモート・システムで処理されます。 MRO 接続と SNA 接続の同等の属性は、ATTACHSEC 属性です。ATTACHSEC 属性、および他の相互通信方式を使用したユーザー・セキュリティーの指定について詳しくは、 284 ページの『リンク定義でのユーザー・セキュリティーの指定』を参照してください。

IPIC 接続の場合、以下のタイプのユーザー認証を指定できます。

LOCAL
CICS は、クライアントからのユーザー ID またはパスワードを受け入れません。すべての要求は、リンク・ユーザー ID またはデフォルトのユーザー ID (リンク・ユーザー ID がない場合) で実行されます。

IDENTIFY
着信接続要求には、ユーザー ID を指定する必要があります。接続元のシステムがユーザーを事前認証できる場合 (例えば、別の CICS または CICS TG システムである場合) は、IDENTIFY を入力します。

SSL クライアント認証を使用するか、接続するシステムが同じシスプレックス内にある必要があります。

VERIFY
着信接続要求には、ユーザー ID とユーザー・パスワードを指定する必要があります。接続するシステムが不明な場合や信頼できない場合は、VERIFY を指定します。

DEFAULTUSER
CICS は、パートナー・システムからのユーザー ID およびパスワードを受け入れません。すべての要求は、デフォルトのユーザー ID で実行されます。

アウトバウンド要求の場合、ユーザー・セキュリティーのレベルはパートナー・システムにインストールされている IPCONN 定義の USERAUTH 属性によって指定されます。CICS は、USERAUTH(IDENTIFY) が指定されるとユーザー ID を送信しますが、USERID(LOCAL) が指定される場合には送信しません。CICS はリモート・システムにパスワードを送信しないので、CICS TS for z/OS システム間での通信では USERAUTH(VERIFY) はサポートされていません。

注: CICS は、ここに示す処理を行う際に、パスワード検証を使用してユーザー ID を検証します。 CICS は、CICS 領域へのログインに使用される各ユーザー ID に対して、1 日 1 回完全検証要求を実施します。 RACROUTE REQUEST=VERIFYX マクロを使用した完全検証要求により、RACF はユーザー ID の最終アクセス日時を記録し、ユーザー統計を書き込むことになります。

USERAUTH(IDENTIFY) を使用するときの信頼関係の確立

IPCONN リソースを USERAUTH(IDENTIFY) で定義することは、接続で非認証ユーザー ID または宣言ユーザー ID を受け入れる用意があることを意味します。つまり、パートナー・システムがそこで既に認証済みのユーザー ID のみを送信することを信頼しています。 CICS は、信頼度の異なる次の 2 タイプのパートナーを識別します。

パートナーがローカル・シスプレックスの外部にある
: パートナーがローカル・シスプレックスの外部にある場合、CICS は、パートナーがトラステッド・デジタル証明書で身元を明らかにしない限り、非認証ユーザー ID をアサートするパートナーを直接信頼することはありません。したがって、そのようなパートナーからの接続は、クライアント認証を使用した Secure Sockets Layer 接続を使用してのみ受け入れられます。このタイプの接続は、TCPIPSERVICE 定義の属性 SSL(CLIENTAUTH) によって指定されます。

  接続を介した暗号化を必要としない場合は、非暗号化の暗号スイートのみを TCPIPSERVICE 定義の属性 CIPHERS(0102) で指定することにより、SSL が提供する通常の暗号化を抑止できます。

パートナーがローカル・シスプレックスの内部にある
: パートナーがローカル・シスプレックス内にある場合、CICS は、デジタル証明書を必要とせずに、非認証ユーザー ID をアサートするパートナーを信頼します。これらのパートナーからの接続は、SSL 接続を必要とせずに許可されます。

  CICS とパートナー・システムとの間にプロキシーが存在しないように TCP/IP ネットワークを構成します。プロキシーがあると、CICS は、パートナーが同じシスプレックス内にあるかどうかを正しく検出できないことがあります。ネットワーク・アクセス・セキュリティー・ゾーンは、同じシスプレックス内の他のどのシステムに CICS との通信を許可するかを制御できるように構成することを強くお勧めします。

  ネットワーク・アクセス制御の詳細については、「z/OS Communications Server IP 構成ガイド」を参照してください。 **NETACCESS** 構成パラメーターの詳細については、「z/OS Communications Server IP 構成解説書」を参照してください。

IPIC を使用したリモート・ユーザー・サインオン状況

IPCONN リソースが USERAUTH(IDENTIFY) を指定すると、最初の接続要求に関連した会話が完了した後に、リモート・ユーザーはサインオンしたままになります。

これ以降 CICS は、以下のいずれかのイベントが発生するまで、同じユーザーからの接続要求を新規サインオンなしで受け入れます。

- このユーザーの接続要求に関連した最後のトランザクションの完了後に、USRDELAY システム初期設定パラメーター システム初期設定パラメーターに指定された期間が経過します。
- RACF は、プロファイルが変更されているという RACF イベント通知 (ENF) を CICS に通知します。詳しくは、 20 ページの『リモート・ユーザーの RACF プロファイルの変更』を参照してください。

ユーザーを取り消すなどしてサインオン・リモート・ユーザーの RACF プロファイルを変更した場合、以下のいずれかの状態となるまで、CICS は最初の接続要求時に設定された許可を引き続き使用します。

- トランザクションが同期点を実行する。
- 接続要求が終了する。
- RACF が CICS にユーザー・プロファイルの変更を通知し、そのサインオン・ユーザー ID に関連付けられている接続要求が、LOCAL を除く ATTACHSEC のすべてのオペランドに対して完了したためにサインオフとなる。
- RACF が CICS にユーザー・プロファイルの変更を通知し、新規接続要求が行われ、**USRDELAY** システム初期設定パラメーターの値が有効期限切れになっていないためにサインオフとなる。このサインオフの後には、サインオンが続きます。

IPIC のトランザクション・セキュリティー、リソース・セキュリティー、およびコマンド・セキュリティー

単一システム環境の場合と同様に、ユーザーには以下を行うための許可が必要です。

- トランザクションを接続する (トランザクション・セキュリティー)。
- トランザクションが使用するようにプログラムされているすべてのリソースにアクセスする。これらのレベルは、リソース・セキュリティー、代理ユーザー・セキュリティー、およびコマンド・セキュリティーと呼ばれます。

トランザクション・セキュリティー

単一システム環境の場合、トランザクションのセキュリティー要件はトランザクションを定義する際に示されます (94 ページの『トランザクション・セキュリティー』 を参照)。

IPIC 環境では、トランザクションを開始する前に、以下の 2 つの基本セキュリティー要件を満たしている必要があります。

- リンク・ユーザー ID には、トランザクションを開始するための十分な権限が必要です (328 ページの『IPIC リンク・セキュリティー』を参照)。
- USERAUTH(LOCAL) 以外のものが指定されている場合、ユーザー・セキュリティーが有効になります。したがって、要求を行っているユーザーは、システムにアクセスしてトランザクションを開始するための十分な権限を持っている必要があります。

リソースおよびコマンド・セキュリティー

相互通信環境でのリソース・セキュリティーおよびコマンド・セキュリティーは、単一システム環境とほぼ同じ方法で処理されます。

リソースおよびコマンドのセキュリティー検査は、インストールされているトランザクション定義でそれらの検査が必須であると (例えば、図 23 に示すように CEDA DEFINE TRANSACTION コマンドなどで) 指定されている場合にのみ実行されます。

```
CEDA DEFINE TRANSACTION
  .
  RESSEC(YES)
  CMDSEC(YES)
  .
```

図 23. トランザクションのリソースおよびコマンド・セキュリティーの指定

トランザクション定義でリソース・セキュリティー検査が RESSEC(YES) を使用して指定されている場合、リンクとユーザーの両方に、接続されたトランザクションがアクセスするリソースに対する十分な権限が備えられている必要があります。

トランザクション定義でコマンド・セキュリティー検査が CMDSEC(YES) を使用して指定されている場合、リンクとユーザーの両方に、接続されたトランザクションが発行する、 162 ページの表 11 に示されているすべてのシステム・プログラミング・コマンドに対して十分な権限が備えられている必要があります。

リソースおよびコマンド・セキュリティーの指定についての詳しい説明は、 100 ページの『リソース・セキュリティー』 および 162 ページの『CICS コマンド・セキュリティー』 を参照してください。

CICS ルーティング・トランザクション CRTE

接続されたリモート・システム上にあるトランザクションをローカル・システムでリモートとして定義する代わりに、CICS ルーティング・トランザクション (CRTE) を IPIC、LU6.2、または MRO リンクと共に使用してそれらのトランザクションを実行できます。

CRTE は、あまり頻繁に使用しないトランザクションや、すべてのシステム上にある CEMT などのトランザクションに対して特に役立ちます。

CRTE を開始した端末が、リモート・システムで定義されているか、またはローカル・システムでシップ可能と定義されていることを確認してください。リモート・システムが保護されている場合、端末オペレーターは RACF 権限を必要とします。

CRTE で実行されるトランザクションに対する AOR でのセキュリティー検査は、ATTACHSEC (MRO リンクと LU6.2 リンクの場合) または USERAUTH (IPIC リンクの場合) で指定されている内容に依存していません。TOR にサインオンしているユーザー ID にも依存していません。その代わり、セキュリティー検査は、ユーザーが CRTE の使用時にサインオンするかどうかに依存しています。

- ユーザーがサインオンしない 場合、作成される代理端末は、AOR デフォルト・ユーザーに関連付けられます。トランザクションが実行されると、このデフォル

トのユーザーに対してセキュリティー検査が実施されます。リンク・ユーザー ID に対しても検査が行われ、ルーティング・アプリケーション自体がリソースにアクセスする権限を持っているかどうかが確認されます。

- ユーザーがサインオンする 場合、CRTE の実行時に CESN トランザクションを使用して、代理がサインオン・ユーザーのユーザー ID を指します。リソースへのアクセスを試行するトランザクションの場合、セキュリティー検査は、代理のサインオン・ユーザーのユーザー ID と、リンク・ユーザー ID に対して行われます。

IPIC を使用した AOR で実行されるセキュリティー検査

これは、LINKAUTH、SECURITYNAME、および USERAUTH が AOR でどのように指定されるかによって異なります。

表 38、 335 ページの表 39、および 335 ページの表 40 に示されているリンク・ユーザー ID は、IPCONN 定義の LINKAUTH および SECURITYNAME の値から決定されます。 LINKAUTH(SECUSER) を指定した場合、リンク・ユーザー ID は SECURITYNAME 属性から決定されます。 LINKAUTH(CERTUSER) を指定した場合、リンク・ユーザー ID は、SSL ハンドシェーク時に TOR によって渡された証明書とユーザー ID を関連付ける RACF などの、外部セキュリティー・マネージャーから決定されます。 LINKAUTH(SECUSER) がデフォルトです。 SECURITYNAME のデフォルト値はデフォルト・ユーザー ID です。

リンク・ユーザー ID が AOR の領域ユーザー ID と同じ場合、リンクは AOR と同じセキュリティーを持っていると見なされ、リンク・セキュリティーは完全に省略されます。省略されたリンク・セキュリティーの影響は、AOR 内の IPCONN の USERAUTH 属性で指定された値によって異なります。

- USERAUTH(LOCAL) を指定した場合、リンク・ユーザー ID のみを使用してセキュリティー検査が行われます。
- USERAUTH(DEFAULTUSER) を指定した場合、AOR のデフォルト・ユーザー ID のみが使用されます。
- USERAUTH(IDENTIFY) または USERAUTH(VERIFY) を指定した場合、リンク・ユーザー ID は使用されません。 TOR から受け取ったユーザー ID のみがセキュリティーの判別に使用されます。

USERAUTH(LOCAL) がデフォルトです。

TOR の領域ユーザー ID、および AOR に対する TOR の IPCONN 定義に関連付けられているリンク・ユーザー ID はどちらも、AOR でのセキュリティー検査には関係しません。

以下の表は、USERAUTH(LOCAL) を指定した場合の検査方法を示しています。

表 38. USERAUTH(LOCAL)

AOR の領域ユーザー **ID**	リンク・ユーザー **ID**	**AOR** での検査
USERIDA	指定されていません	AOR DFLTUSER に対する検査
USERIDA	USERIDA	AOR DFLTUSER に対する検査

表 38. USERAUTH(LOCAL) (続き)

AOR の領域ユーザー **ID**	リンク・ユーザー **ID**	**AOR** での検査
USERIDA	USERIDB	USERIDB に対する検査

以下の表は、USERAUTH(DEFAULTUSER) を指定した場合の検査方法を示しています。

表 39. USERAUTH(DEFAULTUSER)

AOR の領域ユーザー **ID**	リンク・ユーザー **ID**	**AOR** での検査
USERIDA	指定されていません	AOR DFLTUSER に対する検査
USERIDA	USERIDA	AOR DFLTUSER に対する検査
USERIDA	USERIDB	USERIDB および AOR DFLTUSER に対する検査

以下の表は、USERAUTH(IDENTIFY) または USERAUTH(VERIFY) を指定した場合の検査方法を示しています。

表 40. USERAUTH(IDENTIFY) および USERAUTH(VERIFY)

AOR の領域ユーザー **ID**	リンク・ユーザー **ID**	**AOR** での検査
USERIDA	指定されていません	送信済みユーザー ID および AOR DFLTUSER
USERIDA	USERIDA	送信済みユーザー ID のみ
USERIDA	USERIDB	送信済みユーザー ID および USERIDB

MRO セキュリティーの実装

このトピックでは、CICS 複数領域操作 (MRO) セキュリティーを実装する方法を説明します。

MRO アクセス方式選択時のセキュリティーへの影響

MVS 仮想記憶間サービスまたは CICS タイプ 3 SVC のいずれかを使用して、領域間通信 (機能シップ、トランザクション・ルーティング、分散トランザクション処理、および非同期処理) を行うことができます。

仮想記憶間サービスを使用する場合、通常は別のアドレス・スペースによって提供される、システム間の全体的な分離がなくなります。

仮想記憶間リンクで接続された 2 つの CICS アドレス・スペース間においては、偶発的な干渉が発生する危険性は低いです。ただし、一連の仮想記憶間命令を使うことよって、一方のシステムのアプリケーション・プログラムから、他方のシステムのストレージにアクセスする (キー制御による保護が条件です) ことが可能になります。

こうした状態によって、ご使用のインストール環境に機密漏れが発生するような場合は、MVS 仮想記憶間サービスではなく、CICS タイプ 3 SVC を領域間通信に使用してください。

MRO のアクセス方式を指定する方法については、複数領域操作を参照してください。

MRO でのバインド時のセキュリティー

CICS 領域間通信 (IRC) 機能は、FACILITY クラス内の DFHAPPL.*applid* プロファイルを使用して MRO をサポートします。

DFHIRP のバインド時のセキュリティー検査には 2 つのフェーズがあり、次の時点で実行されます。

- ログオン時
- 接続時

SAF インターフェースへの RACROUTE 呼び出しを介するこれらのセキュリティー検査は、MRO パートナー領域が CICS リソース・セキュリティー検査用にアクティブな外部セキュリティーを使用して実行しているかどうかに関係なく (つまり SEC=YES と SEC=NO のどちらでも)、常に実行されます。 MRO 接続が 2 つの領域間で確立されるようにするには、両方のシステムでログオン・セキュリティー検査と接続セキュリティー検査の両方が正常に実行される必要があります。

MRO を使用したログオン・セキュリティー検査

ログオン・セキュリティー検査は、CICS 領域が CICS 提供の領域間通信 (IRC) プログラムの DFHIRP にログオンするときにはいつでも実行されます。

CICS 領域間通信は、外部セキュリティー・マネージャーを使用して、IRC にログオンしている CICS 領域がその主張どおりの領域であることを検査します。

IRC アクセス方式を使用する各領域は、RACF FACILITY クラス内の DFHAPPL.*applid* プロファイルで RACF に対して許可されている必要があります。これには、DFHIRP にログオンする各領域の DFHAPPL.*applid* プロファイルの定義が必要です。さらに、各 CICS 領域ユーザー ID が、その固有の DFHAPPL.*applid* プロファイルに対して UPDATE 権限を持っている必要があります。

バッチ・ジョブが IRC を使用して CICS 領域に接続する場合、CICS 領域は IRC にログオンし、本資料で説明するとおり、その固有の DFHAPPL *applid* プロファイルに対する UPDATE 権限を必要とします。

ログオン検査の例については、 338 ページの図 24 を参照してください。

接続セキュリティー

MRO 接続セキュリティー検査を実行するために、DFHIRP は、接続内の各 CICS 領域に、そのパートナーの DFHAPPL.*applid* プロファイルへの読み取り権限があることを確認します。

CICS Transaction Server for z/OS, バージョン 5 リリース 5 DFHIRP がインストールされる場合、MVS イメージ内の以前の CICS リリースを使用するすべての

領域は、DFHAPPL.*applid* という形式の MRO 接続セキュリティーを使用します。さらに、MRO には CONNECTION 定義の SECURITYNAME パラメーターは使用されず、無視されます。

バインド・セキュリティー目的で MRO パートナー領域を許可するには、RACF FACILITY クラスで適切な DFHAPPL プロファイルを定義する必要があります。つまり、MRO 領域間通信リンク内の各 CICS 領域に、そのパートナーの DFHAPPL.*applid* プロファイルへのアクセス権限と、READ 権限が付与される必要があります。例えば、ユーザー ID CICSRTOR (APPLID CICSATOR が指定されている) で実行する CICS TOR が、ユーザー ID CICSRAOR (APPLID CICSAAOR が指定されている) で実行する AORに接続する場合の、接続を許可する RACF コマンドを 338 ページの図 24 に示します。

CICS は必ず RACROUTE 呼び出しを発行するので、MRO 接続のセキュリティー検査を接続するかどうかを CICS に指定することはできません。

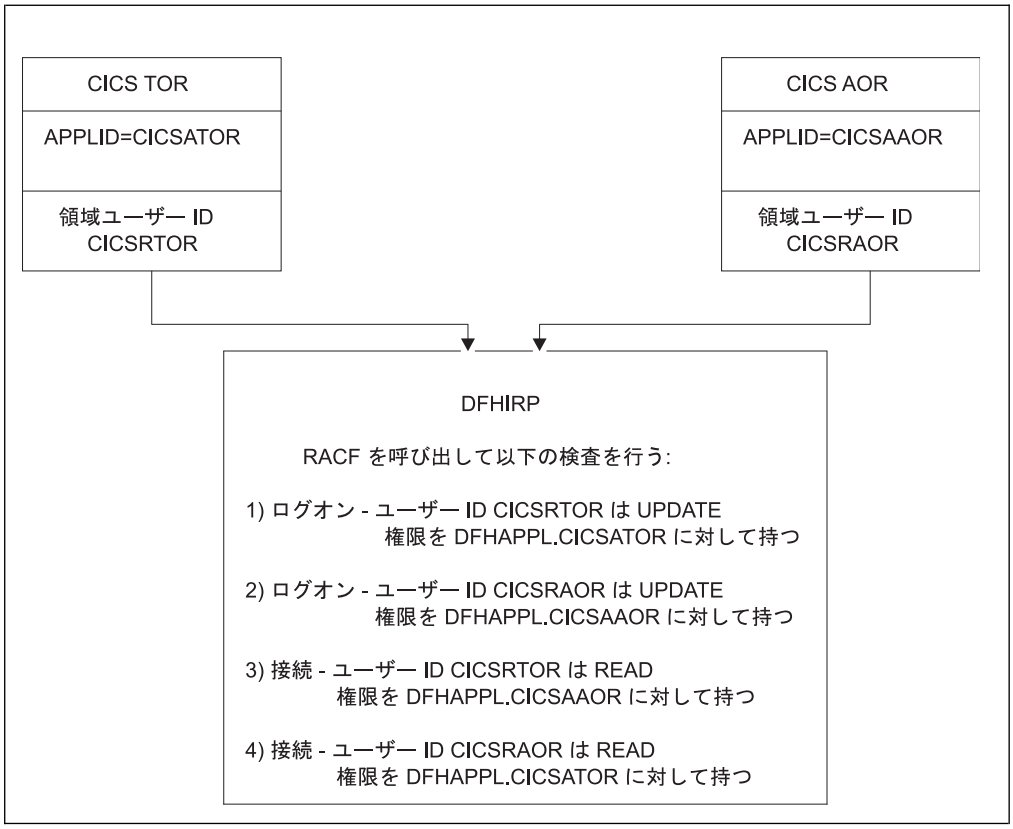


図 *24. DFHIRP* ログオンと接続セキュリティー検査の図：

この図は、CICS 端末専有領域 (TOR) と CICS アプリケーション専有領域 (AOR) が DFHIRP にログオンし、相互に接続している様子を示しています。

- TOR には APPLID として CICSATOR、領域ユーザー ID として CICSRTOR があります。
- AOR には APPLID として CICSAAOR、領域ユーザー ID として CICSRAOR があります。

ログオン・プロセス中に、DFHIRP は RACF を呼び出して、以下のことを確認します。

1. ユーザー ID の CICSRTOR には、DFHAPPL.CICSATOR に対する UPDATE 権限があります。
2. ユーザー ID の CICSRAOR には、DFHAPPL.CICSAAOR に対する UPDATE 権限があります。

接続プロセス中に、DFHIRP は RACF を呼び出して、以下のことを確認します。

1. ユーザー ID の CICSRTOR には、DFHAPPL.CICSAAOR に対する READ 権限があります。
2. ユーザー ID の CICSRAOR には、DFHAPPL.CICSATOR に対する READ 権限があります。

図 24 に示されている TOR および AOR (それぞれ、領域ユーザー ID の CICSRTOR および CICSRAOR で実行し、APPLID が CICSATOR および CICSAAOR である) は、DFHIRP へのログオンを許可する以下の RACF 定義が必要です。

- MRO ログオンおよび接続プロセスの場合は、次のようにします。

```
RDEFINE FACILITY (DFHAPPL.CICSATOR) UACC(NONE)
RDEFINE FACILITY (DFHAPPL.CICSAAOR) UACC(NONE)

PERMIT DFHAPPL.CICSATOR  CLASS(FACILITY) ID(CICSRTOR) ACCESS(UPDATE)
PERMIT DFHAPPL.CICSAAOR  CLASS(FACILITY) ID(CICSRAOR) ACCESS(UPDATE)
```

- 接続の場合は、次のようにします。

```
PERMIT DFHAPPL.CICSAAOR  CLASS(FACILITY) ID(CICSRTOR) ACCESS(READ)
PERMIT DFHAPPL.CICSATOR  CLASS(FACILITY) ID(CICSRAOR) ACCESS(READ)
```

System Authorization Facility (SAF) からの応答

指定されたリソースのセキュリティー・プロファイルを取得できない場合、SAF はアクセス要求を認可することも拒否することもしません。このような状態では、次のようになります。

IRC は、次の場合にログオン要求または接続要求を拒否します。

- セキュリティー・マネージャーはインストールされていますが、MVS イメージが持続している間、一時的に非アクティブまたは操作不能になります。これは、セキュリティー・マネージャーがアクティブであるとアクセスを許可しないプロファイルを取得する可能性があるという理由に基づく、フェイルセーフ動作です。

IRC は、次の場合にログオン要求または接続要求を受け入れます。

- インストールされているセキュリティー・マネージャーがありません。または、
- アクティブなセキュリティー・マネージャーがありますが、FACILITY クラスが非アクティブであるか、または FACILITY クラスにプロファイルがありません。 CICS APPLID へのアクセスを制御することをユーザーが希望しているという証拠がないので、この場合にはログオンは許可されます。

特定の DFHAPPL.*applid* プロファイルおよび適用できる総称プロファイルがない CICS 領域は、すべてのログオン要求および接続要求を許可します。これを示すメッセージは発行されません。潜在的な機密漏れが発生しないようにするために、特定の領域に対するセキュリティー対策の前またはそれに並行して、総称プロファイルを使用して、すべての領域または特定の領域のグループを保護できます。例えば、以下のように指定します。

```
RDEFINE FACILITY (DFHAPPL.*) UACC(NONE)
```

これにより、さらに具体的なプロファイルを持たないすべての領域は、バインドされなくなります。

MRO とのリンク・セキュリティー

リンク・セキュリティーは、ユーザーがアクセスできるリソースを、アクセス元のリモート・システムに応じて制限します。 リンク・セキュリティーの実際的な効果として、リモート・ユーザーによるトランザクションへの接続や、リンク・ユーザー ID が権限を持たないリソースへのアクセスを抑止します。

システム間の各リンクには、リンク・ユーザー ID によって定義されたアクセス権限が付与されます。 MRO のリンク・ユーザー ID は、この接続のためのセッション定義で定義されたユーザー ID です。 MRO の場合、LU6.2 とは異なり、接続あたり 1 セッション定義しか持つことができず、接続あたり 1 リンク・ユーザー ID しか存在させることができないことに注意してください。事前設定セッション・ユ

ーザー ID がない場合、リンク・ユーザー ID は TOR 領域の領域ユーザー ID であると見なされます。MRO では、接続定義の SECURITYNAME フィールドは無視されます。

CICS へのトランザクション・ルーティングおよび機能シップを実行するには、少なくとも 1 つのセキュリティー検査が必要です。ただし、リンク・ユーザー ID がローカル領域ユーザー ID と一致する場合、実行されるセキュリティー検査は最小限に抑えられます。

- ユーザー ID が一致する場合は、必ずセキュリティー検査を 1 つのみ行うことになります。これは、ローカル領域のデフォルト・ユーザー (ATTACHSEC=LOCAL の場合)、または受信した FMH-5 接続要求のユーザー ID (ATTACHSEC=IDENTIFY の場合) のいずれかに対して行われます。
- ユーザー ID が一致せず ATTACHSEC=LOCAL の場合、リソース検査はリンク・ユーザー ID に対してのみ行われます。ATTACHSEC=IDENTIFY の場合、2 つのリソース検査が必ず行われます。 1 つの検査はリンク・ユーザー ID に対して、もう 1 つの検査は接続要求でリモート・ユーザーから受信したユーザー ID に対して行われます。

リンク・セキュリティーを確立する際に障害が生じると、ローカル領域のデフォルト・ユーザーに対して定義された同じセキュリティー許可がリンクに付与されます。これは例えば、事前設定セッションのユーザー ID が取り消された場合に起きる可能性があります。

インバウンド・トランザクションがアクセスする必要がある任意の保護リソースへのアクセス権限を持つ RACF ユーザー・プロファイルと、SESSIONS 定義を関連付けます。プロファイルの定義の手順については、RACF ファシリティーを参照してください。

サインオンが失敗した場合は、サインオン失敗メッセージが CSCS セキュリティー宛先に送信され、受信側システムで DFLTUSER のセキュリティーがリンクに付与されます。つまり、デフォルト・ユーザーがアクセス権限を持つリソースのみにアクセスできます。

CICS 領域ユーザー ID の取得

MRO のログオン・セキュリティー検査および接続セキュリティー検査のために、DFHIRP は、CICS ジョブまたはタスクを実行している CICS 領域ユーザー ID を認識する必要があります。DFHIRP は、RACROUTE REQUEST=EXTRACT マクロを発行することによって CICS 領域ユーザー ID を抽出します。

外部セキュリティー・マネージャーとして RACF を使用していない場合は、MVS セキュリティー・ルーター出口の ICHRTX00 を使用して、RACROUTE REQUEST=EXTRACT マクロからの応答をカスタマイズする必要があります。

CICS は、SAF 応答コードを調べて、セキュリティー・マネージャーが存在しているかどうかを判別します。

MRO 接続のリンク・セキュリティーの指定

このタスクについて

MRO 接続の場合、すべてのセッションには同じリンク・ユーザー ID があります。 SESSIONS リソースについて詳細情報が必要な場合は、SESSIONS リソースを参照してください。

手順

SESSIONS リソース定義の USERID 属性にリンク・ユーザー ID を指定します。MRO 接続では、CONNECTION リソース定義の SECURITYNAME フィールドは無視されます。

MRO でのユーザー・セキュリティー

ユーザー・セキュリティーにより、CICS は、 339 ページの『MRO とのリンク・セキュリティー』で説明されているリンク・セキュリティー検査に加え、端末にサインオンしているユーザーに対して 2 回目の検査を行います。ユーザー・セキュリティーが提供する特別なレベルのセキュリティー検査を必要とするかどうかを検討してください。

LOCAL (ユーザーは検査されない) または IDENTIFY (ユーザー ID は必要であるが、パスワードは送信されない) のいずれかを指定できます。

CONNECTION 定義の ATTACHSEC オペランドを使用して、接続ごとにサインオン・サポートを指定します。これについては、『リンク定義でのユーザー・セキュリティー』で説明されています。

リンク定義でのユーザー・セキュリティー

リモート・システムに必要なユーザー・セキュリティーのレベルは、CONNECTION リソース定義の ATTACHSEC 属性で指定されます。

CICS は、ここで説明されているとおりに、ATTACHSEC 属性のパラメーターを解釈します。ただし、CRTE を使用する CICS トランザクション・ルーティングには、 345 ページの『MRO でのトランザクション・ルーティング・セキュリティー』で説明するとおりに、特別なルールが適用されます。

ATTACHSEC 属性は、着信要求のサインオン要件を指定します。ご使用のシステムからリモート・システムに発行される要求には、影響がありません。そうした要求は、リモート・システムで処理されます。

ATTACHSEC 属性の以下の値は、MRO で有効です。

LOCAL

リモート・システムからのユーザー ID が不要であり、それを受け取った場合でも無視することを示します。ここで、CICS は、リンク・セキュリティー・プロファイルと同等のユーザー・セキュリティー・プロファイルを作成します。リモート・ユーザーに対して RACF プロファイルを指定する必要はありません。(LOCAL がデフォルト値です。)

リンク・セキュリティー・プロファイルのみで十分なセキュリティーがシステムに提供されると判断した場合は、ATTACHSEC(LOCAL) を指定します。

IDENTIFY
すべての接続要求でユーザー ID が予期されることを示します。システムのすべてのリモート・ユーザーは RACF で識別される必要があります。

例えば、リモート・ユーザーが別の CICS であるときに、リモート・システムがそのユーザーを検査することを CICS が信頼できると分かっている場合は、ATTACHSEC(IDENTIFY) を指定します。

以下のルールが IDENTIFY に適用されます。

- パスワードが、ATTACHSEC(IDENTIFY) が指定されたリンク上の、ユーザー ID が指定された接続要求に含まれている場合、CICS は接続要求を拒否し、セッションをアンバインドします。
- ヌルのユーザー ID または不明のユーザー ID を受け取った場合、CICS は接続要求を拒否します。
- ユーザー ID を受け取らない場合、USEDFLTUSER(YES) が接続に指定されていない限り、接続は拒否されます。この場合、CICS は、DFLTUSER システム初期設定パラメーターに指定されているとおり、デフォルト・ユーザーのセキュリティー機能を適用します。詳しくは、CICS デフォルト・ユーザー IDを参照してください。

注**:** 分散トランザクション処理 (DTP) トランザクションの場合は、MRO SEND コマンドまたは CONVERSE コマンドの前に BUILD ATTACH 要求を発行して、接続要求に端末ユーザーのユーザー ID を含める必要があります。

MRO でのリモート・ユーザー・サインオン状況**:**

ATTACHSEC(IDENTIFY) パラメーターを使用すると、最初の接続要求に関連した会話が完了した後に、リモート・ユーザーはサインオンしたままになります。

これ以降 CICS は、以下のいずれかのイベントが発生するまで、同じユーザーからの接続要求を新規サインオンなしで受け入れます。

- このユーザーの接続要求に関連した最後のトランザクションの完了後に、USRDELAY システム初期設定パラメーター システム初期設定パラメーターに指定された期間が経過します。
- CICS に通知されます。詳細については、 20 ページの『リモート・ユーザーの RACF プロファイルの変更』を参照してください。
- CICS 領域がシャットダウンします。

ユーザーを取り消すなどしてサインオン・リモート・ユーザーの RACF プロファイルを変更した場合、以下のいずれかの状態となるまで、CICS は最初の接続要求時に設定された許可を引き続き使用します。

- トランザクションが同期点を実行する。
- 接続要求が終了する。
- RACF が CICS にユーザー・プロファイルの変更を通知し、そのサインオン・ユーザー ID に関連付けられている接続要求が、LOCAL を除く ATTACHSEC のすべてのオペランドに対して完了したためにサインオフとなる。

- RACF が CICS にユーザー・プロファイルの変更を通知し、新規接続要求が行われ、**USRDELAY** システム初期設定パラメーターの値が有効期限切れになっていないためにサインオフとなる。このサインオフの後には、サインオンが続きます。

リモート・ユーザーに関する情報

MRO リンクを使用すると、ユーザーに関する情報は、リモート・システムから接続要求を使用して送信できます。

つまり、どのリモート・システムが要求を出しているかだけでなく、リモート・システムでどの実ユーザーが要求を出しているかにも基づいてリソースを保護できます。

このセクションでは、リモート・ユーザー・セキュリティーに関連するいくつかの概念と、CICS がユーザー情報を送受信する方法を説明しています。

ユーザーを RACF に定義する必要があります。リモート・ユーザーが RACF に定義されていない場合、そのリモート・ユーザーからのすべての接続要求は拒否されます。

CICS は ATTACHSEC(IDENTIFY) の会話でユーザー ID を送信します。 表 41 は、CICS が送信するユーザー ID を決定する方法を示しています。

表 *41. MRO* 接続時ユーザー *ID*

ローカル・タスクの特性	**TOR** によって **AOR** に送信されるユーザー **ID**
関連する端末があるタスク - ユーザー ID	端末ユーザー ID
関連する端末があるタスク - ユーザーのサインオンなし、および端末定義に USERID の指定なし	TOR からのデフォルト・ユーザー ID
インターバル制御 START コマンドによって開始された、関連する端末および USERID がないタスク (機能シップまたは DTP を使用する場合)	START コマンドを発行したタスクのユーザー ID
USERID オプションを指定して開始されたタスク	START コマンドに指定されたユーザー ID
CICS 内部システム・タスク	CICS 領域のユーザー ID
一時データ・トリガーによって開始された、関連する端末がないタスク	キューを定義する一時データ宛先定義に指定されたユーザー ID
一時データ・トリガーによって開始された、関連する端末があるタスク	端末ユーザー ID
PLTPI から開始されたタスク	PLTPIUSR システム初期設定パラメーターによって指定されたユーザー ID

MRO でのトランザクション・セキュリティー、リソース・セキュリティー、およびコマンド・セキュリティー

単一システム環境の場合と同様に、ユーザーには以下を行うための許可が必要です。

- トランザクションを接続する。
- トランザクションが使用するようにプログラムされているすべてのリソースにアクセスする。これにより、トランザクション・セキュリティー、リソース・セキュリティー、およびコマンド・セキュリティーと呼ばれるセキュリティー・レベルが実現します。

トランザクション・セキュリティー

単一システム環境の場合、トランザクションのセキュリティー要件はトランザクションを定義する際に示されます (トランザクション・セキュリティー を参照)。

MRO 環境では、トランザクションを開始する前に、以下の 2 つの基本セキュリティー要件を満たしている必要があります。

- リンクにはトランザクションを開始するための十分な権限が必要です。
- 要求を行っている「ユーザー」は、システムにアクセスしてトランザクションを開始するための十分な権限を持っている必要があります。

リソースおよびコマンド・セキュリティー

相互通信環境でのリソース・セキュリティーおよびコマンド・セキュリティーは、単一システム環境とほぼ同じ方法で処理されます。

リソースおよびコマンドのセキュリティー検査の実行時: リソースおよびコマンドのセキュリティー検査は、インストールされているトランザクション定義でそれらの検査が必須であると (例えば、図 25 に示すように CEDA DEFINE TRANSACTION コマンドなどで) 指定されている場合にのみ実行されます。

```
CEDA DEFINE TRANSACTION
  .
  RESSEC(YES)
  CMDSEC(YES)
  .
```

図 *25.* トランザクションのリソースおよびコマンドのセキュリティーの指定

トランザクションでリソース・セキュリティー検査が RESSEC(YES) を使用して指定されている場合、リンクとユーザーの両方ともに、接続されたトランザクションがアクセスするリソースに対する十分な権限が備えられている必要があります。

トランザクションでコマンド・セキュリティー検査が CMDSEC(YES) を使用して指定されている場合、リンクとユーザーの両方ともに、接続されたトランザクションが発行するコマンド (162 ページの表 11 を参照) に対する十分な権限が備えられている必要があります。

リソースおよびコマンド・セキュリティーの指定についての詳しい説明は、 100 ページの『リソース・セキュリティー』 および 162 ページの『CICS コマンド・セキュリティー』 を参照してください。

NOTAUTH 例外状態:

トランザクションがリソースへのアクセスを試行したが、リソース・セキュリティー検査に失敗した場合、NOTAUTH 条件が発生します。

トランザクションが CICS ミラー・トランザクションである場合、NOTAUTH 条件は、それを通常の方法で処理できる要求元トランザクションに返されます。

MRO でのトランザクション・ルーティング・セキュリティー

トランザクション・ルーティングにおいて、トランザクションにアクセスするためのユーザーの権限は、TOR および AOR の両方で検査できます。

TOR では、ローカル・トランザクションの場合と同じように、リモートとして定義されたトランザクションへのアクセス権限をユーザーが持っていることを確認するために、通常検査が行われます。この検査によって、中継プログラムの実行がユーザーに許可されるかどうかが決まります。

AOR では、トランザクションはその基本機能として、TOR 内の「実際の」端末を表すリモート端末 (「代理」端末) を持ちます。リモート端末が定義される方法 (トランザクション・ルーティングのためのリモート・リソースの定義を参照) は、ユーザー・セキュリティーの適用方法に影響を与えます。

- リモート端末の定義で USERID パラメーターが指定されない場合は、以下のようになります。
  - ATTACHSEC(IDENTIFY) が指定されたリンクの場合、ユーザーのトランザクション・セキュリティーおよびリソース・セキュリティーは、リモート・ユーザーのサインオン時に確立されます。ユーザーがサインオンで使用するユーザー ID は、(DFLTUSER システム初期設定パラメーター内で) 明示的または暗示的のいずれで表される場合であっても、このセキュリティー機能を有します (リモート・システムに割り当てられます)。
  - ATTACHSEC(LOCAL) が指定されたリンクの場合、トランザクション・セキュリティー、コマンド・セキュリティー、およびリソース・セキュリティーは、リンクの権限によって制限されます。

いずれの場合も、 339 ページの『MRO とのリンク・セキュリティー』に説明されている方法で、リンク・セキュリティーに対する検査が行われます。

注: トランザクション・ルーティングの際に、3 文字のオペレーター識別子が TOR から AOR の代理端末エントリーへ送信されます。この ID はセキュリティーのために使用されませんが、メッセージおよび監査証跡内で参照されることがあります。

トランザクション・ルーティング PSB が要求を出す際には、次の条件が両方とも満たされている必要があります。

- 接続定義の ATTACHSEC は LOCAL であってはならない (つまり、IDENTIFY、PERSISTENT、MIXIDPE、または VERIFY のいずれかになる)。
- リモート・システムのシステム初期設定パラメーターとして、PSBCHK=YES が指定されている。

CICS ルーティング・トランザクション CRTE

接続されたリモート・システム上にあるトランザクションをローカル・システムでリモートとして定義する代わりに、CICS ルーティング・トランザクション (CRTE) を IPIC、LU6.2、または MRO リンクと共に使用してそれらのトランザクションを実行できます。

CRTE は、あまり頻繁に使用しないトランザクションや、すべてのシステム上にある CEMT などのトランザクションに対して特に役立ちます。

CRTE を開始した端末が、リモート・システムで定義されているか、またはローカル・システムでシップ可能と定義されていることを確認してください。リモート・システムが保護されている場合、端末オペレーターは RACF 権限を必要とします。

CRTE で実行されるトランザクションに対する AOR でのセキュリティー検査は、ATTACHSEC (MRO リンクと LU6.2 リンクの場合) または USERAUTH (IPIC リンクの場合) で指定されている内容に依存していません。TOR にサインオンしているユーザー ID にも依存していません。その代わり、セキュリティー検査は、ユーザーが CRTE の使用時にサインオンするかどうかに依存しています。

- ユーザーがサインオンしない 場合、作成される代理端末は、AOR デフォルト・ユーザーに関連付けられます。トランザクションが実行されると、このデフォルトのユーザーに対してセキュリティー検査が実施されます。リンク・ユーザー ID に対しても検査が行われ、ルーティング・アプリケーション自体がリソースにアクセスする権限を持っているかどうかが確認されます。
- ユーザーがサインオンする 場合、CRTE の実行時に CESN トランザクションを使用して、代理がサインオン・ユーザーのユーザー ID を指します。リソースへのアクセスを試行するトランザクションの場合、セキュリティー検査は、代理のサインオン・ユーザーのユーザー ID と、リンク・ユーザー ID に対して行われます。

MRO での機能シップ・セキュリティー

CICS が機能シップ要求を受け取るときに、呼び出されるトランザクションはミラー・トランザクションです。

ミラー・トランザクションの CICS 提供の定義はすべて、リソース・セキュリティー検査を指定しますが、コマンド・セキュリティー検査は行いません。これは、リンクまたは他のシステムのユーザー・プロファイルのいずれかが必要な権限を持っていない場合に、リモート・リソースにアクセスできないことを意味します。

CICS 提供のミラー・トランザクションの定義がセキュリティー戦略のニーズに適していない場合は、グループ DFHISC 内の定義を独自のグループにコピーし、変更し、再インストールすることで変更できます。詳しくは、Security for CICS-supplied transactionsを参照してください。

リソース定義にリモート・リソースを含める場合、ローカル・リソースの場合と同じように、ローカルでセキュリティー検査が実行されるように調整できます。さらに、リソースを所有するシステムは、ユーザー ID を受け取ることができる場合、独立検査を適用するようにすることができます。したがって、セキュリティー制限の適用は、両方の側または一方の側に行うか、あるいはどちらにも行わないという選択ができます。

注: 機能シップされた要求に対して SYSID オプションを指定する場合、セキュリティー検査は、リモート・システムでは実行されますが、ローカル・システムではバイパスされます。 347 ページの図 26 では、どのような動作になるかを要約しています。

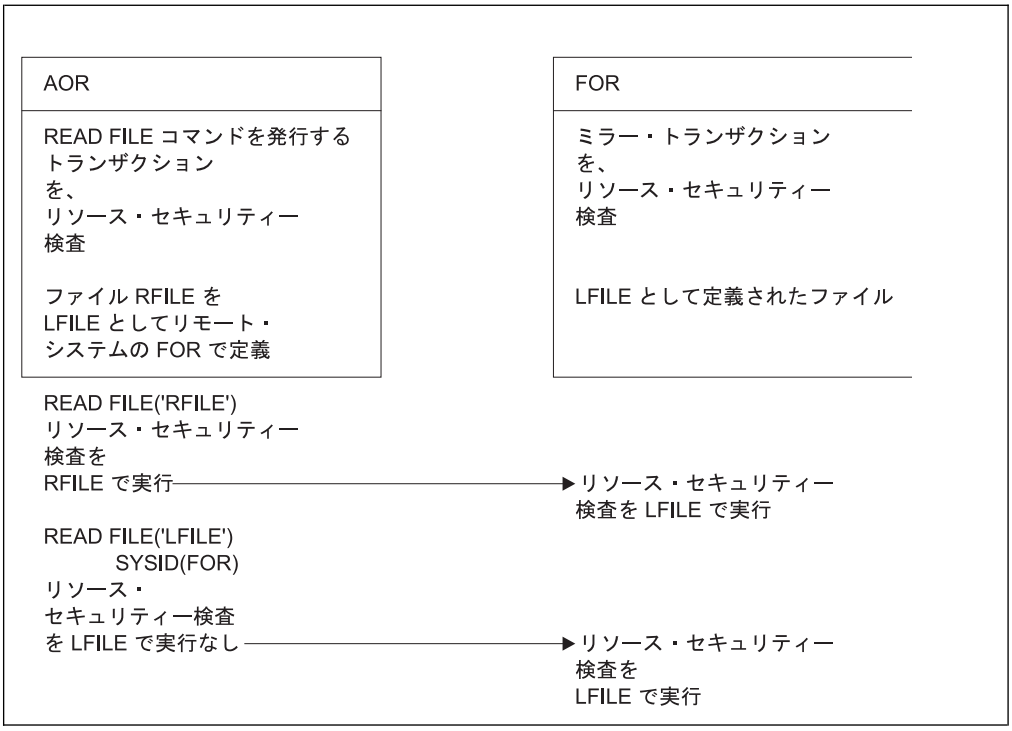


図 *26. SYSID* の指定ありまたは指定なしで実行されるセキュリティー検査：

この例は、アプリケーション専有領域内のトランザクションがリモート・ファイルに対してファイル制御要求を発行するときに、どのようなセキュリティー検査が実行されるかを示しています。

- アプリケーション専有領域 (AOR) 内で、ファイル RFILE は、ファイル専有領域 (FOR) 内の LFILE という名前で、リモートとして定義されます。ファイル制御要求を発行するトランザクションに対して、リソース・セキュリティー検査がアクティブです。
- FOR 内では、ミラー・トランザクションに対してリソース・セキュリティー検査がアクティブです。

これには 2 つのケースがあります。最初のケースは次のとおりです。

1. AOR 内のトランザクションが EXEC CICS READ FILE('RFILE') を発行します。リソース・セキュリティー検査がファイル RFILE に対して実行されます。
2. 要求は FOR に送信され、そこではリソース・セキュリティー検査がファイル LFILE に対して実行されます。

2 番目のケースは次のとおりです。

1. AOR 内のトランザクションが、EXEC CICS READ FILE('RFILE') を *SYSID* オプションを指定して 発行します。リソース・セキュリティー検査はファイル RFILE に対して実行されません。
2. 要求は FOR に送信され、そこではリソース・セキュリティー検査がファイル LFILE に対して実行されます。

SYSID オプションの指定に関するプログラミング情報については、READを参照してください。

MRO での分散プログラム・リンク・セキュリティー

CICS 分散プログラム・リンク (DPL) 機能により、プログラム (クライアント・プログラム) はリモート CICS 領域内の CICS プログラム (サーバー・プログラム) を呼び出すことができます。クライアント・プログラムは、CICS プログラムでも非 CICS プログラムでも構いません。

CICS クライアント・プログラムは、**EXEC CICS LINK PROGRAM** コマンドで SYSID オプションを指定することで DPL を使用します。または、PROGRAM リソースの REMOTESYSTEM オプションがリモート CICS 領域を既に指定している場合は、SYSID オプションを省略することで DPL を使用します。 **EXEC CICS LINK** コマンドの SYSID オプションがリモート CICS システムを指定する場合、クライアント領域はリソース・セキュリティー検査を実行しませんが、サーバー領域のリソース検査はそのまま実行されます。

非 CICS クライアント・プログラムは、DFHXCIS への呼び出しを使用して CICS システムへのラインをオープンし、その後、CICS プログラムにリンクします。これは、外部 CICS インターフェース (EXCI) と呼ばれます。リンク呼び出しのパラメーターの 1 つは、サーバー・プログラムの実行トランザクション ID です。このトランザクションを CICS に対して、プログラム DFHMIRS を実行し、プロファイル DFHCICSA を使用するように定義します。リンク呼び出しのもう 1 つのパラメーターはクライアントのユーザー ID であり、これは MRO 接続が ATTACHSEC(IDENTIFY) が指定されて定義されている場合に検証されます。

DFHXCIC 呼び出しでユーザー ID パラメーターを使用するには、クライアント・プログラムは、指定されたユーザー ID の代理ユーザー権限を持っている必要があります。詳しくは、 158 ページの『EXCI 呼び出しの代理ユーザー検査』を参照してください。

外部 CICS インターフェース・コマンドがセキュリティー検査に失敗した場合、クライアント・プログラムは USER_ERROR エラーを受け取ります。ただし、このエラーには他の原因がある可能性もあります。USER_ERROR 応答の各理由コード値は、コマンドを直接再発行できるかどうか、または使用中のパイプをまずクローズして再オープンする必要があるかどうかを示します。

サーバー・プログラムは、他の機能シップ CICS 要求に対するのと同じように、ミラー・トランザクションによって実行されます。ただし、ミラーに関連付けられるトランザクション名は、クライアント領域でプログラム・リンクがどのように呼び出されるかに依存しています。通常の接続セキュリティーがミラー・トランザクションに適用されるため、ユーザーはトランザクション名について認識しておく必要があります。

- トランザクション ID がリンク要求で指定されている場合、指定されたトランザクション名がミラーに使用されます。
- トランザクション ID がリンク要求で省略されていても、TRANSID オプションがクライアント領域のプログラム・リソース定義で使用されている場合、ミラーの名前はプログラムの TRANSID 仕様から取られます。
- そうでない場合、ミラー・トランザクションにはデフォルト名 CSMI が使用されます。

ミラーの実行トランザクション名へのアクセス権限をユーザーに付与します。ユーザー ID に権限が付与されるかどうかは、LOCAL または IDENTIFY の接続セキュリティーが使用されているかどうかによって異なります。これについては、『MRO での AOR で実行されるセキュリティー検査』で説明されています。サーバー領域で RESSEC(YES) を指定してミラー・トランザクションを定義する場合は、これらのユーザー ID に、ミラーによってリンクされるサーバー・プログラムにアクセスすることを許可します。サーバー・プログラムがいずれかの CICS リソースにアクセスする場合は、複数の同じユーザー ID に、それらのリソースへのアクセスを許可します。サーバー・プログラムがいずれかの SP タイプのコマンドを呼び出し、ミラー・トランザクションがサーバー領域内で CMDSEC(YES) が指定されて定義されている場合、複数の同じユーザー ID に、そのコマンドへのアクセスを許可します。

セキュリティー上の理由でミラー・トランザクションが接続できない場合、NOTAUTH 条件は発生しませんが、TERMERR 条件がクライアント領域内の発行元アプリケーションに返されます。ミラー・トランザクションが正常に接続されたが、サーバー領域内の分散プログラムへのリンクを許可されない場合、NOTAUTH 条件が発生します。セキュリティー上の理由からサーバー・プログラムがいずれかの CICS リソースへのアクセスに失敗した場合も、NOTAUTH 条件が発生します。

サーバー・プログラムは、サーバー領域で実行されるときは、CICS API コマンドの DPL サブセットに制限されます。サポートされないコマンドは、セキュリティー関連情報を返すコマンドなどです。制限されるコマンドに関するプログラミング情報については、LINK コマンドの例外条件を参照してください。 DPL の詳細については、DPL の概要を参照してください。

MRO での AOR で実行されるセキュリティー検査

このセクションでは、AOR でのセキュリティー検査の実行方法について概要を示します。

フロントエンド CICS 領域のユーザー ID が、デフォルトとして割り当てられます。ただし、ユーザー ID が SESSIONS 定義で指定され、リンク検査が行われる場合、使用されるユーザー ID は SESSIONS 定義のユーザー ID です。

表 42 から 350 ページの表 43 で示されている領域ユーザー ID は、SESSIONS 定義のユーザー ID です。この場合に示されているユーザー ID は、ジョブの実行ユーザー ID です。通常、このユーザー ID は、セキュリティー・マネージャー・ドメインによって返されます。

ATTACHSEC(LOCAL) が指定されている場合

表 42 は、ATTACHSEC(LOCAL) が指定されている場合の AOR での検査方法を示しています。

表 42. AOR—ATTACHSEC(LOCAL) を指定して実行されるセキュリティー検査

AOR の領域ユーザー **ID**	**AOR** のセッション定義のユーザー **ID**	**TOR** の領域ユーザー **ID**	**AOR** での検査
USERIDA	指定されていません	USERIDA	AOR DFLTUSER に対する検査

表 *42. AOR—ATTACHSEC(LOCAL)* を指定して実行されるセキュリティー検査 *(*続き*)*

AOR の領域ユーザー **ID**	**AOR** のセッション定義のユーザー **ID**	**TOR** の領域ユーザー **ID**	**AOR** での検査
USERIDA	USERIDA	任意	AOR DFLTUSER に対する検査
USERIDA	指定されていません	USERIDB	USERIDB に対する検査
USERIDA	USERIDB	任意	USERIDB に対する検査

ATTACHSEC(IDENTIFY) が指定されている場合

表 43 は、ATTACHSEC(IDENTIFY) が指定されている場合の AOR での検査方法を示しています。

表 *43. AOR—ATTACHSEC(IDENTIFY)* を指定して実行されるセキュリティー検査

AOR の領域ユーザー **ID**	**AOR** のセッション定義のユーザー **ID**	**TOR** の領域ユーザー **ID**	**AOR** での検査
USERIDA	指定されていません	USERIDA	FMH-5 ATTACH 検査のみ
USERIDA	USERIDA	任意	FMH-5 ATTACH 検査のみ
USERIDA	指定されていません	USERIDB	FMH-5 ATTACH 検査および USERIDB
USERIDA	USERIDB	任意	FMH-5 ATTACH 検査および USERIDB

データ・テーブルのセキュリティー

このトピックでは、CICS 共用データ・テーブルおよびカップリング・ファシリティー・データ・テーブルにセキュリティーを提供する方法を説明します。

CICS 共用データ・テーブルのセキュリティー

仮想記憶間サービスの使用時に共用データ・テーブルのセキュリティーを提供するには、以下について確認します。

- 共用データ・テーブル・サーバーとして機能しているファイル専有領域 (FOR) では、偽名を使用できません。これを確認する方法の詳細については、 351 ページの『SDT サーバー許可セキュリティー検査』を参照してください。
- アプリケーション専有領域 (AOR) は、アクセスを意図していないデータへのアクセス権限は取得できません。これは、接続時に、AOR が FOR へのアクセスを許可されていることと、ファイル・セキュリティーが実施されている場合は、要求されたファイルへのアクセスを AOR が許可されていることを確認することで防止できます。

これらのセキュリティー検査は、System Authorization Facility (SAF) によって実行され、RACF または同等のセキュリティー・マネージャーを呼び出します。

注: 領域は、共用データ・テーブル・サーバーとして機能する権限がない場合でも、ローカルにデータ・テーブルを使用することができます。

CICS 共用データ・テーブル (SDT) 機能は、領域レベルで作動する機能シップ・セキュリティーの主要な特性を再現します。ただし、以下の相違点に注意してください。

- SDT は、トランザクション・レベルでセキュリティー検査を実行するための FOR のメカニズムを提供しません (ATTACHSEC(IDENTIFY) または ATTACHSEC(VERIFY) に相当するものはありません)。したがって、AOR によって実行されるトランザクション・レベルの検査が一部のファイルに対して不適切であると考えられる場合は、それらのファイルが FOR 内のデータ・テーブルに関連付けられていないことを確認してください。
- SDT はセッションを使用しないため、SESSIONS での同等の事前設定セキュリティーはサポートしません。
- SDT は、インストール・パラメーター・リスト (INSTLN) 情報をセキュリティー・ユーザー出口に渡しません。

データ・テーブルのセキュリティー検査

データ・テーブルに関連付けられているファイルを共用する前に、セキュリティー検査の影響を考慮する必要があります。

SDT セキュリティーは既存の CICS ファイル・セキュリティー定義を使用しますが、SDT サーバー APPLID を保護リソースとして扱うことにも依存しています。SDT サーバーの APPLID は、RACF FACILITY リソース・クラス内の DFHAPPL.*applid* プロファイルによって表されます。

SDT サーバー許可セキュリティー検査

ある領域が SDT サーバーとして機能しようとする場合、RACF を呼び出して、そのユーザー ID に APPLID への必須アクセス権限があるかどうかを検査します。

呼び出しが失敗すると、領域はサーバーになるために必要な SDT サポートを初期設定できません。これにより、AOR が、SDT サーバーとして機能することを適切に許可されていない FOR からの偽データ・レコードを受け入れるリスクは最小化されます。この検査は、システムの初期設定時に SEC=NO が指定されている場合でも、バイパスされることは決してありません。

保護された APPLID のサーバーとして機能するには、SDT FOR のユーザー ID は、FACILITY クラス内の DFHAPPL.*applid* プロファイルに対する UPDATE (またはそれ以上の) 権限を持っている必要があります。以下の定義の例では、FOR の APPLID は CICSHF01 であり、そのユーザー ID は CICSSDT1 です。

```
RDEFINE FACILITY (DFHAPPL.CICSHF01) UACC(NONE)

PERMIT DFHAPPL.CICSHF01 CLASS(FACILITY) ID(CICSSDT1) ACCESS(UPDATE)
```

最初の例では、1 つの FOR に、ユーザー ID CICSSDT1 で実行している APPLID CICSHF01 を持つサーバーとして機能することを許可します。以下の例は、グループ SDTGRP1 のメンバーとして定義されたユーザー ID を持つ FOR のグループを許可し、FACILITY クラス内の総称プロファイルを使用して SDT サーバーとして機能する方法を示しています。

```
RDEFINE FACILITY (DFHAPPL.CICSTST*) UACC(READ)

PERMIT DFHAPPL.CICSTST* CLASS(FACILITY) ID(SDTGRP1) ACCESS(UPDATE)
```

SAF がアクセス要求を認可も拒否もしない場合**:**

指定されたリソースのセキュリティー・プロファイルが取得されない場合、SAF はアクセス要求を認可することも拒否することもしません。

このような状態では、次のようになります。

- セキュリティー・マネージャーがインストールされているものの、この MVS IPL の間に一時的に非アクティブまたは操作不能である場合は、要求は失敗します。この決定は、セキュリティー・マネージャーがアクティブであった場合、アクセスを拒否するプロファイルを取得した可能性があるという根拠に基づいて下されます。
- 要求は次の場合に成功します。
  - セキュリティー・マネージャーがありません。
  - アクティブなセキュリティー・マネージャーがありますが、FACILITY クラスは未定義または非アクティブです。
  - 問題となる APPLID を対象として含むプロファイルがありません。

  特定の FOR APPLID へのアクセスを制御することをユーザーが希望しているという証拠がないため、これらの場合には要求は許可されます。

AOR の CONNECT セキュリティー検査

CONNECT 時に実行されるセキュリティー検査では、バインド・セキュリティーとファイル・セキュリティーという 2 つのレベルのセキュリティーを利用できます。

- バインド・セキュリティーにより、CICS ファイル・セキュリティーなしで実行する FOR は、選択された AOR への共用アクセスを制限することができます。(ファイル・セキュリティーなしで実行すると、ランタイム・オーバーヘッドとセキュリティー定義の数が最小限に抑えられます。)
- ファイル・セキュリティーは、AOR 全体に適用される検査を SDT に実施させる場合に、FOR でアクティブ化できます。

ATTACHSEC(IDENTIFY) または ATTACHSEC(VERIFY) が機能シップで使用されている場合、SDT は、FOR がトランザクション・レベルで行うそれらのセキュリティー検査を実施する手段を提供していないことに注意してください。

バインド・セキュリティー**:**

FOR のデータ・テーブルのいずれかへの共用アクセスが許可されるには、AOR のユーザー ID は、FACILITY クラス内の FOR の DFHAPPL.*applid* に対して READ (またはそれ以上の) 権限を持っている必要があります。

この検査は、システムの初期設定時に SEC=NO が指定されている場合でも、バイパスされることは決してありません。以下の定義例では、3 つの CICS AOR (ユーザー ID が CICSAOR1、CICSAOR2、および CICSAOR3) のすべてが、DFHAPPL.CICSHF01 プロファイルによって表される、FOR に対する SDT アクセス権限を必要とします。

```
PERMIT DFHAPPL.CICSHF01 CLASS(FACILITY) ID(CICSAOR1 CICSAOR2 CICSAOR3)
       ACCESS(UPDATE)
```

SAF がアクセスを許可も拒否もしない場合は、サーバー LOGON の場合と同じように解決されます (352 ページの『SAF がアクセス要求を認可も拒否もしない場合』を参照)。結果が拒否である場合、CICS は AOR による FOR の APPLID に対する共用アクセスを許可しません。

同じリソースへのさまざまな (ただし階層的な) レベルのアクセスを使用して SDT サーバー許可セキュリティーおよびバインド・セキュリティーを制御すると、以下のような結果になることに注意してください。

- サーバーと同じユーザー ID を持つ領域はすべて、必ずそのサーバーにバインドできます。
- どのユーザー ID が特定の APPLID にバインドできるかの制御は、どのユーザー ID がその APPLID のサーバーとしてログオンできるかを制御せずに行うことはできません。

SDT バインド時のセキュリティーは、IPIC、ISC、および (事前設定セッションを使用する場合は) MRO によって採用されているものの中から、さまざまな定義を使用します。したがって、これらを一貫性のあるものにしない限り、機能シップの試行が拒否される場合は、SDT アクセスが付与される可能性があります (この逆も成り立ちます)。 MRO と SDT はいずれも同じクラスを使用するため、ISC の場合のみ、SDT CONNECT セキュリティーは、機能シップよりも前または後のいずれかにセキュリティー定義の変更に反応する可能性があります。

FOR でファイル・セキュリティーが強制されていない場合 (つまり、システム初期設定時に SEC=NO または XFCT=NO が指定された場合)、FOR へのバインドが許可されている AOR も、その FOR のすべての共用データ・テーブルにアクセスすることが許可されます。

ファイル・セキュリティーが強制されている場合、バインドが許可されている AOR は、AOR と FOR のユーザー ID が同じである場合には引き続きフリー・アクセスが許可されます (未定義のユーザー ID は同じであるとは見なされません)。

ファイル・セキュリティー**:**

バインド・セキュリティー検査の後、ファイル・セキュリティーが FOR で強制されている場合、FOR は AOR が FOR に「サインオン」することを許可されているかどうかを検査します。

このセキュリティー検査はオプションであり、AOR のユーザー ID が FOR のユーザー ID とは異なる場合にのみ適用されます。これは MRO 環境内の ATTACHSEC(LOCAL) と同等です (341 ページの『MRO でのユーザー・セキュリティー』を参照)。さらに AOR には、FOR 内のアクセスしようとしているファイルへの READ 権限が必要です。

FOR によってファイル・セキュリティー検査を実装するには、次のようにします。

- システム初期設定パラメーター SEC=YES で FOR を初期設定します。
- APPL 一般リソース・クラス内の FOR の APPLID プロファイルへの READ 権限を AOR に付与します。

- XFCT システム初期設定パラメーターに適切な値を指定します。
- AOR の領域ユーザー ID に、XFCT システム初期設定パラメーターで指定されたファイル・リソース・プロファイル内の必須ファイルへの READ 権限を付与します。

例えば、以下のようにして、APPL プロファイルを FOR に対して APPLID CICSHF01 で定義し、PERMIT コマンドで AOR がユーザー ID CICSAOR1 および CICSAOR2 で CICSHF01 にサインオンできるように定義します。

```
RDEFINE APPL CICSHF01 UACC(NONE) NOTIFY(sys_admin_userid)
PERMIT CICSHF01 CLASS(APPL) ID(CICSAOR1 CICSAOR2) ACCESS(READ)
```

ファイルへのアクセスの許可については、 112 ページの『ファイルのセキュリティー』を参照してください。

SAF が要求を許可も拒否もしない場合は、サーバー LOGON の場合と同じように解決されます (352 ページの『SAF がアクセス要求を認可も拒否もしない場合』を参照)。

ユーザー ID が FOR のアプリケーションにサインオンすることが許可されている場合は、AOR のユーザー ID が指定のファイルの読み取りを許可されていない場合を除き、CONNECT 要求は成功します。許可されていない場合、CONNECT 要求は AOR のユーザー ID が未定義の場合と同じように処理されます。

ファイル・セキュリティーが FOR で強制されており、AOR のユーザー ID が未定義 の場合は、FOR のデフォルト・ユーザー ID (DFLTUSER システム初期設定パラメーターで指定されている) が指定のファイルの読み取りを許可されている場合を除き、CONNECT 要求は失敗します。

機能シップは、FOR でファイル制御リソース・クラスの再作成が完了すると、AOR のファイルへのアクセスが取り消されていることを検出します。ただし、有効な接続がある場合は、何らかの原因で接続が切断されるまで、SDT は引き続きアクセスを許可します。 主記憶域内のリソース・プロファイルのリフレッシュを参照してください。

注意**:** 機能シップに MRO の代わりに ISC を使用する場合は、FOR の SECURITYNAME パラメーターの値が AOR のユーザー ID と同じであることを確認します。機能シップに MRO の代わりに IPIC を使用する場合は、IPCONN SECURITYNAME パラメーターの値、または (該当する場合は) FOR 内の AOR の証明書のユーザー ID が、AOR のユーザー ID と同じであることを確認します。同じでない場合は、SDT CONNECT と機能シップ・セキュリティー検査に不整合があります。

カップリング・ファシリティー・データ・テーブルのセキュリティー

CICS と MVS は、RACF 機能を使用して、カップリング・ファシリティー・データ・テーブルのセキュリティーを以下の範囲で提供します。

1. カップリング・ファシリティー・リスト構造へのサーバーのアクセスの許可
2. サーバーの許可

3. カップリング・ファシリティー・データ・テーブル・プールへの CICS 領域のアクセスの許可
4. CFDT への CICS 領域の許可
5. ファイル・リソース・セキュリティー検査

オプションである項目 4 および 5 は例外として、それ以外のセキュリティー検査は自動的に実行され、決してバイパスされることはありません。このリスト内の項目 2 および 3 では、System Authorization Facility (SAF) が許可も拒否もしない場合、アクセスは CICS 共用データ・テーブル・サポートの LOGON セキュリティー検査と同じ方法で解決されます (詳細については、 351 ページの『SDT サーバー許可セキュリティー検査』を参照してください)。

サーバー始動パラメーターによって制御される、オプションのセキュリティー検査は、カップリング・ファシリティー・データ・テーブル・プール内の特定のテーブルへのアクセスを制御するために提供されています。これについては、 356 ページの『Authorizing a CICS region to a coupling facility data table』で詳しく説明しています。

Authorizing server access to a list structure

各カップリング・ファシリティー・データ・テーブル・サーバーは、カップリング・ファシリティー・データ・テーブルのプールが含まれているカップリング・ファシリティー・リスト構造にアクセスする必要があります。

アクセスを許可するには、サーバー領域ユーザー ID に、IXLSTR.*structure_name* という FACILITY クラスの一般リソース・プロファイルへの ALTER 権限が必要です。

- サーバー領域ユーザー ID とは、ジョブまたは開始タスクの実行ユーザー ID のことです。
- カップリング・ファシリティー・データ・テーブルの構造名は、DFHCFLS_*poolname* という形式になります。

例えば、カップリング・ファシリティー・データ・テーブルが PRODCFT1 というプール内で定義されている場合、このプールのリスト構造は、CFRM ポリシー内では DFHCFLS_PRODCFT1 という名前になります。このリスト構造にアクセスするには、プール PRODCFT1 のサーバー・ユーザー ID に、次のように定義される IXLSTR プロファイルへの ALTER 権限が必要です。

```
RDEFINE FACILITY IXLSTR.DFHCFLS_PRODCFT1 UACC(NONE)
PERMIT IXLSTR.DFHCFLS_PRODCFT1 CLASS(FACILITY) ID(server_userid) ACCESS(ALTER)
```

サーバーの許可

カップリング・ファシリティー・データ・テーブル (CFDT) サーバーが、特定のカップリング・ファシリティー・データ・テーブル・プールに対して起動すると、CICS 許可仮想記憶間 (AXM) サービスは RACF を呼び出して、サーバーがそのプールのサーバーとして機能することが許可されるようにします。

CFDT サーバーがその指定されたプールのサーバーとして機能することを許可するには、サーバー領域ユーザー ID に、DFHCF.*poolname* という FACILITY クラス一般リソース・プロファイルへの CONTROL 権限を付与します。

例えば、プールが PRODCFT1 の場合、プロファイルおよび必要な PERMIT ステートメントを次のように定義します。

```
RDEFINE FACILITY DFHCF.PRODCFT1 UACC(NONE)
PERMIT DFHCF.PRODCFT1 CLASS(FACILITY) ID(server_userid) ACCESS(CONTROL)
```

CFDT プールに対する CICS 領域の許可

各 CICS 領域には、カップリング・ファシリティー・データ・テーブル・プールに接続するための許可が必要です。 CICS 領域がサーバーとそのプールに接続することを許可するには、CICS 領域に、サーバーのプール用の FACILITY クラス・プロファイルへの UPDATE 権限を付与します。

例えば、プールが PRODCFT1 の場合、必要な PERMIT ステートメントを次のように定義します。

```
PERMIT DFHCF.PRODCFT1 CLASS(FACILITY) ID(CICS_region_userid) ACCESS(UPDATE)
```

Authorizing a CICS region to a coupling facility data table

CICS 領域からのカップリング・ファシリティー・データ・テーブル (CFDT) プールへのアクセスを制御できるのに加えて、オプションでプール内の各 CFDT へのアクセスも制御できます。

このセキュリティー検査は、アクティブである場合、CICS 領域がカップリング・ファシリティー・データ・テーブルに初めて接続するたびに、サーバーによって実行されます。 CFDT サーバー・セキュリティー・パラメーターについて詳しくは、Security parametersを参照してください。

このリソース・セキュリティー検査は、SECURITYCLASS サーバー初期設定パラメーターで指定された一般リソース・クラスで定義されているプロファイルを使用して、カップリング・ファシリティー・データ・テーブル・サーバー領域によって所有されている CICS ファイルに対する場合と同じように実行されます。このデフォルトは FCICSFCT クラスです。プロファイル名には、ファイル・リソース定義で定義されているテーブル名を使用します。

オプションで、SECURITYPREFIX=YES をサーバー初期設定パラメーターとして指定して、サーバー領域ユーザー ID を接頭部として使用することで、プロファイル名に接頭部を付けることができます。このセキュリティー検査用の接頭部は、サーバー初期設定パラメーター SECURITYPREFIXID を使用してカスタマイズできます。

以下の例では、PRODCFT1 というプールのセキュリティー・パラメーターを示しています。

```
POOLNAME=PRODCFT1
SECURITY=YES
SECURITYCLASS=FCICSFCT
SECURITYPREFIX=NO
```

例えば、プールが PRODCFT1 の場合、プロファイルおよび必要な PERMIT ステートメントを次のように定義します。

```
RDEFINE FCICSFCT PRODCFT1 UACC(NONE)
PERMIT PRODCFT1 CLASS(FCICSFCT) ID(region_userid) ACCESS(UPDATE)
```

カップリング・ファシリティー・データ・テーブル・サーバーは、グローバル・ストレージ内セキュリティー・プロファイルの使用を必要とする、仮想記憶間モード FASTAUTH 検査を発行することで、テーブル・セキュリティー検査を実行します。 FASTAUTH 検査への応答でゼロ以外の戻りコードをサーバーが受け取ると、アクセスは失敗します。外部セキュリティー・マネージャーが仮想記憶間モード FASTAUTH およびグローバル・ストレージ内プロファイルをサポートしていない場合、カップリング・ファシリティー・データ・テーブル・セキュリティー検査は実行できず、テーブル・セキュリティー検査が指定されている場合は、サーバー初期設定時にエラー・メッセージが発行されます。指定可能なすべてのサーバー初期設定パラメーターについては、カップリング・ファシリティー・データ・テーブル・サーバーのパラメーターを参照してください。

ファイル・リソース・セキュリティー検査

カップリング・ファシリティー・データ・テーブルでは、ファイルの通常の CICS リソース・セキュリティーがサポートされています。 CICS は、XFCT システム初期設定パラメーターで指定された一般リソース・クラスで定義されたプロファイルを使用して、カップリング・ファシリティー・データ・テーブルにアクセスするトランザクションのサインオン・ユーザーに対して、通常のファイル・リソース・セキュリティー検査を実行します。

CICS ファイル・セキュリティーの詳細については、 112 ページの『ファイルのセキュリティー』を参照してください。

第 8 章 TCP/IP クライアントのセキュリティー

この部分では、CICS が TCP/IP 通信プロトコルを使用してクライアント/サーバー構成に参加するときにアプリケーションを保護するための方法を解説しています。

TCP/IP クライアントのセキュリティーについて

クライアントとサーバー間の TCP/IP 接続は、特にインターネットを使用する場合は、悪意のあるパーティーによる攻撃に対して脆弱になります。

攻撃者は以下を試行する可能性があります。

- クライアントとサーバーの間をフローする機密情報を読み取ります。

  クライアントとサーバーの間をフローするデータを暗号化することにより、システムを保護できます。
- クライアントとサーバーの間をフローする機密情報を改ざんします。

  クライアントとサーバーの間をフローするデータを暗号化し、改ざんされたデータを検出するすることにより、システムを保護できます。
- クライアント・システムの正当なユーザーになりすまします。

  ユーザーを認証することにより、システムを保護できます。

CICS は、ユーザーを認証するためのいくつかのスキームをサポートします。詳細については、 362 ページの『識別と認証』を参照してください。

CICS は、Secure Sockets Layer (SSL) セキュリティー・プロトコルまたは Transport Layer Security (TLS) プロトコルを使用して、セキュアな TCP/IP 接続をサポートできます。詳細については、 368 ページの『セキュリティー・プロトコルのサポート』を参照してください。

メッセージ保護

メッセージ保護 は、機密メッセージが転送中に監視されたり、不正に変更されたりできないようにするために使用される技法を示す用語です。

メッセージ保護には、以下の 2 つの局面があります。

機密性
: メッセージの内容が傍受されないように保護します。

保全性
: メッセージを不正な変更から保護します

機密性 は、公開鍵暗号化方式を使用してメッセージ (またはその一部) を暗号化し、メッセージの対象とする受信者のみが読めるようにすることで実現されます。

© Copyright IBM Corp. 1989, 2018

保全性 は、メッセージにデジタル署名して、対象とする受信者が、メッセージが不正に変更されていないことを確信できるようにすることで実現されます。

公開鍵暗号化

公開鍵暗号化は、2 つの鍵を使用する暗号システムです。それは、すべてのユーザーに知られる可能性がある公開鍵 と、情報交換のために特定のパーティーにのみ知られる関連した秘密鍵 です。

公開鍵の暗号化に使用される秘密鍵と公開鍵は、以下のように互いに関連しています。

- 公開鍵から秘密鍵の値を推測したり、秘密鍵から公開鍵を推測することはできません。

  秘密鍵は安全に保管され、所有者以外に知らされないようにする必要がありますが、公開鍵はどのユーザーでも自由に入手でき、秘密鍵のセキュリティーが損なわれるリスクはありません。
- 公開鍵を使って暗号化された情報は、秘密鍵だけを使って暗号化解除できます。

  どのユーザーでも情報を暗号化して、それを秘密鍵の所有者に安全に送信することができます。公開鍵を使用して暗号化されたデータは、秘密鍵の所有者のみが読み取り可能です。
- 秘密鍵を使って暗号化された情報は、公開鍵だけを使って暗号化解除できます。

  公開鍵を使って暗号化解除される情報を暗号化できるのは、秘密鍵の所有者だけです。どのパーティーでも公開鍵を使用して、暗号化された情報を読み取ることができます。ただし、公開鍵を使用して暗号化解除できるデータは、秘密鍵の所有者から発信されていることが保証されます。

公開鍵を知っていても、対応する秘密鍵の所有者の身元を保証するわけではありません。そのため、公開鍵を使用した情報の暗号化では、暗号化された情報が悪人の手に渡るのを自動的に防ぐことはできません。公開鍵を安全に使用して情報の暗号化または暗号化解除を実行するには、事前に秘密鍵の所有者の身元が保証されている必要があります。この保証は、公開鍵を秘密鍵の所有者の身元にバインドするデジタル証明書 によって提供されます。

デジタル署名

デジタル署名とは、データが変更されていないこと、およびデータがメッセージの署名者から発信されたことをデータの受信者に保証するために、データに添付される情報です。デジタル署名は、紙文書の手書きの署名と同等の機能を実行します。

デジタル署名は、メッセージ送信者の秘密鍵で暗号化されたメッセージ・ダイジェスト で構成されます。メッセージ・ダイジェストは元のメッセージよりずっと短く、ハッシュ化 と呼ばれるプロセスを使用してメッセージから作成されます。メッセージ・ダイジェストから元のメッセージを再構成することはできません。メッセージを署名と組み合わせたものが、署名付きメッセージ です。

署名されたメッセージの受信者は、送信者の公開鍵を使用して署名を暗号化解除し、このようにしてメッセージ・ダイジェストに戻します。成功の場合、送信者のみが秘密鍵を持っているため、メッセージが送信者によって署名されたことを示し

ます。受信者は文書データをメッセージ・ダイジェストにハッシュ化し、そのメッセージ・ダイジェストを、署名を暗号化解除して取得したメッセージ・ダイジェストと比較します。両方のダイジェストが同じ場合、受信者は、署名されたメッセージが変更されていないことを確信できます。

デジタル署名は機密性を確保しません。言い換えれば、暗号化データでないデータにはデジタル署名を付けることができます。

公開鍵を知っていても、対応する秘密鍵の所有者の身元を保証するわけではありません。そのため、公開鍵を使用した情報の暗号化では、暗号化された情報が悪人の手に渡るのを自動的に防ぐことはできません。公開鍵を安全に使用して情報の暗号化または暗号化解除を実行するには、事前に秘密鍵の所有者の身元が保証されている必要があります。この保証は、公開鍵を秘密鍵の所有者の身元にバインドするデジタル証明書 によって提供されます。

デジタル証明書

デジタル証明書は、公開鍵を秘密鍵の所有者の ID にバインドする、デジタル署名されたデータ構造です。デジタル証明書を使用すると、公開鍵のユーザーが、対応する秘密鍵の所有権を確信できるようにします。

デジタル証明書は、認証局 (CA) と呼ばれる信頼できる機関によって発行され、通常はメッセージの送信者と受信者で別々に発行されます (ただし、送信者と受信者の間に信頼関係がある場合、証明書はいずれか一方が発行できます)。証明書は、CA の秘密鍵を使用して暗号化されます。また、CA の公開鍵を使用して暗号化解除でき、証明書を読み取る必要があるユーザーは誰でも自由に使用できます。

暗号化解除の後、有効な証明書は、証明書が実際に CA によって発行されたこと、および証明書が改ざんおよび偽造されていないことを読者に保証します。

デジタル証明書には、証明書の所有者を特定する情報と証明書の所有者の公開鍵が含まれており、CA によってデジタル署名されています。証明書を含むメッセージの受信者は、CA の公開鍵を使用して証明書を暗号化解除し、それが CA によって発行されたことを確認してから、送信者の公開鍵と証明書に保持された識別情報を取得します。

X.509 証明書**:**

ITU-T 勧告 X.509 は、デジタル証明書に幅広く使用されている形式を定義します。

X.509 証明書には、以下の情報が含まれています。

- 2 つの識別名。これは、証明書を発行した認証局 (CA) と、対象 (証明書の発行先の個人または組織) を一意的に識別します。識別名には、次のようないくつかのオプションのコンポーネントが含まれます。
  - 共通名
  - 組織単位
  - 組織
  - 局所性
  - 都道府県
  - 国

- デジタル署名。署名は、次の公開鍵暗号化技法を使用して、認証局によって作成されます。
  1. 安全なハッシュ・アルゴリズムを使用して、証明書の内容のダイジェストを作成します。
  2. ダイジェストは、認証局の秘密鍵を使用して暗号化されます。
  3. 署名は、CA の公開鍵を使用して暗号化解除されます。
  4. 証明書の内容の新しいダイジェストが作成され、暗号化解除された署名と比較されます。何らかの不一致がある場合、証明書が改変された可能性があることを示唆しています。このように、証明書が発行された後にその証明書に対して変更が行われていないことが、デジタル署名によって受信者に保証されます。
- サブジェクトのドメイン・ネーム。受信者は、これを証明書の実際の送信者と比較します。
- サブジェクトの公開鍵。

識別と認証

識別 とはユーザーの ID を確立するプロセスのことであり、認証 とは特定の ID を使用したいというユーザーからの要求を、サービスが資格情報 (通常はパスワードまたは証明書) を使って確認するプロセスのことです。

識別

識別は、ユーザーの ID を確立するのに使用されるプロセスです。

CICS では、識別はいくつかの方法で実行できます。

- クライアントはユーザー ID を直接提供できます。通常、これは認証プロセスの一環として行われます。

  基本認証を ECI アプリケーション・プロトコルおよび HTTP アプリケーション・プロトコルとともに使用する場合、この方法でユーザーを識別できます。
- クライアントは、認証プロセス中にユーザー ID 以外の情報 (例えば、SSL クライアント証明書) を提供できます。情報は、セキュリティー・マネージャーでユーザー ID にマップされます。

  SSL クライアント証明書認証を HTTP アプリケーション・プロトコルとともに使用する場合、この方法でユーザーを識別できます。
- ユーザー ID は、インバウンド要求ごとに呼び出されるユーザー置き換え可能プログラムで提供できます。

  HTTP アプリケーション・プロトコルでは、アナライザー・プログラムがユーザー ID を提供できます。
- ユーザー ID はインバウンド要求の URIMAP 定義で提供できます。URIMAP 定義を使用して HTTP アプリケーション・プロトコルで要求を処理する場合、この方法でユーザーを識別できます。

これらのいずれかの方法を使用してユーザー ID を提供しない場合、デフォルト・ユーザー ID が使用されます。

HTTP ユーザーの識別**:**

識別は、ユーザーの ID を確立するのに使用されるプロセスです。これは、HTTP アプリケーション・プロトコルのユーザー の ID を確立する方法です。

このタスクについて

HTTP アプリケーション・プロトコルの場合、次の方法でユーザーを識別できます。

- ユーザー ID は、HTTP 基本認証を使用して Web クライアントから取得できます。
- Web ブラウザーからクライアント証明書が送信される場合は、その証明書に関連付けられたユーザー ID を使用できます。

  次の 2 つの方法で、証明書と RACF ユーザー ID を関連付けることができます。

  – RACF コマンドを使用して、証明書をユーザー ID に関連付けることができます。
  – CICS で、自動的に RACF コマンドを実行して、証明書をユーザー ID (HTTP 基本認証を使用して Web クライアントから取得される) に関連付けることができます。

  これを行う方法については、 382 ページの『RACF ユーザー ID と証明書との関連付け』を参照してください。

アプリケーション生成の応答のみの場合、次のように CICS で Web クライアントの代わりにユーザー ID を提供することもできます。

- 要求に対する処理パスで使用されるアナライザー・プログラムで提供します。
- 要求に対する URIMAP 定義の USERID 属性で提供します。
- CICS デフォルト・ユーザー ID として提供します。

クライアントによって提供されなかったユーザー ID を設定するために、URIMAP 定義またはアナライザー・プログラムを使用する場合、または CICS デフォルト・ユーザー ID の使用を許可する場合、クライアント ID の認証はありません。この処置を行うのは、ユーザー独自のクライアント・システムで通信する際に、すでにそのシステムのユーザーが認証済みであり、セキュアな環境でサーバーと通信する場合のみです。

HTTP 応答がアプリケーションによって提供される場合 (アプリケーション生成の応答)、ユーザー ID の優先順位は、次のとおりです。

1. アナライザー・プログラムを使用して設定したユーザー ID。 このユーザー ID は、Web クライアントから取得された、または URIMAP 定義によって提供されたユーザー ID をオーバーライドできます。
2. 基本認証を使用して Web クライアントから取得したユーザー ID、または Web クライアントによって送信されたクライアント証明書に関連付けられたユーザー ID。 接続に認証が必要なのに、クライアントが認証済みユーザー ID を提供していない場合、要求はリジェクトされます。
3. 要求の URIMAP 定義で指定したユーザー ID。
4. 他に判別できるユーザー ID がない場合、CICS デフォルト・ユーザー ID。

HTTP 応答が、CICS 文書テンプレートまたは z/OS UNIX ファイルを指定する URIMAP 定義によって提供される場合 (静的応答)、 Web クライアント用に使用されるユーザー ID は、基本認証を使用して Web クライアントから取得したユーザー ID であるか、または Web クライアントによって送信されたクライアント証明書に関連付けられたユーザー ID です。 静的応答の場合、Web クライアントの代わりにユーザー ID を提供することはできず、Web クライアントから取得した認証済みユーザー ID をオーバーライドすることもできません。

静的応答の場合に、トランザクション用にリソース・セキュリティー検査を指定した場合、CICS では、提供されたユーザー ID のみが使用されます。 静的応答には、デフォルト・ユーザー ID は必要ありません。 Web クライアントからユーザー ID が提供されない場合、リソース・セキュリティーがトランザクションに対してアクティブであっても、リソース・セキュリティー検査は実行されません。

注**:** CICS は、ここに示す処理を行う際に、パスワード検証を使用してユーザー ID を検証します。 CICS は、CICS 領域へのログインに使用される各ユーザー ID に対して、1 日 1 回完全検証要求を実施します。 RACROUTE REQUEST=VERIFYX マクロを使用した完全検証要求により、RACF はユーザー ID の最終アクセス日時を記録し、ユーザー統計を書き込むことになります。

ユーザーの識別に使用される方式は、TCPIPSERVICE 定義の AUTHENTICATE 属性および SSL 属性によって決まります。

表 *44. HTTP* クライアントのユーザーの識別方法

AUTHENTICATE	**SSL**	ユーザーの識別方法
NO	NO または YES	クライアントは、ユーザー ID を提供しません。 ユーザー ID は、アナライザー・プログラムまたは URIMAP 定義によって提供されるか、または該当する場合には、デフォルトの CICS デフォルト・ユーザー ID となることが許可されます。
NO	CLIENTAUTH、または ATTLSAWARE	クライアントが、ユーザー ID に関連付けられている証明書を送信する場合、そのユーザー ID がアナライザー・プログラムによってオーバーライドされない限り、そのユーザー ID が適用されます。 クライアントが、ユーザー ID に関連付けられていない証明書を送信する場合、ユーザー ID は、アナライザー・プログラムまたは URIMAP 定義によって提供されるか、または該当する場合には、デフォルトの CICS デフォルト・ユーザー ID となることが許可されます。

表 44. *HTTP* クライアントのユーザーの識別方法 (続き)

AUTHENTICATE	**SSL**	ユーザーの識別方法
BASIC	すべての値	クライアントが、ユーザー ID に関連付けられている証明書を送信する場合、そのユーザー ID がアナライザー・プログラムによってオーバーライドされない限り、そのユーザー ID が適用されます。 クライアントが、ユーザー ID に関連付けられていない証明書を送信する場合、ユーザー ID は HTTP 基本認証を使用してクライアントから取得され、そのユーザー ID が証明書に登録されます。 クライアントが証明書を送信しない場合、このユーザー ID はクライアントから HTTP 基本認証を使用して取得され、アナライザー・プログラムによってオーバーライドできます。
CERTIFICATE	CLIENTAUTH、または ATTLSAWARE	クライアントが、ユーザー ID に関連付けられている証明書を送信する場合、そのユーザー ID がアナライザー・プログラムによってオーバーライドされない限り、そのユーザー ID が適用されます。 クライアントが、ユーザー ID に関連付けられていない証明書を送信する場合、または証明書を送信しない場合は、接続がリジェクトされます。
AUTOREGISTER	CLIENTAUTH、または ATTLSAWARE	クライアントが、ユーザー ID に関連付けられている証明書を送信する場合、そのユーザー ID がアナライザー・プログラムによってオーバーライドされない限り、そのユーザー ID が適用されます。 クライアントが、ユーザー ID に関連付けられていない証明書を送信する場合、ユーザー ID は HTTP 基本認証を使用してクライアントから取得され、そのユーザー ID が証明書に登録されます。 クライアントが証明書を送信しない場合、接続がリジェクトされます。
AUTOMATIC	NO または YES	ユーザー ID は、HTTP 基本認証を使用してクライアントから取得されます。 これは、アナライザー・プログラムによってオーバーライドすることができます。

表 44. *HTTP* クライアントのユーザーの識別方法 (続き)

AUTHENTICATE	SSL	ユーザーの識別方法
AUTOMATIC	CLIENTAUTH、または ATTLSAWARE	クライアントが、ユーザー ID に関連付けられている証明書を送信する場合、そのユーザー ID がアナライザー・プログラムによってオーバーライドされない限り、そのユーザー ID が適用されます。 クライアントが、ユーザー ID に関連付けられていない証明書を送信する場合、ユーザー ID は HTTP 基本認証を使用してクライアントから取得され、そのユーザー ID が証明書に登録されます。 クライアントが証明書を送信しない場合、ユーザー ID は、HTTP 基本認証を使用してクライアントから取得されます。

注:

1. このテーブルは、無効な AUTHENTICATE 属性および SSL 属性の値の組み合わせをリストしていないので、その組み合わせは TCPIPSERVICE 定義内に指定できません。
2. HTTP 基本認証が使用される場合、CICS はパスワードを検査します。パスワードが無効である場合、接続はリジェクトされます。

ECI ユーザーの識別:

ECI プロトコルの場合、基本認証を使用してユーザーを識別できます。

ECI クライアントの TCPIPSERVICE 定義に ATTACHSEC(VERIFY) を指定します。ユーザーを識別しないようにするには、ATTACHSEC(LOCAL) を指定します。

IPIC ユーザーの識別:

識別は、ユーザーの ID を確立するのに使用されるプロセスです。これは、IPIC プロトコルに対してユーザーの ID を確立する方法です。

IPIC プロトコルの場合、基本認証を使用してユーザーを識別できます。IPCONN リソース定義で USERAUTH(VERIFY) または USERAUTH(IDENTIFY) を指定します。ユーザーを識別しない場合は、USERAUTH(LOCAL) を指定します。

認証

多くのシステムでは、ユーザーの認証性はユーザーが提供したパスワードを確認することによって検証されます。

パスワードが傍受される可能性がないシステムでは、このレベルの認証で十分です。ただし、セキュアでないネットワークでは、パスワードが傍受されて、システムの正当なユーザーになりすますために使用される可能性があります。

アプリケーションがインターネット経由でユーザーによってアクセスされたり、組織の制御の範囲外のユーザーによってアクセスされたりする環境では、よりセキュアな認証方式が必要です。

他方、制限されたレベルの認証で十分な状況もあります。クライアント・システムがそのユーザーを認証し、セキュアな環境でサーバーと通信している場合、サーバーでユーザーを認証する必要はありません。ただし、クライアントの認証メカニズムに全面的に依存することになります。

CICS は、以下の認証スキームをサポートします。

基本認証
: クライアントの ID はパスワードによって認証されます。このレベルの認証は、パスワードが傍受されてユーザーのなりすましに使用される可能性がない環境に適しています。

  基本認証は、HTTP、ECI、および IPIC の各アプリケーション・プロトコルとともに使用できます。

SSL クライアント証明書による認証
: クライアントの ID は、信頼のおける第三者機関 (または認証局) が発行したクライアント証明書を使用して認証されます。このレベルの認証は、ネットワークをフローする情報が傍受されてユーザーのなりすましに使用される可能性がある環境に適しています。

  SSL クライアント証明書認証は、HTTP アプリケーション・プロトコルおよび IPIC アプリケーション・プロトコルとともに使用できます。

CICS は、ここに示す処理を行う際に、パスワード検証を使用してユーザー ID を検証します。

CICS は、毎日初めて特定のユーザー ID が CICS 領域へのログインに使用されるか、**VERIFY PASSWORD** コマンドを使用して検証されたときに、完全検証要求を実施します。完全検証要求により、ユーザー ID の最終アクセス日時が記録され、ユーザー統計が書き込まれます。完全検証は、正しくないパスワードが入力された場合、および次の成功した要求時にも実施されます。それ以外の場合、**VERIFY PASSWORD** は高速パス方式を使用してパスワードを検証します。使用されている SAF インターフェースの詳細については、CICS セキュリティー制御点を参照してください。

HTTP ユーザーの認証**:**

HTTP 基本認証または SSL クライアント証明書認証を使用して、HTTP ユーザーを認証できます。

このタスクについて

認証スキームは、TCPIPSERVICE 定義の AUTHENTICATE 属性および SSL 属性によって指定されます。

認証スキーム	**AUTHENTICATE**	**SSL**	注
認証を使用しない HTTP	NO	NO、YES、CLIENTAUTH、または ATTLSAWARE	
基本認証を使用する HTTP	BASIC	NO、YES、CLIENTAUTH、または ATTLSAWARE	

認証スキーム	**AUTHENTICATE**	**SSL**	注
基本認証を使用する HTTP	AUTOMATIC	NO、YES、CLIENTAUTH、または ATTLSAWARE	SSL(CLIENTAUTH | ATTLSAWARE) が指定されており、クライアントが証明書を送信する場合、SSL クライアント証明書認証が使用されます。
SSL クライアント証明書認証を使用する HTTP	CERTIFICATE または AUTOREGISTER	CLIENTAUTH または ATTLSAWARE	クライアントが証明書を送信しない場合、接続は確立されません。
SSL クライアント証明書認証を使用する HTTP	AUTOMATIC	CLIENTAUTH または ATTLSAWARE	クライアントが証明書を送信しない場合、基本認証が使用されます。

ECI ユーザーの認証**:**

ECI プロトコルの場合、基本認証を使用してユーザーを認証できます。

ECI クライアントの TCPIPSERVICE 定義で ATTACHSEC(VERIFY) を指定します。ユーザーを認証しない場合は、ATTACHSEC(LOCAL) を指定します。

IPIC ユーザーの認証**:**

基本認証または SSL クライアント証明書認証を使用して、IPIC ユーザーを認証できます。

IPIC 接続の場合、基本ユーザー認証を使用できます。IPCONN リソース定義で USERAUTH(VERIFY) を指定します。ユーザーを認証しない場合は、USERAUTH(LOCAL) を指定します。

セキュリティー・プロトコルのサポート

CICS は、Secure Sockets Layer および Transport Layer Security のプロトコルをサポートします。

具体的には、CICS は、TLS 1.0、TLS 1.1、および TLS 1.2 をサポートします。これらのプロトコルの詳細については、該当する RFC を参照してください。

TLS 1.0: RFC 2246

TLS 1.1: RFC 4346

TLS 1.2: RFC 5246

注**:** いずれかのプロトコルに関する特定のポイントが必要である場合を除き、用語 SSL は、文書内では Secure Sockets Layer と Transport Layer Security の両方のプロトコルを指すのに使用されます。

これらのセキュリティー・プロトコルの主な機能は、次のとおりです。

プライバシー
: クライアントとサーバー間で交換されるデータは暗号化されます。詳細については、『SSL 暗号化』を参照してください。

保全性
: SSL プロトコルを使用して送信されるデータは、**Message Authentication Code** (MAC) によって改ざんから保護されます。 MAC は、セキュア・ハッシュ・アルゴリズムを使用してデータ内容から計算され、そのデータと一緒に送信されます。 MAC は、受信側によって再び計算され、送信側によって送信された値と比較されます。 MAC の 2 つの値での不一致は、データが改ざんされた可能性があることを示します。

認証
: SSL は、デジタル証明書を使用してサーバーからクライアントへの認証を行い、オプションでクライアントからサーバーへの認証を行います。詳細については、『 370 ページの『SSL 認証』』を参照してください。

SSL 暗号化

SSL プロトコルは、アプリケーション層と TCP/IP 層の間で機能します。これによりデータ・ストリーム自体の暗号化が可能になり、任意のアプリケーション層プロトコルを使ってそれを安全に伝送することができます。

データの暗号化およびメッセージ認証コードの計算に使用できるアルゴリズムは多種あります。アルゴリズムによっては、ハイレベルのセキュリティーを提供するが、暗号化と暗号化解除のために大量の計算が必要になるものがあります。また、セキュリティーはそれより劣るものの、暗号化と暗号化解除を短時間で行えるものもあります。暗号化に使用される鍵の長さは、セキュリティーのレベルに影響を及ぼします。鍵が長いほど、データのセキュリティーが向上します。SSL は、SSL 接続中に使用される暗号アルゴリズムを指定するために暗号スイートを定義します。

SSL 暗号化技法

SSL では、2 つの暗号化技法が使用されます。

- Public Key Cryptography Standard (PKCS) は、SSL ハンドシェーク中に証明書を暗号化および暗号化解除します。暗号鍵は、公開鍵とそれに関連する秘密鍵というペアで作成されます。特定の公開鍵で暗号化されたデータは、それに関連する秘密鍵だけを使って暗号化解除できます。つまり、意図された受信者だけがデータを読み取ることができます。特定の秘密鍵で暗号化されたデータは、それに関連する公開鍵だけを使って暗号化解除できます。つまり、認証データの発信元は秘密鍵の所有者であることが確証されます。
- DES (Data Encryption Standard)、Triple-DES などの相互合意による対称暗号化技法は、ハンドシェーク後のデータ転送で使用されます。

SSL で使用される PKCS は、要約すると次のように機能します。

1. 証明書が作成されるとき、2 つの乱数に基づくアルゴリズムを使って、証明書の所有者の秘密鍵と公開鍵が作成されます。結果として作成される秘密鍵と公開鍵は、次のように互いに関連しています。
   - 公開鍵から秘密鍵の値を推測したり、秘密鍵から公開鍵を推測することはできません

秘密鍵は安全に保管されます。所有者以外に知らされることはありません。公開鍵は、任意のユーザーが自由に入手できます。その際、秘密鍵の暗号が漏えいするセキュリティー・リスクはありません。

- 公開鍵を使って暗号化された情報は、秘密鍵だけを使って暗号化解除できます

  任意のユーザーが情報を暗号化して、それを秘密鍵の所有者に安全に送ることができます。第三者は公開鍵を使って情報を読み取ることができません。
- 秘密鍵を使って暗号化された情報は、公開鍵だけを使って暗号化解除できます

  公開鍵を使って暗号化解除される情報を暗号化できるのは、秘密鍵の所有者だけです。第三者が情報の送信者を偽装することはできません。

SSL 認証

環境をセキュアにするために、通信はすべて、身元を確認できるトラステッド・サイトと行う必要があります。 SSL は、認証に証明書を使用します。これらは、デジタル署名された文書であり、公開鍵を秘密鍵所有者の ID にバインドします。

認証は接続時に行われ、アプリケーションやアプリケーション・プロトコルから独立しています。認証の際には、通信相手のサイトの身元が本当かどうかの確認が行われます。 SSL では証明書 (ITU-T 規格 X.509 で記述された形式のデータ・ブロック) を交換することによって認証が行われます。 X.509 証明書は認証局という外部組織によって発行され、デジタル署名されます。

証明書には、以下の情報が入っています。

- 2 つの識別名。これらは発行者 (証明書を発行した認証局) およびサブジェクト (証明書の発行の対象となった個人または組織) を一意的に識別します。識別名には、次のようないくつかのオプションのコンポーネントが含まれます。
  - 共通名
  - 組織単位
  - 組織
  - 局所性
  - 都道府県
  - 国
- デジタル署名。署名は、次の公開鍵暗号化技法を使用して、認証局によって作成されます。
  1. 安全なハッシュ・アルゴリズムを使用して、証明書の内容のダイジェストを作成します。
  2. ダイジェストは、認証局の秘密鍵を使用して暗号化されます。

  次に示すように、デジタル署名は、証明書の発行後にその内容がまったく変更されていないことを受信者に確証します。
  1. 認証局の公開鍵を使って署名が暗号化解除されます。
  2. 証明書の内容の新しいダイジェストが作成され、暗号化解除された署名と比較されます。何らかの不一致がある場合、証明書が改変された可能性があることを示唆しています。

- サブジェクトのドメイン・ネーム。受信者は、これを証明書の実際の送信者と比較します。
- サブジェクトの公開鍵。

証明書を使用して、クライアントがサーバーに対して、またサーバーがクライアントに対して認証されます。どちらの場合も基本的に同じ仕組みが使用されます。ただし、サーバー証明書は必須です (つまりサーバーは自分の証明書をクライアントに送る必要があります) が、クライアント証明書はオプションです。クライアント証明書をサポートしないクライアントや、証明書がインストールされていないクライアントもあります。サーバーは、接続のためにクライアント認証が必要かどうかを決定できます。

認証局

あるシステムが、別のシステムから受け取った証明書が本物であることを確信できるようにするために、証明書を保証できる信頼のおける第三者機関が必要です。

認証局とは、他者が使用するための証明書を発行することにより、信頼のおける第三者機関としての役割を果たす独立した実体です。証明書を発行する前に、認証局は、証明書を要求した個人または組織の資格情報を検査します。証明書が発行されると、その情報が公的にアクセス可能なリポジトリーに保持されます。ユーザーはリポジトリーを調べて、受信した証明書のステータスと妥当性を検査できます。

認証局は、各種の目的に応じたさまざまなレベルのセキュリティー証明書を発行します。以下に例を示します。

- 電子メールの保護
- クライアント認証
- サーバー認証

CICS は、クライアントから受信するすべての証明書を調べ、証明書失効リストを使用して失効ステータスかどうかを確認できます。証明書失効リストには、特定の認証局の失効したすべての証明書に関する詳細が含まれています。このリストはインターネットから無料でダウンロードできます。証明書が失効ステータスである場合、CICS は即時に SSL 接続を閉じます。証明書失効リストをセットアップする方法については、 394 ページの『証明書失効リスト (CRL) の使用』を参照してください。

暗号スイートおよび暗号スイート仕様ファイル

データの暗号化およびメッセージ認証コードの計算に使用できるアルゴリズムは多種あります。ユーザーがそれぞれのニーズを満たすセキュリティーのレベルを選択できるようにするために、また、ニーズが異なる可能性のある他のユーザーと通信できるようにするために、SSL は暗号スイート、つまり暗号のセットを定義します。 SSL 接続時に使用される暗号スイートのリストを SSL 暗号スイート仕様ファイルで指定できます。

暗号スイート

SSL 接続を確立するときには、SSL ハンドシェークで、クライアントとサーバーが、共通して所有する TLS プロトコルと暗号スイートに関する情報を交換します。その後、最高レベルのセキュリティーを提供するプロトコルおよび共通の暗号スイ

ートを使用して通信します。共通のプロトコルまたは暗号スイートがない場合、セキュアな接続はできないので、CICS は接続を閉じます。

使用可能な暗号スイートは、システム初期設定パラメーター **MINTLSLEVEL** と **NISTSP800131A**、および z/OS® System SSL によってサポートされる暗号によって異なります。また、該当するリソース定義の CIPHERS 属性の暗号スイート・リストを編集するか、またはリソース定義の SSL 暗号スイート仕様ファイルを編集することで、使用される暗号を制限できます。

各 CICS 領域からの SSL インバウンド接続で、どの暗号スイートが選択されているかを確認できます。 DFHSOCK グループ内のパフォーマンス・データ・フィールド SOCIPHER (320) には、各 SSL インバウンド接続に使用された暗号スイートのコードが表示されます。この情報は、CICS 領域によって提供されているが、SSL 接続用に選択されていない暗号スイートを特定するために使用します。また、SSL 接続用に選択されている効果の少ない暗号スイートまたはセキュリティーの低い暗号スイートを特定することもできます。さらに、このような暗号スイートを除去するかどうか決定できます。

必要な暗号化レベルを指定するには、次のようにします。

インバウンド **HTTP** の場合
: TCPIPSERVICE リソース定義の CIPHERS 属性を使用します。

アウトバウンド **HTTP** および **Web** サービス要求の場合
: URIMAP リソース定義の CIPHERS 属性を使用します。

インバウンド **IPIC** の場合
: TCPIPSERVICE リソース定義の CIPHERS 属性を使用します。

アウトバウンド **IPIC** の場合
: IPCONN リソース定義の CIPHERS 属性を使用します。

インバウンド **CICSPlex SM Web** ユーザー・インターフェース要求の場合
: **TCPIPSSLCIPHERS** Web ユーザー・インターフェース・サーバー初期設定パラメーターを使用します。この値の構文は、TCPIPSERVICE リソースの CIPHERS 属性の構文と同じですが、最大 22 の暗号コードに制限されます。

サポートされる各セキュリティー・プロトコルに対して z/OS および CICS でサポートされている暗号スイートについては、『z/OS Cryptographic Services System SSL Programming』の『Cipher Suite Definitions』で説明されています。

SSL 暗号スイート仕様ファイル

SSL 暗号スイート仕様ファイルは、SSL 接続で使用できる暗号スイートのリストが含まれる XML ファイルです。

ファイル名の長さは、拡張子 (`.xml` でなければなりません) を含めて最大 28 文字です。指定される値は、UNIX ファイルに有効な名前でなければならず、使用できる文字は `A`〜`Z`、`a`〜`z`、`0`〜`9`、`#`、`-`、`.`、`@`、`_` のみです。大/小文字の区別があります。

SSL 暗号スイート仕様ファイルは *ussconfig*/security/ciphers ディレクトリーになければなりません (ここで、*ussconfig* は SIT パラメーター **USSCONFIG** の値です)。

CICS 領域は、z/OS UNIX へのアクセス権、このファイルが含まれるディレクトリーへの読み取り/実行権限、およびこのファイル自体への読み取り権限を持っている必要があります。

サンプル・ファイルが *usshome*/security/ciphers ディレクトリーで提供されています (ここで、*usshome* は SIT パラメーター **USSHOME** の名前です)。また、スキーマ・ファイルが *usshome*/schemas/security ディレクトリーで提供されています。ファイル名は ciphersfile.xsd です。

SSL 暗号スイート仕様ファイルの構造

それぞれの暗号スイートは、 **cipher** エレメントの **number** 属性として指定されます。暗号番号は 4 文字のコードです。2 文字のコードを使用する場合は、先行ゼロを埋め込みます。

サンプル・ファイルには各暗号のコメントも含まれており、そのコメントには暗号スイートについて説明するテキスト・ストリングが含まれています。ただし、CICS はこのエレメントを検証することも、このエレメントに対して何らかのアクションを取ることもありません。

以下の例は、暗号ファイルの構造を示しています。

```
<?xml version="1.0"?>
<cipher_list xmlns="http://www.ibm.com/software/htp/cics/ciphers">
    <cipher number="000A">
        <!-- SSL_RSA_WITH_3DES_EDE_CBC_SHA -->
    </cipher>
    <cipher number="000D">
        <!-- SSL_DH_DSS_WITH_3DES_EDE_CBC_SHA -->
    </cipher>
    ...
</cipher_list>
```

SSL ハンドシェーク

SSL ハンドシェーク とは、接続が確立されたときにクライアントとサーバーの間で行われる情報の交換です。ハンドシェーク中に、クライアントとサーバーは、使用する暗号化アルゴリズムをネゴシエーションし、相互に認証します。

SSL ハンドシェークの主な機能は、次のとおりです。

- クライアントとサーバーは、両方がサポートする SSL バージョン番号および暗号スイートに関する情報を交換します。
- サーバーは、その証明書とその他の情報をクライアントに送信します。情報の一部は、サーバーの秘密鍵を使用して暗号化されます。クライアントがサーバーの公開鍵を使用して情報を正常に暗号化解除できる場合は、サーバーの ID が保証されます。
- クライアント認証が必要な場合、クライアントはその証明書とその他の情報をサーバーに送信します。情報の一部は、クライアントの秘密鍵を使用して暗号化されます。サーバーがクライアントの公開鍵を使用して情報を正常に暗号化解除できる場合は、クライアントの ID が保証されます。

- クライアントとサーバーは、それぞれが生成するランダムな情報を交換し、それはセッション鍵の確立に使用されます。このセッション鍵は、SSL セッション中に情報の暗号化と暗号化解除に使用される対称鍵です。その鍵は、データの保全性を検証するためにも使用されます。

SSL キャッシュ

SSL キャッシュは、クライアントと CICS の間の SSL セッションのセッション ID を保管するために使用されます。これらのセッション ID を再利用すると、CICS は、以前に認証したクライアントとの部分的ハンドシェークを実行できます。

ローカル CICS 領域では、デフォルトで、SSL キャッシュは S8 TCB のエンクレーブに保管されます。SSL キャッシュは、このエンクレーブ内に存在する SSL 環境の一部として z/OS System SSL によって管理されます。PERFORM SSL REBUILD コマンドを CICS 領域に対して発行すると、新しいキャッシュが作成されます。新しいキャッシュには、CICS 領域で確立された新規 SSL セッションによってデータが取り込まれます。古いキャッシュは、それを使用する最後の接続がドロップされると、削除されます。

SSL キャッシュをシスプレックス全体で共用するオプションがあります (ご使用の CICS システムに適している場合)。SSL 接続を同じ IP アドレスで受け入れる複数のソケット専有の CICS 領域がある場合は、SYSPLEX のキャッシングを使用できます。

領域間でキャッシュを共用する場合、z/OS System SSL 開始タスク GSKSRVR をアクティブにし、CICS 領域のシステム初期設定パラメーター **SSLCACHE=SYSPLEX** を指定します。このシステム初期設定パラメーターを指定すると、CICS は SSL キャッシュをカップリング・ファシリティーで使用してセッション ID を保管します。PERFORM SSL REBUILD コマンドを CICS 領域に対して発行する場合、シスプレックス・レベルで保持されたキャッシュは影響を受けません。

SSL プール

CICS は、DFHDDAPX XPI インターフェースを使用して LDAP への SSL 接続および要求を管理するために、オープン・トランザクション環境 (OTE) を使用します。

CICS で SSL 接続の数およびパフォーマンスを改善するために、各 SSL 接続は SSL プールから S8 TCB を使用します。SSL 接続のすべての CICS 処理は S8 TCB で行われます。Web サーバーの HTTP 接続タスク (デフォルトでは CWXN) は、すべてのデータが送信または受信されるまで、S8 TCB 上に留まります。送信または受信されているメッセージが非常に大きい場合、タスクはランナウェイ制限に達して終了する可能性があります。タスクの異常終了を避けるために、Web サーバー HTTP 接続トランザクション (デフォルトでは CWXN)、Web サーバー別名トランザクション、および Web クライアント API コマンドの SEND、RECEIVE、CONVERSE を発行するすべてのトランザクションのトランザクション **RUNAWAY** 値を大きくすることが必要な場合があります。

S8 TCB は SSL プールに格納され、CICS ディスパッチャーによって管理されます。S8 TCB は新しい SSL プールから割り振られますが、SSL 機能または LDAP 機能を実行するのに必要な期間は、トランザクションにロックされます。SSL 要求

または LDAP 要求が完了した後、TCB は解放され、再利用のために SSL プールに戻されます。**MAXSSLTCBS** システム初期設定パラメーターは、SSL プール内の S8 のオープン TCB の最大数を指定します。デフォルト値は 8 ですが、最大 1024 を指定できます。

DFH0STAT および DFHSTUP からのディスパッチャー・レポートを使用して、SSL プールおよび S8 TCB のパフォーマンスをモニターできます。統計には、S8 TCB の最大数に到達する頻度、TCB が割り振られる前の遅延、および SSL プール内の TCB の実際の数に関する情報が含まれます。

SSL を使用するための CICS の構成

CICS は Secure Sockets Layer (SSL) またはトランスポート層セキュリティー (TLS) セキュリティー・プロトコルを使用してセキュアな TCP/IP 接続をサポートすることができます。クライアントに対してサーバーを認証するには、RACF で証明書および鍵リングを作成し、セキュリティーをサポートするように CICS 領域およびリソースが正しく構成されていることを確認します。

始める前に

CICS の構成を開始する前に、SSL ハンドシェークで使用する証明書のタイプを決定します。

このタスクについて

RACF を使用して証明書を作成することはできますが、クライアントが RACF サーバー証明書を認識できるようにクライアントを構成する必要があります。クライアントをこのように構成できない場合 (クライアントが組織の外部にある場合など) は、外部の認証局によって署名された証明書を使用します。

手順

1. 鍵リングの作成、署名証明書 (認証局証明書) の作成、鍵リングへの証明書の追加を行うための正しい許可を RACF で設定します。
2. オプション: 認証局からの証明書を使用することを決定した場合、RACF を使用して認証要求を作成し、それを認証局に送信します。 認証局から署名証明書を受け取るまでに、数日間待機しなければならない場合があります。選択した認証局に、RACF に組み込まれた証明書がない場合は、インポートしなければならないことがあります。
3. 鍵リングを作成します。 RACF データベースに鍵リングを作成する必要があります。鍵リングには、以下のものが含まれます。
   - 公開鍵と秘密鍵
   - サーバー証明書
   - サーバー証明書の署名証明書
   - クライアント証明書が有効な RACF ユーザー ID に関連付けられていない場合、鍵リングに、クライアント認証を使用して CICS が通信することが予測されるクライアントが所有するクライアント証明書の署名証明書を追加する必要があります。
4. 証明書を作成して、鍵リングに追加します。

5. CICS 領域が z/OS システムの SSL ライブラリー SIEALNKE にアクセスできることを確認します。 STEPLIB ステートメントまたは JOBLIB ステートメントを使用するか、システム・リンク・ライブラリーを使用できます。
6. セキュリティーに関連する CICS システム初期設定パラメーターを定義します。 特に、**KEYRING** システム初期設定パラメーターで作成した鍵リングの名前を指定します。
7. TCPIPSERVICE リソースを定義します。 また、**CIPHERS** 属性を使用して SSL 接続のセキュリティーのレベルを指定することもできます。この属性は、暗号スイート・コードのリストとして解釈されるストリングまたは暗号スイート仕様ファイルの名前で指定できます。

例

CICS はサンプル REXX プログラムを提供します。これには、鍵リングの作成、署名証明書の作成、追加の証明書の作成、および鍵リングへの証明書の追加を行うためのすべての RACF コマンドが含まれています。DFH$RING には、テスト鍵リングの作成に適したサンプル値のみ含まれています。実稼働環境に適した鍵リングを作成する場合は、すべての値を編集する必要があります。

RACF でのプロファイルのセットアップ

CICS 領域での使用に適した RACF データベースで RACF 鍵リングを作成するには、FACILITY クラスの適切なプロファイルへのアクセス権限を付与する必要があります。

このタスクについて

このアクセス権限は、CICS システムを管理するユーザーにのみ付与する必要があり、一般の CICS ユーザーに付与してはなりません。使用可能なプロファイルは以下のとおりです。

CONTROL

- IRR.DIGTCERT.GENCERT (証明書が CERTAUTH 証明書によって署名されることを許可する)
- IRR.DIGTCERT.ADD (初回実行時) (CERTAUTH 証明書が生成されることを許可する)
- IRR.DIGTCERT.CONNECT (他のユーザーの CERTAUTH 証明書に接続する)

UPDATE

- IRR.DIGTCERT.CONNECT (CERTAUTH 証明書をユーザー自身の鍵リングに接続する)
- IRR.DIGTCERT.* (他のユーザーの証明書を管理する)

READ

IRR.DIGTCERT.* (ユーザー自身のユーザー ID の証明書を管理する)

IRR.DIGTCERT.* にはワイルドカードのアスタリスクが含まれ、総称プロファイルとして使用されます。総称プロファイルを FACILITY クラスで作成できるようにするには、以下のようにします。

手順

1. **SETROPTS GENERIC(FACILITY)** コマンドを発行します。
2. 以下のコマンドを発行します。

```
RDEFINE FACILITY(IRR.DIGTCERT.*)
RDEFINE FACILITY(IRR.DIGTCERT.ADD)
RDEFINE FACILITY(IRR.DIGTCERT.CONNECT)
RDEFINE FACILITY(IRR.DIGTCERT.GENCERT)
```

3. FACILITY クラスが RACLISTed かどうかに応じて、以下のいずれかのコマンドを発行します。

```
SETROPTS RACLIST(FACILITY) REFRESH
SETROPTS GENERIC(FACILITY) REFRESH
```

4. ユーザー IDまたはグループ *ringuser* に、DFH$RING に含まれるコマンドの使用を許可するには、以下のコマンドを発行します。

```
PERMIT IRR.DIGTCERT.*       CLASS(FACILITY) ID(ringuser) ACCESS(READ)
PERMIT IRR.DIGTCERT.CONNECT CLASS(FACILITY) ID(ringuser) ACCESS(UPDATE)  (for self)
PERMIT IRR.DIGTCERT.CONNECT CLASS(FACILITY) ID(ringuser) ACCESS(CONTROL)  (for another user)
PERMIT IRR.DIGTCERT.GENCERT CLASS(FACILITY) ID(ringuser) ACCESS(CONTROL)
```

   DFH$RING は CICS で提供されるサンプルで、適切な鍵リングをセットアップするのに役立ちます。

5. DFH$RING の最初のユーザーに、認証局証明書を作成するための *certauser* 権限を与える必要があります。

```
PERMIT IRR.DIGTCERT.ADD CLASS(FACILITY) ID(certauser) ACCESS(CONTROL)
```

   この証明書は、DFH$RING によって作成された他のすべての証明書に署名するために使用されます。

タスクの結果

FACILITY クラス内の IRR.DIGTCERT.ADD プロファイルに対する READ 権限を持っている場合は、ユーザー自身のユーザー ID の証明書情報を追加できます。FACILITY クラス内の IRR.DIGTCERT.ADD プロファイルに対する UPDATE 権限を持っている場合は、他のユーザー ID の証明書情報を追加できます。RACF SPECIAL 権限を持っている場合は、任意のユーザー ID に対して RACDCERT ADD を実行できます。また、RACF 定義ユーザーのデジタル証明書または SPECIAL 権限を持つ認証局証明書あるいはサイト証明書のデジタル証明書を生成することもできます。

認証局からの証明書の要求

RACF を使用して、Verisign などの認証局から署名証明書 (認証局証明書) を要求できます。RACF 証明書を認識できないクライアントに対してサーバーを認証するには、外部証明書を使用します。

始める前に

RACDCERT コマンドを使用するための許可が必要です。このコマンドは、RACF でデジタル証明書、鍵リング、およびデジタル証明書マッピングをインストールして維持します。

このタスクについて

RACF は、さまざまな認証局の証明書を提供するため、自分で定義する必要はありません。これらの証明書は、『z/OS Security Server RACF Security Administrator's Guide』の『提供されているデジタル証明書』にリストされています。

手順

1. RACF で自己署名証明書をプレースホルダーとして作成します。

```
RACDCERT ID(foruser) GENCERT,
 SUBJECTSDN(CN('username')
           T ('username''s certificate')
           OU('department')
           O ('organization')
           L ('city')
           SP('state')
           C ('country'))
 NOTBEFORE(DATE(start) TIME(00:00:00))
 NOTAFTER (DATE(finish) TIME(23:59:59))
 WITHLABEL(self-signed-certlabel)
 SIZE     (1024)
```

2. 外部の認証局に送信するための認証要求をプレースホルダー証明書に基づいて生成します。**RACDCERT GENREQ** コマンドを使用します。

```
RACDCERT ID(cics-region-userid) GENREQ(LABEL('label'))
         DSN('request.dataset')
```

   ここで、*label* は、プレースホルダーの自己署名証明書です。 RACF は、**DSN** パラメーターで指定されたデータ・セットに認証要求を保管します。

3. 認証局が受け入れる方法を使用して、認証局に認証要求を送信します。
4. 証明書を受信したら、それを新規データ・セットに保管します。
5. オプション: デフォルトの認証局のいずれでもない認証局を使用しており、その証明書が鍵データベースに既に保管されている場合、その認証局の証明書を RACF データベースにインポートする必要があります。
6. 自己署名証明書を、CA が署名した新しい証明書に置き換えます。

```
RACDCERT ID(cics-region-userid) ADD('response.dataset') TRUST
```

次のタスク

RACF データベースに鍵リングを作成し、CA が署名した証明書を追加します。

手動での鍵リングの作成

CICS では、必要なサーバー証明書および認証局に関する関連情報が RACF データベースの鍵リング に保持されます。鍵リングには、システムの秘密鍵と公開鍵のペアが入るとともに、サーバー証明書も、クライアントから受け取った証明書に署名した可能性のあるすべての認証局の証明書も一緒に入ります。

始める前に

SSL を CICS で使用するには、その前に、秘密鍵と公開鍵のペアおよびサーバー証明書を含む鍵リングを作成する必要があります。鍵リングを作成するには、FACILITY クラスの IRR.DIGTCERT.ADDRING リソースに対する UPDATE 権限

が必要です。鍵リング内の証明書を CICS 領域間で共用する場合、CICS 領域は同じユーザー ID を持つ必要があり、ユーザー ID は鍵リングを所有している必要があります。

このタスクについて

RACDCERT コマンドは、公開鍵インフラストラクチャー (PKI) の秘密鍵と証明書を RACF にインストールして保持します。**RACDCERT** コマンドを手動で発行して新しい鍵リングを作成するか、または DFH$RING サンプル・プログラムを使用できます。DFH$RING を使用した証明書付きの鍵リングの作成を参照してください。

鍵リングを手動で作成するには、以下の手順を実行します。

手順

次の **RACDCERT** コマンドを発行します。

```
RACDCERT ID(cics-region-userid) ADDRING(ringname)
```

鍵リングを CICS 領域ユーザー ID に関連付ける必要があります。

タスクの結果

RACF は鍵リングを RACF データベースに作成します。同じ名前の鍵リングが既に RACF データベースに既に存在する場合、新しい鍵リングに置き換えられます。

次のタスク

署名証明書 (認証局証明書) を作成し、それを鍵リングに追加します。

DFH$RING を使用した証明書付きの鍵リングの作成

DFH$RING は、鍵リングの作成、署名証明書 (認証局証明書) の作成、追加の証明書の作成、鍵リングへの証明書の追加を行う、サンプルの REXX プログラムです。

始める前に

RACF コマンドを実行するには、必要な許可を持っていなければなりません。ユーザー ID には、プログラムの最初の実行時に署名証明書を作成するための CONTROL 権限が必要です。プログラムを再度実行する場合は、UPDATE 権限のみが必要です。

このタスクについて

DFH$RING はライブラリー CICSTS55.CICS.SDFHSAMP にあります。適切な鍵リングおよび証明書を作成するには、DFH$RING の値を編集します。

手順

1. *firstname*、*lastname*、および *hostname* の各変数の値を入力します。 *firstname* の値と *lastname* の値が連結されて、鍵リングの名前を構成します。*hostname* 変数に Web サーバーのホスト名を入力します。
2. オプション: 別のユーザー ID (CICS 領域ユーザー ID など) の鍵リングを作成している場合は、*FORUSER* 変数の値を入力します。

3. 署名証明書 (認証局証明書) がある場合、*certifier* 変数にラベルを入力します。
4. 署名証明書がない場合、**RACDCERT CERTAUTH GENCERT** コマンドの変数を適切な値に置き換えると、RACF がそれを自動作成します。

```
"RACDCERT CERTAUTH GENCERT",
" SUBJECTSDN(CN('CICS Sample Certification Authority' ) ",
            "OU('"department"'                          ) ",
            "O ('"organization"'                        ) ",
            "L ('"city"'                                ) ",
            "SP('"state"'                               ) ",
            "C ('"country"'                             ))",
" NOTBEFORE(DATE("start")  TIME(00:00:00)                )",
" NOTAFTER (DATE("finish") TIME(23:59:59)                )",
" WITHLABEL("certifier"                                  )",
" SIZE     (2048                                          )"
```

これらの値により、生成された証明書の識別名で適切なフィールドが定義されます。*country* 変数の国別コードは ISO 3166-1 コードでなければなりません。有効なコードのリストについては、国際標準化機構の国別コード - ISO 3166を参照してください。*start* と *finish* は、証明書の妥当性を判断します。*certifier* は、他の証明書に署名するために使用される、自己署名付きの認証局証明書のラベルです。**SIZE** パラメーターは、証明書に関連付けられている秘密鍵のサイズをビット単位で指定します。サイズが大きくなるほど、鍵のセキュリティーは高くなります。少なくとも 2048 をお勧めします。

DFH$RING は、署名証明書がまだ存在していない場合にのみ、それを作成します。

5. 適切な証明書を作成して鍵リングに追加するには、**RADCERT GENCERT** RACF コマンドの変数を編集します。 DFH$RING には、編集、追加、または削除できる 4 つの例があります。**SIGNWITH** パラメーターの *certifier* 変数が署名証明書のラベルと一致することを確認します。
6. 証明書に一致するように、**RACDCERT CONNECT** RACF コマンドのラベルを編集します。署名証明書は他のすべての証明書に署名するため、最初に鍵リングに追加されるようにしてください。
7. DFH$RING を実行して、以下のように鍵リングと証明書を作成します。

```
EXEC 'CICSTS55.CICS.SDFHSAMP(DFH$RING)' 'firstname lastname webservername  [ FORUSER(userid) ] '
```

ここで、*userid* は、CICS 領域ユーザー ID です。

タスクの結果

DFH$RING プログラムは、*userid* ユーザー ID によって所有される *firstname.lastname* という名前の鍵リングを作成します。その名前を持つ既存の鍵リングは置き換えられます。**FORUSER** パラメーターを省略する場合、鍵リングはプログラムの実行に使用されたユーザー ID によって所有されます。DFH$RING は必要に応じて署名証明書を作成し、それを鍵リングに追加した後、他の証明書を作成します。

例

デフォルト値を使用して DFH$RING を実行した場合、DFH$RING は以下のラベルが付いた証明書を作成します。

***lastname*-Web-Server**
この証明書は、TCPIPSERVICE の CERTIFICATE 属性で PROTOCOL(HTTP) とともに使用できます。証明書内の識別名には、*webservername* という共通名があります。これは、接続に関連付けられたホスト名と同じでなければなりません。Web ブラウザーは、通常、証明書内の共通名が受信元のサーバーのホスト名と一致することを確認します。

***lastname*-IP-CONNECTION**
この証明書は、IP 相互接続 (IPIC) に使用できます。これは、CICS 領域が IPIC を使用するために必要なリソース定義の CERTIFICATE 属性で使用できます。このサンプル証明書は、IPCONN が取得されたときに発生する SSL ハンドシェーク中に、クライアント証明書およびサーバー証明書として使用される CICS 領域用の証明書です。これは、クライアント証明書の場合は IPCONN 定義の CERTIFICATE 属性で使用でき、サーバー証明書の場合は TCPIPSERVICE 定義の CERTIFICATE 属性で PROTOCOL(IPIC) とともに使用できます。

***lastname*-2048-Certificate**
この証明書は、強度の高い証明書を必要とする CICS システムに使用できます。これは、TCPIPSERVICE、IPCONN、および URIMAP の各定義の CERTIFICATE 属性、および EXEC CICS WEB OPEN コマンドで使用できます。

***lastname*-Default-Certificate**
この証明書は鍵リングのデフォルト証明書としてマーク付けされており、CERTIFICATE 属性を指定していないすべての TCPIPSERVICE リソースに使用される証明書です。この証明書には、*webservername* という共通名も含まれます。

Verisign Class 1 Primary CA

Verisign Class 2 Primary CA

IBM World Registry CA
これらの証明書は、認証局によって署名済みの、受け取る可能性がある証明書を確認するために必要です。他の認証局によって署名されたクライアント証明書、または自分自身で作成した証明書を受け入れる予定の場合、**RACDCERT CONNECT** コマンドを使用して、それらの証明書を鍵リングに手動で追加する必要があります。このようにして証明書を鍵リングに追加する場合、USAGE(PERSONAL) を指定する必要があります。

次のタスク

証明書をさらに作成して、鍵リングに追加できます。

新規 RACF 証明書の作成

新規証明書を作成して鍵リングに追加するには、**RACDCERT** コマンドを使用します。

このタスクについて

鍵リング内の証明書は、CICS 領域ユーザー ID に関連付けられている必要があります。鍵リングは、それを使用している CICS 領域ユーザー ID によって所有されている必要があります。

注: 同じ **KEYRING** 上の同じ識別名を持つ複数の証明書はサポートされていません。

手順

1. CICS 領域ユーザー ID を指定して、証明書を作成します。 **RACDCERT GENCERT** コマンドを以下のように入力します。 変数の値を入力します。 *country* 変数の国別コードは、ISO 3166-1 コードでなければなりません。有効なコードのリストについては、国際標準化機構の国別コード - ISO 3166を参照してください。 *certifier* の値は、鍵リング内の署名証明書のラベルです。

```
RACDCERT ID(foruser) GENCERT
 SUBJECTSDN(CN('username')
           T ('username''s certificate')
           OU('department')
           O ('organization')
           L ('city')
           SP('state')
           C ('country'))
 NOTBEFORE(DATE(start) TIME(00:00:00))
 NOTAFTER (DATE(finish) TIME(23:59:59))
 SIGNWITH (CERTAUTH LABEL('certifier'))
 WITHLABEL('certlabel')
 SIZE     (1024)
```

2. RACDCERT CONNECT コマンドを使用して、鍵リングに証明書を追加します。
   a. 証明書を複数の CICS 領域間で共用する場合は、その CICS 領域の KEYRING システム初期設定パラメーターで指定されている鍵リングに証明書を追加して、USAGE(PERSONAL) を指定します。 その証明書には、同じ領域ユーザー ID を持っており、同じ鍵リングを使用しているすべての CICS 領域がアクセスできます。

      ```
      RACDCERT ID(foruser) CONNECT( RING(ringname) LABEL('label') USAGE('PERSONAL'))
      ```

   b. 証明書をデフォルトの証明書として鍵リングに追加する場合は、その CICS 領域の **KEYRING** システム初期設定パラメーターで指定されている鍵リングに証明書を追加して、DEFAULT を指定します。

      ```
      RACDCERT ID(foruser) CONNECT( RING(ringname) LABEL('label') DEFAULT)
      ```

      クライアントまたはサーバーが CICS から証明書を要求すると、特に別の指定をしていない限り、デフォルトの証明書が使用されます。インバウンド HTTP 要求の場合は、TCPIPSERVICE リソース内の証明書を指定します。
3. 証明書または鍵リングを更新する任意の RACDCERT コマンドの実行後に、DIGTCERT クラスおよび DIGTRING クラスを RACLIST する場合は、以下のコマンドを発行する必要があります。

   ```
   SETROPTS RACLIST(DIGTCERT DIGTRING) REFRESH
   ```

4. 鍵リング内の証明書に何らかの更新または追加を行った後には、CICS 領域に対する **PERFORM SSL REBUILD** コマンドを発行します。 このコマンドにより、CICS 領域の SSL 環境が再構築され、証明書のキャッシュが鍵リングからの新しい情報でリフレッシュされます。

RACF ユーザー ID と証明書との関連付け

クライアント証明書は、RACF ユーザー ID に関連付けられている場合にのみ、CICS トランザクションのユーザー ID を判別するために使用できます。

次の 2 つの方法で、証明書と RACF ユーザー ID を関連付けることができます。

- ユーザーは、Web ブラウザー・プログラムを介して自分の証明書をオンラインで登録できます。クライアントを使用可能にし、TCPIPSERVICE 定義で AUTHENTICATE(AUTOREGISTER) を指定してクライアントの証明書そのものを登録します。このような TCPIPSERVICE を介して CICS に接続するユーザーは、クライアント証明書を保有している必要があります。この証明書がすでにユーザー ID に対して登録されている場合は、そのユーザー ID が使用されます。登録されていない場合は、クライアントでは、HTTP 基本認証でユーザー ID とパスワードのプロンプトが出されます。クライアントで有効なユーザー ID とパスワードを入力すると、そのユーザー ID が証明書に登録され、クライアントでは、パスワードのプロンプトが再び出されることはありません。この規則の要約が 363 ページの『HTTP ユーザーの識別』に示されています。

  証明書がこの方法で登録されると、証明書をすべてのインバウンド TCP/IP 接続に使用できます。

- RACDCERT コマンドを使用できます。クライアント自体の証明書の登録をクライアントに許可しない場合は、RACDCERT コマンドを使用してクライアントの証明書を登録する必要があります。RACDCERT を実行する前に、TSO からアクセスできる RECFM=VB を使用して、MVS 順次ファイル内に処理する証明書をダウンロードする必要があります。RACDCERT の構文は、次のとおりです。

  ```
  RACDCERT ADD('datasetname') TRUST [ ID(userid) ]
  ```

  ここで、*datasetname* は、クライアント証明書を含むデータ・セットの名前であり、*userid* は、証明書に関連付けられるユーザー ID です。オプションの ID(userid) パラメーターが省略された場合、証明書は RACDCERT コマンドを実行するユーザーに関連付けられます。

FACILITY クラス内の IRR.DIGTCERT.ADD プロファイルに対する READ 権限を保有している場合は、ユーザー自身のユーザー ID の証明書情報を追加できます。FACILITY クラス内の IRR.DIGTCERT.ADD プロファイルに対する UPDATE 権限を保有している場合、または RACF SPECIAL 権限を保有している場合は、他のユーザー ID の証明書情報を追加できます。

ダウンロードした証明書データ・セット内で許可されるデータの形式を含め、RACDCERT コマンドについて詳しくは、z/OS Security Server RACF コマンド言語解説書を参照してください。

CICS 領域ユーザー ID によって所有されていない既存の証明書の使用

適切な RACF ファシリティーを使用することにより、単一の証明書を CICS システム間で共用できます。

このタスクについて

CERTIFICATE 属性を持つ CICS リソースおよび Web Services Security 用の CICS リソースの場合、使用される証明書は、デフォルトで CICS 領域ユーザー ID によって所有されている必要があります。CICS が所有していない証明書を CICS が使用する必要がある場合 (例えば、各システムが異なる領域ユーザー ID を持つ複数の CICS システムによって共用される単一の証明書の場合)、RACF Facility ク

ラス RDATALIB を使用して、複数の CICS システムが単一の証明書を共用することを許可できます。

手順

1. PERSONAL 使用オプションを指定して、証明書をその鍵リングに接続します。
2. 証明書が USER 証明書である場合、証明書を使用する CICS 領域ユーザー ID に、RDATALIB クラスの *ring_owner.ring_name*.LST リソースに対する UPDATE 権限を付与します。
3. **RACLIST** コマンドを使用して、RDATALIB クラスをアクティブ化します。

タスクの結果

CICS は、他のユーザー ID によって所有されている証明書を使用できます。詳しくは、z/OS Security Server RACF 呼び出し可能サービスを参照してください。

CICS TS で使用するための RACF サイト証明書の構成

CICS Transaction Server for z/OS 領域で SSL を有効にするが、各領域に別個の SSL 証明書を定義しない場合は、サイト証明書 を使用できます。

手順

378 ページの『手動での鍵リングの作成』または 379 ページの『DFH$RING を使用した証明書付きの鍵リングの作成』の説明に従って、鍵リングを作成します。
鍵リングは完全に構成する必要があります。つまり、TCPIPSERVICE RDO 定義が指す証明書だけでなく、その証明書の署名に使用されたすべての証明書が含まれている必要があります。これらの署名証明書は USAGE=CERTAUTH を指定した鍵リングに含まれている必要があります。
次の図は、次の RACF コマンドによってリストされた、完全に構成された鍵リングの例を示しています。 RACDCERT ID(*ring_owner*)
LISTRING(*ring_name*)

```
Ring:  ring_name

 Certificate Label Name          Cert Owner  USAGE     DEFAULT
--------------------------------- ----------- --------  -------
  Verisign Class 1 Primary CA     CERTAUTH    CERTAUTH  NO
  IBM World Registry CA           CERTAUTH    CERTAUTH  NO
  CICS-Sample-Certification       CERTAUTH    CERTAUTH  NO
  Verisign Class 2 Primary CA     CERTAUTH    CERTAUTH  NO
  SITECERT                        SITE        PERSONAL  YES
```

サイト証明書 (SITECERT) として使用する証明書は、SITE によって所有され、PERSONAL を使用できる必要があります。この証明書は、SSL 暗号化を必要とする TCPIPSERVICE 定義によって使用されるものです。
鍵リングは、CICS 領域ユーザー ID によって所有される必要があります。複数の CICS 領域が同じ領域ユーザー ID を使用する場合、同じ鍵リングを共用できます。異なる領域ユーザー ID で実行する場合は、別個の鍵リングを作成する必要があります。ただし、各リング内で同じサイト証明書を使用できます。
サイト証明書には秘密鍵が必要です。そうしないと、TCPIPSERVICE がインストールに失敗するか、それを使用しようとすると失敗します。
CICS 領域ユーザー ID には、FACILITY クラスのプロファイル IRR.DIGTCERT.GENCERT に対して CONTROL アクセス以上の権限が必要です。

詳しくは、z/OS Security Server RACF 呼び出し可能サービスの『RACF Callable Services Authorization』および『RACF Callable Services Usage Notes』のセクションを参照してください。

証明書を Untrusted としてマーク付けする

証明書が RACF データベースに登録されているが、それをクライアントに使用させないようにする場合は、RACDCERT コマンドを使用して、UNTRUSTED としてマーク付けすることができます。

手順

1. コマンド `RACDCERT ID(userid) LIST` を入力して、証明書に関連付けられているラベルを検出します。
2. コマンド `RACDCERT ID (userid) ALTER(LABEL(label)) NOTRUST` を使用して、証明書を Untrusted としてマーク付けします。
3. 実行中の CICS 領域が証明書を含む鍵リングを使用しているときに証明書を修正した場合、その CICS 領域に対して PERFORM SSL REBUILD コマンドを発行します。 このコマンドにより、CICS 領域の SSL 環境が再構築され、証明書のキャッシュが鍵リングからの新しい情報でリフレッシュされます。

   注: **PERFORM SSL REBUILD** コマンドは、CICS が SSL(ATTLSAWARE) で定義された TCPIPSERVICE を使用 (AT-TLS で保護されたクライアント接続を指示) している SSL/TLS 環境には適用されません。このような SSL 環境およびキャッシュをリフレッシュする場合は、Application Transparent Transport Layer Security (AT-TLS) の概要の説明に従ってください。

タスクの結果

クライアントは、この証明書を使用した CLIENTAUTH 接続を確立できなくなります。

SSL のシステム初期設定パラメーター

SSL に関連するシステム初期設定パラメーターの説明。

以下のシステム初期設定パラメーターは SSL と関連があります。

CRLPROFILE システム初期設定パラメーター
: LDAP サーバーに保管されている認証取り消しリストへのアクセスを CICS に許可するプロファイルの名前を指定します。認証取り消しリストおよびこのプロファイルのセットアップについて詳しくは、 394 ページの『CRL のための LDAP サーバーの構成』を参照してください。

KEYRING システム初期設定パラメーター
: CICS によって使用される鍵および証明書を含む RACF データベース内の鍵リングの名前を指定します。これは CICS 領域ユーザー ID によって所有される必要があります。最初の鍵リングは、DFH$RING exec を使用して CICSTS55.CICS.SDFHSAMP に作成できます。

MAXSSLTCBS システム初期設定パラメーター
: DFHDDAPX XPI インターフェースを使用して LDAP への Secure Sockets Layer 接続および要求を処理するために、CICS が使用可能な S8 TCB の最大

数を指定します。この値は 0 から 999 までの範囲の数値で、デフォルト値は 8 です。S8 TCB は SSL プールで作成および管理されます。 S8 TCB は、SSL 処理または LDAP 処理中のみ、タスクによって使用されます。

MINTLSLEVEL システム初期設定パラメーター
CICS がセキュア TCP/IP 接続で使用する最小 TLS プロトコルを指定します。

SSLCACHE システム初期設定パラメーター
CICS が CICS 領域に SSL セッションのローカル・キャッシュを使用するか、またはカップリング・ファシリティーを使用して複数の CICS 領域にわたってキャッシュを共用するかを指定します。シスプレックスにおけるキャッシングは、領域が同じ IP アドレスでの SSL 接続を受け入れる場合のみ行うことができます。キャッシュには、CICS が以前に認証したクライアントとの部分的ハンドシェークを実行できるようにするセッション ID が含まれています。
PERFORM SSL REBUILD コマンドを CICS 領域に対して発行すると、ローカル・キャッシュは置き換えられますが、シスプレックス・キャッシュは影響を受けません。

SSLDELAY システム初期設定パラメーター
CICS がローカル CICS 領域でセキュア・ソケット接続のセッション ID を保持する時間の長さを秒単位で指定します。セッション ID は、クライアントと SSL サーバーの間のセキュア接続を表すトークンです。SSLDELAY の期間中、セッション ID が CICS によって保持される間、SSL ハンドシェークによる重大なオーバーヘッドが生じることなく、CICS はクライアントと通信を続行できます。この値は 0 から 86400 の範囲の秒数です。デフォルトは 600 です。

SSL の TCPIPSERVICE 属性

SSL に関連する TCPIPSERVICE リソースの属性について説明します。

このタスクについて

TCPIPSERVICE リソースの以下の属性は、SSL と関連があります。

AUTHENTICATE
HTTP プロトコルのインバウンド TCP/IP 接続で使用する認証および識別スキームを指定します。 HTTP プロトコルは、以下の認証スキームをサポートします。

NO クライアントは認証または識別情報を送信する必要がありません。

BASIC
HTTP 基本認証を使用して、クライアントからユーザー ID およびパスワードが取得されます。

CERTIFICATE
SSL クライアント証明書認証を使用して、クライアントが認証および識別されます。

AUTOREGISTER
SSL クライアント証明書認証を使用して、クライアントが認証されます。クライアントがセキュリティー・マネージャーに登録されていない有効な証明書を送信すると、CICS はその証明書を登録します。

AUTOMATIC
クライアントが証明書を送信すると、SSL クライアント証明書認証を使用して、クライアントが認証されます。クライアントがセキュリティー・マネージャーに登録されていない有効な証明書を送信すると、CICS はその証明書を登録します。クライアントが証明書を送信しない場合、HTTP 基本認証を使用して、クライアントからユーザー ID およびパスワードを入手します。

CERTIFICATE
SSL ハンドシェーク中に使用するサーバー証明書のラベルを指定します。この属性が省略された場合は、CICS 領域ユーザー ID の鍵リングに定義されているデフォルトの証明書が使用されます。

CIPHERS

CIPHERS 属性は、以下の 2 とおりの方法のいずれかで指定できます。

- 最大 28 個の 2 桁の暗号スイート・コードを示すリストとして解釈される 56 桁までの 16 進数字で構成されるストリング。
- SSL 暗号スイート仕様ファイルの名前。この z/OS UNIX ファイルは、**USSCONFIG** システム初期設定パラメーターによって指定されるディレクトリーの `security/ciphers` サブディレクトリー内にあります。例えば、**USSCONFIG** が `/var/cicsts/dfhconfig` に設定され、**CIPHERS** が `strongciphers.xml` に設定されている場合、完全修飾ファイル名は `/var/cicsts/dfhconfig/security/ciphers/strongciphers.xml` です。詳しくは、SSL 暗号スイート仕様ファイルの作成を参照してください。

CEDA トランザクションを使用してリソースを定義すると、CICS は、デフォルトの許容コード・リストに従ってその属性を自動的に初期化します。この属性を CICS で初期化できるようにするためには、CEDA を実行する CICS 領域で KEYRING システム初期設定パラメーターを指定する必要があります。KEYRING を設定しなかった場合、CICS はその属性を初期化しません。デフォルトのコード・リストは `35363738392F303132330A1613100D15120F0C` ですが、システム初期設定パラメーターで **NISTSP800131A=CHECK** が設定されている場合は `35363738392F303132330A1613100D` です。

暗号コードを再配列したり、初期リストから削除したりできます。ただし、指定された暗号化レベルのデフォルトのリストに存在しない暗号コードを追加することはできません。コードのデフォルト・リストに値をリセットするには、暗号スイート・コードをすべて削除します。このフィールドはデフォルト・リストから自動的に再生成されます。

詳しくは、暗号スイートおよび暗号スイート仕様ファイルを参照してください。

PORTNUMBER
CICS が着信クライアント要求を listen するポートの番号を指定します。 CICS がサポートする SSL サービス用のウェルノウン・ポートは、SSL を使用した HTTP の場合は 443 です。

SSL
TCP/IP サービスで暗号化と認証のために SSL を使用するかどうかを指定します。

NO SSL は使用されません。

YES SSL セッションが使用されます。CICS は、クライアントにサーバー証明書を送信します。

CLIENTAUTH
SSL セッションを使用します。CICS はサーバー証明書をクライアントに送信し、クライアントはクライアント証明書を CICS に送信する必要があります。

ATTLSAWARE
CICS は、クライアント接続を照会して、AT-TLS がアクティブであるかどうかを判別します。 CICS は、パートナーによって提供されている場合は、TCP/IP からクライアント証明書を取得します。

注**:** SSL(ATTLSAWARE) を指定する場合は、PROTOCL(HTTP) も指定する必要があります。

SSL 暗号スイート仕様ファイルの作成

SSL によって使用される暗号スイートのリストを指定するために、SSL 暗号スイート仕様ファイルを作成できます。SSL が TCP/IP 接続に使用される場合、TCP/IP 接続を定義するリソースの **CIPHERS** 属性で暗号スイート仕様ファイルの名前を指定できます。

手順

1. SSL 暗号スイート仕様ファイルを作成するには、サンプル仕様ファイルを編集するか、独自の仕様ファイルを作成します。
   - サンプル SSL 暗号スイート仕様ファイルを変更するには、*usshome*/security/ciphers ディレクトリーにあるサンプル・ファイルの 1 つを *ussconfig*/security/ciphers ディレクトリーにコピーします。ここで、それぞれの値は以下のとおりです。

     usshome
     SIT パラメーター **USSHOME** の値です。

     ussconfig
     SIT パラメーター **USSCONFIG** の値です。

     注**:** SSL 暗号スイート仕様ファイルは *ussconfig*/security/ciphers ディレクトリーになければなりません。
   - 独自の SSL 暗号スイート仕様ファイルを作成するには、*ussconfig*/security/ciphers ディレクトリーに XML ファイルを作成し、次の規則に従って、ファイルに名前を付けます。
     - ファイル名の長さは、.xml 拡張子を含めて、最大 28 文字です。
     - ファイル名は UNIX ファイルに有効な名前でなければならず、使用できる文字は A〜Z、a〜z、0-9、#、-、.、@、_ のみです。大/小文字の区別があります。
2. 暗号スイートおよび暗号スイート仕様ファイルの説明に従って、暗号スイートのリストを仕様ファイルに指定します。 サンプル・ファイルを編集する場合、セキュリティー要件を満たさない不要な暗号スイート、またはハードウェアによっ

てサポートされない不要な暗号スイートを削除できます。暗号スイートを追加することもできますが、CICS および z/OS によってサポートされる暗号スイートに限ります。

3. SSL 接続でファイルを有効にするには、CICS 領域が、z/OS UNIX へのアクセス権、この仕様ファイルが含まれるディレクトリーへの読み取り/実行権限、およびこのファイル自体への読み取り権限を持っている必要があります。

タスクの結果

暗号スイート仕様ファイルが作成されました。SSL 暗号スイート・ファイルは複数のリソースが使用できます。仕様ファイルを使用するリソースが初めてインストールされるときに、ファイルが zFS から読み取られ、解析されます。この解析中に、エラーがあればフラグが立てられます。ファイルが有効である場合、リソースがインストールされ、そのファイルに関連付けられている新しい制御ブロックに暗号情報が保管されます。同じ暗号ファイルを使用する後続のリソースがインストールされると、制御ブロックにキャッシュされた情報が使用されます。

次のタスク

暗号スイート仕様ファイルの暗号スイートのリストを更新する場合、ファイルを直接編集できますが、更新されたリストを有効にするには、CICS を再始動する必要があります。**START** システム初期設定パラメーターが INITIAL、COLD、AUTO のいずれに設定されている場合でも、開始のタイプに合わせてファイルの再読み取りが行われます。

CICS を再始動せずにリソース用の暗号スイートのリストを更新するには、新しい仕様ファイルを使用する必要があります。

1. 新しい暗号スイート仕様ファイルを作成します。ファイル名がこの CICS システムによってロードされていないことを確認してください。
2. 既存のリソース定義を、新規ファイルを参照するように更新します。例えば、CIPHERS(newciphers.xml) を指定して **CREATE TCPIPSERVICE** コマンドを発行します。
3. リソース定義を再インストールします。

暗号化ネゴシエーションのカスタマイズ

SSL 接続の暗号化ネゴシエーション・プロセスで使用される暗号スイートを選択して、最小レベルおよび最大レベルの暗号化を設定できます。

このタスクについて

リソース定義 TCPIPSERVICE、IPCONN、および URIMAP の中の CIPHERS 属性は、それぞれの暗号化レベルで使用可能な暗号スイートを指定します。この属性のデフォルト値は、暗号化折衝で使われる暗号スイートを示す 2 桁のコードのリストです。オプションで、この暗号スイート・リストをカスタマイズすることにより、CICS とクライアントとの折衝における独自の暗号化レベル優先順位を含めることができます。また、リストから暗号スイートを除去するよう選択することもできます。このオプションは、非常に高いレベルの暗号化のみを確実に使用したい場合に役立ちます。

コードのリストを直接編集するか、または CIPHERS 属性を、使用する暗号スイートが指定されている SSL 暗号スイート仕様ファイルの名前に設定することにより、リストをカスタマイズできます。SSL 暗号スイート仕様ファイルは、**USSCONFIG** システム初期設定パラメーターによって指定されたディレクトリーの security/ciphers サブディレクトリーにある z/OS UNIX ファイルです。詳細については、SSL 暗号スイート仕様ファイルの作成を参照してください。

各 CICS 領域からの SSL インバウンド接続で、どの暗号スイートが選択されているかを確認できます。 DFHSOCK グループ内のパフォーマンス・データ・フィールド SOCIPHER (320) には、各 SSL インバウンド接続に使用された暗号スイートのコードが表示されます。この情報は、CICS 領域によって提供されているが、SSL 接続用に選択されていない暗号スイートを特定するために使用します。また、SSL 接続用に選択されている効果の少ない暗号スイートまたはセキュリティーの低い暗号スイートを特定することもできます。さらに、このような暗号スイートを除去するかどうか決定できます。

CICS および z/OS でサポートされている暗号スイートのリストについては、暗号スイートおよび暗号スイート仕様ファイルを参照してください。

手順

1. 変更の対象となるリソース定義を選択します。 CIPHERS 属性のデフォルト値が表示されます。 CICS でデフォルト値を表示させるためには、リソース定義の操作対象となっている CICS 領域で **KEYRING** システム初期設定パラメーターを指定しておく必要があります。
2. 属性値を編集して、暗号スイートを除去したり、その順序を並べ替えたり、SSL 暗号スイート仕様ファイルの名前を指定したりします。 例えば、352F0A または strongciphers.xml を指定できます。
3. リソース定義を保管します。

例

352F0A を指定した場合、このリソースを使用する SSL 接続で CICS は 128 ビットを下回る暗号化での折衝を行わないことを意味します。属性内の 2 桁のコード (例えば 35、2F、0A) はそれぞれ、少なくとも 128 ビット暗号化を使用する暗号スイートを指します。 CICS は、AES 暗号スイート 35 と 2F を使用してネゴシエーションを試みることから始めます。これらの暗号コードがリストの最初にあるためです。クライアント側にこのレベルの暗号化がない場合、CICS は接続を閉じます。

strongciphers.xml を指定することは、CICS が *ussconfig*/security/ciphers/strongciphers.xml (ここで、*ussconfig* は SIT パラメーター **USSCONFIG** の値です) にある暗号ファイルにリストされている暗号を使用することを意味します。

CICS TS システムを NIST SP800-131A 準拠にする

システムを SP800-131A 準拠にするには、さまざまな SIT パラメーターとリソース属性を更新して、適合する暗号スイートと証明書を使用するようにします。

このタスクについて

米国連邦情報・技術局 (NIST) SP800-131A セキュリティー標準に規格適合すると、より強力な暗号鍵とより堅固なアルゴリズムの使用が必要となるため、セキュリティーが強化されます。

NIST 標準の詳細については、NIST Computer Security Resource Center (crc.nist.gov)を参照してください。

手順

システムを NIST SP800-131A 準拠にするには、以下の手順を実行します。

1. **NISTSP800131A** システム初期設定パラメーターを NISTSP800131A=CHECK に設定します。
2. **MINTLSLEVEL** システム初期設定パラメーターを TLS12 に設定します。
3. **KEYRING** システム初期設定パラメーターを、NIST SP800-131A 準拠の証明書が取り込まれた鍵リングの名前に設定します。
4. **USSCONFIG** システム初期設定パラメーターを、z/OS UNIX 上の CICS Transaction Server 構成ファイルのルート・ディレクトリーの名前とパスに設定します。 このディレクトリーには、SSL 暗号スイート仕様ファイルが少なくとも 1 つ含まれる /security/ciphers/ サブディレクトリーがなければなりません。詳しくは、 371 ページの『暗号スイートおよび暗号スイート仕様ファイル』を参照してください。
5. TCPIPSERVICE 定義、IPCONN 定義、または URIMAP 定義を更新し、**CIPHERS** 属性を、SP800-131A 準拠の暗号スイートが含まれる SSL 暗号スイート仕様ファイルの名前に設定します。 サンプル・ファイルの fipsciphers.xml は、適切なファイルです。
6. TCPIPSERVICE 定義、IPCONN 定義、または URIMAP 定義を更新し、**CERTIFICATE** 属性を、SP800-131A 準拠の証明書ラベルの名前に設定します。また、SSL を使用するアウトバウンド HTTP アプリケーションは、どのような **EXEC CICS WEB OPEN** コマンドでも SP800-131A 準拠の証明書を使用する必要があります。これらのリソース定義のいずれかまたは **WEB OPEN** コマンドと共に鍵リング・デフォルト証明書を使用する場合は、鍵リング・デフォルト証明書を SP800-131A 準拠にしておく必要があります。
7. CICSPlex SM Web ユーザー・インターフェース (WUI) を使用して CICS に接続する場合は、**TCPIPSSLCIPHERS** WUI サーバー初期設定パラメーターを、SP800-131A 準拠の暗号スイートが含まれる SSL 暗号スイート仕様ファイルの名前に設定します。 サンプル・ファイルの fipsciphers.xml は、適切なファイルです。
8. CICSPlex SM WUI を使用して CICS に接続する場合、**TCPIPSSLCERT** WUI サーバー初期設定パラメーターを、SP800-131A 準拠の証明書ラベルの名前に設定します。 鍵リング・デフォルト証明書を使用する場合は、鍵リング・デフォルト証明書を SP800-131A 準拠にしておく必要があります。

次のタスク

CICS に接続するすべてのクライアントは、SP800-131A に準拠して TLS 1.2 をサポートしている必要があります。準拠のためには、それらのクライアントは

SP800-131A 準拠の暗号スイートを使用できる必要があり、証明書を使用する場合は SP800-131A 準拠の証明書を使用できる必要があります。

パートナー CICS システムは、MINTLSLEVEL=TLS12 システムと通信するために MINTLSLEVEL=TLS12 または ENCRYPTION=ALL を使用する必要があり、SP800-131A 準拠の暗号スイートと証明書を使用するように構成されている必要があります。

注: NISTSP800131A=CHECK を設定する場合、CICS は以下のアクションを行います。

- JVM サーバーが始動すると、Java が NIST SP800-131A 準拠となるように、Java のいくつかのプロパティーが CICS によって設定されます。
- SAML を使用し、アウトバウンド・メッセージに署名する場合、CICS はメッセージ DFHXS1300 を発行して、使用される証明書が準拠しているかどうか検査するように警告します。
- WS-Security を使用する場合、WS-Security の CICS サポートは NIST SP800-131A に準拠していないため、CICS はメッセージ DFHXS1301 を発行します。
- SSL(ATTLSAWARE) で定義された TCPIPSERVICE がオープンされている場合、CICS はメッセージ DFHSO0148 を発行して、この TCPIPSERVICE を保護するために使用される AT-TLS ポリシーは **NIST SP800-131A** に準拠している必要があることを警告します。「z/OS Communications Server: IP 構成ガイド」の『AT-TLS ポリシー構成』を参照してください。

CICS を使用するための LDAP の構成

LDAP を使用して、CRL (証明書失効リスト) または基本認証資格情報を保管できます。証明書失効リストまたは資格情報が LDAP サーバーに保管されている場合、CICS にそれらへのアクセスを許可する必要があります。

このタスクについて

証明書失効リストおよびパスワードは、アクセス・クラスを *critical* に設定して、LDAP サーバーに保管されており、LDAP バインド時に認証資格情報を提供したユーザーのみがアクセスできます。これらの資格情報は、ユーザーの識別名および関連するパスワードです。これらの詳細は、LDAPBIND RACF クラス内の専用プロファイルに保管できます。このプロファイルをセットアップするには、次のステップに従います。

手順

1. プロファイルで使用されるパスワードは、RACF データベースに保管される前に、暗号化される必要があります。パスワードを暗号化するには、次の RACF コマンドのいずれかを発行して、パスワード暗号鍵を KEYSMSTR RACF クラスに保管する必要があります。
   - ```
     RDEFINE KEYSMSTR LDAP.BINDPW.KEY OWNER(userid)
             SSIGNON(KEYENCRYPTED(keyvalue))
     ```

     このコマンドは、パスワード暗号鍵が統合暗号化サービス機能 (ICSF) によって保管される場合に使用します。
   - ```
     RDEFINE KEYSMSTR LDAP.BINDPW.KEY OWNER(userid)
             SSIGNON(KEYMASKED(keymask))
     ```

このコマンドは、ICSF がアクティブでない場合に使用します。

2. 次の RACF コマンドを使用してプロファイルを作成します。

```
RDEFINE LDAPBIND profile-name
                PROXY(LDAPHOST(ldap-url)
                      BINDDN('ldap-distinguished-name')
                      BINDPW(password))
                UACC(NONE)
```

ここで、

profile-name
RACF プロファイルの名前です。その PROXY セグメントには、次の LDAP バインド・パラメーターが含まれています。

ldap-url
アクセスされる LDAP サーバーの完全修飾 URL です (例えば、LDAP://EXAMPLE.COM:3389)。

ldap-distinguished-name
サーバーから証明書失効リストの属性を問い合わせることを許可されている LDAP ユーザーの識別名です (例えば、CN=LDAPADMIN)。

password
LDAP ユーザーを認証するパスワードです。このパスワードは、大/小文字が区別されます。

3. 次の形式の 1 つ以上のコマンドを実行して、LDAPBIND クラス内の適切なバインド資格情報にアクセスするための各 CICS 領域ユーザー ID を許可します。

```
PERMIT profile-name CLASS(LDAPBIND)
                    ACCESS(READ)
                    ID(region-userid)
```

4. 該当する各 CICS 領域のシステム初期設定パラメーター **CRLPROFILE** にプロファイル名を指定します。

タスクの結果

CICS 領域が、**CRLPROFILE** システム初期設定パラメーターで指定されたプロファイル名で始まる場合、LDAP サーバーのバインド情報は CICS 領域の SSL 環境にキャッシュされ、その領域は z/OS System SSL によって管理されます。PERFORM SSL REBUILD コマンドを CICS 領域に対して発行すると、LDAP サーバーのバインド情報の外部セキュリティー・マネージャーからリフレッシュされます。

次のタスク

CRLPROFILE パラメーターが CICS 領域に指定されているが無効である場合、または指定されたプロファイルに無効なデータが含まれている場合、またはプロファイルによって特定された LDAP サーバーが CICS 領域の開始時に使用できない場合、CICS 領域は LDAP サーバーに対するそれ自体のアクセスを使用不可にします。この問題は、メッセージ DFHSO0128 および DFHSO0129 で報告されます。

アクセスを復元するには、エラーを修正して、CICS 領域を再始動する必要があります。CICS 領域が LDAP サーバーを使用不可にしている場合、PERFORM SSL REBUILD コマンドは LDAP サーバーへのアクセスを復元できません。コマンドの

発行時に CICS 領域が使用可能であった LDAP サーバーにのみ、リフレッシュが行われます。

証明書失効リスト (CRL) の使用

証明書失効リスト (CRL) を使用して SSL ネゴシエーションで使用されているクライアント証明書の妥当性を確認するように、CICS を構成できます。

始める前に

証明書失効リストを使用するには、LDAP サーバーをインストールして構成する必要があります。z/OS Cryptographic Services PKI サービス ガイドおよび解説書を参照してください。また、CICS で使用するための LDAP の構成の説明に従って、CICS に LDAP サーバーへのアクセスを許可する必要もあります。

このタスクについて

証明書失効リストには、認証局からの失効した証明書に関する詳細が含まれています。認証局はこれらのリストをワールド・ワイド・ウェブ上の使用可能な CRL リポジトリーに保持します。これらのリストはダウンロードして、LDAP サーバーに格納できます。LDAP サーバーにデータを追加し、証明書失効リストを更新するには、CICS 提供のトランザクション CCRL を使用します。

CRL のための LDAP サーバーの構成

証明書失効リスト (CRL) を使用するには、LDAP サーバーが稼働している必要があります。また、CRL をダウンロードする前に、いくつかの構成ステップを実行する必要もあります。

始める前に

LDAP サーバーをインストールして構成する必要がある場合は、z/OS Cryptographic Services PKI サービス ガイドおよび解説書を参照してください。

このタスクについて

手順

1. LDAP サーバーが稼働していることを確認します。デフォルトの開始タスク名は LDAPSRV です。
2. `etc/ldap` のファイル・システムで、次のように構成ファイル slapd.conf を編集します。
   a. adminDN および adminPW の値を指定して、管理者の識別名およびパスワードを作成します。 CICS 提供の CCRL トランザクションは、LDAP サーバーを証明書失効リストで更新するために、この情報を必要とします。
   b. CCRL を使用して CRL をダウンロードするすべての認証局について接尾部項目を作成します。それぞれの接尾部に、構文 "O=*certificate authority*" を使用します。 接尾部は、組織つまりキーワード "O=" を含む認証局の識別名と、この右にあるその他のキーワードで構成されます。接尾部に 特殊文字 **<**、**+**、**;**、**>**、**¥**" のいずれかが含まれる場合、**2** 個の円記号 (¥) 文字を使用して、その特殊文字をエスケープする必要があります。z/OS LDAP サーバーを使用しているときに、必要な 1047 コード・ページに載っていない文

字が接尾部に含まれる場合、その文字をエスケープする必要があります。そうするには、アンパーサンドが前に付いた Unicode 表記の 3 桁の 8 進数として、その文字をエンコードします。

例

例えば、ファイル slapd.conf では次の接尾部を指定できます。

```
suffix "O=CompanyName"
suffix "O=CompanyName plc"
suffix "O=CompanyName,L=CompanyLocation,ST=CompanyArea,C=CompanyCountry"
suffix "O=CompanyName¥¥, Inc."
suffix "O=CompanyName¥¥, Inc.,C=CompanyCountry"
```

次のタスク

すべての認証局を含むように LDAP サーバーを構成した場合は、CCRL トランザクションを実行します。詳しくは、『CCRL トランザクションの実行』を参照してください。

CCRL トランザクションの実行

CICS 提供のトランザクション CCRL によって、証明書失効リスト (CRL) のダウンロードと保管ができるようになります。これは、クライアント証明書が有効かどうかを判断するために SSL ハンドシェークで使用することができるものです。

始める前に

使用する認証局を指定するように LDAP サーバーを構成し、管理者 ID とパスワードを作成する必要があります。詳しい手順については、 394 ページの『CRL のための LDAP サーバーの構成』を参照してください。

このタスクについて

証明書失効リストは、Verisign などの認証局から入手できます。これらは、ワールド・ワイド・ウェブ上の使用可能な CRL リポジトリーに保持され、LDAP サーバーでダウンロードおよび格納できます。LDAP サーバーにデータを追加し、証明書失効リストを更新するには、CICS 提供のトランザクション CCRL を使用します。CCRL トランザクションは、端末から実行するか、または **START** コマンドを使用して実行することができます。定期更新をスケジュールするには、**START** コマンドを使用します。

手順

1. システム初期設定パラメーター **CRLPROFILE** で LDAP サーバーの名前を指定します。
2. CCRL トランザクションを実行します。
   - 端末からの場合、 396 ページの『端末からの CCRL の実行』を参照してください。
   - コマンドを使用する場合、 396 ページの『START コマンドからの CCRL の実行』を参照してください。

端末からの **CCRL** の実行:

端末を使用して CICS 提供のトランザクション CCRL を実行することにより、証明書失効リスト (CRL) をダウンロードできます。

始める前に

このトランザクションを端末から実行する前に、 395 ページの『CCRL トランザクションの実行』を読んで前提条件を確認してください。

手順

1. 端末から、URL のリストを大/小文字混合で入力できるように、コマンド `CEOT TRANIDONLY` を入力します。
2. `CCRL` *`url-list`* を入力します。ここで、*url-list* は、ダウンロードする証明書失効リスト・ファイルの場所を指定する URL です。リスト内のそれぞれの URL の間にスペースを入れることにより、複数の URL を指定できます。 例えば、次のように指定します。`CCRL http://crl.verisign.com/ATTClass1Individual.crl http://crl.verisign.com/ATTClass2Individual.crl`.
3. LDAP サーバーの管理者の識別名とパスワードの入力を求めるプロンプトが出されます。これにより、CICS は LDAP サーバーを、ダウンロードした CRL で更新できます。 管理者名とパスワードは、ファイル slapd.conf で指定されています。このファイルの構成について詳しくは、 394 ページの『CRL のための LDAP サーバーの構成』を参照してください。

タスクの結果

CICS は、指定された URL から CRL をダウンロードし、それを LDAP サーバーに保管します。すべてのリストがダウンロードされたことの確認が送信されます。CICS で問題が発生した場合 (例えば、URL が無効である場合)、エラー・メッセージが表示されます。

次のタスク

定期更新をセットアップするには、START コマンドを使用できます。『START コマンドからの CCRL の実行』を参照してください。

START コマンドからの **CCRL** の実行:

START コマンドを使用して、CCRL トランザクションを定期的に実行するようにスケジュールできます。

始める前に

このトランザクションを端末から実行する前に、 395 ページの『CCRL トランザクションの実行』を読んで前提条件を確認してください。

手順

START コマンドを使用するには、`EXEC CICS START TRANSID(CCRL) FROM (admin://`*`adminDN:adminPW url-list`*`) LENGTH (`*`url-list-length`*`) [INTERVAL(`*`hhmmss`*`)|TIME(`*`hhmmss`*`)]` を入力します。ここで、*url-list* は、証明書失効

リストのダウンロード元の URL のスペースで区切られたリストです。*url-list-length* は、URL リストの長さ (admin:// を含む) です。*hhmmss* は、CCRL トランザクションがスケジュールされる間隔または有効期限です。 例えば、次のように指定できます。

```
EXEC CICS START TRANSID(CCRL)
FROM('admin://cn=ldapadmin:cics31ldap
      http://crl.verisign.com/ATTClass1Individual.crl
      http://crl.verisign.com/ATTClass2Individual.crl')
      LENGTH(124) INTERVAL(960000)
```

この例では、CCRL トランザクションを 96 時間の間隔で実行するようにスケジュールしています。

第 9 章 データ・ソースのセキュリティー

Db2 のセキュリティー

CICS Db2 環境には、セキュリティー検査を実行できる 4 つの主なステージがあります。

それら 4 つのステージは以下のとおりです。

- CICS ユーザーが CICS 領域にサインオンするとき。CICS サインオンでは、ユーザーが有効なユーザー ID とパスワードを指定したことを確認することで、そのユーザーを認証します。
- CICS ユーザーが Db2 に関連している CICS リソースを使用または変更しようとしているとき。このリソースは、以下のものである可能性があります: DB2CONN、DB2ENTRY、または DB2TRAN のリソース定義。あるいは、データを取得するために Db2 にアクセスする CICS トランザクション。あるいは、CICS Db2 接続機能または Db2 自体にコマンドを発行する CICS トランザクション。このステージでは、RACF または同等の外部セキュリティー・マネージャーによって管理される CICS セキュリティー・メカニズムを使用して、リソースへの CICS ユーザーのアクセスを制御できます。
- CICS 領域を Db2 に接続するとき、およびトランザクションが Db2 に送信されたスレッドを取得するとき。CICS 領域とトランザクションの両方で Db2 に対する許可 ID を提供する必要があり、これらの許可 ID は RACF または同等の外部セキュリティー・マネージャーによって検証されます。
- CICS ユーザーが CICS トランザクションを使用して Db2 リソースを実行または変更しようとしているとき。これは、計画、Db2 コマンド、または動的 SQL を実行するために必要なリソースである場合があります。このステージでは、Db2 自体、あるいは RACF または同等の外部セキュリティー・マネージャーによって管理される Db2 のセキュリティー検査を使用して、リソースへの CICS ユーザーのアクセスを制御できます。

また、RACF、あるいは同等の外部セキュリティー・マネージャーを使用して、CICS と Db2 を構成するコンポーネントを無許可アクセスから保護することもできます。この保護は、Db2 のデータベース、ログ、ブートストラップ・データ・セット (BSDS)、Db2 の範囲外のライブラリー、および CICS のデータ・セットとライブラリーに適用できます。RACF で提供される保護を部分的に置き換えるものとして、VSAM パスワード保護を使用することもできます。詳細については、CICS システム・リソースのセキュリティーを参照してください。

注**:** ここでは、RACF は CICS で使用される外部セキュリティー・マネージャーとして言及されています。明示的な RACF の例を除き、一般的な説明が、機能的に同等のすべての非 IBM 外部セキュリティー・マネージャーに同様に適用されます。

400 ページの図 27 は、CICS Db2 環境で使用されるセキュリティー・メカニズムを示しています。

© Copyright IBM Corp. 1989, 2018

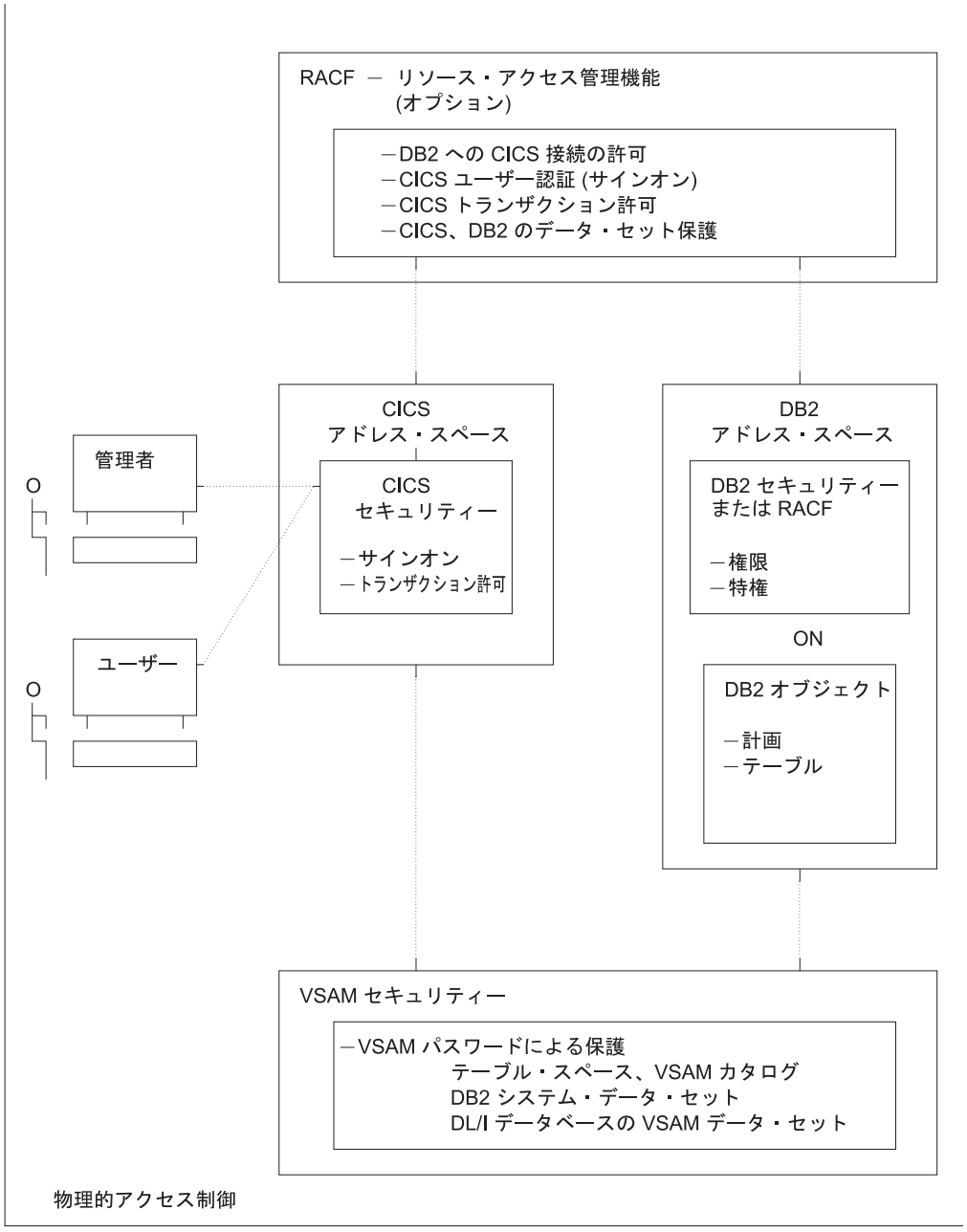


図 *27. CICS Db2* セキュリティー・メカニズムの概要

CICS での Db2 関連リソースへのアクセスの制御

外部セキュリティー・マネージャーおよび適切な CICS セキュリティー・メカニズムを有効にすることによって、CICS 領域内の Db2 リソースへのアクセスを制御し、リソースのセキュリティー検査を開始することができます。

このタスクについて

CICS ユーザーは、Db2 に関係する以下のアクティビティーを実行することが必要になる場合があります。

- DB2CONN、DB2ENTRY、および DB2TRAN の各リソース定義の照会、変更、作成、または破棄。
- Db2 にアクセスするトランザクションを使用したデータの取得、または DSNC トランザクションの使用による CICS Db2 接続機能コマンドまたは Db2 コマンドの発行。

RACF または同等の外部セキュリティー・マネージャーを使用して、CICS 領域でセキュリティー検査を実行します。ユーザーが保護リソースにアクセスしようとすると、CICS は外部セキュリティー・マネージャーを呼び出してセキュリティー検査を実行します。 RACF は、ユーザーが CICS にサインオンするときに認証される、CICS ユーザーの ID を使用してセキュリティー検査を行います。ユーザー ID が事前設定セキュリティーによって端末に永続的に関連付けられている場合を除き、ユーザーが CICS にサインオンしない場合は、デフォルト・ユーザー ID が付与されます。

CICS 領域に対する適切なセキュリティー・メカニズム (トランザクション接続セキュリティー、リソース・セキュリティー、コマンド・セキュリティー、または代理セキュリティー) を有効にします。

CICS セキュリティーの詳細については、保護の概要を参照してください。

DB2CONN、DB2TRAN、および DB2ENTRY リソース定義に対するユーザー・アクセスの制御

さまざまな CICS セキュリティー・メカニズムを有効にすることで、DB2CONN、DB2TRAN、および DB2ENTRY の各リソース定義に対するユーザーのアクセスを制御できます。

このタスクについて

以下のセキュリティー・メカニズムを使用して、Db2 リソース定義に対するユーザー・アクセスを制御できます。

- CICS リソース・セキュリティー・メカニズムを使用して、特定のリソースにユーザーがアクセスできるかを制御します。リソース・セキュリティーはトランザクション・レベルで実装されます。例えば、一部のユーザーに、特定の DB2ENTRY 定義を変更させないようにすることができます。このセキュリティー・メカニズムの使用法については、 402 ページの『リソース・セキュリティーを使用して、DB2ENTRY リソース定義および DB2TRAN リソース定義へのアクセスを制御する』の説明を参照してください。
- CICS コマンド・セキュリティー・メカニズムを使用して、Db2 関連リソースに対してユーザーが特定の SPI コマンドを発行できるかを制御します。コマンド・セキュリティーもトランザクション・レベルで実装されます。例えば、特定のユーザーのみに、DB2ENTRY リソース定義に対して CREATE コマンドと DISCARD コマンドを発行することを許可できます。このセキュリティー・メカニズムの使用法については、 404 ページの『コマンド・セキュリティーを使用して、DB2CONN、DB2ENTRY、および DB2TRAN の各リソース定義に対する SPI コマンドの発行を制御する』の説明を参照してください。
- CICS 代理セキュリティーおよび **AUTHTYPE** セキュリティーのメカニズムを使用して、CICS が Db2 に提供する許可 ID をユーザーが変更できるかを制御し

ます。許可 ID は、Db2 のセキュリティー検査に使用され、Db2 関連リソース定義の AUTHID、COMAUTHID、AUTHTYPE および COMAUTHTYPE の各属性と、CICS 領域の DB2CONN 定義の SIGNID 属性によって設定されます。CICS は、許可 ID を変更を求めるユーザーが、リソース定義で指定された既存の許可 ID に代わる役割を果たすことが許可されるかどうかを検査します。これらのセキュリティー・メカニズムの使用法については、406 ページの『代理セキュリティーおよび AUTHTYPE セキュリティーを使用して、CICS が DB2 に提供する許可 ID へのアクセスを制御する』の説明を参照してください。

リソース・セキュリティーを使用して、DB2ENTRY リソース定義および DB2TRAN リソース定義へのアクセスを制御する

CICS リソース・セキュリティー・メカニズムは、指定された CICS リソースへのユーザーのアクセスを制御します。例えば、このメカニズムを使用して、あるリソース (例えば、特定の DB2ENTRY 定義) が特定のユーザーによって変更されないようにすることができます。

このタスクについて

CICS コマンド・セキュリティーは、ユーザーがリソースのタイプ (「すべての DB2ENTRY 定義」など) に対して特定のアクションを実行しないようにすることはできますが、リソース・タイプ内の個々の項目を保護することはできません。

CICS 領域には 1 つの DB2CONN 定義しかないため、リソース・セキュリティーを使用して保護する必要はありません。コマンド・セキュリティーを使用して DB2CONN 定義へのアクセスを制御できます。また、リソース・セキュリティーのために、DB2TRAN 定義はそれが参照する DB2ENTRY 定義の拡張と見なされ、それだけでリソース・セキュリティーに定義されることはありません。ユーザーに DB2ENTRY 定義へのアクセス権限を与える場合、それを参照する DB2TRAN 定義へのアクセス権限も与えます。 (トランザクションが、DB2TRAN 定義が関連付けられている DB2ENTRY の名前を変更する場合、二重セキュリティー検査が実行されます。この検査で、DB2TRAN 定義が参照した古い DB2ENTRY と、これから参照する新しい DB2ENTRY の両方を変更するユーザーの権限を確認します。) そのため、リソース・セキュリティーでは、DB2ENTRY 定義を RACF に対して定義する必要のみがあります。

リソース・セキュリティーがトランザクションに対して有効にされている場合、外部セキュリティー・マネージャーは、そのトランザクションに関連付けられているユーザー ID が、関係するリソースの変更を許可されているかどうかを検査します。 リソース定義のセキュリティーには、このプロセスに関する詳細情報があります。

リソース・セキュリティーを使用して Db2 関連リソースを保護するには、以下の手順を実行します。

手順

1. RACF または同等の外部セキュリティー・マネージャーを有効にし、CICS 領域でリソース・セキュリティーを有効にするには、CICS 領域のシステム初期設定パラメーターとして SEC=YES を指定します。

2. RACF で、Db2 関連のリソースを含める一般リソース・クラスを作成します。メンバー・クラスとグループ化クラスが必要です。 CICS にデフォルトの RACF リソース・クラス名があるのとは異なり、DB2ENTRY には IBM 提供のデフォルトのクラス名はありません。RACF クラス記述子テーブル (CDT) のインストール定義部分 (モジュール ICHRRCDE) に新規のクラス記述子を追加して、ユーザー独自のインストール定義クラス名を作成します。この方法の例については、CICSTS55.CICS.SDFHSAMP のメンバー DFH$RACF で提供されている、IBM 提供のサンプル・ジョブ、RRCDTE を参照してください。ここには、XCICSDB2 というメンバー・クラスと、ZCICSDB2 というグループ化クラスの例があります。この例は、CICS のデフォルトのリソース・クラス名と同じ命名規則を使用します。Db2 関連のリソース定義に既存の CICS クラス名を使用しないでください。代わりに、同様の命名規則を使用して新しいクラス名を作成してください。
3. 作成したリソース・クラスで、DB2ENTRY 定義のプロファイルを定義します。例えば、DB2ENTRY 名の番号を XCICSDB2 リソース・クラスに追加するには、RDEFINE コマンドを以下のように使用します。

```
RDEFINE  XCICSDB2  (db2ent1, db2ent2, db2ent3.., db2entn) UACC(NONE)
                   NOTIFY(sys_admin_userid)
```

   DB2ENTRY リソース定義を保護すると、関連する DB2TRAN 定義へのアクセスも保護されます。DB2TRAN はそれが参照する DB2ENTRY への拡張と見なされるからです。そのため、リソース・セキュリティーを使用して、DB2CONN 定義を保護する必要はありません。
4. Db2 関連リソースのリソース・セキュリティーをアクティブにするには、CICS 領域のシステム初期設定パラメーターとして XDB2=*name* を指定します (ここで、*name* は、Db2 関連リソースに定義された一般リソース・クラス名です)。
5. リソース・セキュリティーを有効にしたい Db2 関連リソースが含まれるすべてのトランザクションに対して、リソース定義で RESSEC=YES を指定します。ユーザーがこれらのいずれかのトランザクションを使用して、保護されている Db2 関連リソースの 1 つにアクセスしようとすると、RACF はユーザー ID がそのリソースへのアクセスを許可されていることを確認します。
6. CICS ユーザーまたはユーザーのグループに、保護されている各 Db2 関連リソースに対して適切なアクションを実行する権限を与えます。 ユーザーが DB2ENTRY 定義に対してアクションを実行する権限を持っている場合、それに関連付けられている DB2TRAN 定義に対して同じアクションを実行することが自動的に許可されることを覚えておいてください。ユーザーが特定のアクションを実行するために必要な権限は以下のとおりです。

   INQUIRE コマンド
   : READ 権限が必要

   SET セット
   : UPDATE 権限が必要

   CREATE コマンド
   : ALTER 権限が必要

   DISCARD コマンド
   : ALTER 権限が必要

例えば、次のように PERMIT コマンドを使用して、クラス XCICSDB2 の保護された DB2ENTRY である db2ent1 を UPDATE 権限で変更することをユーザーのグループに許可できます。

```
PERMIT db2ent1 CLASS(XCICSDB2) ID(group1) ACCESS(UPDATE)
```

コマンド・セキュリティーを使用して、DB2CONN、DB2ENTRY、および DB2TRAN の各リソース定義に対する SPI コマンドの発行を制御する

CICS コマンド・セキュリティー・メカニズムを使用して、DB2CONN、DB2ENTRY、および DB2TRAN の各リソース定義を保護します。

このタスクについて

CICS コマンド・セキュリティー・メカニズムは、Db2 関連リソースのタイプに対して特定の SPI コマンドをユーザーが発行する機能を制御します。例えば、このメカニズムを使用して、どのユーザーが DB2ENTRY リソース定義に対して CREATE コマンドおよび DISCARD コマンドを発行することが許可されるかを制御できます。リソース・セキュリティーとは異なり、CICS コマンド・セキュリティーは個々の指定されたリソースを保護できません。これはリソースのタイプを保護するように設計されています。コマンド・セキュリティーを使用して、DB2CONN リソース定義、DB2ENTRY リソース定義、および DB2TRAN リソース定義を保護できます。

コマンド・セキュリティーがトランザクションに対して有効にされている場合、外部セキュリティー・マネージャーは、そのトランザクションに関連付けられているユーザー ID が、そのコマンドを使用して関係するリソースのタイプを変更することを許可されているかどうかを検査します。CICS command securityには、このプロセスに関する詳細情報があります。

特定のトランザクションに対してリソース・セキュリティーとコマンド・セキュリティーの両方が有効にされている場合、RACF はユーザー ID に対して 2 つのセキュリティー検査を実行します。例えば、DB2ENTRY 定義 db2ent1 に対して DISCARD を発行するユーザーがトランザクションに含まれている場合、RACF は以下のことを検査します。

1. ユーザー ID が DB2ENTRY リソース・タイプに対して DISCARD コマンド (ALTER 権限) を発行することが許可されている。
2. ユーザー ID が ALTER 権限を使用して DB2ENTRY 定義 db2ent1 にアクセスすることが許可されている。

コマンド・セキュリティーを使用して Db2 関連リソースを保護するには、以下の手順を実行します。

手順

1. RACF または同等の外部セキュリティー・マネージャーを CICS 領域に対して有効にするには、その領域のシステム初期設定パラメーターとして SEC=YES を指定します。
2. Db2 リソース名 DB2CONN、DB2ENTRY、および DB2TRAN を CICS コマンド用の IBM 提供の RACF リソース・クラスである CCICSCMD または

VCICSCMD のいずれかにリソース ID として追加します。 あるいは、CICS コマンドにユーザー定義の一般リソース・クラスを使用することもできます。コマンド・セキュリティー検査の対象となる CICS リソースでは、これについて詳しく説明しています。 例えば、次のように、REDEFINE コマンドを使用してデフォルト・クラス VCICSCMD に CMDSAMP というプロファイルを定義し、ADDMEM オペランドを使用して Db2 リソース・タイプがこのプロファイルによって保護されることを指定できます。

```
RDEFINE  VCICSCMD CMDSAMP UACC(NONE)
                  NOTIFY(sys_admin_userid)
                  ADDMEM(DB2CONN, DB2ENTRY, DB2TRAN)
```

3. コマンド・セキュリティーを CICS 領域に対して有効にするには、次のようにします。
   a. CICS コマンド・プロファイルに IBM 提供の RACF リソース・クラス CCICSCMD または VCICSCMD を使用した場合、その領域のシステム初期設定パラメーターとして XCMD=YES を指定します。 YES を指定することは、CCICSCMD および VCICSCMD が RACF のストレージ内のプロファイルの作成に使用されることを意味します。
   b. CICS コマンドにユーザー定義の一般リソース・クラスを使用した場合、その領域の初期設定パラメーターとして XCMD=*user_class* を指定します (ここで、*user_class* はユーザー定義の一般リソース・クラスの名前です)。
4. コマンド・セキュリティーを有効にしたい Db2 関連リソースが含まれるすべてのトランザクションに対して、リソース定義で CMDSEC=YES を指定します。ユーザーがこれらのいずれかのトランザクションを使用して、保護されている Db2 関連リソースの 1 つを変更しようとすると、RACF はユーザー ID がそのリソースのタイプに対してそのコマンドを発行することを許可されていることを確認します。
5. CICS ユーザーまたはユーザーのグループに、Db2 関連リソースのそれぞれのタイプに対して適切なコマンドを発行する権限を与えます。コマンド・セキュリティーの場合、DB2TRAN リソース・タイプおよび DB2ENTRY リソース・タイプに関連する別個の権限を与える必要があります。また、DB2CONN リソース・タイプ (すなわち、CICS 領域の DB2CONN 定義) を保護することもできます。

   ユーザーが特定のコマンドを発行するために必要な権限は以下のとおりです。

   INQUIRE コマンド
   READ 権限が必要

   SET セット
   UPDATE 権限が必要

   CREATE コマンド
   ALTER 権限が必要

   DISCARD コマンド
   ALTER 権限が必要

   例えば、ステップ 2 の例のように CMDSAMP プロファイルに Db2 リソース・タイプを定義した場合、次のように PERMIT コマンドを使用して、Db2 リソース・タイプに対して EXEC CICS INQUIRE コマンドを発行することをユーザーのグループに許可できます。

```
PERMIT CMDSAMP CLASS(VCICSCMD) ID(operator_group) ACCESS(READ)
```

トランザクション内で、**EXEC CICS QUERY SECURITY RESTYPE(SPCOMMAND)** コマンドを、DB2CONN、DB2ENTRY、または DB2TRAN を指定する RESID パラメーターとともに使用することで、ユーザー ID が Db2 リソース・タイプへのアクセス権限を持っているどうか照会できます。

代理セキュリティーおよび AUTHTYPE セキュリティーを使用して、CICS が DB2 に提供する許可 ID へのアクセスを制御する

CICS 代理セキュリティー・メカニズムおよび AUTHTYPE セキュリティー・メカニズムは、CICS が DB2 に提供する許可 ID をユーザーが変更する機能を制御します。

このタスクについて

代理セキュリティーおよび AUTHTYPE セキュリティーを使用して、特定のユーザーのみに許可 ID の変更が許可されるようにします。この許可 ID は DB2 独自のセキュリティー検査に使用されます。代理セキュリティーおよび AUTHTYPE セキュリティーは CICS 領域全体に対して設定され、許可 ID の変更を含むトランザクションはそれらの影響を受けます。

410 ページの『CICS 領域用および CICS トランザクション用に、許可 ID を Db2 に提供する』では、これらの許可 ID を選択および変更する方法が説明されています。要約すると、CICS が DB2 に提供する許可 ID は、DB2 関連リソース定義では AUTHID 属性、COMAUTHID 属性、AUTHTYPE属性、および COMAUTHTYPE 属性によって設定され、CICS 領域の DB2CONN 定義では SIGNID 属性によって設定されます。許可 ID を変更するには、まず DB2CONN 定義および DB2ENTRY 定義を変更する権限が必要です。これらの定義はコマンド・セキュリティーまたはリソース・セキュリティーによって保護されている可能性があります。代理セキュリティーは追加の保護レイヤーを提供します。代理セキュリティーには DB2 に代わって機能する CICS が含まれており、許可 ID を変更するユーザーがリソース定義で指定された既存の許可 ID の代理として機能することを許可されているかどうか検査するからです。

真の代理セキュリティーは、ユーザーが DB2CONN 定義または DB2ENTRY 定義で SIGNID 属性、AUTHID 属性、または COMAUTHID 属性を変更しようとするときにセキュリティー検査を行います。これらの属性はすべて、プロセスが DB2 にサインオンするときに使用される許可 ID を指定します。CICS は、RACF の代理ユーザー機能を使用して、この検査を実行します。代理ユーザーとは、他のユーザーのパスワードを知らなくてもそのユーザーに代わって作業を行う権限を持っているユーザーのことです。ユーザーが SIGNID 属性、AUTHID 属性、または COMAUTHID 属性のいずれかを変更しようとすると、CICS は RACF を呼び出して、そのユーザーが SIGNID 属性、 AUTHID 属性、または COMAUTHID 属性で現在指定されている許可 ID の代理として許可されていることを確認します。

AUTHTYPE 属性および COMAUTHTYPE 属性は、許可 ID そのものを指定するのではなく、使用される許可の ID のタイプを提供するため、CICS は真の代理セキュリティーを使用できません。代わりに、AUTHTYPE セキュリティーと呼ばれるメカニズムを使用します。ユーザーが AUTHTYPE 属性または COMAUTHTYPE 属性のいずれかを変更しようとすると、CICS は RACF を呼び

出して、ユーザーが RACF FACILITY 一般リソース・クラスでリソース定義に対して定義されているプロファイルによって許可されていることを確認します。AUTHTYPE セキュリティーが真の代理セキュリティーでない場合でも、同じシステム初期設定パラメーターによって有効にされており、それを代理セキュリティーに加えて使用する可能性があります。そのため、このトピックの説明には両方のタイプのセキュリティーのセットアップ方法が含まれています。

DB2CONN リソース定義および DB2ENTRY リソース定義が CICS のコールド・スタートまたは初期始動の一環としてインストールされているときに、代理セキュリティーと AUTHTYPE セキュリティーが有効にされている場合、RACF は CICS 領域ユーザー ID に対して代理セキュリティー検査と AUTHTYPE セキュリティー検査を行います。このようにして DB2CONN リソース定義および DB2ENTRY リソース定義をインストールする場合、CICS 領域ユーザー ID がリソース定義で指定された許可 ID の代理ユーザーとして定義されていること、およびそれが RACF FACILITY 一般リソース・クラスの正しいプロファイルによって許可されていることを確認してください。

CICS が DB2 に提供する許可 ID を保護するために代理セキュリティーおよび AUTHTYPE セキュリティーを実装するには、以下の手順を実行します。

手順

1. RACF または同等の外部セキュリティー・マネージャーを CICS 領域に対して有効にするには、その領域のシステム初期設定パラメーターとして SEC=YES を指定します。
2. CICS 領域の代理セキュリティーおよび AUTHTYPE セキュリティーをアクティブにするには、その領域のシステム初期設定パラメーターとして XUSER=YES を指定します。このシステム初期設定パラメーターは、両方のセキュリティー・メカニズムを有効にします。セキュリティー・メカニズムが有効にされている場合、DB2CONN リソース定義および DB2ENTRY リソース定義の SIGNID、AUTHID、COMAUTHID、AUTHTYPE、および COMAUTHTYPE の各属性に対して機能する EXEC CICS SET コマンド、CREATE コマンド、および INSTALL コマンドがトランザクションに含まれているときはいつも、CICS は RACF を呼び出してセキュリティー検査を実行します。 SIGNID 属性、AUTHID 属性、および COMAUTHID 属性の場合、RACF は代理セキュリティー検査を実行し、AUTHTYPE 属性または COMAUTHTYPE 属性の場合、RACF は AUTHTYPE セキュリティー検査を実行します。
3. 代理セキュリティーのために、適切な CICS ユーザーまたはユーザーのグループを、DB2CONN 定義および DB2ENTRY 定義の SIGNID 属性、AUTHID 属性、または COMAUTHID 属性で指定された許可 ID の代理として定義する必要があります。ユーザー ID を許可 ID の代理として定義するには、以下のようにします。
   a. RACF SURROGAT クラスに許可 ID のプロファイルを作成します。名前の形式は *authid*.DFHINSTL とし、許可 ID は所有者として定義します。例えば、DB2AUTH1 を SIGNID 属性、AUTHID 属性、または COMAUTHID 属性で許可 ID として指定した場合、次のコマンドを使用してプロファイルを作成します。

```
RDEFINE  SURROGAT  DB2AUTH1.DFHINSTL  UACC(NONE) OWNER(DB2AUTH1)
```

b. 作成したプロファイルの READ 権限を適切な CICS ユーザーに付与することで、そのユーザーが許可 ID の代理として機能することを許可します。例えば、ID が CICSUSR1 のユーザーに、許可 ID が DB2AUTH1 の代理として機能することを許可し、そのために既存の許可 ID として DB2AUTH1 を指定している SIGNID 属性、AUTHID 属性、または COMAUTHID 属性をインストールあるいは変更するには、次のコマンドを使用します。

```
PERMIT   DB2AUTH1.DFHINSTL  CLASS(SURROGAT) ID(CICSUSR1)  ACCESS(READ)
```

SIGNID 属性、AUTHID 属性、または COMAUTHID 属性に指定されたすべての許可 ID に対して、このプロセスを繰り返します。

c. SIGNID 属性、AUTHID 属性、または COMAUTHID 属性が CICS のコールド・スタートまたは初期始動の一環として含まれる DB2CONN リソース定義および DB2ENTRY リソース定義をインストールする必要がある場合、それらの属性によって指定された許可 ID の代理として機能することを CICS 領域ユーザー ID に許可します。 DB2CONN リソース定義および DB2ENTRY リソース定義のデフォルトには、AUTHID 属性および COMAUTHID 属性は含まれていません。インストール済みの DB2CONN 定義のデフォルト SIGNID は、CICS 領域のアプリケーション ID です。

4. AUTHTYPE セキュリティーのために、RACF FACILITY 一般リソース・クラスの DB2CONN リソース定義または DB2ENTRY リソース定義ごとにプロファイルを作成し、そのプロファイルに対する READ アクセス権限を適切な CICS ユーザーまたはユーザーのグループに与える必要があります。(このプロセスは真の代理セキュリティー・メカニズムを模倣しますが、特定の許可 ID の使用は含まれません。代わりに、各リソース定義を保護します。) 以下はその方法です。

a. RACF FACILITY 一般リソース・クラスに DB2CONN リソース定義または DB2ENTRY リソース定義のプロファイルを作成します。名前の形式は DFHDB2.AUTHTYPE.*authname* とします (ここで、*authname* は DB2CONN リソース定義または DB2ENTRY リソース定義の名前です)。例えば、DB2CONN1 という名前の DB2CONN リソース定義のプロファイルを定義するには、次のコマンドを使用します。

```
RDEFINE  FACILITY  DFHDB2.AUTHTYPE.DB2CONN1   UACC(NONE)
```

b. 適切な CICS ユーザーに、作成したプロファイルに対する READ 権限を与えます。 例えば、ID が CICSUSR2 のユーザーに、DB2CONN1 という名前の DB2CONN リソース定義の AUTHTYPE 属性または COMAUTHTYPE 属性をインストールあるいは変更することを許可するには、次のコマンドを使用します。

```
PERMIT  DFHDB2.AUTHTYPE.DB2CONN1 CLASS(FACILITY) ID(CICSUSR2) ACCESS(READ)
```

このプロセスを DB2CONN リソース定義および DB2ENTRY リソース定義のそれぞれに対して繰り返します。また、AUTHTYPE 属性または COMAUTHTYPE 属性が CICS のコールド・スタートまたは初期始動の一環として含まれる DB2CONN リソース定義および DB2ENTRY リソース定義をインストールする必要がある場合は、そのリソース定義のプロファイルの CICS 領域ユーザー ID に READ 権限を与えます。

タスクの結果

Db2 関連の CICS トランザクションへのユーザー・アクセスの制御

CICS トランザクション接続セキュリティー・メカニズムを使用して、データを取得するために Db2 にアクセスする CICS トランザクション、DSNC トランザクション、および CICS Db2 接続機能コマンドと Db2 コマンドを発行する他のすべてのトランザクションに対するユーザーのアクセスを制御します。

このタスクについて

トランザクション接続セキュリティーが使用可能な場合、RACF または同等の外部セキュリティー・マネージャーは、CICS ユーザーが要求したトランザクションを実行する権限がその CICS ユーザーにあるかを検査します。

トランザクション接続セキュリティーを使用する Db2 関連トランザクションを保護するには、トランザクション・セキュリティーの説明に従ってください。プロセスはすべての CICS トランザクションについて同じです。トランザクション接続セキュリティー・メカニズムに関する限り、Db2 関連トランザクションに特別な考慮事項はありません。その指示は、以下の方法を示しています。

- トランザクション接続セキュリティーをアクティブ化するために、CICS 領域用の適切なシステム初期設定パラメーターをセットアップする (トランザクション接続セキュリティーを制御する CICS パラメーターを参照)。
- 保護したいトランザクションについて、トランザクション・プロファイルを RACF に定義する (RACF profilesを参照)。

CICS Db2 接続機能コマンドと Db2 コマンドを発行する、DSNC 以外のトランザクションを定義した場合 (例えば、各コマンドを実行するための別個のトランザクションを定義した場合) には、これらのトランザクションも同様に RACF に定義することを覚えておいてください。

どの CICS ユーザーが、Db2 にアクセスするトランザクションを使用できるかを制御できるようになりました。トランザクション・プロファイル用のアクセス・リストに、適切なユーザー、またはユーザーのグループを READ 権限と共に追加してください。 RACF profilesには、これに関するいくつかの推奨事項があります。

CICS Db2 接続機能コマンドと Db2 コマンドを発行するトランザクションについて、以下の点に留意してください。

- CICS Db2 接続機能コマンドは、CICS と Db2 間の接続上で作動し、それらは完全に CICS 内で実行されます。 Db2 コマンドは Db2 自体の中で作動し、それらは Db2 に差し向けられます。 Db2 コマンドを CICS Db2 接続機能コマンドから区別するために、ハイフン (-) 文字を Db2 コマンドと一緒に入力します。
- DSNC トランザクションへのアクセス権限がある場合、CICS は、すべての CICS Db2 接続機能コマンドと Db2 コマンドを発行することを許可します。
- 個々の CICS Db2 接続機能コマンドと Db2 コマンドを実行するための別個のトランザクションを定義した場合は、別の CICS ユーザーに、これらのトランザクション・コードのサブセットに対する権限と、それに伴ってコマンドのサブセットに対する権限を与えることができます。例えば、何人かのユーザーに CICS Db2 接続機能コマンドを発行する権限を与えるけれども Db2 コマンドについて

はそうしない、ということも可能です。CICS Db2 用の CICS 提供トランザクション には、CICS Db2 接続機能コマンドと Db2 コマンドに対して CICS が供給する、別個のトランザクション定義の名前があります。

CICS Db2 接続機能コマンドが Db2 に流れることはないので、それらは、これ以上のセキュリティー検査の対象にはなりません。それらは CICS トランザクション接続セキュリティーのみによって保護されます。ただし、Db2 コマンド、およびデータを取得するために Db2 にアクセスする CICS トランザクションは、以下のように Db2 のセキュリティー・メカニズムによるそれ以降の段階のセキュリティー検査の対象になります。

- トランザクションが Db2 にサインオンするときには、有効な許可 ID を Db2 に提供しなければなりません。許可 ID は RACF によって、または同等の外部セキュリティー・マネージャーによって検査されます。
- トランザクションは Db2 コマンドを発行したり、Db2 データにアクセスしたりするので、トランザクションが提供した許可 ID には、これらのアクションを Db2 内で実行する権限が必要です。 Db2 では、GRANT ステートメントを使用して、アクションを実行する権限を許可 ID に与えることができます。

加えて、CICS 領域自体、Db2 サブシステムに接続する権限が必要です。

『CICS 領域用および CICS トランザクション用に、許可 ID を Db2 に提供する』 には、Db2 サブシステムに接続するために CICS 領域に権限を与える方法と、トランザクションに有効な許可 ID を提供する方法が記されています。

419 ページの『ユーザーに対する Db2 内のリソースへのアクセス許可』 には、トランザクションが Db2 に提供した許可 ID に権限付与する方法が記されています。

CICS 領域用および CICS トランザクション用に、許可 ID を Db2 に提供する

CICS には、Db2 に許可 ID を提供する必要のある 2 つのタイプのプロセスがあります。その 1 つは CICS 領域と Db2 の間の全体的な接続で、もう 1 つは Db2 へのスレッドを獲得する CICS トランザクションです。

このタスクについて

セキュリティーの目的で、Db2 では、データへのあらゆる形式のアクセスを表すために「プロセス」という語を使用します。これは、ユーザーが直接 Db2 と対話するものと、CICS を含む他のプログラム経由でユーザーが Db2 と対話するもののどちらかになります。 Db2 に接続するプロセス、またはサインオンするプロセスは、Db2 アドレス・スペースのセキュリティー検査に使用できる許可 ID という名前の短い Db2 ID を、1 つ以上提供する必要があります。 すべてのプロセスは 1 次許可 ID を提供する必要があり、オプションとして 2 次許可 ID を 1 つ以上提供することもできます。 Db2 特権および権限を 1 次または 2 次の許可 ID に付与することができます。例えばユーザーは、2 次許可 ID を使用して表を作成することができます。すると、その表はその 2 次許可 ID によって所有されます。同じ 2 次許可 ID を Db2 に提供する他のすべてのユーザーは、その表に対して関連した特権を持ちます。ユーザーから特権を取り去るために、管理者はそのユーザーをその許可 ID から切断することができます。

CICS には、Db2 に許可 ID を提供する必要のある、以下の 2 つのタイプのプロセスがあります。

- CICS 領域と Db2 との全体的な接続。これは CICS Db2 接続機能によって作成されます。このプロセスは、Db2 の接続処理を行って Db2 に許可 ID を提供する必要があります。
- Db2 へのスレッドを獲得する CICS トランザクション。これらは例えば、Db2 データベースからデータを取り出すトランザクション、あるいは Db2 コマンドを発行する DSNC トランザクションです。それぞれの CICS トランザクションごとに、Db2 から見える実際のプロセスはスレッド TCB です。これは、Db2 に入れるトランザクションのスレッドを制御するために CICS が使用するものです。これらのプロセスは、Db2 のサインオン処理を行って Db2 に許可 ID を提供する必要があります。

接続処理およびサインオン処理の間に、Db2 は、Db2 アドレス・スペースで使用するプロセスのために、1 次および 2 次の許可 ID を設定します。デフォルトで Db2 は、プロセスが提供した許可 ID を使用します。しかし、接続処理とサインオン処理の両方には出口ルーチンが関与しており、これらの出口ルーチンは、1 次および 2 次の許可 ID の設定にユーザーが影響を与えることを可能にします。 Db2 には、デフォルトの接続出口ルーチンとデフォルトのサインオン出口ルーチンがあります。これらを独自の出口ルーチンに置き換えることができます。これを支援するために、サンプルの接続出口ルーチンとサインオン出口ルーチンが Db2 に付属しています。

CICS 領域用の許可 ID を Db2 に提供する

CICS は、1 次許可 ID および 1 つ以上の 2 次許可 ID を Db2 に提供します。

このタスクについて

CICS Db2 接続機能で CICS 領域と Db2 の間の全体的な接続を作成する場合、そのプロセスには Db2 の接続処理が含まれます。 CICS 領域は、以下を提供することができます。

- 1 次許可 ID。 1 次許可 ID は、Db2 での CICS 領域の 1 次 ID になります。CICS 領域と Db2 の間の接続の場合、Db2 の接続処理に最初に渡される 1 次許可 ID を選択することはできません。これは、CICS 領域のユーザー ID です。ただし、独自の接続出口ルーチンを作成することで、接続処理中に Db2 が設定する 1 次 ID を変更できます。 RACF または同等の外部セキュリティー・マネージャーがアクティブである場合、それに対して CICS 領域のユーザー ID が定義されている必要があります。 412 ページの『CICS 領域用の 1 次許可 ID を提供する』は、CICS 領域に対して指定できる 1 次許可 ID について説明しています。
- 1 つ以上の 2 次許可 ID。 RACF グループの名前、またはグループのリストを、CICS 領域の 2 次許可 ID として使用できます。これを実行する場合、デフォルトの Db2 接続出口ルーチン DSN3@ATH (1 次許可 ID を Db2 に渡すことのみを行う) を置き換える必要があります。サンプル Db2 接続出口ルーチン DSN3SATH は、RACF グループの名前を 2 次許可 ID として Db2 に渡します。あるいは、CICS 領域の 2 次 ID を設定する、独自の接続出口ルーチンを作成できます。 412 ページの『CICS 領域用の 2 次許可 ID を提供する』は、CICS 領域の 2 次許可 ID のセットアップ方法を説明しています。

CICS 領域用の 1 次許可 ID を提供する

Db2 に渡される 1 次許可 ID は、CICS が開始タスク、開始ジョブ、ジョブのいずれとして稼働しているかによって異なります。

このタスクについて

CICS が Db2 から要求する接続タイプは、単一アドレス・スペース・サブシステム (SASS) です。CICS 領域と Db2 の間の接続の場合、最初に Db2 の接続処理に渡される 1 次許可 ID を選択することはできません。 CICS 領域の 1 次許可 ID として Db2 に渡される ID は次のいずれかです。

- CICS が開始タスクとして稼働している場合は、RACF 開始済みプロシージャー・テーブル、ICHRIN03 から取得されたユーザー ID。
- CICS が開始ジョブとして稼働している場合は、STARTED 一般リソース・クラス・プロファイルの STDATA セグメントのユーザー・パラメーター。
- CICS がジョブとして稼働している場合は、JOB カードの USER パラメーターで指定されたユーザー ID。

CICS 領域が使用する可能性があるユーザー ID は、RACF または同等の外部セキュリティー・マネージャー (外部セキュリティー・マネージャーがアクティブの場合) に対して定義する必要があります。ユーザー ID を RACF に対して USER プロファイルとして定義します。これを RESOURCE プロファイルとして定義するだけでは十分ではありません。

CICS 領域のユーザー ID を RACF に対して定義したら、以下のようにして Db2 へのアクセスを許可します。

1. 接続のタイプが単一アドレス・スペース・サブシステム (SASS) である Db2 サブシステムのプロファイルを RACF クラス DSNR に定義します。例えば、次の RACF コマンドでは、Db2 サブシステム DB2A への SASS 接続用のプロファイルをクラス DSNR に作成します。

   ```
   RDEFINE DSNR (DB2A.SASS) OWNER(DB2OWNER)
   ```

2. CICS 領域のユーザー ID が Db2 サブシステムにアクセスすることを許可します。例えば、次の RACF コマンドでは、ユーザー ID が CICSHA11 の CICS 領域に Db2 サブシステム DB2A への接続を許可します。

   ```
   PERMIT DB2A.SASS CLASS(DSNR) ID(CICSHA11) ACCESS(READ)
   ```

Db2 の接続出口ルーチンは、CICS 領域によって提供された 1 次許可 ID (ユーザー ID) を取得し、それを CICS 領域の 1 次 ID として Db2 に設定します。デフォルトの Db2 接続出口ルーチン DSN3@ATH とサンプルの Db2 接続出口ルーチン DSN3SATH はどちらもこのように動作します。独自の接続出口ルーチンを作成することによって、Db2 が設定する 1 次 ID を変更することができます。サンプル接続出口ルーチン、および出口ルーチンの作成について詳しくは、Db2 for z/OS 製品資料内の『Db2 の保護』を参照してください。しかし、CICS 領域の 2 次許可 ID を提供し、1 次許可 ID ではなく、それらの 2 次許可 ID に基づいて CICS 領域に権限を付与する方がより簡単です。

CICS 領域用の 2 次許可 ID を提供する

CICS 領域が Db2 に接続するとき、1 次許可 ID に加えて、1 つ以上の 2 次許可 ID を Db2 に提供できます。CICS 領域が接続されている RACF グループまたは

グループのリストの名前を 2 次許可 ID として使用できます。これにより、CICS 領域ごとに考えられるすべての 1 次許可 ID に Db2 特権を付与する代わりに、RACF グループに Db2 特権および権限を付与してから、複数の CICS 領域を同じグループに接続できます。

このタスクについて

CICS 領域が接続されている RACF グループまたはグループのリストの名前を Db2 に 2 次許可 ID として提供するには、以下の手順を実行します。

手順

1. CICS が RACF を使用するようにするために、CICS 領域のシステム初期設定パラメーターとして SEC=YES を指定します。
2. CICS 領域を適切な RACF グループまたはグループのリストに接続します。RACF グループ・プロファイルを参照してください。
3. デフォルトの Db2 接続出口ルーチン DSN3@ATH を置き換えます。 この出口ルーチンは接続処理中に駆動されます。デフォルトの接続出口ルーチンは 2 次許可 IDをサポートしないため、それをサンプルのサインオン出口ルーチンである DSN3SATH に置き換える必要があります。これは、Db2 に付属するものか、またはユーザー独自のルーチンです。DSN3SATH は Db2 SDSNSAMP ライブラリー内のソースの形で出荷されるもので、それをベースにして独自のルーチンを作成できます。 DSN3SATH は、CICS 領域が接続されている RACF グループの名前を 2 次許可 ID として Db2 に渡します。「RACF グループ・リスト」オプションがアクティブである場合、DSN3SATH は、CICS 領域が接続されているすべてのグループの名前を取得して、それらを 2 次許可 ID として使用します。「RACF グループ・リスト」オプションがアクティブでない場合、 DSN3SATH は、CICS 領域の現在接続されているグループの名前を唯一の 2 次許可 ID として使用します。

タスクの結果

CICS 領域が Db2 に接続するときに、サンプルの接続出口ルーチンは CICS 領域の 1 次許可 ID (領域のユーザー ID) を 1 次 ID として設定し、CICS 領域が接続されている RACF グループの名前を 2 次 ID として設定します。

次のタスク

RACF グループの名前を 2 次許可 ID として Db2 に提供する代わりの方法として、CICS 領域の 2 次許可 ID を設定する独自の接続出口ルーチンを作成することもできます。接続出口ルーチンの作成については、Db2 for z/OS 製品資料内の『Db2 の保護』を参照してください。

CICS トランザクション用の許可 ID を Db2 に提供する

CICS は Db2 に許可 ID を渡すことができるため、CICS は RACF を使用している必要があります。また、SIT で SEC=YES が指定されている必要があります。これは、CICS が RACF アクセス制御環境エレメント (ACEE) を Db2 に渡す必要があるためです。

このタスクについて

CICS トランザクションのスレッド TCB は、Db2 にサインオンして Db2 のサインオン処理を実行すると、以下を提供することができます。

- 1 次許可 ID。CICS トランザクションの場合、1 次許可 ID を選択できます。これは CICS ユーザーのユーザー ID またはオペレーター ID、端末 ID、トランザクション ID のいずれか、または指定した ID にすることができます。1 次許可 IDとして使用される ID は、DB2ENTRY 定義 (エントリー・スレッドの場合) または DB2CONN 定義 (プール・スレッドおよびコマンド・スレッドの場合) の属性によって決定されます。 415 ページの『CICS トランザクション用の 1 次許可 ID を提供する』は、CICS トランザクションの 1 次許可 ID を選択する方法を説明しています。
- 1 つ以上の 2 次許可 ID。RACF グループまたはグループのリストの名前を 2 次許可 ID として使用できます。これには、個々の CICS ユーザーにそれぞれ Db2 特権および権限を付与するのではなく、RACF グループにそれらを付与できるという利点があります。2 次許可 IDを使用するには、DB2ENTRY 定義の AUTHTYPE 属性 (エントリー・スレッドの場合)、あるいは DB2CONN 定義の AUTHTYPE 属性または COMAUTHTYPE 属性 (プール・スレッドまたはコマンド・スレッドの場合) を使用して、GROUP オプションを指定します。また、デフォルトの Db2 サインオン出口ルーチン DSN3@SGN を置き換える必要があります。デフォルトのルーチンが 2 次許可 ID を Db2 に渡すことはないからです。GROUP オプションを指定する際に、1 次許可 ID がトランザクションに関連付けられた CICS ユーザーのユーザー ID として自動的に定義されます。418 ページの『CICS トランザクション用に 2 次許可 ID を提供する』では、2 次許可 ID のセットアップ方法および使用方法を説明しています。

CICS トランザクションが Db2 に提供する許可 ID を選択する際の主要な考慮事項は、Db2 アドレス・スペースでのセキュリティー検査用に選択したセキュリティー・メカニズムです。このセキュリティー検査は、Db2 コマンド、計画、および動的 SQL へのアクセスを対象とします。このセキュリティー検査が以下によって実行されることを選択できます。

- Db2 内部セキュリティー。
- RACF または同等の外部セキュリティー・マネージャー。
- 一部は Db2、一部は RACF。

Db2 アドレス・スペース内の一部またはすべてのセキュリティー検査に RACF を使用している場合、Db2 にサインオンする CICS トランザクションは、以下のいずれかの方法で許可 ID を提供する必要があります。

- スレッド (DB2ENTRY または DB2CONN) の適切な定義で AUTHTYPE(USERID) または COMAUTHTYPE(USERID) を指定して、トランザクションに関連付けられた CICS ユーザーのユーザー ID を 1 次許可 ID として Db2 に提供します。
- スレッド (DB2ENTRY または DB2CONN) の適切な定義で AUTHTYPE(GROUP) または COMAUTHTYPE(GROUP) を指定して、トランザクションに関連付けられた CICS ユーザーのユーザー ID を 1 次許可 ID として、RACF グループまたはグループのリストの名前を 2 次許可 ID として Db2 に提供します。

- スレッド (DB2ENTRY または DB2CONN) の適切な定義で AUTHTYPE(SIGN) を指定し、DB2CONN の SIGNID 属性で CICS 領域ユーザー ID を指定して、 CICS 領域 ID を 1 次許可 ID として Db2 に提供します。

CICS 領域の RACF アクセス制御環境エレメント (ACEE) が、CICS Db2 接続機能に影響を与えるような方法で変更される場合、サインオンが行われるまで Db2 は変更を認識しません。CEMT コマンドまたは EXEC CICS SET DB2CONN SECURITY(REBUILD) コマンドを使用して、スレッドが次回再利用されるとき、またはスレッドが既にサインオン済みの TCB で作成されるときに、CICS Db2 接続機能が Db2 サインオンを発行するように指定できます。これにより、Db2 にセキュリティー変更を認識させることができます。

CICS トランザクション用の 1 次許可 ID を提供する

CICS トランザクションのスレッド TCB が Db2 にサインオンするとき、Db2 に 1 次許可 ID を提供する必要があります。トランザクションがその 1 次許可 ID として使用する ID は、トランザクションが Db2 へのアクセスに使用するスレッドのリソース定義内の属性によって決定されます。

このタスクについて

これは、同じタイプのスレッド (同じタイプのエントリー・スレッド、プール・スレッド、またはコマンド・スレッドのいずれか) を使用するすべてのトランザクションが、同じタイプの 1 次許可 ID を使用する必要があることを意味します。各 CICS 領域で、以下に対して 1 次許可 ID を設定する必要があります。

- DB2ENTRY 定義を使用する、各タイプのエントリー・スレッド。
- DB2CONN 定義を使用するプール・スレッド。
- DB2CONN 定義を使用するコマンド・スレッド (DSNC トランザクションに使用される)。

1 次許可 ID の設定を開始する前に、それを実行する権限があることを確認してください。 DB2CONN 定義または DB2ENTRY 定義を変更するための権限に加え、CICS 領域に対して代理ユーザー検査が強制される場合 (つまり、システム初期設定パラメーター XUSER が YES に設定されている場合)、Db2 許可 ID が関係する操作を実行するための特殊権限を取得する必要があります。これらの操作は、DB2ENTRY 定義または DB2CONN 定義上の AUTHID、COMAUTHID、AUTHTYPE、または COMAUTHTYPE の各属性の変更、および DB2CONN 定義上の SIGNID 属性の変更です。 406 ページの『代理セキュリティーおよび AUTHTYPE セキュリティーを使用して、CICS が DB2 に提供する許可 ID へのアクセスを制御する』は、これらの操作を実行する権限をユーザーに付与する方法を説明しています。

特定のタイプのスレッドに 1 次許可 ID を設定するには、次の 2 とおりの方法があります。

1. DB2ENTRY 定義の AUTHID 属性 (エントリー・スレッドの場合)、または DB2CONN 定義の AUTHID 属性あるいは COMAUTHID 属性 (プール・スレッドまたはコマンド・スレッドの場合) を使用して、1 次許可 ID を指定します。例えば、`AUTHID=test2` を定義できます。この場合、CICS Db2 接続機能は、文字 `TEST2` を 1 次許可 ID として Db2 に渡します。

AUTHID および COMAUTHID の使用は、2 次許可 ID の使用を許可するものではありません。さらにその使用は、Db2 アドレス・スペースのセキュリティー検査で、RACF または同等の外部セキュリティー・マネージャーの使用と両立しません。

2. DB2ENTRY 定義の AUTHTYPE 属性 (エントリー・スレッドの場合)、または DB2CONN 定義の AUTHTYPE 属性あるいは COMAUTHTYPE 属性 (プール・スレッドまたはコマンド・スレッドの場合) を使用して、トランザクションに関連する既存の ID を 1 次許可 ID として使用するように CICS に指示します。この ID は、CICS ユーザー ID、オペレーター ID、端末 ID、またはトランザクション ID でも構いません。あるいは、CICS 領域の DB2CONN 定義で指定した ID でも構いません。

   AUTHTYPE または COMAUTHTYPE の使用は、USERID または GROUP オプションを使用する場合、Db2 アドレス・スペースのセキュリティー検査での RACF (または同等の外部セキュリティー・マネージャー) の使用と両立します。また、GROUP オプションを使用する場合、2 次許可 ID の使用と両立します。

1 次許可 ID を決定する 2 とおりの方法は相互排他的です。AUTHID と AUTHTYPE の両方、および COMAUTHID と COMAUTHTYPE の両方を同じリソース定義で指定することはできません。

セキュリティー・マネージャーが Db2 サブシステムに対してアクティブである場合、1 次許可 ID として選択したすべての ID は、RACF または同等の外部セキュリティー・マネージャーに対して定義する必要があることに注意してください。RACF に対して、1 次許可 ID は、単に RESOURCE プロファイルとして (例えば、端末またはトランザクションとして) ではなく、RACF USER プロファイルとして定義する必要があります。

DB2CONN リソースおよびDB2ENTRY リソースの説明に従って、DB2CONN 定義および DB2ENTRY 定義をセットアップまたは変更します。 AUTHTYPE 属性または COMAUTHTYPE 属性を使用してスレッド・タイプのための 1 次許可 ID を決定する場合は、 417 ページの表 45 を使用して、必要な許可 ID の提供と必要な機能のサポートを行うオプションを特定します。考慮すべきキーポイントは、次のとおりです。

- 1 次許可 ID だけでなく 2 次許可 ID も Db2 に提供する場合は、GROUP オプションを選択する必要があります。 GROUP オプションを指定すると、1 次許可 ID が CICS ユーザー ID として自動的に定義されますが、セキュリティー検査は 2 次許可 ID に基づいたものにできます。
- Db2 アドレス・スペースでのセキュリティー検査に RACF を使用する場合、GROUP オプションまたは USERID オプションのいずれかを選択する必要があります。 RACF アクセス制御環境エレメント (ACEE) を Db2 に渡すことができるのはこれらのオプションのみであり、セキュリティー検査に RACF を使用する場合は必須になります。
- 許可 ID の選択による、パフォーマンスと保守への影響を考慮してください。これについては、パフォーマンスおよびメンテナンスのための許可 ID の選択に概説されています。 USERID、OPID、TERM、TX、または GROUP オプションを使用すると、より多くの許可 ID にアクセス権を付与する必要があるため、サインオン処理がさらに頻繁に実行され、保守にもより多くの時間がかかります。

SIGN オプションを使用する場合、または AUTHTYPE 属性の代わりに AUTHID 属性を使用する場合、サインオン処理は減り、保守の複雑さは低減されます。ただし、標準許可 ID を使用すると、Db2 のセキュリティー検査の精度は低下します。

- 許可 ID の選択による、アカウンティングへの影響を考慮してください。許可 ID は、各 Db2 アカウンティング・レコードで使用されます。アカウンティングの観点から、最も詳細な情報は USERID、OPID、GROUP、または TERM を使用する場合に取得されます。ただし、ACCOUNTREC の指定に応じて、どのようなケースでも個別のユーザー・レベルでアカウンティングを行うということは不可能である場合があります。 CICS Db2 環境でのアカウンティングの詳細については、『モニター』の『CICS Db2 環境のアカウンティングとモニター』を参照してください。

表 45 は、AUTHTYPE 属性または COMAUTHTYPE 属性の各オプションを選択するときに CICS Db2 接続機能が Db2 に渡す、1 次許可 ID を示しています。

表 *45. AUTHTYPE* 属性および *COMAUTHTYPE* 属性で使用可能なオプション

オプション	**Db2** に渡される **1** 次許可 **ID**	**Db2** に対する **RACF** 検査のサポート	**2** 次許可 **ID** のサポート
USERID	RACF に定義されて CICS サインオンで使用される、CICS トランザクションに関連付けられたユーザー ID	はい	いいえ
OPID	RACF ユーザー・プロファイルの CICS セグメントで定義された、ユーザーの CICS オペレーター ID	いいえ	いいえ
SIGN	CICS 領域に対する DB2CONN 定義の SIGNID 属性で指定された ID。デフォルトは CICS 領域のアプリケーション ID	はい (DB2CONN リソースの SIGNID 属性が CICS 領域ユーザー ID と一致する場合)。	いいえ
TERM	トランザクションと関連付けられた端末の端末 ID	いいえ	いいえ
TX	トランザクション ID	いいえ	いいえ
GROUP	CICS サインオンで使用される、ユーザーの CICS RACF ユーザー ID	はい	はい

CICS トランザクションに 2 次許可 ID を提供する計画がない場合は、デフォルトの Db2 サインオン出口ルーチン DSN3@SGN を置き換える必要はありません。デフォルトのサインオン出口ルーチンは、1 次許可 ID を処理します。ただし、接続先の Db2 サブシステムは、他の何らかの理由で別のサインオン出口ルーチンを使用する可能性があります。 Db2 サブシステムがサンプル・サインオン出口ルーチン DSN3SSGN を使用する場合、以下のすべての条件が当てはまれば、DSN3SSGN への変更が必要になる可能性があります。

- GROUP ではなく AUTHID オプションまたは AUTHTYPE オプションを選択している。
- グループ処理の RACF リストがアクティブである。

- 1 次許可 ID が RACF に定義されていないトランザクションがある。

このような場合、サンプル・サインオン出口ルーチンに加える必要のある変更については、Db2 for z/OS 製品資料内の『Db2 の保護』を参照してください。

CICS トランザクション用に 2 次許可 ID を提供する

CICS トランザクションに属するスレッド TCB が Db2 にサインオンするときに、1 次許可 ID に加えて、1 つ以上の 2 次許可 ID を Db2 に提供することができます。

このタスクについて

CICS ユーザーの RACF グループの名前、またはグループのリストを、2 次許可 ID として Db2 に提供できます。これには、個々の CICS ユーザーにそれぞれ Db2 特権および権限を付与するよりも、RACF グループに権限付与できるという利点があります。その後 CICS ユーザーを必要に応じて RACF グループに接続したりグループから除去したりすることができます。 RACF グループ・プロファイルでは、ユーザーを RACF グループに配置する方法を説明しています。

CICS トランザクション用の 2 次許可 ID を Db2 に提供できるのは、DB2ENTRY 定義内の AUTHTYPE 属性 (エントリー・スレッドの場合)、あるいは DB2CONN 定義内の AUTHTYPE または COMAUTHTYPE 属性 (プール・スレッドまたはコマンド・スレッドの場合) に GROUP オプションを指定した場合だけです。AUTHTYPE または COMAUTHTYPE に他の何らかのオプションを指定した場合は、2 次許可 ID はブランクに設定されます。 GROUP オプションを指定するときは、スレッド・タイプに 1 次許可 ID を選択することはできません。それは自動的に、トランザクションに関連付けられた CICS ユーザーのユーザー ID として定義されます。代わりに、2 次許可 ID をセキュリティー検査のベースにしてください。

ユーザーの RACF グループの名前を Db2 に 2 次許可 ID として提供するには、以下のステップ全体を実行します。

1. CICS が RACF を使用するようにするため、CICS 領域のシステム初期設定パラメーターとして SEC=YES を指定します。 CICS トランザクション・プロファイル名が接頭部付きで定義されている場合は、システム初期設定パラメーター SECPRFX=YES または SECPRFX=*prefix* も指定してください。
2. CICS 領域が MRO を使用している場合:
   a. 接続されたそれぞれの CICS 領域も RACF セキュリティーを使用していること (SEC=YES) を確認します。
   b. サインオン情報が TOR から AOR に伝搬されるようにするため、TOR の CONNECTION 定義に ATTACHSEC=IDENTIFY を指定します。
3. デフォルトの Db2 サインオン出口ルーチンである DSN3@SGN を置き換えます。このサインオン出口ルーチンはサインオン処理中に駆動されます。デフォルトのサインオン出口ルーチンは 2 次許可 IDをサポートしないので、それをサンプルのサインオン出口ルーチンである DSN3SSGN に置き換える必要があります。DSN3SSGN は Db2 に付属するものか、またはユーザー独自のルーチンです。 DSN3SSGN は Db2 SDSNSAMP ライブラリー内のソースの形で出荷されるもので、それをベースにして独自のルーチンを作成できます。サインオン出口、および出口ルーチンの作成については、Db2 for z/OS 製品資料内の

『Db2 の保護』を参照してください。「RACF グループ・リスト」オプションがアクティブでない場合、 DSN3SSGN は、現在接続されているグループの名前を 2 次許可 ID として渡します。「RACF グループ・リスト」オプションがアクティブである場合は、DSN3SSGN は、ユーザーが接続されているすべてのグループの名前を取得して、それらを 2 次許可 ID として Db2 に渡します。

4. DB2ENTRY 定義内の AUTHTYPE 属性 (エントリー・スレッドの場合)、または DB2CONN 定義の AUTHTYPE または COMAUTHTYPE 属性 (プール・スレッドまたはコマンド・スレッドの場合) を使用して、 2 次許可 ID を提供したいスレッドの各タイプごとに、GROUP オプションを許可タイプとして指定します。 CICS 領域に対して代理ユーザー検査が強制される場合 (つまり、システム初期設定パラメーター XUSER が YES に設定されている場合) には、Db2 許可 ID が関係する操作を実行するための特殊権限を取得する必要があります。これを行う方法については 406 ページの『代理セキュリティーおよび AUTHTYPE セキュリティーを使用して、CICS が DB2 に提供する許可 ID へのアクセスを制御する』を参照してください。

上記にリストされているすべてのステップを正常に完了した場合は、 GROUP オプションを付けて定義したスレッドのタイプを使って CICS トランザクションのスレッド TCB が Db2 にサインオンするときに、CICS ユーザーのユーザー ID が 1 次 許可 ID として Db2 に渡され、ユーザーの RACF グループまたはグループ・リストが 2 次許可 ID として Db2 に渡されます。すべてのステップを正常に完了していない場合は、CICS Db2 接続機能は、許可の失敗メッセージを出します。

RACF グループの名前を 2 次許可 ID として Db2 に提供する代わりの方法として、CICS トランザクションの 2 次許可 ID を設定する独自のサインオン出口ルーチンを書くこともできます。 Db2 for z/OS 製品資料内の『Db2 の保護』を参照してください。

ユーザーに対する Db2 内のリソースへのアクセス許可

ユーザーが CICS トランザクションから Db2 アドレス・スペースにアクセスすると、Db2 コマンドを発行したり計画を実行したりするための許可が必要になる場合があります。

このタスクについて

CICS ユーザーが Db2 コマンド発行や計画実行の際に必要とする Db2 リソースへのアクセスは、Db2 のセキュリティー・メカニズムによるセキュリティー検査の対象になります。このセキュリティー検査が以下によって実行されることを選択できます。

- Db2 内部セキュリティー。
- RACF、または同等の外部セキュリティー・マネージャー。
- 一部は Db2、一部は RACF。

Db2 アドレス・スペースでセキュリティー検査を実行するための RACF のセットアップについて詳しくは、Db2 for z/OS 製品資料内の『Db2 の保護』を参照してください。

Db2 アドレス・スペース内の一部またはすべてのセキュリティー検査に RACF を使用している場合は、Db2 にサインオンする CICS トランザクションが許可 ID を

提供する必要があることに注意してください。詳しくは、 413 ページの『CICS トランザクション用の許可 ID を Db2 に提供する』を参照してください。さらに、CICS は RACF を使用している必要もあります (SIT で SEC=YES を指定する必要があります)。これは、RACF を Db2 アドレス・スペース内のセキュリティー検査に使用する場合、CICS は RACF アクセス制御環境エレメント (ACEE) を Db2 に渡す必要があるためです。 CICS は RACF がアクティブである場合にのみ ACEE を生成でき、GROUP、SIGN、または USERID の各オプションで定義されたスレッドのみが ACEE を Db2 に渡すことができます。

ACEE は、Db2 に渡されると、Db2 出口の DSNX@XAC によって使用され、そこで RACF、IBM 以外の同等の外部セキュリティー・マネージャー、または Db2 内部セキュリティーをセキュリティー検査に使用するかどうかが決定されます。DSNX@XAC は、スレッドが Db2 にサインオンしているトランザクションが API 要求を発行すると駆動します。 DSNX@XAC を変更できます。詳しくは、Db2 for z/OS 製品資料内の『Db2 の保護』を参照してください。

Db2 または外部セキュリティー・マネージャーは、CICS トランザクションが、使用スレッドが Db2 にサインオンしたときに Db2 に提供した許可 ID を使用して、セキュリティー検査を実行します。許可 ID は、個々の CICS ユーザー (例えば、CICS ユーザーのユーザー ID や、ユーザーが接続されている RACF グループ) に関連しているか、トランザクション (例えば、端末 ID やトランザクション ID) に関連しているか、または CICS 領域全体に関連している場合があります。詳しくは、 410 ページの『CICS 領域用および CICS トランザクション用に、許可 ID を Db2 に提供する』を参照してください。

Db2 または外部セキュリティー・マネージャーは、Db2 内の関連アクションを実行する権限を、ユーザーが許可 ID に付与済みであることを検査します。 Db2 で GRANT ステートメントを使用することによって、この権限を許可 ID に付与できます。許可 ID に対する Db2 許可の付与、および取り消し方法については、Db2 for z/OS 製品資料内の『Db2 の保護』を参照してください。

Db2 コマンドへのユーザーのアクセスの制御

CICS ユーザーの場合、ユーザーが Db2 コマンドを発行する CICS トランザクションにアクセスしようとすると、Db2 コマンドに関連する最初のセキュリティー検査が CICS アドレス・スペースで実行されます。これは、DSNC トランザクション、または DFHD2CM1 を呼び出して個々の Db2 コマンドを実行するユーザー定義のトランザクションである可能性があります。

このタスクについて

409 ページの『Db2 関連の CICS トランザクションへのユーザー・アクセスの制御』では、CICS アドレス・スペースで Db2 コマンドを発行するトランザクションに対するユーザーのアクセスを制御する方法を説明しています。

ユーザーが CICS トランザクションで Db2 コマンドを発行すると、そのユーザーは Db2 のセキュリティー検査の対象にもなります。その検査で、Db2 でコマンドを発行する許可があるかどうかが検証されます。このセキュリティー検査では、トランザクションによって CICS から渡された許可 ID (1 次または 2 次) が使用されます。 410 ページの『CICS 領域用および CICS トランザクション用に、許可 ID を Db2 に提供する』では、これらの許可 ID を選択して、Db2 に提供する方

法を説明しています。DFHD2CM1 を使用して Db2 コマンドを発行するトランザクションの場合、許可 ID は CICS 領域の DB2CONN 定義の COMAUTHID 属性または COMAUTHTYPE 属性によって設定されます。 Db2 コマンドを発行するその他のアプリケーションの場合、許可 ID は、トランザクションによって使用されるスレッドのタイプ (プール・スレッドまたはエントリー・スレッド) に対する CICS 領域のリソース定義の AUTHID 属性または AUTHTYPE 属性によって設定されます。これらの属性は、そのタイプのスレッドを使用するトランザクションによって Db2 に渡される許可 ID (許可 ID のタイプ) を制御します。

したがって、Db2 コマンドは 2 つのセキュリティー検査の対象になります。1 つは CICS アドレス・スペース、もう 1 つは Db2 アドレス・スペースで行われます。 図 28 は、このプロセスを示しています。

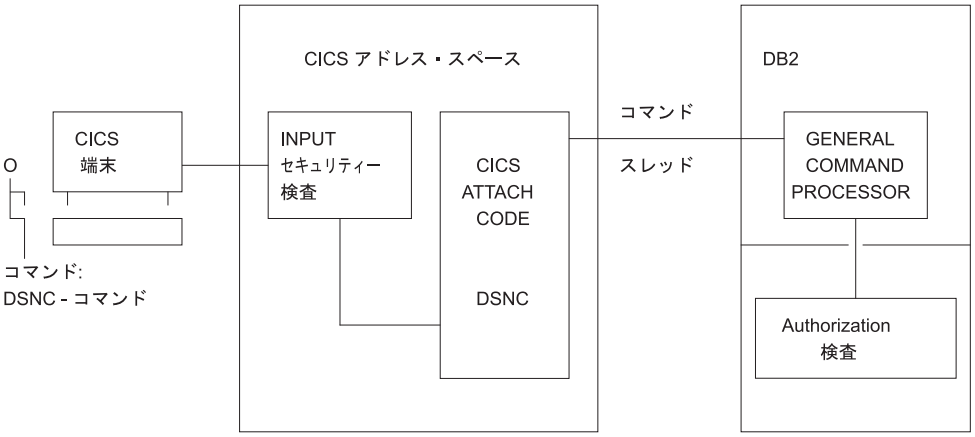


図 *28. Db2* コマンドのセキュリティー・メカニズム

ほとんどの場合、Db2 コマンドの実行が許可されているのは、限られた数のユーザーだけです。有用な解決策は、DB2CONN 定義で COMAUTHTYPE(USERID) を指定することです。そうすると、Db2 での許可 ID として 8 バイトの CICS ユーザー ID に解決されます。この方法を使用すると、異なる Db2 特権を CICS ユーザー ID に明示的に与えることができます。例えば、GRANT DISPLAY を使用して、-DIS コマンドのみを使用する権限を特定の CICS ユーザー ID に与えることができます。

ユーザーに Db2 コマンドを発行する権限を与えるには、GRANT コマンドを使用して、トランザクションによって CICS から渡された許可 ID に Db2 コマンド特権を付与します。許可 ID に対する Db2 許可の付与、および取り消し方法については、Db2 for z/OS 製品資料内の『Db2 の保護』を参照してください。

計画へのユーザー・アクセスの制御

ユーザーが Db2 の計画にアクセスする前に、いくつかの検査が行われます。この検査は CICS アドレス・スペースで始まり、その後 Db2 に要求が渡されてさらに検査が行われます。

このタスクについて

Db2 コマンドでは、CICS が、計画を実行するトランザクションへのアクセスがユーザーに許可されているかどうか確認するときに、計画へのユーザー・アクセスに

対する最初のセキュリティー検査が CICS アドレス・スペースで行われます。 Db2 が、トランザクションによって提供される許可 ID に計画の実行許可があるかどうか確認するときに、2 番目のセキュリティー検査が Db2 アドレス・スペースで行われます。 図 29はこの処理を示しています。

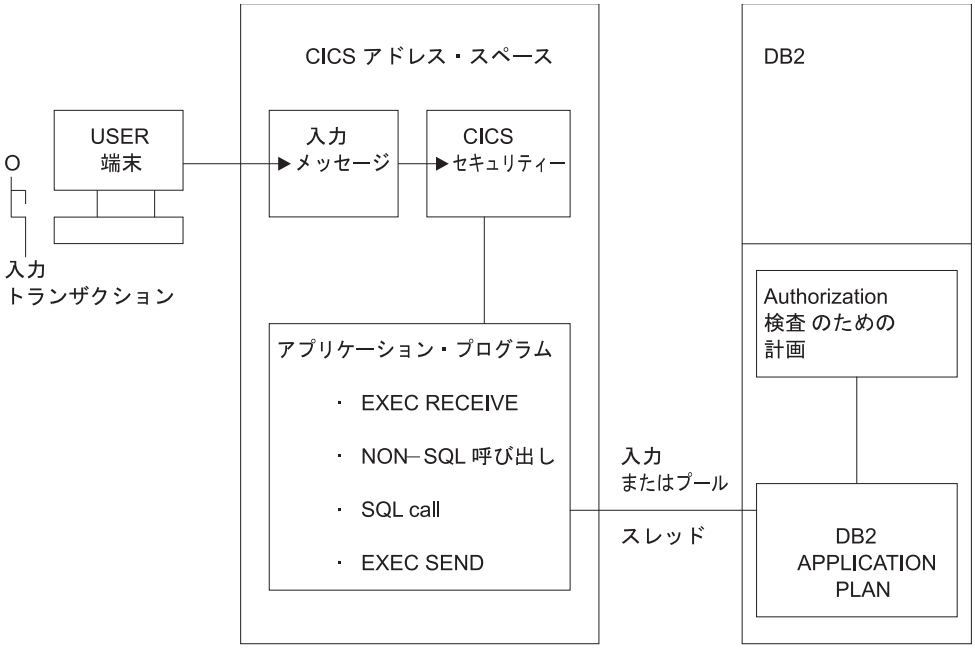


図 *29.* 計画の実行におけるセキュリティー・メカニズム

ユーザーに計画の実行許可を与えるには、GRANT コマンドを使用して、トランザクションによって CICS から渡された許可 ID に Db2 コマンド特権を付与します。許可 ID に対する Db2 許可の付与、および取り消し方法については、Db2 for z/OS 製品資料内の『Db2 の保護』を参照してください。

計画に動的 **SQL** が含まれる場合**:**

静的 SQL を使用する場合、計画のバインダーには、データへのアクセスに必要な特権がなければなりません。CICS から Db2 に渡される許可 ID には、計画を実行するための特権があれば十分です。

このタスクについて

ただし、計画に動的 SQL の使用が含まれている場合、CICS から Db2 へ渡される許可 ID には、関係するすべての Db2 リソース (計画とデータの両方) にアクセスするために必要な特権がなければなりません。例えば、AUTHTYPE(USERID) を指定する場合、CICS ユーザー ID には、動的 SQL に関係する Db2 リソースに対する Db2 特権を付与する必要があります。このユーザー ID が TSO ユーザー ID でもある場合は、SPUFI、QMF™、およびその他のユーティリティーから Db2 リソースに直接アクセスできます。

動的 SQL の使用に関係する計画をトランザクションが実行する場合に、Db2 特権の付与にあまり時間を費やしたくない場合は、以下の、許可 ID を Db2 に提供する方式のいずれかを使用することを検討してください。

- トランザクションによって使用されるスレッドの DB2ENTRY 定義の AUTHTYPE 属性で、SIGN オプションを使用します。これにより、トランザクションは、CICS 領域の DB2CONN 定義の SIGNID 属性に指定された 1 次許可 ID を持つことになります。 (この方式は、RACF が Db2 アドレス・スペースでのセキュリティー検査に使用される場合は適していません。)
- トランザクションによって使用されるスレッドの DB2ENTRY 定義の AUTHID 属性を使用して、標準許可 ID を指定します。動的 SQL にアクセスする必要があるすべてのトランザクションに対して、同じ許可 ID を使用します。 (この方式は、RACF が Db2 アドレス・スペースでのセキュリティー検査に使用される場合は適していません。)
- RACF グループを作成し、CICS ユーザーをこの RACF グループに接続します。トランザクションで使用されるスレッドの DB2ENTRY 定義の GROUP 属性を使用して、RACF グループが Db2 に渡されるセカンダリー ID の 1 つになるようにします。

それぞれのケースで、すべての動的 SQL に関係する Db2 リソースに対する Db2 特権を、単一の ID (DB2CONN 定義または AUTHID 属性からの標準許可 ID であるか、あるいは RACF グループの名前のいずれか) に付与できます。 410 ページの『CICS 領域用および CICS トランザクション用に、許可 ID を Db2 に提供する』は、これらのメソッドによって許可 ID を提供する方法を説明しています。

Db2 マルチレベル・セキュリティーおよび行レベル・セキュリティー

DB2® バージョン 8 では、マルチレベル・セキュリティーのサポートが導入されました。CICS はマルチレベル・セキュリティーのための固有のサポートを提供しませんが、構成が気になる場合には、マルチレベル・セキュリティー環境で CICS を使用できます。

マルチレベル・セキュリティーについて詳しくは、以下の資料を参照してください。

- Db2 for z/OS 製品資料内の『Db2 の保護』
- z/OS マルチレベル・セキュリティーの計画および Common Criteria
- IBM Redbooks: z/OS での DB2 の保護と MLS の実装

DB2 バージョン 8 以降で、マルチレベル・セキュリティーがデータに対して行レベル (行レベルのセキュリティー) で実装されている場合、RACF SECLABEL クラスがアクティブ化され、一連のセキュリティー・レベルがユーザーおよび Db2 表の行に対して定義されます。RACF オプションである SETR MLS と MLACTIVE は、アクティブである必要はありません。 Db2 行レベルのセキュリティーは、MVS システムの残りの部分に影響を与えることなく使用できます。

CICS は、このような方法で保護された Db2 行にアクセスできます。CICS の場合、Db2 行へのアクセスを必要とする CICS ユーザーの RACF ユーザー・プロファイルが、デフォルトの SECLABEL を含むように RACF に定義されていることを確認する必要があります。詳しくは、z/OS Security Server RACF セキュリティー管理者のガイドを参照してください。

CICS ユーザーが SIT に SEC=YES を指定して CICS 領域にサインオンすると、RACF はデフォルトの SECLABEL をユーザーの RACF アクセス制御環境エレメント (ACEE) に関連付けます。DB2ENTRY 定義 (プールが使用されている場合は DB2CONN 定義) は、AUTHTYPE=USERID または AUTHTYPE=GROUP を指定する必要があります。これにより、ACEE が今後のセキュリティー検査のために Db2 に渡されます。そのため、個々の CICS ユーザーは、関連付けられた SECLABEL を 1 つしか持つことができません。

PLT プログラムなどの非端末タスクまたはプログラムの場合、 PLTPIUSR システム初期設定パラメーターが指定されておらず、PLTPISEC=NONE システム初期設定パラメーターが指定されている場合、PLT プログラムは CICS 領域ユーザー ID で実行されます。この場合、デフォルトの SECLABEL を使用して CICS 領域ユーザー ID を定義する必要があります。トランザクションにさまざまな SECLABEL を定義する必要がある場合、異なる CICS 領域ユーザー ID および関連 SECLABEL を持つ別個の CICS 領域で、それぞれのトランザクションを実行する必要があります。

DBCTL のセキュリティー

CICS を DBCTL とともに使用する場合は、いくつかのセキュリティー機能を使用できます。

以下のオプションのセキュリティー機能を 1 つ以上使用できます。

- 『CICS による PSB 許可検査』
- DBCTL によるリソース・アクセス・セキュリティー検査
- DBCTL パスワード・セキュリティー検査

DBCTL および DBCTL パスワード・セキュリティー検査によるリソース・アクセス・セキュリティー検査について詳しくは、「IMS 製品資料内の『システム管理』」を参照してください。

IMS セキュリティーを使用して保護できるリソースのうち、考慮する必要があるのは、PSB、データベース、およびコマンドのみです。

CICS による PSB 許可検査

PSB スケジューリング時に、CICS はセキュリティー検査を呼び出して、端末ユーザーが PSB へのアクセスを許可されているかどうかを判別します。実際の検査は、外部セキュリティー・マネージャー (RACF または独自のセキュリティー・プログラムが可能) によって実行されます。

PSB スケジューリング要求は処理のために DBCTL に送信されますが、CICS は PSB 許可検査を実行します。独自のセキュリティー・プログラムの作成に関するプログラミング情報については、外部セキュリティー・マネージャーの起動を参照してください。

データ・セットのセキュリティー: 暗号化

z/OS データ・セット暗号化がサポートされている、CICS で使用するあらゆるデータ・セットを暗号化できます。そうしたデータには、CICS ファイル制御 API を介してアクセスされるユーザー・データ・セット、CICS 区画外一時データに使用される待機順次アクセス方式 (QSAM) データ・セット、CICS で使用される基本順次アクセス方式 (BSAM) データ・セット、暗号化の適切な候補となる CICS システム・データ・セットが含まれます。

CICS TS for z/OS の任意のサービス中リリースで、データ・セット暗号化を使用することができます。 z/OS データ・セット暗号化は、z/OS 2.2 (APAR OA50569 の PTF 適用済み) 以降のリリースでサポートされます。

暗号化データ・セットは、ストレージ管理サブシステム (SMS) で管理する必要があり、拡張フォーマットでなければなりません。暗号化データ・セットを作成するには、新規データ・セットを割り振るときに、そのデータ・セットに鍵ラベルを割り当てます。その鍵ラベルは、データの暗号化または復号に使用される、Integrated Cryptographic Service Facility (ICSF) 暗号鍵データ・セット (CKDS) の AES-256 ビット暗号鍵を指している必要があります。鍵ラベルは機密情報ではありませんが、鍵ラベルによって識別する暗号鍵は機密情報です。

CICS ユーザー・データ・セットの暗号化サポート

CICS に定義されるユーザー・データ・セットにおける暗号化サポートには、基本 VSAM および VSAM レコード・レベル共用 (RLS) を介してアクセスされる、キー順データ・セット (KSDS)、入力順データ・セット (ESDS)、相対レコード・データ・セット (RRDS)、および可変相対レコード・データ・セット (VRRDS) が含まれます。また、共用データ・テーブルまたはカップリング・ファシリティー・データ・テーブルで使用されるバッキング VSAM キー順データ・セットでも、暗号化がサポートされます。

CICS システム・データ・セットの暗号化サポート

暗号化がサポートされている、あらゆる CICS システム・データ・セットを暗号化できます。しかし、システム・データ・セットの中には暗号化に適しているものと、適していないものがあります。

- 機密データを含む可能性がある データ・セットは、暗号化に適しています。機密データを含む可能性があるのはどのデータ・セットか、またそれらを暗号化するかどうかを判断する必要があります。
- 機密データを含まないデータ・セットと、含む可能性が低いデータ・セットは、暗号化に適していません。

以下の表では、暗号化が可能な CICS システム・データ・セットをリストし、暗号化に適しているかどうかを示します。すべてを暗号化する方式を採用する場合は、この表にリストされているすべてのデータ・セット・タイプを暗号化することができます。

表 46. 暗号化に適しているシステム・データ・セット

CICS システム・データ・セット	暗号化に適しているか**?**	特別な考慮事項
一時記憶域データ・セット (DFHTEMP)	はい。機密データを含む可能性があります。	DFHTEMP は拡張フォーマットをサポートしますが、拡張アドレッシングをサポートしません。
区画内一時データ (DFHINTRA)	はい。機密データを含む可能性があります。	DFHINTRA は拡張フォーマットをサポートしますが、拡張アドレッシングをサポートしません。
区画外一時データ	はい。機密データを含む可能性があります。	区分データ・セット (PDS) では、暗号化がサポートされません。
補助トレース・データ・セット (DFHAUXT および DFHBUXT)	はい。診断で機密データを含む可能性があります。別の方法として、CONFDATA システム初期設定パラメーターを使用して、十分な保護が得られる可能性があります。	トレース・データが暗号化されていて、それを診断用に IBM に送信する必要がある場合は、CICS トレース・フォーマット設定か、または別の方法を使用して、暗号化解除されたデータを送信してください。
CICS ダンプ・データ・セット (DFHDMPA および DFHDMPB)	はい。診断で機密データを含む可能性があります。別の方法として、CONFDATA システム初期設定パラメーターを使用して、十分な保護が得られる可能性があります。	ダンプ・データが暗号化されていて、それを診断用に IBM に送信する必要がある場合は、CICS ダンプ・フォーマット設定か、または別の方法を使用して、暗号化解除されたデータを送信してください。 CICS ダンプ・データ・セットは拡張フォーマットをサポートしますが、拡張アドレッシングをサポートしません。
Doctemplate リソース	はい、機密データを含む可能性があります。	暗号化がサポートされるタイプのデータ・セットを使用していることが前提です。
静的配信に使用される URIMAP リソース	はい、機密データを含む可能性があります。	暗号化がサポートされるタイプのデータ・セットを使用していることが前提です。
BTS リポジトリー・データ・セットおよび BTS ローカル要求キュー (LRQ) データ・セット	いいえ。制御データのみが含まれます。	なし
グローバル・カタログ・データ・セットおよびローカル・カタログ・データ・セット (DFHGCD および DFHLCD)	いいえ。構成データのみが含まれます。	CICS 構成データが機密情報であると考えられる場合のみ、検討します。

表 46. 暗号化に適しているシステム・データ・セット (続き)

CICS システム・データ・セット	暗号化に適しているか**?**	特別な考慮事項
CICS システム定義データ・セット (DFHCSD)	いいえ。リソース構成に関する情報のみが含まれます。	リソース定義に機密情報が含まれると考えられる場合のみ、検討します。
CMAC メッセージ・データ・セット (DFHCMACD)	いいえ。メッセージの詳細のみが含まれます。	機密データを含むと考えられる、独自のメッセージを追加した場合のみ、検討します。
バンドル定義および Web サービスに使用する zFS ファイル	いいえ。構成情報のみが含まれます。	バンドル定義に機密情報が含まれると考えられる場合のみ、検討します。

関連情報:

データ・セットの暗号化

第 10 章 外部インターフェースのセキュリティー

EXCI のセキュリティー

CICS は、MVS クライアント・プログラムから受け取った要求に対して、複数の方法でセキュリティー検査を適用します。

MRO ログオン・セキュリティーおよびバインド時のセキュリティーの使用

CICS 領域間通信プログラムである DFHIRP は、IRP にログオンしようとする (特定の接続のみ) ユーザーまたは CICS 領域に接続しようとする (バインド時のセキュリティーとも呼ばれる) ユーザーに対して 2 つのセキュリティー検査を実行します。

このタスクについて

汎用 **EXCI** 接続**:** このセクションのログオン・セキュリティー検査に関する説明は、SPECIFIC として定義された EXCI 接続にのみ適用されます。MRO ログオン・セキュリティー検査は、汎用接続に対して実行されません。

MRO ログオン・セキュリティー検査および接続 (バインド時の) セキュリティー検査に関する限り、MVS クライアント・プログラムは別の CICS 領域とまったく同じように処理されます。これは、クライアント・プログラムが領域間通信プログラムにログオンすると、IRP はクライアント・プログラムの実行ユーザー ID に対してログオン・セキュリティー検査とバインド時のセキュリティー検査を実行することを意味します。このセクションの残りの部分では、このユーザー ID はバッチ領域のユーザー ID と呼ばれます。

クライアント・プログラムが IRP に正常にログオンし、ターゲット・サーバー領域に接続できるようにするには、最初にユーザー・プロファイルのバッチ領域のユーザー ID を RACF に対して定義します。バッチ領域のユーザー ID を RACF に対して定義すると、バッチ領域に適切なログオン許可とバインド時の許可を与えることができます。

手順

- ログオン許可

  バッチ領域のユーザー ID を DFHAPPL.*user_name* RACF FACILITY クラス・プロファイルに UPDATE 権限付きで許可します。プロファイル名の *user_name* の部分は、INITIALIZE_USER コマンドで定義されたユーザー名です。

  バッチ領域のユーザー ID を、IRP にログオンしている特定のユーザー ID の DFHAPPL プロファイルに許可できないと、Allocate_Pipe 処理は失敗し、RESPONSE(SYSTEM_ERROR) REASON(IRC_LOGON_FAILURE) が出されます。ログオン・セキュリティー検査が失敗すると、subreason フィールド 1 は 10 進数 204 を返します。

© Copyright IBM Corp. 1989, 2018

EXCI クライアント・プログラムの FACILITY クラス・プロファイルについては、『EXCI 領域の DFHAPPL FACILITY クラス・プロファイルの定義』を参照してください。

- バインド時の許可

  バッチ領域のユーザー ID をターゲット CICS サーバー領域の DFHAPPL.*applid* RACF FACILITY クラス・プロファイルに READ 権限付きで許可します。

  バッチ領域のユーザー ID を CICS サーバー領域の DFHAPPL.*applid* プロファイルに許可できないと、Open_Pipe 処理は失敗し、RESPONSE (SYSTEM_ERROR) REASON(IRC_CONNECT_FAILURE) が出されます。バインド時のセキュリティー検査が失敗すると、subreason フィールド 1 は 10 進数 176 を返します。

  MRO ログオン・セキュリティー検査およびバインド時のセキュリティー検査と、RACF DFHAPPL プロファイルの定義方法の例については、バインド・セキュリティーを参照してください。

EXCI 領域の DFHAPPL FACILITY クラス・プロファイルの定義

DFHAPPL プロファイル名の *user_name* の部分を定義するには、次のようにします。

- EXCI CALL インターフェースの場合、*user_name* は、INITIALIZE_USER コマンドの *user_name* パラメーターに指定する名前でなければなりません。

  クライアント・プログラムに複数のユーザー名に対する INITIALIZE_USER コマンドがある場合、そのプログラムで指定されたユーザー名ごとに、適切な許可を持つ FACILITY クラス・プロファイルを定義します。

  例えば、INITIALIZE_USER コマンドで定義されている *user_name* が DCEUSER1 の場合、次のように FACILITY クラスで DFHAPPL プロファイルを定義します。

  ```
  RDEFINE FACILITY (DFHAPPL.DCEUSER1) UACC(NONE)
  ```

  バッチ領域のユーザー ID が CLIENTA の場合、次のようにしてバッチ領域が IRP にログオンするのを許可します。

  ```
  PERMIT DFHAPPL.DCEUSER1  CLASS(FACILITY) ID(CLIENTA)
         ACCESS(UPDATE)
  ```

- EXEC CICS LINK コマンドの場合、*user_name* は外部 CICS インターフェースによって DFHXCEIP として事前設定されています。これには IRP ログオンの許可は不要です。EXEC CICS LINK インターフェースはログオン・セキュリティー検査が適用されない汎用接続を使用するからです。

リンク・セキュリティー

ターゲット CICS サーバー領域は、クライアント・プログラムからの要求に対してリンク・セキュリティー検査を実行します。

これらのセキュリティー検査は、トランザクション接続セキュリティー (ミラー・トランザクションを接続している場合)、およびサーバー・アプリケーション・プロ

グラム内のリソースおよびコマンドのセキュリティー検査を対象としています。CICS がこれらのセキュリティー検査に使用するリンク・ユーザー ID は、バッチ領域のユーザー ID です。

これらのリンク・セキュリティー検査が原因でセキュリティー障害が発生することがないようにするには、必要に応じて、リンク・ユーザー ID が以下のリソース・プロファイルに対して許可されていることを確認する必要があります。

- ミラー・トランザクションのプロファイル。CSMI (デフォルトの場合) または *transid* パラメーターで指定されたミラー・トランザクションのいずれか。これはトランザクション接続セキュリティー検査では必須です。
- CICS サーバー・アプリケーション・プログラムがアクセスするすべてのリソースのプロファイル。ファイル、キュー (一時データ・キューおよび一時ストレージ・キュー)、プログラムなど。これはリソース・セキュリティー検査では必須です。
- CICS サーバー・アプリケーション・プログラムによって発行される SPI コマンドの CICS コマンド・プロファイル。INQUIRE、SET、DISCARD など。これはコマンド・セキュリティー検査では必須です。

MRO リンク・セキュリティー検査については、 352 ページの『バインド・セキュリティー』を参照してください。

ユーザー・セキュリティー

ターゲット CICS サーバー領域は、DPL_ Request 呼び出しで渡されるユーザー ID に対してユーザー・セキュリティー検査を実行します。ユーザー・セキュリティー検査は、接続で ATTACHCSEC(IDENTIFY) を指定した場合のみ実行されます。

ユーザー・セキュリティーは、リンク・セキュリティーに加えて実行されます。

ユーザー・セキュリティーの場合、リンク・セキュリティーに対して行う許可に加えて、DPL_Request 呼び出しで指定されたユーザー ID も許可する必要があります。

EXEC CICS LINK コマンドでユーザー ID を指定する方法はないことに注意してください。この場合、外部 CICS インターフェースがバッチ領域のユーザー ID を渡します。したがって、接続定義で ATTACHSEC(IDENTIFY) が指定されている場合、ユーザー・セキュリティー検査は、バッチ領域のユーザー ID に対して実行されます。

注**:** 外部 CICS インターフェースの接続リソース定義で ATTACHSEC(IDENTIFY) が指定される場合、RACF または同等の外部セキュリティー・マネージャー (ESM) がインストールされておらずアクティブになっていない環境でサーバー・プログラムを実行すると、サーバー・プログラムは失敗し、ATCY 異常終了が出されます。

セキュリティーがアクティブになっていない外部 CICS インターフェースのサーバー・プログラムを実行する場合、ATTACHSEC(LOCAL) を指定する必要があります。

代理ユーザー検査

代理ユーザー検査は、バッチ領域のユーザー ID が別のユーザーの DPL 呼び出しを発行することを許可されている (つまり、DPL_Request 呼び出しで指定されたユーザー ID の代理として許可されている) ことを確認します。

SURROGCHK=YES (デフォルト) が EXCI オプション・テーブル DFHXCOPT で指定されている場合、EXCI クライアント・ジョブは代理ユーザー検査の対象となります。SURROGCHK=YES を指定する場合 (またはデフォルトにすることを許可する場合)、バッチ領域のユーザー ID を、すべての DPL_Request 呼び出しで指定されたユーザー ID の代理として許可します。これは、バッチ領域のユーザー ID には、SURROGAT 一般リソース・クラスの *userid*.DFHEXCI というプロファイルに対する READ 権限が必要であることを意味します (ここで、*userid* は DPL 呼び出しで指定されたユーザー ID です)。例えば、次のコマンドは DPL ユーザー ID の代理プロファイルを定義し、EXCI バッチ領域に READ 権限を付与します。

```
RDEFINE  SURROGAT dpl_userid.DFHEXCI UACC(NONE) OWNER(DPL_userid)
PERMIT   userid.DFHEXCI CLASS(SURROGAT) ID(batch_region_userid)
                    ACCESS(READ)
```

代理ユーザー検査が有効である (SURROGCHK=YES) が、ユーザー ID が DPL_Request 呼び出しで指定されていない場合、代理ユーザー検査は実行されません。DPL_Request 呼び出しのユーザー ID はデフォルトでバッチ領域のユーザー ID になるからです。この代理ユーザー検査のバイパスを正常に実行するには、DPL_Request 呼び出しでユーザー ID を正しく省略したことを確認します。EXCI 呼び出しパラメーターを省略するときに NULL ポインターを正しく指定する方法については、 EXCI CALL インターフェースでヌル・パラメーターを持つ EXCI CALL の例を参照してください。

代理ユーザー・セキュリティー検査を行わない場合は、DFHXCOPT オプション・テーブルで SURROGCHK=NO を指定します (SURROGCHK=YES がデフォルトであることに注意してください)。

代理ユーザー検査は、バッチ領域のユーザー ID が CICS サーバー領域ユーザー ID と同じ場合に有用です。この場合、リンク・セキュリティー検査 (430 ページの『リンク・セキュリティー』を参照) がバイパスされるからです。この場合、DPL_Request 呼び出しで指定されるユーザー ID は認証されたユーザー ID ではないため (パスワードは渡されません)、代理ユーザー検査が推奨されます。

バッチ領域のユーザー ID と CICS 領域ユーザー ID が異なる場合、リンク・セキュリティー検査が実行されます。リンク・セキュリティーを使用する場合、DPL_Request 呼び出しで渡される非認証ユーザー ID は、リンク・セキュリティー検査によって許可されているものより多くの権限を獲得できません。それは、リンク・セキュリティー検査で許可されているのと同じ (またはそれより低い) 権限しか獲得できません。

CICS セキュリティーの詳細については、保護の概要を参照してください。

ONC RPC のセキュリティー

重要: この情報には、プロダクト・センシティブ・プログラミング・インターフェースと関連ガイダンス情報が含まれています。

CICS 環境で ONC RPC サポートを提供する際には、セキュリティーが重要な関心事になります。その理由は、CICS ONC RPC は CICS へのオープン・システムの通信インターフェースを提供するからです。

ONC RPC には独自のセキュリティー方式 (RPC の認証と呼ばれる) があります。ONC RPC 呼び出し内に専用フィールドがあり、応答メッセージ・ヘッダーもあります。RPC 認証には以下の 3 つのタイプがあります。

- **UNIX** 認証。クライアントの UNIX ユーザー ID、グループ ID、およびその他の ID 情報の伝送に使用されます。
- データ暗号化規格 **(DES)** 認証。ONC RPC バージョン 3.9 では使用できないので、CICS ONC RPC と併用することはできません。
- ヌル認証。セキュリティー検査が提供されません。

このセクションでは、CICS ONC RPC が ONC RPC および CICS のセキュリティー機能と対話する方法について説明します。

ONC RPC のセキュリティー

ONC RPC には独自のセキュリティー方式 (RPC の認証と呼ばれる) があります。ONC RPC 呼び出し内に専用フィールドがあり、応答メッセージ・ヘッダーもあります。

RPC 認証には以下の 3 つのタイプがあります。

- **UNIX** 認証。クライアントの UNIX ユーザー ID、グループ ID、およびその他の ID 情報の伝送に使用されます。
- データ暗号化規格 **(DES)** 認証。ONC RPC バージョン 3.9 では使用できないので、CICS ONC RPC と併用することはできません。
- ヌル認証。セキュリティー検査が提供されません。

CICS のセキュリティーとそれが CICS ONC RPC の稼働に及ぼす影響

CICS ONC RPC の稼働中に、さまざまな CICS コマンドを使用して、外部セキュリティー・マネージャー (ESM) によるセキュリティー検査が行われます。

SEC=NO がシステム初期設定パラメーターとして指定されている場合、検査では常に肯定的な結果が返されます。SEC=YES が指定されているが、CICS の稼働中に ESM が異常終了した場合、検査では常に否定的な結果が返されます。CICS セキュリティー・コマンドの使用に関する以下の説明では、SEC=YES が指定されており、ESM がアクティブであることを想定しています。

- ユーザー ID が *userid1* であるトランザクションが EXEC CICS START USERID(*userid2*) を発行すると、代理ユーザー検査が ESM によって行われ、

userid1 に *userid2* を使用する許可があることが確認されます。XUSER=YES がシステム初期設定パラメーターとして指定されている場合にのみ、検査が行われます。

接続マネージャーがサーバー・コントローラーを開始するとき、およびサーバー・コントローラーが別名トランザクションを開始するたびに、このコマンドが発行されます。最初のケースでは、使用されるユーザー ID は、パネル DFHRP02 で CRPM ユーザー ID として接続マネージャーに提供されたものです。2 番目のケースでは、使用されるユーザー ID は、デコードから出力されたものです。

- EXEC CICS VERIFY PASSWORD は、クライアント要求にサービスを提供する CICS プログラムにリンクする前に、別名によって発行されます。ESM により、ユーザー ID とパスワードが許容できる組み合わせであることが検査されます。
- EXEC CICS QUERY SECURITY が別名によって使用され、その実行ユーザー ID に CICS プログラムを使用する権限があることが検査されます。XPPT=YES がシステム初期設定パラメーターとして指定されている場合にのみ、検査が行われます。
- CICS プログラムの稼働中に、保護リソースにプログラムがアクセスしようとするたびに、セキュリティー検査が行われます。RESSEC(YES) が別名トランザクションの定義で指定されており、リソース・タイプのセキュリティー検査を制御するシステム初期設定パラメーターが YES に設定されている場合にのみ、検査が行われます。
- CICS プログラムの稼働中に、プログラムが CICS SPI (システム・プログラミング・インターフェース) からコマンドを使用しようとするたびに、セキュリティー検査が行われます。CMDSEC(YES) が別名トランザクションの定義で指定されており、XCMD=YES がシステム初期設定パラメーターとして指定されている場合にのみ、検査が行われます。

435 ページの図 30 は、CICS セキュリティーと CICS ONC RPC の稼働との相互作用の仕組みを示しています。

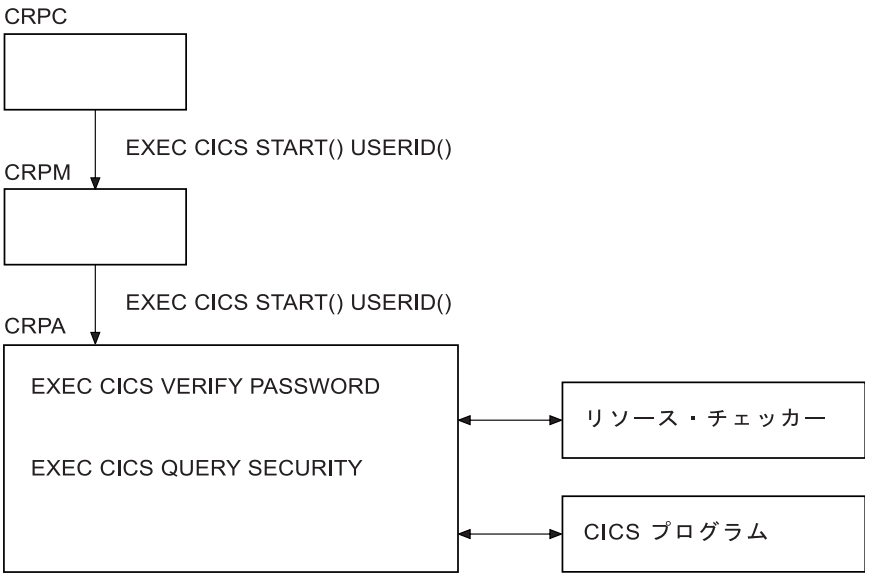


図 *30. CICS* セキュリティーと *CICS ONC RPC* の稼働との相互作用の仕組み

この図は、ユーザー提供のリソース・チェッカー・プログラムが構成されている場合、別名がそのプログラムにリンクすることを示しています。ただし、リソース・チェッカー・プログラムの使用は推奨されていません。CICS セキュリティー機能を使用し、ESM で適切な定義を行う必要があります。

ONC RPC クライアント用の RACF セキュア・サインオン

RACF セキュア・サインオンのサポートでは、パスチケットを送信することにより、RPC クライアントが CICS 機能へのセキュリティー・アクセスを取得できるようにします。これにより、パスワードがネットワーク上を平文で送信されるというセキュリティー上の危険が回避されます。

詳しくは、z/OS Security Server RACF システム・プログラマーのガイドを参照してください。RPC クライアントがパスチケットを生成するために使用する必要があるアルゴリズムの詳細が記載されています。このアルゴリズムには DES アルゴリズムが含まれます。

ONC RPC クライアント用のパスチケットの生成**:** ONC RPC クライアント用にパスチケットを生成するアルゴリズムは、以下の項目の機能です。

- クライアントの CICS ユーザー ID。
- CICS ONC RPC を実行している CICS 領域の CICS アプリケーション ID。
- 双方に認識されている、保護されたサインオン・アプリケーション・キー。
- タイム・スタンプまたは日付スタンプ。

パスチケットを生成するには、RPC クライアントは以下を行う必要があります。

- その CICS ユーザー ID、サーバーの CICS アプリケーション ID、およびアプリケーション・キーを認識する。
- そのクロックをサーバーと 10 分以内に同期する。
- そのマシン上の暗号化アルゴリズムにアクセスする。DES DES アルゴリズムのみ使用できます。

リソース・チェッカーの作成

リソース・チェッカー・プログラムの名前は DFHRPRSC でなければなりません。リソース・チェッカーは、CICS 領域ごとに 1 つのみ可能です。

リソース・チェッカーを使用することにより、インバウンド・クライアント要求の資格情報を検査できます。

リソース・チェッカーは、入力パラメーターとして渡されるクライアント・アドレスを、要求を受け取ったホストで認識されているクライアントのリストと突き合わせて検査することができます。リソース・チェッカーに渡されるパスワードはブランクです。

リソース・チェッカーに関する参照情報

リソース・チェッカーは、クライアント要求にサービスを提供する CICS プログラムへのリンクを試行する前に、必要に応じて別名によって呼び出されます。リソース・チェッカーは、クライアント要求を続行できるかどうか示す必要があります。

リソース・チェッカーに関する参照情報は、以下のように提示されます。

- パラメーターの要約表。それぞれが入力専用と出力専用のどちらであるかが示されます。
  - 入力とは、リソース・チェッカーが参照できるが、変更できないパラメーターのことです。
  - 出力とは、リソース・チェッカーが参照できないが、変更できるパラメーターのことです。
- リソース・チェッカーが行うことになっている処理の説明。
- アルファベット順のパラメーターのリスト、および CICS ONC RPC が入力をセットアップする方法と出力の作成に使用するものの説明。
- リソース・チェッカーが返す可能性がある応答と理由コードのリスト、およびそれぞれの応答と理由コードに対して CICS ONC RPC が行うアクションの説明。

説明では、C 形式で表されるプログラム要素の名前を示しています。COBOL の場合は名前はすべて大文字になり、下線はハイフンに置き換えられます。

パラメーターの要約

リソース・チェッカー・パラメーターが含まれる連絡域の形式は、C ヘッダー・ファイル DFHRPRDH、および COBOL コピーブック DFHRPRDO にあります。また、C ヘッダー・ファイル DFHRPUCH、または COBOL コピーブック DFHRPUCO で定義された値も必要になります。

入力	出力
res_check_alias_transid **res_check_cics_password_ptr** **res_check_cics_userid** **res_check_client_ip_address** **res_check_eyecatcher** **res_check_host_ip_address** **res_check_server_program_name**	**res_check_reason** **res_check_response**

パラメーター

res_check_alias_transid
(入力のみ)

リソース・チェッカーにリンクされた別名トランザクションの 4 文字の名前。

res_check_cics_password_ptr
(入力のみ)

要求側クライアントから渡されたか、デコードによって提供された 8 文字のパスワードを指し示すポインター。このフィールドの値はブランクであり、CICS ONC RPC の旧バージョンとの互換性のために提供されています。

res_check_cics_userid
(入力のみ)

別名を実行する 8 文字の CICS ユーザー ID。。

res_check_client_ip_address
(入力のみ)

クライアントのフルワード・インターネット・アドレス。

res_check_eyecatcher
(入力のみ)

長さが 8 のストリング (この値は、ヘッダー・ファイル DFHRPUCH およびコピーブック DFHRPUCO に定義されています)。

res_check_host_ip_address
(入力のみ)

サーバー・コントローラーが通信中である z/OS Communications Server ホストのフルワード・インターネット・アドレス。

res_check_reason
(出力のみ)

別名に対して返される理由。

res_check_response
(出力のみ)

別名に対して返される応答。

res_check_server_program_name
(入力のみ)

クライアントによって要求されたサーバー機能を実行するために呼び出される CICS プログラムの 8 文字の名前。

応答および理由コード

以下のいずれかの値を **res_check_response** フィールドで返す必要があります。

URP_OK
別名は、クライアント要求の処理を続行します。

URP_EXCEPTION
別名は、例外トレース項目 (トレース・ポイント 9F0E) を書き込み、以下のように理由コードに応じてメッセージを発行します。

- URP_AUTH_BADCRED: メッセージ DFHRP0130

  svcerr_auth 呼び出しと理由値 AUTH_BADCRED を使用してクライアントに応答を送信します。
- URP_AUTH_TOOWEAK: メッセージ DFHRP0184

  svcerr_auth 呼び出しと理由値 AUTH_TOOWEAK を使用してクライアントに応答を送信します。
- その他の値: メッセージ DFHRP0185

  svcerr_systemerr 呼び出しを使用してクライアントに応答を送信します。

URP_INVALID
別名は例外トレース項目 (トレース・ポイント 9F0E) を書き込み、メッセージ (DFHRP0186) を発行します。

svcerr_systemerr 呼び出しを使用してクライアントに応答を送信します。

URP_DISASTER
別名は例外トレース項目 (トレース・ポイント 9F0E) を書き込み、メッセージ (DFHRP0187) を発行します。

svcerr_systemerr 呼び出しを使用してクライアントに応答を送信します。

res_check_response でその他の値を返す場合、別名は例外トレース項目 (トレース・ポイント 9F0E) を書き込み、メッセージ (DFHRP0188) を発行します。
svcerr_systemerr 呼び出しを使用してクライアントに応答を送信します。

エラーの場合、さらに情報を提供するために、 32 ビットの理由コードと応答値を一緒に提出できます。CICS ONC RPC は、URP_EXCEPTION に以前に示されている事例を除いて、リソース・チェッカーによって返された理由コードに対して何の処置も取りません。理由コードは、リソース・チェッカーから生じるトレースまたはメッセージ内に出力され、デバッグの補助として使用できます。

トレース出力内の応答と CICS 定義の理由コードの数値については、応答コードと理由コードの数値を参照してください。

第 11 章 Java アプリケーションのセキュリティー

Java アプリケーションを保護して、許可されたユーザーだけがアプリケーションのデプロイやインストールを行ったり、これらのアプリケーションに Web からアクセスしたり CICS を経由してアクセスしたりできるようにすることが可能です。Java セキュリティー・マネージャーを使用して、Java アプリケーションが安全でない可能性のあるアクションを実行しないようにすることもできます。

Java アプリケーションのライフサイクル内のさまざまな時点で、以下のようなセキュリティーを追加できます。

- Java アプリケーション・リソースの定義やインストールに関するセキュリティー検査を実装します。Java アプリケーションは CICS バンドルにパッケージ化されているので、JVM サーバーにアプリケーションをインストールすることを許可されているユーザーが、このタイプのリソースをインストールできることを確認しなければなりません。
- 許可されたユーザーだけがアプリケーションにアクセスできるようにするための、アプリケーション・ユーザーに関するセキュリティー検査を実装します。
- CICSExecutorService を使って開始される CICS Java タスクに関するセキュリティー検査を実装します。このような CICS タスクはすべて、CJSA トランザクションおよびデフォルト・ユーザー ID の下で実行されます。
- Java セキュリティー・マネージャーを使用して、Java API に対するセキュリティー制限を実装します。

Java アプリケーションは、OSGi フレームワーク内か Liberty サーバー内で実行できます。Liberty は、Web アプリケーションをホストするように設計されており、OSGi フレームワークが組み込まれています。Liberty には独自のセキュリティー・モデルがあるので、Liberty サーバーのセキュリティー構成は異なります。

OSGi アプリケーションのセキュリティーを構成するには、CICS リソース・セキュリティーを使用して、JVMSERVER と Java アプリケーションのライフサイクルを管理できるユーザーを許可します。 CICS トランザクション・セキュリティーを使用して、アプリケーションにアクセスできるユーザーを判別します。

OSGi アプリケーションに関するセキュリティーの構成

CICS リソース・セキュリティーを使用して、JVMSERVER と Java アプリケーションのライフサイクルを管理できるユーザーを許可します。 CICS トランザクション・セキュリティーを使用して、アプリケーションにアクセスできるユーザーを判別します。

手順

- 必要に応じて、JVMSERVER リソースや BUNDLE リソースの作成、表示、更新、削除を行うための権限を、アプリケーション開発者やシステム管理者に与えます。JVMSERVER リソースは JVM サーバーの可用性を制御します。

© Copyright IBM Corp. 1989, 2018

BUNDLE リソースは、Java アプリケーションのデプロイメントの単位で、このアプリケーションの可用性を制御します。

- ユーザーにアプリケーションを実行する権限を与えます。これを行うには、アプリケーションの実行に使用されるトランザクションに、関連するユーザー ID で接続できるようにします。

タスクの結果

OSGi フレームワーク内で実行される Java アプリケーションに関するセキュリティーを正常に構成しました。

Liberty JVM サーバーに関するセキュリティーの構成

CICS Liberty セキュリティー機能を使用すると、Java Platform, Enterprise Edition ロール (Java EE ロール) を介して、ユーザーを認証したり、Web アプリケーションへのアクセスを許可したりできます。このようにして、CICS トランザクションおよびリソース・セキュリティーとの統合が可能になります。また、CICS リソース・セキュリティーを使用して、CICS BUNDLE リソースにデプロイされる JVMSERVER リソースと Java Web アプリケーションの両方のライフサイクルを管理する権限を適切なユーザーに与えることができます。このトピックでは、認証によって所定のユーザーの ID が検証されます。一般的には、ユーザーにユーザー名とパスワードの入力を要求することによって行われます。許可によって、認証済みユーザーの ID に基づいてアクセス制御権限が付与されます。

始める前に

1. CICS 領域で、SAF セキュリティーを使用するための構成が完了していることと、システム初期設定パラメーターとして SEC=YES が定義されていることを確認します。
2. アプリケーション開発者とシステム管理者に、Web アプリケーションを Liberty JVM サーバーにデプロイするために、JVMSERVER リソースと BUNDLE リソースの作成、表示、更新、および除去を行う権限を与えます。

JVMSERVER リソースは JVM サーバーの可用性を制御します。BUNDLE リソースは Java アプリケーションのデプロイメント単位であり、このアプリケーションの可用性を制御します。 CICS TS セキュリティー機能 `cicsts:security-1.0` のデフォルトの動作は SAF レジストリーを使用することです。LDAP レジストリーを使用する場合、SAF レジストリーは作成されません。詳しくは、 分散 ID マッピングを使用した Liberty JVM サーバーのセキュリティーの構成を参照してください。基本ユーザー・レジストリー (`quickStartSecurity` でも使用) は、単純なセキュリティー・テストにのみ適しています。基本ユーザー・レジストリーを構成し、それを使用して実行し、`cicsts:security-1.0` に切り替える必要がある場合は、セッション・トークンを削除する必要があることに注意してください。

このタスクについて

このタスクでは、Liberty JVM サーバー用セキュリティーの構成方法、および Liberty セキュリティーと CICS セキュリティーの統合方法について説明します。Link to Liberty のセキュリティーの構成方法については、CICS プログラムから Java EE アプリケーションへのリンクを参照してください。

Web 要求を実行するためのデフォルトのトランザクション ID は CJSA です。ただし、JVMSERVER タイプの URIMAP を使用して、別のトランザクション ID で Web 要求を実行するように CICS を構成することもできます。通常、URIMAP を Web アプリケーションの一般コンテキスト・ルート (URI) に一致するように指定し、トランザクション ID の有効範囲が、そのアプリケーションを構成するサーブレットのセットになるようにします。あるいは、具体性の高い URI を使用して、個々のサーブレットを別々のトランザクションで実行することも可能です。

Web 要求を実行するためのデフォルトのトランザクション ID は、CICS のデフォルト・ユーザー ID です。URIMAP が使用可能で、静的ユーザー ID が含まれる場合、それがデフォルト・ユーザー ID よりも優先して使用されます。Web 要求のセキュリティー・ヘッダーにユーザー ID が含まれる場合、他のすべてのメカニズムより優先されます。

Liberty から発生し、Web 要求として分類されないタスクは、デフォルトでは CJSU トランザクションで実行されます。このようなタイプのタスクには URIMAP スタイルのメカニズムはありませんが、JVM プロファイル・プロパティー `com.ibm.cics.jvmserver.unclassified.tranid` を使用してデフォルト・トランザクション ID をオーバーライドし、JVM プロファイル・プロパティー `com.ibm.cics.jvmserver.unclassified.userid` を使用してデフォルト・ユーザー ID をオーバーライドすることができます。

注**:** ユーザー ID には、指定されたトランザクションに接続する権限が必要です。詳しくは、トランザクション・セキュリティーを参照してください。

手順

1. Liberty のエンジェル・プロセスを、認証サービスおよび許可サービスを Liberty JVM サーバーに提供するように構成します。 Liberty サーバーのエンジェル・プロセスを参照してください。

   ヒント**:** 名前付きエンジェル・プロセスがある場合は、JVM プロファイルに次の行を追加して、このエンジェル・プロセスに接続するように Liberty JVM サーバーを構成する必要があります。

   ```
   -Dcom.ibm.ws.zos.core.angelName=<named_angel>
   ```

2. オプション: JVM プロファイルに次の行を追加することによって、Liberty JVM サーバーが使用可能にされるときに Liberty エンジェル・プロセスに接続するという要件を適用します。

   ```
   -Dcom.ibm.ws.zos.core.angelRequired=true
   ```

   このオプションを使用すると、エンジェル・プロセスを使用できない場合には Liberty JVM サーバーが始動しなくなります。 これは CICS に対して、Liberty エンジェル検査 API を呼び出して、Liberty JVM サーバーの始動にエンジェル・プロセスを使用できるかどうかを検証するよう指示します。

   エンジェル・プロセスを使用できない場合、CICS は次の処理を行います。

   - Liberty JVM サーバーが CEMT トランザクションを介して使用可能にされる場合は、メッセージが発行され、Liberty JVM サーバーが使用不可になります。

- Liberty JVM サーバーが **SET JVMSERVER** SPI コマンドによって使用可能にされるか、または CICS Explorer を介した CMCI を使用して使用可能にされる場合は、メッセージが発行され、Liberty JVM サーバーが使用不可になります。
- Liberty JVM サーバーが CICS CREATE SPI、BAS、または GRPLIST によって使用可能にされる場合は、メッセージが発行され、CICS は 30 秒待機してから Liberty エンジェル検査 API 呼び出しを再試行します。 5 回試行してもエンジェル・プロセスを使用できない場合は、WTOR メッセージが発行され、そのまま待機するか JVMSERVER リソースを使用不可にするかを選択するオプションがオペレーターに示されます。

3. `cicsts:security-1.0` 機能を、`server.xml` の `featuremanager` リストに追加します。

```
<featureManager>
   ...
   <feature>cicsts:security-1.0</feature>
</featureManager>
...
```

4. 次の例を使用して、System Authorization Facility (SAF) レジストリーを `server.xml` に追加します。

```
<safRegistry id="saf" enableFailover="false"/>
```

5. `server.xml` に対する変更を保管します。
6. オプション: あるいは、Liberty JVM サーバーを自動構成する場合に CICS 領域で **SEC** システム初期設定パラメーターが YES に設定されていると、Liberty JVM サーバーの再始動時に、Liberty JVM セキュリティーをサポートするように Liberty JVM サーバーが動的に構成されます。詳しくは、Liberty JVM サーバーの構成を参照してください。 **SEC** システム初期設定パラメーターが NO に設定されていれば、認証または SSL サポートに依然として Liberty セキュリティーを使用できます。CICS セキュリティーがオフになっている場合に、Liberty セキュリティーを使用するには、`server.xml` ファイルを手動で構成する必要があります。
   a. `appSecurity-2.0` 機能を `featuremanager` リストに追加します。
   b. ユーザーを認証するユーザー・レジストリーを追加します。 Liberty セキュリティーでは、SAF、LDAP、および基本ユーザー・レジストリーがサポートされます。詳しくは、Liberty のユーザー・レジストリーの構成を参照してください。
   c. セキュリティー・ロール定義を追加して、アプリケーション・リソースへのアクセスを許可します。 449 ページの『ユーザーに Liberty JVM サーバー内のアプリケーションを実行する権限を与える』を参照してください。

タスクの結果

Web コンテナーは、Liberty の z/OS® セキュリティー機能を使用するように自動的に構成されます。 SAF レジストリーが認証に使用され、Java EE ロールが許可のために考慮されます。許可制約とセキュリティー・ロールによって、アプリケーションにアクセスできるユーザーが決定されます。これらは通常、アプリケーションのデプロイメント記述子 (`web.xml`) で定義されますが、ソース・コードのセキュリティー・アノテーションとして定義されている場合もあります。通常、ユーザーとグループは、`server.xml` 内のアプリケーションの *<application-bnd>* エレメント

によってロールにマップされます。あるいは、*<safAuthorization>* エレメントが `server.xml` で構成されている場合、マッピングは SAF に保持されます (RACF の EJBROLES として)。

次のタスク

- Liberty アプリケーション・セキュリティー認証規則を構成します。 446 ページの『Liberty JVM サーバーでのユーザー認証』を参照してください。
- Web アプリケーションの許可規則を定義します。 449 ページの『ユーザーに Liberty JVM サーバー内のアプリケーションを実行する権限を与える』および 455 ページの『SAF ロール・マッピングを使用した許可』を参照してください。
- Liberty 認証キャッシュを変更します。

Secure Sockets Layer (SSL) を使用するための詳細については、 461 ページの『Java 鍵ストアを使用した Liberty JVM サーバー用の SSL (TLS) の構成』を参照してください。

Liberty のエンジェル・プロセス

`cicsts:security-1.0` 機能を組み込むと、CICS Liberty JVM サーバーは、エンジェル・プロセスを使用して、System Authorization Facility (SAF) などの z/OS 許可サービスを呼び出します。

オプションで、エンジェル・プロセスに名前を付けることができます。エンジェル・プロセスに名前を指定しない場合、それがデフォルト・エンジェル・プロセスになります。デフォルトのエンジェル・プロセスは 1 つだけ作成できます。別のプロセスを作成しようとすると、その開始時に失敗します。 z/OS イメージ上で実行されるすべての Liberty サーバーで、1 つのエンジェル・プロセスを共用できます。各サーバーが実行しているコードのレベルや、各サーバーが CICS JVM サーバーで実行されるかどうかは関係ありません。名前付きエンジェル・プロセスについて詳しくは、名前付きエンジェルを参照してください。

重要: バンドルされている製品に関係なく、最新バージョンのエンジェル・プロセスをインストールします。最新バージョンは他の IBM ソフトウェアにバンドルされている可能性があります。また、CICS にバンドルされているバージョンを置き換えている可能性があります。

エンジェル・プロセスの開始タスクの実行

1. USSHOME ディレクトリー内の開始タスクの JCL プロシージャーを見つけます (例えば、`/usr/lpp/cicsts55/wlp/templates/zos/procs/bbgzangl.jcl`)。
2. JCL プロシージャーを変更し、JES プロシージャー・ライブラリーにコピーします。 **ROOT** には、`USSHOME/wlp` の値を設定できます。例えば、`ROOT=/usr/lpp/cicsts55/wlp` のようにします。
3. エンジェル・プロセスを開始します。以下の例では、`[.identifier]` は最大 8 文字のオプションの ID を示します。
   a. エンジェル・プロセスに名前を付けずに開始するには、次のコマンドを使用します。

      ```
      START BBGZANGL[.identifier]
      ```

b. エンジェル・プロセスを名前付きエンジェル・プロセスとして開始するには、オペレーターの START コマンドに NAME パラメーターを指定します。以下に例を示します。

```
START BBGZANGL[.identifier],NAME=<named_angel>
```

エンジェル・プロセス名は 1 文字から 54 文字までで、次の文字のみを使用してください。A-Z 0-9 ! # $ + - / : < > = ? @ [ ] ■ _ ` { } | ■

注: Liberty サーバーでは、独自の名前付きエンジェル・プロセスを使用できます。この分離の利点の 1 つは、エンジェル・プロセスが LPAR 上の他の Liberty サーバー・インスタンスに影響を与えずにサービスを提供できることです。エンジェル・プロセスは、Liberty JVM サーバーが始動する前に実行する必要があります。

4. Liberty JVM サーバーを始動します。デフォルトでは、サーバーは名前のないエンジェル・プロセス (使用可能な場合) に接続します。特定のエンジェル・プロセスに接続するには、com.ibm.ws.zos.core.angelName プロパティーを設定します。以下に例を示します。

```
-Dcom.ibm.ws.zos.core.angelName=named_angel
```

5. com.ibm.ws.zos.core.angelRequired プロパティーを true に設定すると、エンジェル・プロセスを有効にする前に、実行中のエンジェル・プロセスの有無を CICS で検査するように指定できます。以下に例を示します。

```
-Dcom.ibm.ws.zos.core.angelRequired=true
```

始動中にエンジェル・プロセスを使用できない場合、サーバーは失敗します。このプロパティーを使用すると、失敗になる状況をより迅速かつスムーズに検出できます。

エンジェル・プロセスの開始タスクとの対話

以下の例では、[.identifier] は最大 8 文字のオプションの ID を示します。

- エンジェル・プロセスを停止します。

```
STOP BBGZANGL[.identifier]
```

- 以下のコンソール・コマンドを使用して、エンジェル・プロセスに接続されている Liberty JVM サーバーを表示します。

```
MODIFY BBGZANGL[.identifier],DISPLAY,SERVERS,PID
```

ジョブ名とプロセス ID (PID) のリストが表示されます。

```
15.48.45 STC82204  CWWKB0067I ANGEL DISPLAY OF ACTIVE SERVERS
15.48.45 STC82204  CWWKB0080I ACTIVE SERVER ASID 5c JOBNAME IYK3ZNA1 PID 83953428
15.48.45 STC82204  CWWKB0080I ACTIVE SERVER ASID 5c JOBNAME IYK3ZNA1 PID 33621002
```

各 Liberty JVM サーバーは、固有の PID の下で実行し、CICS コマンド INQUIRE JVMSERVER によって返されます。

エンジェル・プロセスで使用される SAF プロファイル

- Liberty JVM サーバーは、CICS 領域ユーザー ID の権限の下で実行されます。許可サービスを使用するには、このユーザー ID でエンジェル・プロセスに接続

できる必要があります。エンジェル・プロセスの実行に使用するユーザー ID には、SAF STARTED プロファイルに対するアクセス権限が必要です。以下に例を示します。

```
RDEFINE STARTED BBGZANGL.* UACC(NONE) STDATA(USER(WLPUSER))
SETROPTS RACLIST(STARTED) REFRESH
```

- Liberty JVM サーバーでエンジェル・プロセスに接続するには、**SERVER** クラスにエンジェル (**BBG.ANGEL**。ただし、名前付きエンジェル・プロセスを使用している場合は **BBG.ANGEL.<namedAngelName>**) 用のプロファイルを作成します。例えば RACF で、それに対するアクセス権限を CICS 領域ユーザー ID (*cics_region_user*) に付与します。

  ```
  RDEFINE SERVER BBG.ANGEL UACC(NONE)
  PERMIT BBG.ANGEL CLASS(SERVER) ACCESS(READ) ID(cics_region_user)
  ```

- Liberty サーバーで z/OS 許可サービスを使用するには、許可モジュール **BBGZSAFM** 用の **SERVER** プロファイルを作成し、プロファイルに対して CICS 領域ユーザー ID (*cics_region_user*) を付与します。

  ```
  RDEFINE SERVER BBG.AUTHMOD.BBGZSAFM UACC(NONE)
  PERMIT BBG.AUTHMOD.BBGZSAFM CLASS(SERVER) ACCESS(READ) ID(cics_region_user)
  ```

- CICS 領域ユーザー ID (*cics_region_user*) の権限の下で実行される Liberty JVM サーバーに対して、**SERVER** クラスの SAF ユーザー・レジストリーと SAF 許可サービス (**SAFCRED**) へのアクセス権限を付与します。例えば、RACF では以下のように指定します。

  ```
  RDEFINE SERVER BBG.AUTHMOD.BBGZSAFM.SAFCRED UACC(NONE)
  PERMIT BBG.AUTHMOD.BBGZSAFM.SAFCRED CLASS(SERVER) ACCESS(READ) ID(cics_region_user)
  ```

- **IFAUSAGE** サービス (**PRODMGR**) 用の **SERVER** プロファイルを作成し、そこへのアクセス権限を CICS 領域ユーザー ID に付与します。これにより、Liberty JVM サーバーは、CICS JVM サーバーが使用可能であれば **IFAUSAGE** に登録し、使用不可であれば登録抹消できるようになります。以下に例を示します。

  ```
  RDEFINE SERVER BBG.AUTHMOD.BBGZSAFM.PRODMGR UACC(NONE)
  PERMIT BBG.AUTHMOD.BBGZSAFM.PRODMGR CLASS(SERVER) ACCESS(READ) ID(cics_region_user)
  ```

- **SERVER** リソースをリフレッシュします。

  ```
  SETROPTS RACLIST(SERVER) REFRESH
  ```

次の表は、CICS JVM サーバーで実行される Liberty サーバーによって使用される SAF セキュリティー・プロファイルを要約したものです。

表 47. *CICS Liberty* セキュリティーの *SAF* プロファイル表

クラス	**Profile (**プロファイル**)**	対象	**CICS** 領域ユーザー **ID** **1**	非認証ユーザー **ID** **2**	認証済みユーザー **ID** **3**
SERVER	BBG.ANGEL	Liberty サーバー始動時のエンジェル・プロセスの登録	READ		
SERVER	BBG.ANGEL.<namedAngelName>	Liberty サーバー始動時のエンジェル・プロセスの登録	READ		

表 47. *CICS Liberty* セキュリティーの *SAF* プロファイル表 (続き)

クラス	**Profile (**プロファイル**)**	対象	**CICS** 領域ユーザー **ID** 1	非認証ユーザー **ID** 2	認証済みユーザー **ID** 3
SERVER	BBG.AUTHMOD.BBGZSAFM	Liberty サーバー始動時のエンジェル・プロセスの登録	READ		
SERVER	BBG.AUTHMOD.BBGZSAFM.SAFCRED	Liberty サーバー始動時のエンジェル・プロセスの登録	READ		
SERVER	BBG.AUTHMOD.BBGZSAFM.PRODMGR	Liberty サーバー始動時のエンジェル・プロセスの登録	READ		
SERVER	BBG.SECPFX.BBGZDFLT 4	認証または許可	READ		
APPL	BBGZDFLT 4	認証または許可		READ	READ
EJBROLE	BBGZDFLT.<resource>.<role> 5	認証または許可			READ

1. CICS ジョブまたは開始タスクに関連付けられているユーザー ID。
2. Liberty で非認証要求に使用するユーザー ID。この値は、<safCredentials> エレメントの unauthenticatedUser 属性を使用して制御されます。この値は、デフォルトで WSGUEST になります。
3. Liberty サーバーによって認証されたユーザー ID。
4. BBGZDFLT は、<safCredentials> エレメントの profilePrefix 属性を使用して設定されたセキュリティー・プロファイル接頭部のデフォルト値です。例えば、<safCredentials profilePrefix="BBGZDFLT"/> のようになります。
5. <safAuthorization> エレメントが構成されている場合は、EJBROLE プロファイルが必要です。プロファイルのデフォルト・パターンは、SAF ロール・マッパー・エレメントによって制御されます。このエレメントは、デフォルトで <safRoleMapper profilePattern="%profilePrefix%.%resource%.%role%"/> になります。

詳しくは、z/OS のプロセス・タイプを参照してください。

Liberty JVM サーバーでのユーザー認証

Liberty JVM サーバーで実行されるすべての Web アプリケーションに対して CICS セキュリティーを構成できますが、Web アプリケーションは、セキュリティー制約が Web アプリケーションに含まれている場合にのみユーザーを認証します。セキュリティー制約は、動的 Web プロジェクトまたは OSGi アプリケーション・プロジェクトのデプロイメント記述子 (web.xml) 内に、アプリケーション開発者が定義します。セキュリティー制約とは、保護対象 (URL) および保護に使用するロールを定義するものです。

<login-config> エレメントは、ユーザーが Web コンテナーへのアクセスを獲得する方法、および認証に使用する方式を定義します。サポートされるメソッドは、

HTTP 基本認証、フォーム・ベース認証、または SSL クライアント認証のいずれかです。 CICS 用のアプリケーション・セキュリティーを定義する方法について詳しくは、CICS Explorer 製品資料内の『SSL security for Explorer connections』を参照してください。web.xml 内のエレメントの例を以下に示します。

```
<!-- Secure the application -->
<security-constraint>
 <display-name>com.ibm.cics.server.examples.wlp.tsq.web_SecurityConstraint</display-name>
  <web-resource-name>com.ibm.cics.server.examples.wlp.tsq.web</web-resource-name>
  <description>Protection area for com.ibm.cics.server.examples.wlp.tsq.web</description>
  <url-pattern>/*</url-pattern>
 </web-resource-collection>
 <auth-constraint>
  <description>Only SuperUser can access this application</description>
  <role-name>SuperUser</role-name>
 </auth-constraint>
 <user-data-constraint>
  <!-- Force the use of SSL -->
  <transport-guarantee>CONFIDENTIAL</transport-guarantee>
 </user-data-constraint>
</security-constraint>

<!-- Declare the roles referenced in this deployment descriptor -->
<security-role>
 <description>The SuperUser role</description>
 <role-name>SuperUser</role-name>
</security-role>

<!--Determine the authentication method -->
<login-config>
  <auth-method>BASIC</auth-method>
</login-config>
```

注: あるサーブレットから別のサーブレットに要求を転送するために RequestDispatcher.forward() メソッドを使用すると、セキュリティー検査は、クライアントから要求された最初のサーブレットだけに行われます。

Liberty セキュリティーを使用して CICS で認証されたタスクでは、CICS でのトランザクションおよびリソース・セキュリティー検査を許可するために Liberty アプリケーション・セキュリティー・メカニズムのいずれかから派生したユーザー ID を使用できます。 CICS ユーザー ID は、以下の基準に従って決定されます。

1. Liberty アプリケーション・セキュリティー認証。

   SAF ユーザー・レジストリーとの統合は、CICS Liberty セキュリティー機能の一部として提供されます (分散 ID マッピングが使用されている場合を除く)。Liberty でサポートされているアプリケーション・セキュリティー・メカニズムは、いずれも CICS でサポートされます。これには、HTTP 基本認証、フォーム・ログイン、SSL クライアント証明書認証に加え、カスタム・ログイン・モジュール、JACC、JASPIC、または Trust Association Interceptor (TAI) を使用した ID アサーションが含まれます。 Liberty によって認証されるすべての SAF ユーザー ID には、Liberty JVM サーバーの APPL クラス・プロファイルへの読み取り権限が付与される必要があります。この名前は、Liberty サーバー構成ファイル server.xml の safCredentials エレメントの profilePrefix 設定によって決定されます。

   ```
   <safCredentials profilePrefix="BBGZDFLT"/>
   ```

APPL クラスは、特定の CICS 領域へのアクセスを制御するために CICS 端末ユーザーによっても使用され、Liberty JVM サーバーは、セキュリティー要件に応じて、CICS APPLID と同じプロファイルを使用できます。このエレメントを指定しない場合、デフォルトの profilePrefix である BBGZDFLT が使用されます。

APPLID は定義する必要があります。また、ユーザーはその APPLID へのアクセス権限を持っている必要があります。 APPL クラスの BBGZDFLT プロファイルを構成し、アクティブ化するには、次のようにします。

```
RDEFINE APPL BBGZDFLT UACC(NONE)
SETROPTS CLASSACT(APPL)
```

ユーザーを認証するには、ユーザーに APPL クラスの BBGZDFLT プロファイルへの読み取り権限が付与されている必要があります。ユーザー AUSER が BBGZDFLT APPL クラス・プロファイルに対して認証することを許可するには、次のようにします。

```
PERMIT BBGZDFLT CLASS(APPL) ACCESS(READ) ID(AUSER)
```

Liberty SAF 非認証ユーザー ID には、APPL クラス・プロファイルへの読み取り権限を与える必要があります。SAF で認証しないユーザー ID を Liberty サーバー構成ファイル server.xml の safCredentials エレメントで指定することができます。

```
<safCredentials unauthenticatedUser="WSGUEST"/>
```

このエレメントを指定しない場合、デフォルトの unauthenticatedUser は WSGUEST です。 SAF 非認証ユーザー ID に APPL クラスの BBGZDFLT プロファイルへの WSGUEST 読み取り権限を付与するには、以下のようにします。

```
PERMIT BBGZDFLT CLASS(APPL) ACCESS(READ) ID(WSGUEST)
```

WLP z/OS システム・セキュリティー・アクセス・ドメイン (WZSSAD) は、Liberty サーバーに付与された権限を参照します。これらの権限によって、サーバーがユーザーの認証と許可を行う際に、どの System Authorization Facility (SAF) アプリケーション・ドメインおよびリソース・プロファイルの照会を許されるかを制御します。 CICS 領域ユーザー ID には、認証呼び出しを行うために WZSSAD ドメイン内で権限を付与する必要があります。認証する権限を付与するには、CICS 領域 ID が SERVER クラスの BBG.SECPFX.<APPL> プロファイルに対する読み取り権限を付与されている必要があります。

```
RDEFINE SERVER BBG.SECPFX.BBGZDFLT UACC(NONE)
PERMIT BBG.SECPFX.BBGZDFLT CLASS(SERVER) ACCESS(READ) ID(cics_region_user)
```

詳しくは、WZSSAD を使用した z/OS セキュリティー・リソースへのアクセスを参照してください。

2. 認証されないサブジェクトが Liberty から提供された場合は、URIMAP に定義されている USERID が使用されます。
3. USERID が URIMAP に定義されていない場合は、CICS のデフォルトのユーザー ID で要求が実行されます。

注:

Liberty トランザクションのセキュリティー処理が CICS トランザクションの接続処理中に据え置かれるという方法のために、CICS Monitoring Facility (CMF) レコード、z/OS Workload Manager (WLM) 種別、タスク関連データ、および XAPADMGR 出口の UEPUSID グローバル・ユーザー出口フィールドで使用されるユーザー ID は次のように決定されます。HTTP セキュリティー・ヘッダーのユーザー ID、またはそれがない場合は、対応する URIMAP から取られたユーザー ID。どちらも存在しない場合、CICS のデフォルト・ユーザー ID が使用されます。

Liberty では、認証済みユーザー ID がキャッシュに入れられ、CICS とは異なり、キャッシュ期間中にユーザー ID の有効期限切れ検査がないことに注意してください。標準 Liberty 構成処理を使用して、キャッシュ・タイムアウトを構成できます。 Liberty の認証キャッシュの構成を参照してください。

ユーザーに Liberty JVM サーバー内のアプリケーションを実行する権限を与える

Java EE アプリケーション・セキュリティー・ロールを使用すると、Java EE アプリケーションへのアクセスを許可できます。また、Liberty JVM サーバーでは、CICS トランザクションおよびリソース・セキュリティーを使用して、トランザクションへのアクセスをさらに制限できます (アプリケーションの一部として実行)。

このタスクについて

アプリケーションは、デプロイメント記述子 (`web.xml`) 内の許可制約 (`<auth_constraint>` エレメント) を提供することによって保護されます。これが提供されている場合、許可されたロールのメンバーであるユーザーのみがアプリケーションにアクセスできるようになります。 Java EE ロールのユーザーまたはグループのメンバーシップは、次の 2 つの方法のいずれかで決定されます。

- `server.xml` の `<application>` エレメント内の `<application-bnd>` エレメントを使用して、ユーザー/グループからロールへのマッピングを XML に直接記述します。
- `server.xml` の `<safAuthorization>` を使用して、ユーザー/グループのロール・メンバーシップを SAF でマップできるようにします (通常は EJBROLES を使用)。

詳しくは、SAF ロール・マッピングを使用した許可を参照してください。

CICS セキュリティーを使用すると、既存のセキュリティー・プロシージャーを再利用できますが、個別の Web アプリケーションにはそれぞれ異なる URIMAP からアクセスする必要があります。ロール・ベースのセキュリティーを使用すると、別の Java EE アプリケーション・サーバーの既存の標準 Java EE セキュリティー定義を使用できます。詳しくは、 446 ページの『Liberty JVM サーバーでのユーザー認証』を参照してください。

CICS トランザクションとリソース許可を排他的に使用する場合、またはコードの詳細な注釈ベースのロールの検査を使用する場合は、以下の例に示すように、特別なサブジェクトの ALL_AUTHENTICATED_USERS ロールを使用して、それらのコ

ンポーネントに対する許可決定を据え置くことができます。 Liberty アプリケーションを CICS バンドルでデプロイする場合、これは CICS によって自動的に構成されます。

注: 宣言セキュリティー・アノテーション、および CICS のトランザクションとリソースのセキュリティーに対するアクセス検査は、構成済みの制約 (web.xml) の検証後にのみ実行されます

```
<application id="com.ibm.cics.server.examples.wlp.tsq.app"
    name="com.ibm.cics.server.examples.wlp.tsq.app" type="eba"
    location="${server.output.dir}/installedApps/com.ibm.cics.server.examples.wlp.tsq.app.eba">
 <application-bnd>
  <security-role name="cicsAllAuthenticated">
   <special-subject type="ALL_AUTHENTICATED_USERS"/>
  </security-role>
 </application-bnd>
</application>
```

この特別なサブジェクトを使用して Web アプリケーションのデプロイメント記述子 (web.xml) 内のすべての URL に cicsAllAuthenticated ロール・アクセスを与えると、認証された任意のユーザー ID を使用して Web アプリケーションにアクセスできるようになるため、トランザクションに対する許可は、CICS トランザクション・セキュリティーを使用して制御する必要があります。アプリケーションを dropins ディレクトリーに直接デプロイする場合、dropins はセキュリティーをサポートしないため、アプリケーションは CICS セキュリティーを使用するようには構成されません。

safAuthorization を使用すると、<application-bnd> はユーザー ID からロールへのマッピングのソースとして機能しなくなります。代わりに、SAF の EJBROLE によって、どの SAF ユーザーがどのロール (EJBROLE) を持つかが決まります。safAuthorization を指定すると、<application-bnd> は無視されます。同じ効果を実現し、すべての認証ユーザーにアプリケーションの実行を許可するには、web.xml の <auth-constraint> で特別なロール ** を使用する必要があります。例を以下に示します。

```
<auth-constraint>
        <description>special role for all authenticated users</description>
        <role-name>**</role-name>
</auth-constraint>
```

- 特別なロール名 ** は、ロールに依存しない認証済みユーザーの省略表現です。
- 特別なロール名 * は、デプロイメント記述子に定義されているすべてのロール名の省略表現です。

特別なロール名 ** が許可制約に含まれている場合は、ロールに依存しない認証済みユーザーが制約付き要求の実行を許可されていることを示します。特別なロールでは、web.xml に追加の <security-role> 宣言は必要ありません。

CICS トランザクションまたはリソースのセキュリティーを使用するには、以下の手順に従う必要があります。

手順

1. 各 Web アプリケーションにタイプ JVMSERVER の URIMAP を定義します。通常、URIMAP を Web アプリケーションの一般コンテキスト・ルート (URI) に一致するように指定し、トランザクション ID の有効範囲が、そのアプリケ

ーションを構成するサーブレットのセットになるようにします。あるいは、より正確な URI を使用して、個別のトランザクションでそれぞれのサーブレットが実行されるようにします。

2. CICS トランザクションまたはリソースのセキュリティー・プロファイルを使用して、URIMAP に指定したトランザクションを使用することを Web アプリケーションのすべてのユーザーに許可します。

OAuth 2.0 を使用したアプリケーションの許可

OAuth 2.0 は、委任された許可のオープン・スタンダードです。 OAuth 許可フレームワークを使用すると、ユーザーは、自分のアクセス権限や自分のデータの全範囲を共用せずに、別の HTTP サービスで保管された情報へのアクセス権限をサード・パーティー・アプリケーションに付与することができます。

WebSphere Liberty は OAuth 2.0 をサポートし、OAuth サービス・プロバイダー・エンドポイントおよび OAuth 保護リソース適用エンドポイントとして使用できます。 Liberty は、持続 OAuth 2.0 サービスをサポートします。 パーシスタント OAuth 2.0 サービスの構成を参照してください。localStore エレメントおよび client エレメントを使用して、クライアントをローカルで定義できます。以下の手順では、ローカル・クライアントを使用して、OAuth 2.0 許可を有効にします。

始める前に

SAF セキュリティーは CICS での一般的なユースケースであり、この手順では例で SAF を使用します。

CICS 領域で、SAF セキュリティーを使用するための構成が完了していることと、システム初期設定パラメーターとして SEC=YES が定義されていることを確認します。

オプションで、SAF EJBROLE
`BBGZDFLT.com.ibm.ws.security.oauth20.clientManager` へのアクセス権限を管理者ユーザーに付与できます。セキュリティー・ロール `clientManager` は、管理インターフェースへのアクセスを制御して、ローカル・クライアントに対する照会と持続ローカル・クライアントの作成を行えるようにします。管理者ユーザーは、OAuth 2.0 ローカル・クライアントを制御します。

認証サービスおよび許可サービスを Liberty JVM サーバーに提供するように、Liberty のエンジェル・プロセスを構成します。 Liberty サーバー・エンジェル・プロセスを参照してください。

OAuth について詳しくは、oauth-2.0を参照してください。

このタスクについて

次の手順では、以下の方法について説明します。

- Liberty JVM サーバーで提供される OAuth 2.0 サービスを作成する。
- ローカルに構成されたクライアントを作成する。

- このローカル・クライアントを使用して、リライング・パーティー・アプリケーション (サード・パーティー Web アプリケーションとも呼ばれる) に OAuth 2.0 トークンを付与する。
- このトークンを使用して、アプリケーション内の保護リソースにアクセスする。

制約事項: Db2 JDBC タイプ 2 接続は、持続 OAuth 2.0 サービスではサポートされません。

手順

1. OAuth 2.0 サービス・プロバイダーを構成します。
   a. oauth-2.0 および cicsts:security-1.0 の各機能を server.xml の featureManager エレメントに追加します。

      ```
      <featureManager>
          ...
          <feature>oauth-2.0</feature>
          <feature>cicsts:security-1.0</feature>
      </featureManager>
      ...
      ```

   b. server.xml で OAuth 2.0 プロバイダーを構成します。

      ```
      <oauthProvider id="myProvider">
      </oauthProvider>
      ```

2. リライング・パーティー・アプリケーションのローカル・クライアントを構成します。ローカル・クライアントは、リライング・パーティー・アプリケーションの詳細 (アプリケーションの名前、秘密パスワード、リダイレクト URI を含む) を定義します。
   a. 分かりやすいローカル・クライアント名を定義し、サーバーが許可に使用する秘密パスワードを作成します。ローカル・クライアント・アプリケーションが URI を listen し、サーバーが許可コードを提供します。
   b. server.xml の oauthProvider エレメントで、ローカル・クライアント ID、秘密パスワード、およびリダイレクト URI を指定して、OAuth 2.0 ローカル・クライアントを構成します。

```
<oauthProvider id="myProvider">
    <localStore>
        <client id="myClient" redirect="https://client.example.ibm.com/webApp/redirect" secret="mySecret" />
    </localStore>
</oauthProvider>
```

   重要:

   この例では示されていませんが、パスワードをエンコードして、server.xml 構成へのアクセスを制限することが重要です。パスワードは、USS_HOME/wlp/bin/securityUtility にある Liberty securityUtility を使用してエンコードできます。詳細については、securityUtility コマンドを参照してください。

   注: localStore エレメント内には、複数のローカル・クライアントを構成できます。

3. リライング・パーティー・アプリケーションでサーバー上の保護リソースへのアクセスが必要になる場合、ユーザーは、最初にこれらのリソースへのアクセスを許可する必要があります。

a. リライング・パーティー・アプリケーションでは、ユーザーが、サーバーで認証を行う必要があり、また、以下のようにユーザーを許可エンドポイントにリンクまたはリダイレクトすることにより、リライング・パーティー・アプリケーションのアクセスのタイプを選択する必要があります。

```
https://hostname:port/oauth2/endpoint/provider_name/authorize
```

または

```
https://hostname:port/oauth2/declarativeEndpoint/provider_name/authorize
```

URL の照会パラメーターには、追加のパラメーターが必要です。ステップ 2 で構成したローカル・クライアントの場合は、以下の GET 要求が必要です (すべて 1 行で記述)。

```
https://zos.example.ibm.com/oauth2/endpoint/myProvider/authorize?response_type=code
   &client_id=myClient&client_secret=mySecret&redirect_uri=https://client.example.ibm.com/webApp/redirect
```

ユーザーは、リライング・パーティー・アプリケーションのアクセス権限を選択すると、以下のリダイレクト URI を使用してリライング・パーティー・アプリケーションにリダイレクトされます。

```
https://client.example.ibm.com/webApp/redirect?code=access_code
```

リライング・パーティー・アプリケーションは、OAuth トークンを要求するためにこのアクセス・コードを保管する必要があります。

注: ローカル・クライアントの場合、Liberty JVM サーバーのユーザー・レジスターにユーザーが存在している必要があります。 Liberty JVM サーバーでのユーザー認証について詳しくは、Liberty JVM サーバーでのユーザー認証を参照してください。

b. リライング・パーティー・アプリケーションは、サーバーに POST 要求を送信して OAuth 2.0 トークンを要求します。

```
https://hostname:port/oauth2/endpoint/provider_name/token
```

リライング・パーティー・アプリケーションは、許可エンドポイントから受け取った許可コード、ローカル・クライアント ID、および秘密パスワードを POST データで送信します (grant_type はすべて 1 行で記述)。

```
POST https://zos.example.ibm.com/oauth2/endpoint/myProvider/token HTTP/1.1
Content-Type: application/www-form-urlencoded

grant_type=authorization_code&code=code&client_id=myClient
   &client_secret=mySecret&redirect_url=https://client.example.ibm.com/webApp/redirect
```

これにより、トークンを含んだ JSON 文書が返されます。

4. このトークンを使用して、保護リソースにアクセスします。
   a. HTTP 要求の Authorization ヘッダーにトークンを追加します。
      `Authorization: Bearer <token>`

タスクの結果

ユーザーは、OAuth 2.0 許可フローを介して、Liberty JVM サーバー内の保護リソースにアクセスする権限をサード・パーティー・アプリケーションに付与できます。 Liberty JVM サーバーは、これらのトークンのプロバイダーを構成して、ローカルに構成されたクライアントを作成することができます。

トークンを付与するためにいくつかの方式を使用できます。詳しくは、OAuth 2.0 サービス呼び出しを参照してください。

持続 OAuth 2.0 サービスの構成

WebSphere Liberty は、データベースでの OAuth 2.0 ローカル・クライアントとトークンの永続化をサポートしています。持続 OAuth 2.0 を使用すると、許可されたローカル・クライアントは、再始動後も引き続き OAuth 2.0 サービスにアクセスできます。

始める前に

SAF セキュリティーは CICS での一般的なユースケースであり、この手順では例で SAF を使用します。

- データベース内の表の作成とこれらの表への読み取り/書き込みに必要なアクセス権限を取得して、Liberty の `server.xml` で構成します。
- SAF EJBROLE `BBGZDFLT.com.ibm.ws.security.oauth20.clientManager` へのアクセス権限を管理者ユーザーに付与して、OAuth 2.0 ローカル・クライアントを制御します。
- Liberty の `server.xml` で OAuth 2.0 プロバイダーを作成します。詳しくは、OAuth 2.0 を使用した許可を参照してください。

このタスクについて

以下の手順で、持続 OAuth 2.0 ローカル・クライアントを作成します。このローカル・クライアントは、OAuth 2.0 トークンを付与するために使用します。

制約事項: Db2 JDBC タイプ 2 接続は、持続 OAuth 2.0 サービスではサポートされません。

手順

1. パーシスタント OAuth サービス用の IBM Db2を参考にして、必要な表を作成します。
2. 次の URL に POST 要求を送信して、持続ローカル・クライアントを作成します。

   ```
   https://hostname:port/oauth2/endpoint/provider_name/registration
   ```

   クライアント登録要求を受け入れるための OpenID Connect プロバイダーの構成の最初の表で説明されている JSON 文書を使用します。以下に例を示します。

   ```
   {
       "client_id": "client_id",
       "client_secret": "client_secret",
       "grant_types": [ "authorization_code", "refresh_token" ],
       "redirect_uris": [ "https://client.example.ibm.com/webApp/redirect" ]
   }
   ```

タスクの結果

持続 OAuth 2.0 ローカル・クライアントが作成されます。このローカル・クライアントを使用してトークンを生成すると、トークンがデータベースに保持されま

す。サーバーが再始動したときに、持続ローカル・クライアントとトークンは引き続き有効になります。

SAF ロール・マッピングを使用した許可

Java EE ロールからユーザーおよびグループへのマッピングは、さまざまな方法で行うことができます。分散システムでは、基本レジストリーまたは LDAP レジストリーを、通常はアプリケーション固有の <application-bnd> エレメントと組み合わせて使用して、これらのレジストリーに基づいてユーザーをロール にマップします。アプリケーションのデプロイメント記述子によって、アプリケーションのどの部分にどのロールがアクセスできるかが決まります。

このタスクについて

z/OS には、追加のレジストリー・タイプである System Authorization Facility (SAF) レジストリーがあります。cicsts:security-1.0 機能がインストールされると、LDAP を使用するように構成されていない限り、Liberty JVM サーバーはこのタイプを認証に暗黙的に使用します。SAF 許可を使用することを選択できます。SAF 許可を使用する場合、SAF ロール・マッパーを使用してロールを EJBROLE リソース・プロファイルにマップするために、ユーザーからロールへのマッピングが使用されます。サーバーは SAF に照会して、EJBROLE リソース・プロファイルに対する必要な READ 権限をユーザーが備えているかどうかを判別します。

Liberty JVM サーバーで SAF 許可なしで Java EE ロールを使用する場合、CICS バンドルを使用してアプリケーションをインストールすることはできません。その理由は、CICS バンドルがインストールされているアプリケーションの場合、<application-bnd> エレメントが自動的に作成され、特別なサブジェクト ALL_AUTHENTICATED_USERS が使用されるので、ユーザー自身でこのエレメントを定義することがないためです。代わりに、server.xml に <application> エレメントを直接作成し、必要なロールとユーザーを含めて <application-bnd> を構成する必要があります。

ただし、Java EE ロールと SAF 許可を使用することを選択した場合は、引き続き CICSバンドルを Web アプリケーションのライフサイクルに対して使用できます。SAF レジストリーで判別されるロール・マッピングを使用することを優先して、Liberty で <application-bnd> が無視されます。ロール・マッピングは、EJB ロールに属するユーザーによって決まります。

ヒント: SAF 許可が使用可能になっている場合、特別なサブジェクト ALL_AUTHENTICATED_USERS と EVERYONE は使用できません。

ヒント: CICS 領域を開始する前に、EJB ロールの作成または更新を行うことが推奨されます。Liberty は、z/OS の最小バージョンをサポートするために、GOBAL=NO を指定して RACROUTE REQUEST=LIST を発行します。再始動 (または開始) されるまで、アドレス・スペースには更新が表示されません。

手順

1. <safAuthorization id="saf"/> エレメントを server.xml に追加します。cicsts:distributedIdentity-1.0 機能を使用している場合、これは自動的に定義されます。

2. オプション: 前のステップで racRouteLog="ASIS" をエレメントに追加できます。 これにより、Liberty からの RACF EJBROLE ロギングを確認できます。
3. 記述されている接頭部スキームへの参照を使用して、RACF で EJB ロールを作成します。
4. ユーザーをこれらの EJB ロールに追加します。

   デフォルトでは、SAF 許可を使用すると、ユーザーが特定のロールに属しているかどうかを判別するために <profile_prefix>.<resource>.<role> というパターンがアプリケーションにより使用されます。profile_prefix 部分はデフォルトでは BBGZDFLT ですが、<safCredentials> エレメントを使用して変更できます。詳しくは、WZSSAD を使用した z/OS セキュリティー・リソースへのアクセスを参照してください。

   ロール・マッピングの設定は、server.xml の <safRoleMapper> エレメントを使用して変更できます。以下に例を示します。

   ```
   <safRoleMapper profilePattern="myprofile.%resource%.%role%" toUpperCase="true"/>
   ```

   以下の RACF コマンドを使用して、特定の EJB ロールに対する権限をユーザーに許可できます。ここで、WEBUSER は認証済みユーザー ID です。

   ```
   RDEFINE EJBROLE BBGZDFLT.MYAPP.ROLE UACC(NONE)
   PERMIT BBGZDFLT.MYAPP.ROLE CLASS(EJBROLE) ACCESS(READ) ID(WEBUSER)
   ```

5. オプション: CICS サーブレット・サンプルをデプロイし、Java EE ロール・セキュリティーを SAF 許可とともに使用する場合は、デプロイしたサーブレットごとに SAF EJBROLE を作成します。例えば、BBGZDFLT のデフォルト APPL クラスを使用する場合は、RACF コマンドを使用して、以下の EJBROLE セキュリティー定義を定義します。

```
RDEFINE EJBROLE BBGZDFLT.com.ibm.cics.server.examples.wlp.hello.war.cicsAllAuthenticated UACC(NONE)
RDEFINE EJBROLE BBGZDFLT.com.ibm.cics.server.examples.wlp.tsq.app.cicsAllAuthenticated UACC(NONE)
RDEFINE EJBROLE BBGZDFLT.com.ibm.cics.server.examples.wlp.jdbc.app.cicsAllAuthenticated UACC(NONE)
SETROPTS RACLIST(EJBROLE) REFRESH
```

   許可を必要とする Web ユーザー ID ごとに、定義したロールに読み取り権限を付与します。

```
PERMIT BBGZDFLT.com.ibm.cics.server.examples.wlp.hello.war.cicsAllAuthenticated
       CLASS(EJBROLE) ID(user) ACCESS(READ)
PERMIT BBGZDFLT.com.ibm.cics.server.examples.wlp.tsq.app.cicsAllAuthenticated
       CLASS(EJBROLE) ID(user) ACCESS(READ)
PERMIT BBGZDFLT.com.ibm.cics.server.examples.wlp.jdbc.app.cicsAllAuthenticated
       CLASS(EJBROLE) ID(user) ACCESS(READ)
SETROPTS RACLIST(EJBROLE) REFRESH
```

タスクの結果

ロールと、ロールに含めるユーザーを定義することにより、CICS セキュリティー、Java EE ロール・セキュリティー、あるいはその両方を使用して、Web アプリケーションに対するアクセス権限を付与できます。

LDAP レジストリーを使用した Liberty JVM サーバーのセキュリティーの構成

Liberty は、ユーザー・レジストリーを使用して、ユーザーを認証し、認証および許可を含むセキュリティー関連操作を実行するためのユーザーおよびグループに関する情報を取得します。CICS Liberty のデフォルト・セキュリティーでは、SAF レジストリーが使用されます。ただし、CICS で実行される多くのトランザクションは、分散アプリケーション・サーバー上で ID を認証するユーザーによって開始されるため、CICS は Liberty での Lightweight Directory Access Protocol (LDAP) レジストリーの使用もサポートします。LDAP を使用するには、`server.xml` を手動で構成する必要があります。

始める前に

- CICS 領域で、SAF セキュリティーを使用するための構成が完了していることと、システム初期設定パラメーターとして `SEC=YES` が定義されていることを確認します。
- アプリケーション開発者とシステム管理者に、Web アプリケーションを Liberty JVM サーバーにデプロイするために、JVMSERVER リソースと BUNDLE リソースの作成、表示、更新、および除去を行う権限を与えます。JVMSERVER リソースは JVM サーバーの可用性を制御します。BUNDLE リソースは Java アプリケーションのデプロイメント単位であり、このアプリケーションの可用性を制御します。

このタスクについて

このタスクでは、Liberty JVM サーバーの LDAP セキュリティーを構成し、Liberty セキュリティーを CICS セキュリティーに統合する方法を説明します。分散 ID マッピングは、SAF ユーザー ID を分散 ID に関連付けるために使用できます。CICS 分散 ID マッピング機能を使用して、分散 ID マッピングをセットアップできます。次に、ユーザーは、LDAP サーバーによって認証された自分の分散 ID を使用して CICS Web アプリケーションにログオンできます。z/OS セキュリティー製品で定義されたフィルター (RACMAP) が、この ID の SAF ユーザー ID へのマッピングを判別します。さらに、この SAF ユーザー ID を使用して、JEE アプリケーション・ロール・セキュリティーを介した Web アプリケーションへのアクセスを許可できます。そのようにして、CICS のトランザクションおよびリソース・セキュリティーとの統合が提供されます。SAF ユーザー ID は 1 つ以上の分散 ID にマップできます。

Web 要求を実行するためのデフォルトのトランザクション ID は CJSA です。JVMSERVER タイプの URIMAP を使用して、別のトランザクション ID で Web 要求を実行するように CICS を構成できます。URIMAP を Web アプリケーションの一般コンテキスト・ルート (URI) と一致するように指定して、トランザクション ID のスコープを、そのアプリケーションを構成するサーブレットのセットにすることができます。あるいは、具体性の高い URI を使用して、個々のサーブレットを別々のトランザクションで実行することを選択できます。

このタスクには、以下の 3 つのシナリオがあります。

- シナリオ 1 – SAF 許可を使用した分散 ID マッピング
- シナリオ 2 – SAF 許可を使用しない分散 ID マッピング

- シナリオ 3 – 認証と許可のための LDAP

手順

1. **SAF** 許可を使用した分散 **ID** マッピング

   CICS 分散 ID マッピング機能 cicsts:distributedIdentity-1.0 を使用して、LDAP 分散 ID が SAF ユーザー ID にマップされるようにすることができます。CICS セキュリティー機能 cicsts:security-1.0 とともに使用すると、Liberty LDAP セキュリティーが認証に使用され、EJB ロール・マッピングからの JEE アプリケーション・ロール・セキュリティーが許可のために考慮されます。 CICS トランザクションは、マップされた SAF ユーザー ID で実行され、CICS トランザクションとリソース・セキュリティーとの統合が提供されます。

   a. 認証サービスと許可サービスを Liberty JVM サーバーに提供するように、WebSphere Liberty エンジェル・プロセスを構成します。詳しくは、Liberty サーバーのエンジェル・プロセスを参照してください。

   b. cicsts:security-1.0 機能および cicsts:distributedIdentity-1.0 機能を、server.xml 内の featureManager リストに追加します。

      ```
      <featureManager>
         ...
         <feature>cicsts:security-1.0</feature>
         <feature>cicsts:distributedIdentity-1.0</feature>
      </featureManager>
      ...
      ```

   c. 例えば、server.xml で LDAP サーバーを定義することにより、Liberty が LDAP 認証を使用するように構成します。

      ```
      <ldapRegistry id="ldap"
          host="host.domain.com" port="389"
          ldapType="IBM Tivoli Directory Server"
          baseDN="ou=users,dc=domain,dc=com"
          ignoreCase="true">
      </ldapRegistry>
      ```

      Liberty での LDAP ユーザー・レジストリーの構成について詳しくは、Liberty での LDAP ユーザー・レジストリーの構成を参照してください。

   d. safRegistry エレメントを削除します (存在する場合)。server.xml に対する変更を保管します。

   e. 必要な RACF 定義を行います。これには、Liberty での LDAP ユーザー・レジストリーの構成の説明に従って、分散 ID を SAF ユーザー ID にマップするように RACMAP をセットアップすることや、 455 ページの『SAF ロール・マッピングを使用した許可』の説明に従って、それらのユーザー ID が適切な EJBROLES にアクセスできるようにすることが含まれます。CICS は、SAF の許可および mapDistributedIdentities 属性を safCredentials 構成エレメントで構成します。

   cicsts:distributedIdentity-1.0 機能が cicsts:security-1.0 機能とともに使用されている場合、Liberty LDAP セキュリティーが認証に使用され、EJB ロール・マッピングからの JEE アプリケーション・ロール・セキュリティーが許可

のために考慮されます。 CICS トランザクションは、RACMAP がマップしたユーザー ID で実行され、CICS トランザクションとリソース・セキュリティーとの統合が提供されます。次のタスク

先頭に戻る

2. **SAF** 許可を使用しない分散 **ID** マッピング

   アプリケーションの <application-bnd> エレメントで構成されたロールを考慮すると同時に、CICS トランザクションに、RACMAP がマップしたユーザー ID で実行することを許可することができます。これは、分散した Liberty から CICS Liberty に作業をマイグレーションする際に役立ちます。CICS バンドルを使用すると、ユーザー定義の <application-bnd> は、CICS 生成の <application-bnd> によって上書きされることに注意してください。ロール・マッピングを使用した SAF 許可を使用することをお勧めします。詳細については、 455 ページの『SAF ロール・マッピングを使用した許可』を参照してください。

   a. 認証サービスおよび許可サービスを Liberty JVM サーバーに提供するように WebSphere Liberty エンジェル・プロセスを構成します。詳しくは、Liberty サーバーのエンジェル・プロセスを参照してください。

   b. cicsts:security-1.0 機能および ldapRegistry-3.0 機能を、server.xml の featureManager リストに追加します。

   ```
   <featureManager>
      ...
      <feature>cicsts:security-1.0</feature>
      <feature>ldapRegistry-3.0</feature>
   </featureManager>
   ...
   ```

   c. 例えば、server.xml で LDAP サーバーを定義することにより、Liberty が LDAP 認証を使用するように構成します。

   ```
   <ldapRegistry id="ldap"
       host="host.domain.com" port="389"
       ldapType="IBM Tivoli Directory Server"
       baseDN="ou=users,dc=domain,dc=com"
       ignoreCase="true">
   </ldapRegistry>
   ```

   Liberty での LDAP ユーザー・レジストリーの構成について詳しくは、Liberty での LDAP ユーザー・レジストリーの構成を参照してください。

   d. 分散 ID フィルターを使用して分散 ID を SAF ユーザー ID にマップするように Liberty を構成します。それには、safCredentials 構成エレメントの mapDistributedIdentities 属性を server.xml で true に設定します。

   e. safRegistry エレメントを削除します (存在する場合)。server.xml に対する変更を保管します。

   f. 必要な RACF 定義を行います。これには、 Liberty での LDAP ユーザー・レジストリーの構成の説明に従って、分散 ID を SAF ユーザー ID にマップするように RACMAP をセットアップすることが含まれます。

   g. EJB ロールからの JEE アプリケーション・ロール・セキュリティーが許可に必要な場合は、 455 ページの『SAF ロール・マッピングを使用した許可』のトピックを参照してください。

Liberty LDAP セキュリティーが認証に使用され、<application-bnd> エレメントの JEE アプリケーション・ロール・セキュリティーが分散 ID の許可のために考慮されます。CICS では、トランザクションは RACMAP がマップしたユーザー ID で実行され、CICS トランザクションとリソース・セキュリティーとの統合が提供されます。次のタスク

先頭に戻る

3. 認証と許可のための **LDAP**

   LDAP セキュリティーは、JEE アプリケーション・ロール・セキュリティーを使用した認証と許可の両方のために、CICS Liberty JVM サーバーで使用できます。その後、URIMAP 定義を使用して、トランザクションの実行ユーザー ID を設定できます。このシナリオでは、mapDistributedIdentities 属性は設定されません。

   このシナリオは、セキュリティー・リソースを大幅に変更する必要なしに、分散アプリケーションを CICS Liberty JVM サーバーにマイグレーションする場合に役立ちます。

   a. cicsts:security-1.0 機能および ldapRegistry-3.0 機能を、server.xml の featureManager リストに追加します。

      ```
      <featureManager>
         ...
         <feature>cicsts:security-1.0</feature>
         <feature>ldapRegistry-3.0</feature>
      </featureManager>
      ...
      ```

   b. 例えば、server.xml で LDAP サーバーを定義することにより、Liberty が LDAP 認証を使用するように構成します。

      ```
      <ldapRegistry id="ldap"
          host="host.domain.com" port="389"
          ldapType="IBM Tivoli Directory Server"
          baseDN="ou=users,dc=domain,dc=com"
          ignoreCase="true">
      </ldapRegistry>
      ```

      Liberty での LDAP ユーザー・レジストリーの構成について詳しくは、Liberty での LDAP ユーザー・レジストリーの構成を参照してください。

   c. safRegistry エレメントを削除します (存在する場合)。server.xml に対する変更を保管します。

   d. EJB ロールからの JEE アプリケーション・ロール・セキュリティーが許可に必要な場合は、 455 ページの『SAF ロール・マッピングを使用した許可』のトピックを参照してください。

   アプリケーションは Liberty LDAP セキュリティーを認証に使用し、<application-bnd> エレメントの JEE アプリケーション・ロール・セキュリティーが許可のために考慮されます。CICS トランザクションでは、必要に応じて、URIMAP ユーザー ID または CICS DFLTUSER ユーザー ID で実行します。次のタスク

   先頭に戻る

次のタスク

以下は 3 つのシナリオすべてに適用されます。

- Liberty 認証キャッシュを変更します。
- Web アプリケーション URL をトランザクション ID にマップするように URIMAP 定義をセットアップします。

以下はシナリオ 1 とシナリオ 2 に適用されます。

- マップされたユーザー ID に基づいて URI へのアクセスを許可するように、CICS トランザクション・セキュリティー定義をセットアップします。

先頭に戻る

Java 鍵ストアを使用した Liberty JVM サーバー用の SSL (TLS) の構成

SSL を使用してデータを暗号化するように Liberty JVM サーバーを構成できます。オプションとして、クライアント証明書を使用してサーバーで認証するように構成することもできます。証明書は、Java 鍵ストアか、または RACF などの SAF 鍵リングに保管できます。

このタスクについて

Liberty JVM サーバーで SSL を使用可能にするには、**transportSecurity-1.0** Liberty フィーチャー、鍵ストア、および HTTPS ポートを追加する必要があります。CICS では自動的に `server.xml` ファイルが作成され、更新されます。自動構成では、必ず Java 鍵ストアが作成されます。

Liberty JVM サーバーへの Web 要求では、CICS ソケット・ドメインではなく、TCP/IP ソケットおよび SSL 処理の JVM サポートが使用されることを理解することが重要です。

手順

SSL を構成するために自動構成を使用するには、以下のステップを実行します。

1. JVM プロファイルで JVM システム・プロパティー **-Dcom.ibm.cics.jvmserver.wlp.autoconfigure=true** を使用して、自動構成を有効にします。
2. JVM プロファイルで JVM システム・プロパティー **-Dcom.ibm.cics.jvmserver.wlp.server.https.port** を設定して、SSL ポートを設定します。
3. JVM サーバーを再始動して、必要な構成エレメントを `server.xml` に追加します。

タスクの結果

Liberty JVM サーバーのための SSL が正常に構成されます。

RACF を使用した Liberty JVM サーバー用の SSL (TLS) の構成

SSL を使用してデータを暗号化するように Liberty JVM サーバーを構成できます。オプションとして、クライアント証明書を使用してサーバーで認証するように構成することもできます。証明書は、Java 鍵ストアか、または RACF などの SAF 鍵リングに保管できます。

このタスクについて

Liberty JVM サーバーで SSL を使用可能にするには、**transportSecurity-1.0** Liberty フィーチャー、鍵ストア、および HTTPS ポートを追加する必要があります。server.xml ファイルを編集して、必要なエレメントと値を追加します。RACF 鍵リングを使用する場合は、手動の手順に従う必要があります。

Liberty JVM サーバーへの Web 要求では、CICS ソケット・ドメインではなく、TCP/IP ソケットおよび SSL 処理の JVM サポートが使用されることを理解することが重要です。

手順

SSL を手動で構成するには、署名証明書を作成する必要があります。この署名証明書を使用して、サーバー証明書を作成します。そして、その署名証明書を、署名証明書を使用してサーバー証明書の認証を行うクライアントの Webブラウザーにエクスポートします。

1. 認証局 (CA) 証明書 (署名証明書) を作成します。RACF コマンドを使用する例を以下に示します。

   ```
   RACDCERT GENCERT
     CERTAUTH
     SUBJECTSDN(CN('CICS Sample Certification Authority')
      O('IBM')
      OU('CICS'))
      SIZE(2048)
     WITHLABEL('CICS-Sample-Certification')
   ```

   証明書の SIZE は、少なくとも 2048 ビットにする必要があります。詳しくは、RACF RACDCERT GENCERT (証明書の生成) コマンドを参照してください。

2. ステップ 2 の署名証明書を使用したサーバー証明書を作成します。ここで、<userid> は CICS 領域ユーザー ID です。hostname は、Liberty サーバー HTTPS ポートで使用するように構成されたサーバーのホスト名です。

   ```
   RACDCERT ID(<userid>) GENCERT
     SUBJECTSDN(CN('<hostname>')
      O('IBM')
      OU('CICS'))
      SIZE(2048)
      SIGNWITH (CERTAUTH LABEL('CICS-Sample-Certification'))
     WITHLABEL('<userid>-Liberty-Server')
   ```

   証明書の SIZE は、少なくとも 2048 ビットにする必要があります。詳しくは、RACF RACDCERT GENCERT (証明書の生成) コマンドを参照してください。

3. 署名証明書およびサーバー証明書を RACF 鍵リングに接続します。 以下のコマンドで RACF を使用できます。<keyring> の値は、使用する鍵リングの名前に置き換えてください。<userid> の値は CICS 領域ユーザー ID に置き換えてください。

```
RACDCERT ID(<userid>) CONNECT(RING(<keyring>)
       LABEL('CICS-Sample-Certification')
       CERTAUTH)

RACDCERT ID(<userid>) CONNECT(RING(<keyring>)
       LABEL('<userid>-Liberty-Server'))
```

署名証明書を CER ファイルにエクスポートします。

```
RACDCERT CERTAUTH EXPORT(LABEL('CICS-Sample-Certification'))
       DSN('<userid>.CERT.LIBCERT')
       FORMAT(CERTDER)
       PASSWORD('password')
```

エクスポートされた証明書を、FTP を使用してバイナリー形式でワークステーションに転送し、認証局証明書としてブラウザーにインポートします。

4. server.xml ファイルを編集し、SSL 機能と鍵ストアを追加します。HTTPS ポート (次の例の値は 9443) を設定し、CICS 領域を再始動します。SAF 鍵リングを URL 形式 safkeyring://<userid>/<keyring> で指定する必要があります。<userid> 値には CICS 領域ユーザー ID を設定し、<keyring> 値には鍵リングの名前を設定する必要があります。パスワード・フィールドは、SAF 鍵リングへのアクセスには使用されないため、password に設定する必要があります。

```
<featureManager>
  ...
       <feature>transportSecurity-1.0</feature>
</featureManager>
...
<httpEndpoint host="*" httpPort="9080" httpsPort="9443"
    id="defaultHttpEndpoint"/>
...
.
<keyStore filebased="false" id="racfKeyStore"
        location="safkeyring://<userid>/<keyring>"
        password="password"
        readOnly="true"
        type="JCERACFKS"/>
<ssl id="defaultSSLConfig" keyStoreRef="racfKeyStore"
      sslProtocol="SSL_TLS"
      serverKeyAlias="<userid>-Liberty-Server" />
```

タスクの結果

Liberty JVM サーバーのための SSL が正常に構成されます。

Liberty JVM サーバーでの SSL (TLS) クライアント証明書認証のセットアップ

SSL クライアント証明書認証を使用することで、クライアントおよびサーバーは証明書を相手に提供して相互に確認できます。この方法は、セキュリティーについての懸念のために追加レベルの認証が必要となる状況でよく使用されます。

始める前に

RACF を使用した Liberty JVM サーバー用の SSL (TLS) の構成のタスクを完了する必要があります。まだ CICS Liberty のセキュリティーをセットアップ済みでない場合は、先に進む前に、Liberty JVM サーバーのセキュリティーの構成 (Configuring security for a Liberty JVM server)を完了させてください。

このタスクについて

以下のセットアップ情報では、RACF 鍵ストアを使用して、SSLクライアント証明書認証の証明書を保管していることを想定しています。

手順

1. 署名証明書を使用して個人証明書を作成し、この個人証明書を RACF ユーザー ID に関連付けます。 次に、個人証明書を CER 形式でデータ・セットにエクスポートしてから、ワークステーションにバイナリーで FTP 転送します。その個人証明書を Web ブラウザーに個人証明書としてインポートします。証明書が Web ブラウザーにインポートされると、SSL クライアント証明書を提供して Liberty サーバーの HTTPS ポートに接続できるようになります。次の RACF コマンドを実行します。ここで <clientuserid> は、RACF ユーザー ID を <hostname> はクライアント・コンピューターのホスト名を表します。

   ```
   RACDCERT ID(<clientuserid>) GENCERT
     SUBJECTSDN(CN('<hostname>')
      O('IBM')
      OU('CICS'))
      SIZE(2048)
      SIGNWITH (CERTAUTH LABEL('CICS-Sample-Certification'))
     WITHLABEL('<clientuserid>-certificate')
   ```

   この手順で既に行ったように、個人証明書をエクスポートします。

   ```
   RACDCERT ID(<clientuserid>)
     EXPORT(LABEL('<clientuserid>-certificate'))
      DSN('USERID.CERT.CLICERT')
      FORMAT(PKCS12DER)
      PASSWORD('password')
   ```

   SSL クライアント証明書認証をサポートするように、server.xml の SSL 要素を更新します。

   ```
     <ssl id="defaultSSLConfig" keyStoreRef="racfKeyStore"
        sslProtocol="SSL_TLS"
        serverKeyAlias="<userid>-Liberty-Server"
        clientAuthenticationSupported="true"/>
   ```

   さらに、すべてのクライアントが有効な SSL クライアント証明書を必ず提供するようにしたい場合は、次のようにして **clientAuthentication** 属性を SSL 要素に追加します。

   ```
     <ssl id="defaultSSLConfig" keyStoreRef="racfKeyStore"
        sslProtocol="SSL_TLS"
        serverKeyAlias="<userid>-Liberty-Server"
        clientAuthenticationSupported="true"
        clientAuthentication="true"/>
   ```

2. 手順 2 のクライアント・ユーザー ID の ID を使用して CICS で Web 要求を認証できます。その後、web.xml に CLIENT-CERT の login-config 要素を指

定して、Web アプリケーションをデプロイします。web.xml ファイルは、デプロイする Web アプリケーションのソース・ファイルにあります。

```
<login-config>
  <auth-method>CLIENT-CERT</auth-method>
</login-config
```

代わりに、SSL クライアント証明書認証が構成されていない場合に HTTP 基本認証へのフェイルオーバーを許可するには、server.xml に webAppSecurity 要素を追加します。

```
<webAppSecurity allowFailOverToBasicAuth="true" />
```

3. 最後に、CICS トランザクション・セキュリティーをセットアップして、CICS トランザクションへのアクセスを認証済みクライアント・ユーザー ID に基づいて許可します。 詳しくは、 449 ページの『ユーザーに Liberty JVM サーバー内のアプリケーションを実行する権限を与える』を参照してください。

Java セキュリティー・マネージャーの有効化

デフォルトで、Java アプリケーションでは、Java API に要求されるアクティビティーにセキュリティーの制限がありません。Java セキュリティーを使用して、安全でない可能性があるアクションを Java アプリケーションが実行しないようにするために、そのアプリケーションが実行される JVM に対してセキュリティー・マネージャーを有効にすることができます。

このタスクについて

セキュリティー・マネージャーは、コード・ソースに割り当てられる 1 組の権限 (システム・アクセス権) であるセキュリティー・ポリシーを実施します。 Java プラットフォームにはデフォルトのポリシー・ファイルが用意されています。ただし、Java セキュリティーがアクティブであるときに、Java アプリケーションを CICS で正常に実行できるようにするには、アプリケーションの実行に必要な権限を CICS に付与する追加のポリシー・ファイルを指定する必要があります。

この追加のポリシー・ファイルは、セキュリティー・マネージャーが有効になっている JVM の種類ごとに指定する必要があります。CICS には、独自のポリシーを作成するために使用できるサンプルが用意されています。

注: Liberty JVM サーバーでは、Java セキュリティー・マネージャーの有効化はサポートされていません。

- OSGi セキュリティー・エージェントのサンプルは、プロジェクトに com.ibm.cics.server.examples.security という名前の OSGi ミドルウェア・バンドルを作成します。その中には、セキュリティー・プロファイルが含まれています。このプロファイルは、インストール先のフレームワーク内にあるすべての OSGi バンドルに適用されます。
- example.permissions ファイルには、JVM サーバーで実行中のアプリケーションに固有の許可が含まれています。それには、アプリケーションが System.exit() メソッドを使用していないことを確認するチェックが含まれます。
- CICS に、zFS の中で OSGi バンドルの配置先となるディレクトリーに対する読み取りアクセス権限と実行アクセス権限が必要です。

JVM サーバーの OSGi フレームワークで実行されるアプリケーションの場合:

手順

1. IBM CICS SDK for Java でプラグイン・プロジェクトを作成して、提供される OSGi セキュリティー・エージェントのサンプルを選択します。
2. プロジェクトで、example.permissions ファイルを選択して、セキュリティー・ポリシーの許可を編集します。
   a. CICS zFS および Db2 のインストール・ディレクトリーが正しく指定されていることを確認します。
   b. 必要に応じて他の権限を追加します。
3. zFS 内の適切なディレクトリー (例えば /u/bundles) に OSGi バンドルをデプロイします。
4. JVM サーバーの JVM プロファイルを編集して、他のすべてのバンドルより前になるように、OSGI_BUNDLES オプションに OSGi バンドルを追加します。

   ```
   OSGI_BUNDLES=/u/bundles/
   com.ibm.cics.server.examples.security_1.0.0.jar
   ```

5. JVM プロファイルに次の Java プロパティーを追加して、セキュリティーを有効にします。

   ```
   -Djava.security.policy=all.policy
   ```

6. JVM プロファイルに次の Java 環境変数を追加して、OSGi フレームワーク内でセキュリティーを有効にします。

   ```
   org.osgi.framework.security=osgi
   ```

7. OSGi フレームワークを Java 2 セキュリティーで開始できるようにするには、以下のポリシーを追加します。

   ```
   grant { permission java.security.AllPermission; };
   ```

8. 変更を保管し、JVMSERVER リソースを使用可能にして、ミドルウェア・バンドルを JVM サーバーにインストールします。
9. オプション: Java 2 セキュリティーをアクティブにします。
   a. Java 2 セキュリティー・ポリシー・メカニズムをアクティブにするには、それを適切な JVM プロファイルに追加します。また、Java 2 セキュリティー・ポリシーを編集して、適切な権限を付与する必要があります。
   b. Java 2 セキュリティー・ポリシー・メカニズムがアクティブな状態で Java アプリケーションから JDBC または SQLJ を使用するには、IBM Data Server Driver for JDBC and SQLJ を使用します。
   c. Java 2 セキュリティー・ポリシー・メカニズムをアクティブにするには、JVM プロファイルを編集します。
   d. JDBC ドライバーに対する権限を付与するには、例 1 に示されている行を追加して、Java 2 セキュリティー・ポリシーを編集します。db2*xxx* の代わりに、すべての db2 ライブラリーが配置されているディレクトリーを指定してください。権限は、このレベルより下のすべてのディレクトリーとファイルに適用されます。これにより、JDBC および SQLJ を使用できるようになります。
   e. 例 2 に示されている行を追加して、Java 2 セキュリティー・ポリシーを編集して読み取り許可を付与します。読み取り許可を追加しないで Java プロ

グラムを実行すると、AccessControlExceptions と予測不能な結果が生成されます。 Java 2 セキュリティー・ポリシーで JDBC および SQLJ を使用できます。

例 **1:**

```
grant codeBase "file:/usr/lpp/db2xxx/-" {
  permission java.security.AllPermission;
};
```

例 **2:**

```
grant {

// allows anyone to read properties
permission java.util.PropertyPermission "*", "read";

};
```

タスクの結果

Java アプリケーションが呼び出されると、JVM は、クラスのコード・ソースを判別し、セキュリティー・ポリシーを調べてから、適切な権限をクラスに付与します。

第 12 章 CICS Web サポートのセキュリティー

CICS がインターネットに接続される場合は、CICS アプリケーションおよびデータへの無許可アクセスを防止するために、また第三者がプライベート情報を取得できないようにするために、セキュリティー手段を講じてください。

CICS Web サポート・アーキテクチャーの開発プロセス全体を通して、セキュリティーは CICS Web サポートのアプリケーションおよびユーティリティー・プログラムの設計の一部であると考えてください。また、関連する CICS 機能のリソース定義を作成する際にも、このように考える必要があります。 サブトピックでは、CICS Web サポート実装環境のセキュリティーを強化するために使用できる手段を要約しています。

HTTP サーバーとしての CICS: 認証および識別

HTTP サーバーとしての CICS の場合、TCPIPSERVICE 定義の AUTHENTICATE 属性で認証スキームを指定します。識別は認証プロセスに関連して取得されます。また、認証が不要の場合は、CICS が提供できます。

Web クライアントからの認証および識別の取得は、許可されていないユーザーによるアクセスから CICS システムを保護するための重要なステップです。

HTTP サーバーとしての CICS に対して適用されるセキュリティー手段を、TCPIPSERVICE リソース定義を使用して指定します。 CICS Web サポートに使用するポートごとに、TCPIPSERVICE リソース定義で以下の属性が指定されます。

- ポートに SSL を使用するか、しないか
- ポートに使用される認証スキーム
- 基本認証のためのレルム

認証

HTTP プロトコルで使用するものとして、CICS では、次の 2 つの認証スキームがサポートされています。

- 基本認証は HTTP の一部であり、クライアントはユーザー ID とパスワードまたはパスワード・フレーズを提供することによって、サーバーに対して自身を認証および識別することができます。 この情報は、デコードが容易な base-64 エンコード方式を使用してエンコードされます。 このため、基本認証を認証の唯一の手段として使用するのが適しているのは、パスワードが傍受されることがない場合のみです。 ほとんどの環境では、基本認証を SSL と組み合わせて使用します。そうすることで、SSL 暗号化によりユーザー ID とパスワードの情報が保護されます。
- **SSL** クライアント証明書認証はより安全なクライアント認証方式で、信頼のおける第三者機関 (または認証局) が発行し、SSL 暗号化を使用して送信されるクライアント証明書が使用されます。 SSL 認証を参照してください。クライアント証明書には、CICS での識別に使用することのできるユーザー ID は含まれません。 識別を行うには、証明書が使用される前か、またはクライアントが要求を

© Copyright IBM Corp. 1989, 2018

行ったときに (基本認証を使用して) 自動的に、クライアント証明書を RACF またはそれと同等のセキュリティー・マネージャー内のユーザー ID と関連付けます。 証明書が使用されるたびに、RACF ユーザー ID がクライアントのユーザー ID になります。これについては、RACF ユーザー ID と証明書との関連付けに記載されています。

CICS Web サポートの TCPIPSERVICE リソース定義の作成には、CICS Web サポートに対して、これらの認証スキームの 1 つを指定する TCPIPSERVICE 定義をセットアップする方法が記載されています。

基本認証またはクライアント証明書認証を使用すると、CICS は、ユーザーからの認証要求のプロセスを処理し、必要に応じて認証情報をデコードし、提供された認証をセキュリティー・マネージャーのデータベースと照合し、認証が受け入れ不可の場合はその要求を拒否します。 アナライザー・プログラムまたはユーザー作成アプリケーション・プログラムは、認証が検査されて受け入れられて初めて呼び出されます。

Web クライアントが使用するすべてのユーザー ID のユーザー・プロファイルは、RACF または相当する外部セキュリティー・マネージャーに作成されている必要があります。 RACF profilesを参照してください。

注: CICS は、ここに示す処理を行う際に、パスワード検証を使用してユーザー ID を検証します。 CICS は、CICS 領域へのログインに使用される各ユーザー ID に対して、1 日 1 回完全検証要求を実施します。 RACROUTE REQUEST=VERIFYX マクロを使用した完全検証要求により、RACF はユーザー ID の最終アクセス日時を記録し、ユーザー統計を書き込むことになります。

基本認証の場合、ユーザーが指定したパスワードまたはパスワード・フレーズが期限切れになっていると、CICS は新規パスワードまたはパスワード・フレーズを求めるプロンプトをユーザーに出して、ユーザーが要求を再実行依頼できるようにします。 CICS 提供のユーティリティー・プログラム DFHWBPW が使用されます。このプロセスで CICS がユーザーに表示する Web ページ上のテキストを、カスタマイズできます。これについては、 472 ページの『HTTP 基本認証のパスワード有効期限管理』に記載されています。

クライアント証明書認証の場合、CICS は、提供された証明書をセキュリティー・マネージャーのデータベースと照合し、また必要な場合は、ユーザーがセットアップしてある証明書取り消しリストとも照合して、その証明書を検査します。 この処理によって取得された情報が要求の処理方法の決定に役立つ場合は、ユーザー作成アプリケーションでその情報を調べることができます。 以下の項目を取得するには、EXTRACT CERTIFICATE コマンドを使用します。

- 発行者または対象の識別名の構成要素。 SSL 認証では、識別名について説明しています。
- 証明書に関連付けられている RACF ユーザー ID。

識別

Web クライアントのユーザー ID を取得するときに、識別が行われます。 ID は以下のようにして Web クライアントから取得されます。

- 基本認証の実行時に取得する。

- ユーザー ID をクライアント証明書に関連付ける。

アプリケーション生成の応答の場合のみ、CICS は Web クライアントの代わりに、次の方法でユーザー ID を提供できます。

- アプリケーション生成の応答の処理パスで使用されるアナライザー・プログラムで、提供する (この ID は、取得された Web クライアントのユーザー ID をオーバーライドできます)。
- 要求の URIMAP 定義で提供する (この ID は、取得された Web クライアントのユーザー ID をオーバーライドできません)。
- 他の ID を決定できない場合、CICS のデフォルトのユーザー ID として、提供する。

Web クライアントの代わりにユーザー ID を提供する場合は、クライアントの ID は認証されないことに注意してください。 ユーザー ID を提供するのは、ユーザー自身のクライアント・システムと通信する際に、クライアント・システムが既にそのユーザーを認証していて、セキュア環境でサーバーと通信する場合のみにしてください。 Identifying HTTP users では、TCPIPSERVICE 定義の設定に応じて、ユーザー ID を決定する方法の詳細を説明しています。

クライアントを識別済みの場合、そのクライアントのユーザー ID には、その他のユーザー ID と同じ様に、RACF またはそれ等価な外部セキュリティー・マネージャーを使用して、 CICS リソースへのアクセスを許可することができます。 CICS 文書テンプレートとして保管されている Web ページや z/OS UNIX ファイル、応答を提供するアプリケーションで使用される CICS コマンドなど、CICS において Web クライアントがアクセスする個々のリソースのいずれかまたはすべてに、リソース・レベルのセキュリティーを適用することもできます。 475 ページの『CICS システムと CICS Web サポートのリソース・セキュリティー』で、これらのリソースを保護する方法、および不要になった場合にリソース・レベルのセキュリティーを除去する方法を説明しています。

HTTP クライアントとしての CICS: 認証および識別

CICS を介して HTTP クライアント要求を行うと、基本認証、プロキシー認証、または SSL クライアント証明書認証を実行するようサーバーまたはプロキシーが要求することがあります。

基本認証は、WEB SEND または WEB CONVERSE コマンドの AUTHENTICATE オプションを使用して行えます。 プロキシー認証は、ユーザー・アプリケーションが行います。 クライアント証明書は、URIMAP 定義を使用して提供します。

クライアント・アプリケーションは、以下の方法で自身の認証を要求されることがあります。

- 基本認証では、特定の情報にアクセスするため、ユーザー名およびパスワードを提供することができます。 サーバーに要求を出すと、そのサーバーは 401 状況コードおよび WWW-Authenticate ヘッダーを含む応答を送信することがあります。 このヘッダーには、基本認証が要求されているレルムが指定されています。 要求した情報を受信するには、ユーザー名とパスワードを提供します。 CICS は、ユーザーがそのレルムにアクセスできるように、ユーザー名とパスワードを指定した Authorization ヘッダーとともに要求を再送信します。 CICS

は、Authorization ヘッダーを予期しているサーバーに、Authorization ヘッダーを直接送信することもできます。これにより、401 応答の必要性がなくなります。 CICS は、ユーザー名とパスワードを ASCII に変換し、基本認証プロトコルで必要とされている base-64 エンコードを適用します。 したがって、WEB SEND または WEB CONVERSE コマンド、あるいは XWBAUTH ユーザー出口を介して、通常の文字で資格情報を提供できます。 基本認証のための資格情報の提供およびHTTP 基本認証を参照してください。

- プロキシー認証は、プロキシー・サーバーによって開始されます。 プロキシー認証の場合、応答の状況コードは 407 で、プロキシー・サーバーからの確認のための質問ヘッダーは Proxy-Authenticate、応答ヘッダーは Proxy-Authorization です。 CICS はこのプロトコルをサポートしていません。
- **SSL** クライアント証明書認証では、信頼のおける第三者機関 (または認証局) が発行したクライアント証明書が使用されます。 HTTPS 要求を行う際、サーバーはこの認証を提供するよう要求する場合もあります。 Configuring CICS to use SSLでは、証明書の取得方法と、RACF データベースまたは同等の外部セキュリティー・マネージャーの鍵リングに証明書を保管する方法を説明しています。サーバーがクライアント証明書を要求した場合は、接続を行うための WEB OPEN コマンドで使用される URIMAP 定義、または WEB OPEN コマンドそのもので、適切な証明書のラベルを指定することができます。 あるいは、WEB OPEN コマンドのオプションとして証明書ラベルを直接指定することもできます URIMAP 定義を使用しているのに、証明書ラベルが指定されていない場合、CICS 領域のユーザー ID の鍵リングで定義されているデフォルトの証明書が使用されます。

サーバーによっては、他のタイプの認証または識別を提供するよう要求する場合があります。 受け入れ可能な認証または識別をサーバーに提供できない場合、要求は拒否されます。 基本認証またはプロキシー認証の場合、サーバーが要求を拒否したときに使用されていた状況コードは、確認のための質問に対する状況コードと同じです (サーバーの場合は 401、プロキシーの場合は 407)。 確認のための質問に応答したにもかかわらず、これらの状況コードの 1 つを含む応答をさらに受信した場合は、使用した許可情報が無効になっています。

HTTP 基本認証のパスワード有効期限管理

HTTP 接続で基本認証が使用されている場合、CICS Web サポートは、外部セキュリティー・マネージャー内のユーザー ID とパスワードを検査します。パスワードの有効期限が切れている場合は、CICS 提供のユーティリティー・プログラム DFHWBPW を使用して、ユーザーに対して新しいパスワードの選択を求めるプロンプトを出します。 DFHWBPW からユーザーに提示されるページは、ユーザー自身がカスタマイズまたは置換することができます。

DFHWBPW は、要求に適用される TCPIPSERVICE 定義が AUTHENTICATE 属性の BASIC、AUTOREGISTER、または AUTOMATIC オプションで定義されている場合の、パスワード有効期限管理にのみ使用されます。 DFHWBPW の構造はコンバーター・プログラムと似ていますが、通常の CICS Web サポート処理パスの一部ではないので、その他の目的でこれにコードを追加する必要はありません。 ユーザーが新しいパスワードを選択すると、DFHWBPW はクライアントを元の要求

の URL にリダイレクトすることによって要求の実行依頼を再開するため、その要求の処理パスはすべて通常通りに発生します。

DFHWBPW は、以下の 2 つの Web ページをユーザーに提示します。

1. パスワード・プロンプト・ページ。 このページには、以下の 2 つの要素が含まれています。
   a. パスワードの有効期間に関するメッセージ。 ユーザーに対して表示される最初のメッセージには、パスワードの有効期限が切れていることが記載されます。 ユーザー ID は、標準パスワードとパスワード・フレーズの両方を持つことができます。 長さが 9 から 100 文字までのパスワードがパスワード・フレーズであり、8 文字以下のパスワードが標準パスワードです。 標準パスワードとパスワード・フレーズは、互いに無関係の働きがあります。 標準パスワードの有効期限が切れた場合は、新規の標準パスワードに置き換える必要があります。同様に、パスワード・フレーズの有効期限が切れた場合も、新規パスワード・フレーズに置き換える必要があります。 ユーザーがパスワードを変更しようとして失敗した場合 (例えば、入力された新規パスワードの 2 つのコピーが一致しない)、さらにメッセージが表示されて問題が示されます。
   b. ユーザーがパスワードを変更するための HTML フォーム。
2. 確認および要求の最新表示ページ。 このページでは、有効期限が切れたパスワードが正常に置換されたことを確認します。また、要求を自動または手動で再作成するための、最新表示タグおよび URL リンクが表示されます。

DFHWBPW は 3 つの CICS 文書テンプレート DFHWBPW1、DFHWBPW2、および DFHWBPW3 を使用して、これらの Web ページを作成します。 CICS 提供の定義は、これらのテンプレートをロード可能プログラム (つまり、タイプ PROGRAM(DFHWBPW1) など) として定義します。 定義は、CICS 提供のリソース定義グループ DFHWEB にあります。 これらの定義は、別のグループにコピーし、リソース定義 ALTER コマンドを使用してテンプレートを別のソースから派生させることによって変更することができます。 また、リソース定義は変更せず、その代わりにロードされるプログラムを変更することもできます。
DFHWBPW1、DFHWBPW2、および DFHWBPW3 の 3 つのプログラムは、アセンブラー言語データのみのモジュールで、そのソースは、対応する CICS サンプル・ライブラリー SDFHSAMP のメンバーに含まれています。これらのサンプルを変更して、DFHRPL データ定義ステートメントに連結されている通常の CICS プログラム・ライブラリーのいずれかの中に、再アセンブルおよびリンク・エディットすることができます。

注**:** 不要な出力を避けるために、DFHWBPW1、DFHWBPW2、および DFHWBPW3 をアセンブルする際は、**NOPROLOG** パラメーターと **NOEPILOG** パラメーターを指定するようにしてください。

各 DFHWBPW テンプレートの内容と機能は、以下のとおりです。

DFHWBPW1
: パスワード・プロンプト・ページの一部。 そのページにつながる HTML ページ見出しを提供し、可能なパスワード妥当性メッセージのシンボルを (サーバー・サイドに組み込まれているシンボル設定用の技法を使用して) 設定します。メッセージは、以下の情報を提供します。

message.1
パスワードの有効期限が切れています。

message.2
入力されたユーザー ID が無効です。

message.3
2 回入力された新規パスワードが一致しません。

message.4
入力された以前のパスワード (有効期限が切れたもの) が正しくありません。

message.5
入力された新規パスワードは、パスワード品質規則に従っていないため、外部セキュリティー・マネージャーによって許可されませんでした。

message.6
そのユーザー ID は現在取り消されています。

DFHWBPW プログラムは、適切なシンボルを選択して、パスワード・プロンプト・ページの文書に挿入します。 DFHWBPW1 をカスタマイズしてページ見出しやタイトルを変更することや、`body` タグを変更してページの色や背景を変更することが可能です。また、メッセージ・シンボルの内容を変更することもできます。

DFHWBPW2
パスワード・プロンプト・ページの一部。 ユーザーがユーザー ID、古い (有効期限が切れた) パスワード (またはパスワード・フレーズ)、および希望する新規パスワード (同じものを 2 回) を入力する HTML フォームを作成します。 DFHWBPW2 をカスタマイズしてユーザーへのプロンプトに使用するテキストを変更するか、そのページのレイアウトを変更することができます。 ただし、`form` タグや `input` タグの内容を変更することはできません。これを変更すると、DFHWBPW が本来の処理を行わない場合があります。

DFHWBPW3
確認および要求の最新表示ページ。 このテキストは、期限切れのパスワードが正常に置換されたことをユーザーに通知し、パスワードの再入力を求めるプロンプトがクライアントからすぐに出されることをユーザーに明らかにします。 このテキスト、およびページのレイアウトは、カスタマイズすることができます。

DFHWBPW3 によって要求プロセスが再開されます。 これには、`meta http-equiv="Refresh"` タグが含まれています。このタグは、10 秒後に、有効期限の切れたパスワードが検出されたときにユーザーが要求していたページに自動的にリダイレクトします。 このタグの制限時間を変更することができます。また、ユーザーが自動リダイレクトされないようにする場合は、このタグを削除できます。 ただし、変更後のページには、本来要求されていたページへのリンクが必ず含まれている必要があります。 そのページの URL は、シンボル `&dfhwbpw_target_url;` に入っています。要求プロセスを再開するということは、Web クライアントが旧パスワードをキャッシュ

に入れている場合は、直ちに新規パスワードに置換できるということを意味します。また、CICS Web サポートの処理パスが影響を受けないということにもなります。

CICS システムと CICS Web サポートのリソース・セキュリティー

CICS が HTTP サーバーである場合、CICS システムを、無許可のユーザーによるアクセスから保護する必要があります。 システムが適切に保護されていない場合、ユーザーが機密データにアクセスしたり、システムを妨害して他のユーザーに対するサービス妨害を発生させたりすることが可能性があります。

CICS Web サポートへのアクセス全般を監視するには、HTTP クライアント要求を行う各ユーザーに ID を要求し、ユーザーが提示した ID を認証します。 インバウンド・ポートの TCPIPSERVICE 定義を使用して、これらの要件を指定します。469 ページの『HTTP サーバーとしての CICS: 認証および識別』を参照してください。

Web クライアントが使用するすべてのユーザー ID のユーザー・プロファイルは、RACF または相当する外部セキュリティー・マネージャーに作成されている必要があります。 RACF profilesを参照してください。

Web クライアントの認証済みユーザー ID を取得したら、この ID を使用して、応答を提供するために使用する CICS 領域のリソースに対するリソース・レベルのセキュリティーを実装できます。 手順は、以下の応答タイプごとに異なります。

- アプリケーションが生成する応答。
- 応答として CICS 文書テンプレートを提供する URIMAP 定義を使用した静的応答。
- z/OS UNIX システム・サービスのファイルを応答として提供する URIMAP 定義を使用した、静的応答。

アプリケーションが生成する応答の場合、CICS システムのデフォルト指定では、リソース・セキュリティー検査は実行されず、トランザクション・セキュリティー検査 (具体的には、別名トランザクションのトランザクション接続セキュリティー) が実行されます。 トランザクション・セキュリティーが CICS 領域でアクティブであると想定した場合、Web クライアントの認証済みユーザー ID をセキュリティー検査に使用しない場合でも、アプリケーション生成の応答に明確に関連したアクションを実行する必要があります。

静的応答の場合、Web クライアントのユーザー ID に対してトランザクション接続セキュリティーは適用されません。 ただし、CICS システムのデフォルト指定では、Web クライアントのユーザー ID が使用可能な場合には、応答レベルのセキュリティー検査が実行されます。 認証済みユーザー ID を Web クライアントから取得する場合は、それらのユーザー ID のリソース権限を設定するか、リソース・レベルのセキュリティー検査を無効にするアクションを実行する必要があります。

CICS Web サポートによって提供されるすべての応答に Web クライアントのユーザー ID を使用したリソース・レベルのセキュリティーを実装するかどうかにかかわらず、以下の保護を備える必要があります。

- 無許可アクセスまたは悪意のあるアクセスからインバウンド・ポートを保護する手段を実装する。
- 無許可のユーザーによる変更から CICS システム・コンポーネントを保護し、認証済みユーザーが正しいアクセス権を持っていることを確認する。

インバウンド・ポートのセキュリティー

TCPIPSERVICE リソース定義で、CICS Web サポートに使用される各ポートを定義します。 TCPIPSERVICE 定義では、SSL を使用するかどうかや、クライアントに要求される認証レベルなど、ポートのセキュリティー・オプションを指定します。ポートは、無許可アクセスや悪意のあるアクセスに対してガードされていなければなりません。

CICS Web サポートの TCPIPSERVICE リソース定義の作成で、ポートの定義の作成方法について説明しています。

ポートの保護状態が維持されるようにするには、以下のようにします。

- すべての TCPIPSERVICE 定義に MAXDATALEN 属性を指定します。これは、CICS が単一の要求に対して受け入れるデータの最大量を制限するオプションであり、大容量データの送信を伴うサービス妨害アタックに対する CICS の防御に役立ちます。
- Web クライアントとの対話を確実に機密の状態に維持し、第三者が傍受できないようにするには、Secure Sockets Layer (SSL) を使用します。 機密データを送信する場合、またはユーザー ID やパスワードなどの許可情報をサーバーに渡す場合は、SSL を使用することが特に重要です。 482 ページの『CICS Web サポートでの SSL』を参照してください。

1 つ以上の CICS Web サポートのポートで異常な活動を発見した場合は、CICS システム・コマンドを使用して、CICS Web サポートをさまざまなレベル (単一の要求、仮想ホスト、ポート、または CICS Web サポート全体) でシャットダウンし、CICS システムをシャットダウンする必要はありません。 『Administering』の『Rejecting HTTP requests』を参照してください。

URIMAP リソース定義では、要求のスキームとして HTTP または HTTPS を指定します。 HTTP が指定された URIMAP は、HTTP またはより安全な HTTPS のどちらかを使用して作成された Web クライアント要求を受け入れます。 HTTPS が指定された URIMAP は、HTTPS を使用して作成された Web クライアント要求のみ受け入れます。

HTTPS が指定された URIMAP 定義が Web クライアントからの要求と一致する場合、CICS は、要求で使用されたインバウンド・ポートが SSL を使用していることをチェックします。 SSL がポートに指定されていない場合、要求は 403 (Forbidden) 状況コードで拒否されます。 URIMAP 定義がすべてのインバウンド・ポートに適用されると、このチェックにより、Web クライアントは非セキュアなポートを使用してセキュアなリソースにアクセスすることができなくなります。HTTP を指定した URIMAP 定義に対しては検査が行われないため、Web クライアントは、非セキュアなポート、またはセキュアな (SSL) ポートのどちらを使用しても、保護されたリソースにアクセスできます。

CICS システム・コンポーネントのセキュリティー

他の CICS リソースと同様に、CICS Web サポートで使用される CICS システム・コンポーネントを、無許可ユーザーによる変更から保護する必要があります。また、許可されているユーザー (特に CICS 領域) に、これらのコンポーネントの使用に必須の権限があることを確認することも必要です。

CICS Web サポートを制御するには、アプリケーション・プログラムやリソース定義など、多数のコンポーネントが使用されます。 CICS Web サポートのコンポーネントを参照してください。これらのコンポーネントを無許可アクセスから保護しないと、CICS Web サポート・アーキテクチャーのセキュリティーが危うくなる可能性があります。 例えば、ポートの TCPIPSERVICE 定義へのアクセス権を持つユーザーによって、SSL の使用または ID の提供に必要な、Web クライアントの要件が削除される可能性があります。 Implementing RACF protection in a single CICS regionでは、CICS トランザクション、リソース、およびコマンドを不正な使用から保護する方法を説明しています。

一部の CICS システム・コンポーネントでは、許可されているユーザーに対して、追加のアクセス権限をセットアップする必要があります。

- URIMAP リソースでは、Web クライアントのユーザー ID を設定するために、追加の権限が必要とある場合があります。 代理ユーザー検査が、(システム初期設定パラメーターとして XUSER=YES が指定されている) CICS 領域で使用可能に設定されている場合、CICS は、URIMAP 定義をインストールするために使用されたユーザー ID が、USERID 属性に指定されているユーザー ID の代理として許可されているかどうかを確認します。
- 文書テンプレートを使用して、HTTP サーバーとしての CICS からの応答のボディを作成したり、HTTP クライアントとしての CICS からの要求のボディを作成したりできます。 文書テンプレートは、DOCTEMPLATE リソース定義で定義します。 文書テンプレートが区分データ・セットで保管される場合、CICS 領域のユーザー ID は、そのデータ・セットの READ 権限を持っている必要があります。
- z/OS UNIX システム・サービス・ファイルを使用して、HTTP サーバーとしての CICS からの静的応答のボディを作成できます。 これらは、それぞれ独自の名前で指定するか、DOCTEMPLATE リソース定義で定義することができます。z/OS UNIX ファイルを使用する場合は、z/OS UNIX へのアクセス権限と、ファイルを含む z/OS UNIX ディレクトリーおよびファイルそのものへのアクセス権限が、CICS 領域に必要です。 CICS 領域に対する z/OS UNIX ディレクトリーおよびファイルへのアクセス権限の付与を参照してください。

アプリケーションが生成する応答のリソース・セキュリティーとトランザクション・セキュリティー

セキュリティー・マネージャーにプロファイルが作成されている Web クライアントの認証済みユーザー ID を取得した場合、このユーザー ID は、アプリケーションが生成する応答に使用される別名トランザクションに適用されます。

このタスクについて

Web クライアントのユーザー ID に適切な権限を与えるか、独自の標準ユーザー ID をオーバーライドとして指定します。 Web クライアントのユーザー ID をリソース・セキュリティー検査に使用するかどうかにかかわらず、別名トランザクション用のユーザー ID が適切な権限を持つようにしておく必要があります。

TRANSACTION リソース定義で別名トランザクションを定義します。 アプリケーションが生成する各応答の別名トランザクションは、要求の URIMAP 定義によって指定するか、またはアナライザー・プログラムによって指定されます。 デフォルトは CICS 提供の別名トランザクション CWBA です。これは、Web 対応アプリケーションまたは COMMAREA アプリケーションのどちらかを使用して応答を提供するときに適用されます。

別名トランザクションを実行するときのユーザー ID には、以下のタスクの実行権限が必要です。

- 別名トランザクションを接続する (CICS 領域に対してトランザクション接続セキュリティーが指定されている場合)。 トランザクション接続セキュリティーは、システム初期設定パラメーターである XTRAN によって制御されます。 デフォルトは、YES (トランザクション接続セキュリティーがアクティブ) です。
- 別名トランザクションで使用される任意の CICS リソースにアクセスする (別名トランザクションに対してリソース・セキュリティーが指定されている場合)。 リソース・セキュリティーは、別名トランザクションの TRANSACTION リソース定義の RESSEC 属性によって制御されます。 デフォルトは NO (リソース・セキュリティーなし) であり、CWBA に対して提供されている設定も NO です。
- 別名トランザクションで使用される任意の CICS システム・プログラミング・コマンドにアクセスする (別名トランザクションに対してコマンド・セキュリティーが指定されている場合)。 これらのシステム・プログラミング・コマンドは、応答を生成するユーザー作成アプリケーション・プログラムで使用されます。 コマンド・セキュリティーは、別名トランザクションの TRANSACTION リソース定義の CMDSEC 属性によって制御されます。 デフォルトは NO (コマンド・セキュリティーなし) であり、CWBA に対して提供されている設定も NO です。

Web クライアントが CICS Web サポートに対して要求を行い、アプリケーションによって応答が提供されると、CICS は以下の優先順に従って別名トランザクション用のユーザー ID を選択します。

1. アナライザー・プログラムを使用して設定したユーザー ID。 このユーザー ID は、Web クライアントから取得された、または URIMAP 定義によって提供されたユーザー ID をオーバーライドできます。
2. 基本認証を使用して Web クライアントから取得したユーザー ID、または Web クライアントによって送信されたクライアント証明書に関連付けられたユーザー ID。 接続に認証が必要なのに、クライアントが認証済みユーザー ID を提供していない場合、要求はリジェクトされます。
3. 要求の URIMAP 定義で指定したユーザー ID。
4. 他に判別できるユーザー ID がない場合、CICS デフォルト・ユーザー ID。

CICS Web サポートのアーキテクチャーに応じて、さまざまな要求に対して 1 つまたは複数のタイプのユーザー ID を使用する場合があります。 Web クライアントの認証済みユーザー ID を取得した場合、その ID をオーバーライドするアクションを実行しない限り、その ID が別名トランザクションに使用されます。

アプリケーションが生成する応答に対して、以下のセキュリティー・アクションを実行します。

手順

1. Web クライアントの認証済みユーザー ID を取得しても、それをアプリケーション生成応答のセキュリティー検査に使用しない 場合は、アナライザー・プログラムを使用して、関連する別名トランザクションの標準ユーザー ID で Web クライアントのユーザー ID をオーバーライドします。 (CICS のデフォルトのユーザー ID を使用できます)。 このオーバーライドを指定する要求の処理パスに、アナライザー・プログラムを含めます。 「システム・プログラムの開発」の「アナライザー・プログラム」を参照してください。このユーザー ID のユーザー・プロファイルがセキュリティー・マネージャーで定義されていることを確認してください。 標準ユーザー ID を設定した場合は、この手順の残りのステップに従って、標準ユーザー ID に必要な権限を与えることができます。
2. Web クライアントの認証済みユーザー ID を取得しない場合は、別名トランザクション用の標準ユーザー ID とする適切なユーザー ID を選択してください。 CICS のデフォルト・ユーザー ID を使用するのでない限り、選択したユーザー ID を要求の URIMAP 定義で指定するか、選択したユーザー ID を指定するアナライザー・プログラムを設定します。 標準ユーザー ID のユーザー・プロファイルがセキュリティー・マネージャーに定義されていることを確認してください。
3. CICS 領域にトランザクション接続セキュリティーが指定されているとすれば、別名トランザクション用の可能性のあるユーザー ID すべてが、トランザクションを接続するための権限を持つようにしておく必要があります。 ユーザー ID をすべて挙げると、Web クライアントのユーザー ID (取得されてオーバーライドされない場合)、URIMAP 定義またはアナライザー・プログラムで指定した標準ユーザー ID、または CICS のデフォルト・ユーザー ID そのものなどが考えられます。 トランザクション・セキュリティーを参照してください。
4. オプション: 別名トランザクションで使用されるリソースに対してリソース・レベルのセキュリティー検査を適用します。
   a. 別名トランザクションで使用されるすべての CICS リソースを識別し、その中のどのリソースが CICS 領域でのリソース・セキュリティー検査の対象になるかを決定します。 CICS Web サポート用のアプリケーション・プログラムで使用される可能性のあるいくつかのリソース、およびそれらのリソースに対するリソース・セキュリティー検査を制御するシステム初期設定パラメーターを以下に示します。
      - CICS 文書テンプレート (XDOC システム初期設定パラメーター)。
      - ビジネス・ロジックを実行するためにメイン・アプリケーション・プログラムによって呼び出される他のアプリケーション・プログラム (XPPT システム初期設定パラメーター)。
      - HTTP 要求シーケンス全体でアプリケーション状態を共用するために使用される一時記憶域キュー (XTST システム初期設定パラメーター)。

- CICS ファイル制御によって管理されるファイル (XFCT システム初期設定パラメーター)。

  zFS ファイルがアプリケーション・プログラムによって使用される場合、zFS ファイルのリソース・セキュリティー検査 (XHFS システム初期設定パラメーター) は適用されません。これらのファイルは、CICS 文書テンプレートとして定義されている場合に限りアプリケーション・プログラムが操作でき、この状態においては、CICS 文書テンプレート・セキュリティーがファイルへのアクセスを制御するためです。

  アナライザー・プログラムを使用する場合、アナライザー・プログラムが別名トランザクションのメインプログラムであり、したがってアナライザー・プログラムはリソース・セキュリティー検査の対象になりません (対象となるのはトランザクション接続セキュリティー検査のみです)。 ただし、ユーザー作成の Web アプリケーション・プログラムそのもの、およびユーザーが使用するなんらかのコンバーター・プログラムはすべて、別個のリソース・セキュリティー検査の対象となることに注意してください。 同様に、コンバーター・プログラムを使用し、アナライザー・プログラムを使用しない場合は、コンバーター・プログラムが別名トランザクションのメインプログラムですが、コンバーター・プログラムによって呼び出されるアプリケーション・プログラムは別個のリソース・セキュリティー検査の対象となります。

b. 別名トランザクションを接続する権限があるすべてのユーザー ID に、別名トランザクションで使用される保護されたリソースを使用する権限を与えます。

c. トランザクションの TRANSACTION リソース定義で RESSEC(YES) を指定します。

5. オプション: 別名トランザクションで使用される CICS システム・プログラミング・コマンドに対してコマンド・セキュリティー検査を適用するには、以下のようにします。

   a. コマンド・セキュリティーが CICS 領域でアクティブであることを確認します。 コマンド・セキュリティーは、XCMD システム初期設定パラメーターによってアクティブ化します。

   b. トランザクションに関連付けられたアプリケーション・プログラム、アナライザー・プログラム (使用する場合)、およびコンバーター・プログラム (使用する場合) で使用する CICS システム・プログラミング・コマンドを識別します。 コマンド・セキュリティー検査の対象となる CICS リソースには、コマンドのチェックリストが記載されています。

   c. 別名トランザクションを接続する権限があるすべてのユーザー ID に、別名トランザクションで使用されるコマンドを使用する権限を与えます。

   d. 別名トランザクションの TRANSACTION リソース定義で CMDSEC(YES) を指定します。

次のタスク

CICS 領域でセキュリティー検査が行われるようにするには、システム初期設定パラメーター SEC=YES を設定します。

文書テンプレートを使用する静的応答のリソース・レベル・セキュリティー

基本認証またはクライアント証明書認証を実装した状態で、特定の Web ページへのユーザーのアクセスを制御する場合は、Web クライアントの認証済みユーザー ID を使用して、静的応答を提供するために使用する個々の CICS 文書テンプレートへのアクセスを制御できます。

このタスクについて

URIMAP 定義で指定された CICS 文書テンプレートを使用して CICS Web サポートが送信する静的応答に対しては、デフォルトでリソース・セキュリティー検査が有効になります。

XRES システム初期設定パラメーターは、CICS 文書テンプレートのリソース・セキュリティーを制御します。 このパラメーターのデフォルトは YES であり、リソース・セキュリティーがアクティブです。 CICS 領域内で目的を問わず使用される CICS 文書テンプレートについて、リソース・セキュリティー検査を行わない場合は、このシステム初期設定パラメーターを NO に設定することで、検査を無効にすることができます。

すべての静的応答のトランザクションは、デフォルトの Web リスナー・トランザクションである CWXN か、または TCPIPSERVICE 定義の TRANSACTION 属性を使用して CWXN の代わりに指定した代替トランザクションです。 CICS 文書テンプレートの場合は、TRANSACTION リソース定義の RESSEC 属性でリソース・セキュリティー検査を制御することもできます。 CWXN の場合、CICS での提供時に RESSEC(YES) が指定されます。これは、リソース・セキュリティーがアクティブになるということです。静的応答に対してリソース・セキュリティー検査を使用しない場合、検査を非アクティブにする最適な方法は、TCPIPSERVICE 定義の CWXN を、プログラム DFHWBXN と RESSEC(NO) を指定した代替トランザクションに置き換えることです。 この設定により、CICS 文書テンプレートに対するリソース・セキュリティー検査が、静的応答についてのみ非アクティブになります HFSFILE 属性で指定した z/OS UNIX ファイルのセキュリティー検査については、RESSEC 属性で制御することはできません。

区分データ・セット、CICS プログラム、CICS ファイル、z/OS UNIX システム・サービス・ファイル、一時記憶域キュー、一時データ・キュー、出口プログラムなど、さまざまなソースから文書テンプレートを取得できます。 リソース・セキュリティー検査が文書テンプレートに対して行われる場合、文書テンプレートを提供するリソースについては、CICS は追加のセキュリティー検査を実行しません。CICS 領域で、そのタイプのリソースに対してリソース・セキュリティーが指定されていても、同様です。

CICS 文書テンプレートを使用して、静的応答のリソース・レベル・セキュリティーを設定するには、以下の手順を実行します。

手順

1. Web クライアントが使用する認証済みユーザー ID を識別します。これらの ID は、リソース・セキュリティー検査の基礎になります (アプリケーションが生成する応答の場合とは異なり、アナライザー・プログラムを使用してオーバー

ライドを提供することはできません)。 認証済みユーザー ID は、すでにセキュリティー・マネージャーにユーザー・プロファイルが定義されています。

2. 静的応答を提供するために使用する CICS 文書テンプレートを、すべて特定します。
3. XRES リソース・セキュリティー・パラメーターを使用したセキュリティーの説明に従って、CICS 領域で CICS 文書テンプレートのセキュリティーを実装します。 静的応答を提供するために使用する CICS 文書テンプレートのそれぞれについて、セキュリティー・マネージャーにプロファイルを定義するとともに、認証済みの各ユーザー ID に対して、適切な CICS 文書テンプレートにアクセスする権限を与える必要があります。
4. CWXN の TRANSACTION リソース定義、または CWXN の代わりに指定した代替トランザクションの TRANSACTION リソース定義に、RESSEC(YES) が指定されていることを確認します。 CICS 提供の CWXN には RESSEC(YES) が指定されていますが、TRANSACTION リソース定義では一般に RESSEC(NO) がデフォルトになります。 このステップにより、静的応答に対するリソース・セキュリティー検査がアクティブになります。そのため、Web クライアントがユーザー ID を提供するときは必ず、適切な権限をセットアップしておく必要があります。

次のタスク

CICS 領域でセキュリティー検査が行われるようにするには、システム初期設定パラメーターの SEC を YES に設定します。

CICS Web サポートでの SSL

HTTP とともに SSL (Secure Sockets Layer) を使用することで、暗号化とメッセージ認証、また、証明書を使用するクライアントおよびサーバーの認証を使用可能にできます。 SSL を使用するように CICS を構成してある場合は、HTTP サーバーとしての CICS、および HTTP クライアントとしての CICS の両方に対して、SSL の機能が使用可能になります。

セキュリティー・プロトコルのサポートでは、SSL が提供する機能を説明し、Configuring CICS to use SSLでは、CICS で SSL を機能させる方法を説明しています。

CICS が HTTP サーバーである場合、SSL を使用して Web クライアントとの相互作用を保護することができます。 CICS がクライアントの要求を受信するポートの TCPIPSERVICE 定義で、適切なセキュリティー・オプションを指定します。

SSL を使用することを指定するだけでなく、基本認証またはクライアント証明書を要求することもできます。 Web クライアントをさらに支援するには、クライアント証明書をクライアントが提供できるようにし、次に、CICS 環境における識別を提供するためにクライアント自体をセキュリティー・マネージャーに登録します。識別を提供するために必要に応じて、クライアントが自己登録または基本認証を使用することを許可することもできます。 これらのアクティビティーはすべて CICS によって処理されるので、アプリケーション生成応答を提供する場合は、アプリケーションがこの登録を処理する必要はありません。 CICS Web サポートの TCPIPSERVICE リソース定義の作成を参照してください。

CICS が HTTP クライアントの場合、サーバーは、いくつかの接続に対しては SSL を使用するように要求することがあります。 そのような場合は、以下のアクションの一部またはすべてを実行する必要があります。

- 接続のスキームとして、HTTPS を使用する。
- 接続に使用する暗号スイートのリストを提供する。これらは、接続を行うための WEB OPEN コマンドで使用する URIMAP 定義で指定することができます。
- クライアント証明書を提供する。 クライアント証明書はすべての SSL トランザクションで必須なわけではありませんが、サーバーが、特定のトランザクションに対してクライアント証明書を要求する可能性があります。 サーバーがクライアント証明書を要求した場合は、接続を行うための WEB OPEN コマンドで使用される URIMAP 定義、または WEB OPEN コマンドそのもので、適切な証明書のラベルを指定することができます。 クライアント証明書は、セキュリティー・マネージャーの鍵リングに保管する必要があります URIMAP 定義を使用しているのに、証明書ラベルが指定されていない場合、CICS 領域のユーザー ID の鍵リングで定義されているデフォルトの証明書が使用されます。

Application Transparent Transport Layer Security (AT-TLS) の概要

Application Transparent Transport Layer Security (AT-TLS) を使用して、CICS に代わってセキュア・ソケット・セッションを作成することができます。 CICS で Transport Layer Security (TLS) を実装する代わりに、AT-TLS はポリシー・エージェントでコーディングされるポリシー・ステートメントに基づいてデータを暗号化および暗号化解除します。 AT-TLS を使用してソケット・セッションを保護する場合、CICS SSL/TLS 開始パラメーター (KEYRING や MINTLSLEVEL/ENCRYPTION など) は必要でなくなります。TLS の実装が AT-TLS ポリシー・ステートメントによって行われ、すべての暗号化/暗号化解除が CICS アドレス・スペースの外部で行われるためです。

AT-TLS のモード

CICS ソケット接続において AT-TLS がアクティブである場合、CICS は平文 (暗号化されていないデータ) 送受信する一方、AT-TLS は TCP トランスポート層でデータを暗号化/暗号化解除します。 AT-TLS および AT-TLS ポリシーのセットアップについて詳しくは、「z/OS Communications Server: IP 構成ガイド」の『AT-TLS ポリシー構成』および「z/OS Communications Server: IP 構成解説書」の『ポリシー・エージェントおよびポリシー・アプリケーション』を参照してください。

CICS などのほとんどのアドレス・スペースでは、TCP/IP が代行するセキュリティー・ネゴシエーションと暗号化を認識する必要がありません。ただし、アドレス・スペースによっては、AT-TLS を認識したり、TCP/IP が行うセキュリティー機能を制御したりする必要が生じることがあります。例えば、アドレス・スペースがクライアント認証を要求するサーバーである場合、クライアント証明書やクライアント証明書に関連付けられたユーザー ID にアクセスする必要が生じることがあります。 **SSL(ATTLSAWARE)** を使って TCPIPSERVICE が定義されている場合、CICS は新しいクライアント接続に対して AT-TLS 照会を発行します。

表 48 で説明されているように、AT-TLS を利用する CICS などのアドレス・スペースを 3 つの異なるタイプ (AT-TLS 基本モード、AT-TLS 認識モード、および AT-TLS 制御モード) に分類できます。この場合は、サービスを認識する必要があるかどうか、またその必要がある場合にアドレス・スペースがセキュリティー機能をどの程度制御できるかに基づいて決まります。

AT-TLS 基本モードの場合、CICS などのアドレス・スペースが AT-TLS 状況をソケットに照会するために AT-TLS 呼び出しを発行することはありません。

AT-TLS 認識モードの場合、アドレス・スペースは AT-TLS 状況をソケットに照会するために AT-TLS 呼び出しを発行します。アドレス・スペースは、クライアント証明書や証明書ユーザー ID などの項目にアクセスできます。

AT-TLS 制御モードの場合、アドレス・スペースはソケットのセキュア・セッションを制御するために AT-TLS 呼び出しを発行します。

表 *48. AT-TLS* モードとその *CICS* サポートに関する詳細な説明

モード・タイプ	発行される **AT-TLS** 呼び出し	**AT-TLS** ポリシーでの **ApplicationControlled** 設定	**CICS TS for z/OS** でのサポート
AT-TLS 基本	アドレス・スペースは AT-TLS 呼び出しを発行しません	Off	すべての CICS リリース
AT-TLS 認識	アドレス・スペースは照会要求を発行します	Off	CICS TS V5.3 以降
AT-TLS 制御	アドレス・スペースは照会および制御の要求を発行します	On	該当する CICS リリースなし

AT-TLS 基本

基本モードの場合、アドレス・スペースは AT-TLS がデータの暗号化/暗号化解除を行っていることを認識しません。 AT-TLS 基本モードは CICS V5.2 以前の旧リリースで使用できる唯一のモードです。

基本モードの場合、CICS などのアドレス・スペースは、TCP/IP が TLS ハンドシェークや、ソケット上のメッセージ・フローの暗号化/暗号化解除を行っていることを検知しません。

AT-TLS の基本モードの場合、CICS はソケットを流れるデータを暗号化/暗号化解除しないため、CICS TCPIPSERVICE は **SSL(NO)** を使って定義します。このモードの欠点は、CICS がクライアント証明書にアクセスできないことです。これは CICS が、ソケットに関連付けられている証明書を認識できないためです。さらに、CICS がこの操作モードで HTTP リダイレクトを使用する場合、CICS はクライアント接続が HTTPS ではなく HTTP であると想定するため、障害が発生します。問題が発生するケースの例として、HTTP TCPIPSERVICE で **AUTHENTICATE (BASIC)** を使用しているときに CICS がユーザー・パスワードの期限切れを検出する場合があります。ユーザーに新規パスワードを要求するダイアログが起動します。元の HTTP 要求の再発信でスキームとして HTTPS ではなく HTTP が指定されるため、このダイアログの終わりにエラーが発生します。

AT-TLS 認識

CICS では、TCPIPSERVICE 定義で追加のオプション **SSL(ATTLSAWARE)** により AT-TLS 認識モードがサポートされます。 **SSL(ATTLSAWARE)** は、TCPIPSERVICE で **PROTOCOL(HTTP)** も指定した場合にのみ使用可能です。

TCPIPSERVICE で **SSL(ATTLSAWARE)** を定義すると、CICS は AT-TLS 照会を発行して、AT-TLS セキュリティー・ステータス、ネゴシエーション CIPHER スイート、パートナー証明書、派生 RACF ユーザー ID などの情報を取得します。

HTTP TCPIPSERVICE で **SSL(ATTLSAWARE)** を定義すると、CICS はクライアント証明書およびそれに関連付けられている RACF USERID にアクセスできます。その結果、証明書に関連するすべての TCPIPSERVICE AUTHENTICATE オプション (`CERTIFICATE | AUTOREGISTER | AUTOMATIC`) が **SSL(ATTLSAWARE)** によってサポートされます。

TCPIPSERVICE で **SSL(ATTLSAWARE)** を定義すると、CICS はクライアントが HTTPS 接続を使用していることを検出します。つまり、AT-TLS を基本モードで使用するときに発生する可能性のあるリダイレクト障害 (パスワード期限切れダイアログなど) は、**SSL(ATTLSAWARE)** を使用することで修正されます。

AT-TLS 制御

CICS では AT-TLS 制御モードがサポートされません。

AT-TLS モード (タイプ) について詳しくは、「z/OS Communications Server: IP 構成ガイド」の『Application Transparent Transport Layer Security のデータ保護』を参照してください。

CICS AT-TLS 照会

SSL(ATTLSAWARE) TCPIPSERVICE でのすべての新規クライアント接続に対して、AT-TLS 属性を抽出するよう照会が出されます。 **SSL(ATTLSAWARE)** を使って定義された CICS TCPIPSERVICE への新しい接続をクライアントが確立するときにクライアントはこの照会をトリガーし、CICS は新しいクライアント接続を受け入れます。

- 接続状況 (AT-TLS 保護/非保護)。
- クライアント認証タイプ (`NONE | PASSTHRU |FULL | REQUIRED | SAFCHECK`)。
- クライアント証明書。クライアント証明書が存在する場合は、CICS 証明書リポジトリーにそれが保管されます。 **`EXEC CICS EXTRACT CERTIFICATE`** を発行するアプリケーションによって、証明書の属性を後で取り出すことができます。
- クライアント証明書 USERID を TCPIPSERVICE の AUTHENTICATE オプションと共に使用して、新しい Web タスクのセキュリティー・コンテキストを確立することができます。例えば **AUTHENTICATE(CERTIFICATE)** の場合、Web 要求を処理できるようにするには CERTIFICATE USERID が存在する必要があります。
- ネゴシエーション CIPHER 番号。この番号は、パフォーマンス・モニター・レコード (既存のフィールド **SOCIPHER** の中) に登録されます。

AT-TLS および CICS TCPIPService の構成

AT-TLS ポリシーと CICS TCPIPService の有効な組み合わせがいくつかあります。この表では、これらの有効な組み合わせと、CICS で予想される結果について示します。さらに、無効な組み合わせも示します。

表 *49. AT-TLS* と *CICS TCPIPService* の組み合わせ

ポートに関する **AT-TLS** ポリシー	**CICS TCPIPSERVICE**	有効な組み合わせ	**CICS** で予想される結果
`HandShakeRole=Server`	**SSL (NO|ATTLSAWARE)**	はい	接続が正常に確立されました。 使用可能なクライアント証明書がありません。

`HandShakeRole=Server`
 	
または
 	
`HandShakeRole=ServerWithClientAuth`
	SSL(YES | ClientAuth)	いいえ	CICS は DFHWB0732 を発行して接続を拒否します。

`HandShakeRole=ServerWithClientAuth` および `ClientAuthType=(REQUIRED | SAFCHECK)`	**SSL(ATTLSAWARE)**	はい	接続が正常に確立されました。 クライアント証明書が使用可能です。
`HandShakeRole=ServerWithClientAuth` および `ClientAuthType=(FULL)`	**SSL(ATTLSAWARE)**	はい	接続が正常に確立されました。 クライアントがサーバーにクライアント証明書を送信する場合は、証明書が使用可能です。
`HandShakeRole=ServerWithClientAuth` および任意の ClientAuthType	**SSL(NO)**	はい	接続が正常に確立されました。 使用可能なクライアント証明書がありません。
ポートに関する AT-TLS ポリシーなし、またはクライアントが AT-TLS ポリシーの使用から除外されている。	**SSL(ATTLSAWARE)**	いいえ	CICS は HTTP 403 エラーで接続を拒否します。 CICS は最初の接続に対して DFHSO0147 を発行し、それぞれの非セキュア接続に対して DFHWB0365 を発行します。
`HandShakeRole=ServerWithClientAuth` および `ClientAuthType=PASSTHRU`	**SSL(ATTLSAWARE)**	いいえ	この構成ではクライアント証明書の検証がバイパスされるため、CICS は接続を拒否します。 CICS は DFHSO0149 を発行し、TCPIPService がクローズされます。

表 49. *AT-TLS* と *CICS TCPIPService* の組み合わせ (続き)

ポートに関する **AT-TLS** ポリシー	**CICS TCPIPSERVICE**	有効な組み合わせ	**CICS** で予想される結果
構成されている AT-TLS ポリシーなし。	**SSL(YES | ClientAuth)**	はい	CICS は記述に従って SSL 接続を処理します。

注: TCPIPService の場合、AT-TLS `HandShakeRole=Client` という構成は間違っています。

SSL(ATTLSAWARE) を指定した TCPIPSERVICE を使用する場合、CICS はすべての接続が AT-TLS によって暗号化され保護されることを想定します。無保護のクライアント接続が到着した場合、HTTP 403 エラーで拒否されます。また、CICS はメッセージ **DFHWB0365** を使ってエラーをログに記録します。

CICS では、`ClientAuthType=PassThru` を使用する AT-TLS ポリシーがサポートされません。この構成はクライアント証明書の検証をバイパスしますが、CICS ではこれが許容されません。クライアント接続を受け取るときに、このタイプのクライアント認証が使われていることを CICS が検出すると、クライアントとの接続が閉じて TCPIPSERVICE が閉じます。このエラーが検出されると、メッセージ DFHSO0149 がコンソールに書き出されます。

AT-TLS の使用中に SSL 環境とキャッシュをリフレッシュする方法

SSL(ATTLSAWARE) を指定した TCPIPSERVICE を使用する場合、**PERFORM SSL REBUILD** コマンドは適用されません。

そのため、CICS で使用される SSL 環境と証明書をリフレッシュするには、以下の手順に従ってください。

1. AT-TLS ポリシーで定義された鍵リングに新規証明書を配置します。
2. セキュリティー・マネージャー内の鍵リングをリフレッシュします。
3. この CICS トラフィックのポリシー・ルールの **EnvironmentUserInstance** 値を変更または追加します。
4. 以下のいずれかの変更コマンドを発行します。

   ```
   F PAGENT,REFRESH
   ```

   または

   ```
   F PAGENT,UPDATE
   ```

AT-TLS の診断

AT-TLS の問題を診断するいくつかのツールがあります。

AT-TLS の問題の診断については、「z/OS Communications Server: IP Diagnosis Guide」の『Diagnosing Application Transparent Transport Layer Security (AT-TLS)』を参照してください。

AT-TLS メッセージに含まれる戻りコードは、問題の診断に役立ちます。 5000 未満の戻りコードはシステム SSL から出されます。戻りコードについて詳しくは、「z/OS Cryptographic Services: System SSL Programming」の『SSL function return codes』を参照してください。

AT-TLS のソケット・ドメイン・トレース・ポイント

AT-TLS に関連するソケット・ドメイン・トレース・ポイントをリストします。詳しくは、ソケット・ドメインのトレース・ポイントを参照してください。

- SO 0CAC (レベル 1)
- SO 0CAB (EXC)
- SO 0CA9 (レベル 2)
- SO 0CAA (レベル 2)

診断メッセージ

AT-TLS に関連するメッセージ DFHSO0147 および DFHSO0149 について詳しくは、DFHSO メッセージを参照してください。さらに、メッセージ DFHWB0365 について詳しくは、DFHWB メッセージを参照してください。

エラーの発生が CICS によって検出されると、CICS から次のようないくつかの診断メッセージが出されます。

DFHWB0365

date time applid tranid SSL(ATTLSAWARE) を使って定義された TCPIPSERVICE にクライアントが接続しますが、接続は AT-TLS で保護されません。ホスト IP アドレス: *hostaddr*。クライアントの IP アドレス: *clientaddr*。TCPIPSERVICE: *tcpipservice*。

DFHSO0147 W

applid ATTLSAWARE TCPIPSERVICE *tcpipservice* に対して非セキュア・クライアント接続を受け取りました。クライアントの IP アドレス: *clientaddr*。TTLS_IOCTL *value X'ttlsioctl'*。

DFHSO0149 W

applid ATTLSAWARE TCPIPSERVICE *tcpipservice* に対して、**CLIENTAUTHTYPE(PASSTHRU)** を使用するクライアント接続が検出されました。 TTLS_IOCTL *value X'ttlsioctl'*。 TCPIPSERVICE はクローズされます。

AT-TLS 接続でのエラーの 2 つの例を以下に示します。

1. クライアントが AT-TLS セキュア・ポートに接続すると、以下の診断が表示されます (これは *HandShakeRole ServerWithClientAuth* および *ClientAuthType Required* を使って構成されます)。この構成では、クライアントが証明書を提供する必要があります。このケースではクライアントが証明書の提供に失敗します。 AT-TLS メッセージ・ログに示される情報は次のとおりです。

   ```
   EZD1287I TTLS Error RC:  403 Initial Handshake  034
   LOCAL: ::FFFF:9.20.5.0..25931
   REMOTE: ::FFFF:9.174.17.124..50077
   JOBNAME: SSYCZCCM RULE: CICSD
   USERID: HORN GRPID: 0000000D ENVID: 00000013 CONNID: 00395D99
   ```

   戻りコード 403 はシステム SSL エラーであり、エラー GSK_ERR_NO_CERTIFICATE (パートナーから証明書を受け取らなかった) に対応し

ます。 CICS ログには何も表示されません。この接続は、AT-TLS によって拒否されるため、CICS で受信されません。

2. TCPIPSERVICE ポートが AT-TLS によって保護されていない場合、**SSL(ATTLSAWARE)** を使って定義された TCPIPSERVICE にクライアント接続を行うと、以下の診断が表示されます。今回、AT-TLS のポリシーが適用されないポートへの接続をクライアントが行うため、AT-TLS 診断はありません。しかし、CICS はクライアント接続が AT-TLS で保護されていないことを検出し、以下のメッセージを出します。
   - DFHSO0147 W IY2CZCCM 041 - ATTLSAWARE TCPIPSERVICE ATTLS2 に対して非セキュア・クライアント接続を受け取りました。クライアントの IP アドレス: 9.174.17.124。TTLS_IOCTL 値 X'0100000102010000'
   - DFHWB0365 06/23/2015 10:14:22 IY2CZCCM CWXN - SSL(ATTLSAWARE) を使って定義された TCPIPSERVICE にクライアントが接続しますが、接続は AT-TLS で保護されません。ホスト IP アドレス: 9.20.5.0。クライアントの IP アドレス: 9.174.17.124。TCPIPSERVICE: ATTLS2。

   最初のメッセージは TCPIPSERVICE に対して一度だけ発行されます。 2 番目のメッセージは、クライアント接続の際、AT-TLS によって接続が保護されていないことを CICS が検出するたびに発行されます。

CICS SSL から AT-TLS への移行

インバウンド・ソケット接続の既存の CICS Transport Layer Security (TLS) (SSL) 実装を Application Transparent Transport Layer Security (AT-TLS) に移行することができます。

インバウンド・ソケット接続の TLS ハンドシェークを実行するために CICS を使用して TLS (SSL) 環境を確立する場合、ハンドシェークで使用される属性は、2 つのソース (領域レベル SIT パラメーターと TCPIPSERVICE リソース属性) から抽出されます。

以下の 2 つの表は、TLS ハンドシェークとそれに対応する AT-TLS レベルの項目に使用される CICS SIT パラメーターと TCPIPSERVICE リソース属性を示しています。

表 *50. SIT* パラメーターおよび対応する *AT-TLS* 項目

SIT パラメーター	対応する **AT-TLS** 項目
MINTLSLEVEL	TLSv1 TLSv1.1 TLSv1.2
ENCRYPTION (非推奨: **MINTLSLEVEL** を使用)	TLSV1 TLSV1.1 TLSV1.2
KEYRING	TTLSKeyRingParms
CRLPROFILE	TTLSGskLdapParms

表 *50. SIT* パラメーターおよび対応する *AT-TLS* 項目 (続き)

SIT パラメーター	対応する **AT-TLS** 項目
SSLDELAY	GSK_V3_SESSION_TIMEOUT
MAXSSLTCBS	AT-TLS で構成することはできません。TCB 番号は動的に増加します。
SSLCACHE=SYSPLEX	GSK_SYSPLEX_SIDCACHE ON
NISTSP800131A=CHECK	FIPS140 ON

表 *51. TCPIPSERVICE* リソース属性および対応する *AT-TLS* 項目

TCPIPSERVICE リソース属性	対応する **AT-TLS** 項目
SSL=YES	HandShakeRole Server
SSL=CLIENTAUTH	HandShakeRole ServerWithClientAuth、ClientAuthType FULL ClientAuthType REQUIRED と ClientAuthType SAFCHECK もサポートされています。
CERTIFICATE	CertificateLabel
CIPHERS	TTLSCipherParms

以下の例は、既存の CICS TLS 実装を AT-TLS に移行してから CICS TLS 実装を削除する方法を示しています。

例 1: TLS/SSL サーバー認証の AT-TLS ポリシー・ルール

CICS を使用してインバウンド HTTP 接続を保護するための構成例では、TCPIPSERVICE リソースでの単純なサーバー認証を使用することもできます (SSL(YES))。この構成では、クライアント証明書はサポートされません。以下の例は、単純なサーバー認証用の CICS-TLS 環境を確立するために必要な CICS 構成ステートメントを示しています。

CICS 始動パラメーターは以下のとおりです。

```
MINTLSLEVEL=TLS10  (or its deprecated equivalent ENCRYPTION=STRONG)
KEYRING=CICSKeyRing (includes the certificate named CICS-2048-certificate)
SSLDELAY=600
MAXSSLTCBS=8
SSLCACHE=CICS
NISTSP800131A=NOCHECK
```

SSL 関連の TCPIPSERVICE リソース属性は、以下のとおりです。

```
TCpipservice   : HTTPSSL
GROup          : JULESWEB
DEScription  ==> CICS WEB TCPIPSERVICE WITH SSL SUPPORT
POrtnumber   ==> 25008
STatus       ==> Open
PROtocol     ==> Http
SSl          ==> Yes
CErtificate  ==> CICS-2048-certificate
CIphers      ==> 35363738392F303132330A1613100D15120F0C
AUthenticate ==> Basic
```

CICS ではなく AT-TLS を使用してインバウンド HTTP 接続を保護する場合は、次の AT-TLS ポリシーを使用して、TCPIPSERVICE リソース定義を SSL(NO) または SSL(ATTLSAWARE) に更新することもできます。

注: 以下の AT-TLS ポリシーの例では、TLSV1.2 オプションを使用します。このオプションは、z/OS 2.1 以降でサポートされています。 TLSV1.2 オプションを使用すると、パフォーマンスの最適化に役立ちます。さらに、NIST SP800-131A に準拠する必要がある場合は、そのオプションを使用することが前提条件です。
NIST 標準の詳細については、NIST Computer Security Resource Center (crc.nist.gov)を参照してください。

HTTPSSL という名前の TCPIPSERVICE 用の CICS 環境を複製する AT-TLS 構成は、以下のとおりです。

```
TTLSRule SIMPLECICS
{
LocalPortRange 25008
Direction Inbound
Priority 256
TTLSGroupActionRef CICSGroupAct1
TTLSEnvironmentActionRef CICSEnvironmentAct1
}
TTLSGroupAction CICSGroupAct1
{
TTLSEnabled On
FIPS140 off
}
TTLSEnvironmentAction CICSEnvironmentAct1
{
HandShakeRole Server
TTLSKeyRingParmsRef CICSKeyRingParms1
TTLSCipherParmsRef CICSCipherParms1
TTLSEnvironmentAdvancedParmsRef CICSEnvAdvParms1
TTLSGskAdvancedParmsRef CICSGskAdvParms1
}
TTLSKeyRingParms CICSKeyRingParms1
{
Keyring CICSKeyRing
}
TTLSCipherParms CICSCipherParms1
{
V3CipherSuites TLS_RSA_WITH_AES_256_CBC_SHA
V3CipherSuites TLS_DH_DSS_WITH_AES_256_CBC_SHA
V3CipherSuites TLS_DH_RSA_WITH_AES_256_CBC_SHA
V3CipherSuites TLS_DHE_DSS_WITH_AES_256_CBC_SHA
V3CipherSuites TLS_DHE_RSA_WITH_AES_256_CBC_SHA
V3CipherSuites TLS_RSA_WITH_AES_128_CBC_SHA
V3CipherSuites TLS_DH_DSS_WITH_AES_128_CBC_SHA
V3CipherSuites TLS_DH_RSA_WITH_AES_128_CBC_SHA
V3CipherSuites TLS_DHE_DSS_WITH_AES_128_CBC_SHA
V3CipherSuites TLS_DHE_RSA_WITH_AES_128_CBC_SHA
V3CipherSuites TLS_RSA_WITH_3DES_EDE_CBC_SHA
V3CipherSuites TLS_DHE_RSA_WITH_3DES_EDE_CBC_SHA
V3CipherSuites TLS_DHE_DSS_WITH_3DES_EDE_CBC_SHA
V3CipherSuites TLS_DH_RSA_WITH_3DES_EDE_CBC_SHA
V3CipherSuites TLS_DH_DSS_WITH_3DES_EDE_CBC_SHA
V3CipherSuites TLS_DHE_RSA_WITH_DES_CBC_SHA
V3CipherSuites TLS_DHE_DSS_WITH_DES_CBC_SHA
V3CipherSuites TLS_DH_RSA_WITH_DES_CBC_SHA
V3CipherSuites TLS_DH_DSS_WITH_DES_CBC_SHA
}
TTLSEnvironmentAdvancedParms CICSEnvAdvParms1
{
```

```
SSLv3 Off
TLSV1 On
TLSV1.1 On
TLSV1.2 On
CertificateLabel CICS-2048-certificate
}
TTLSGskAdvancedParms CICSGskAdvParms1
{
GSK_SYSPLEX_SIDCACHE off
GSK_V3_SESSION_TIMEOUT 600

}
```

この AT-TLS ポリシーをアクティブ化する前に、CICS TCPIPSERVICE リソースを以下のように変更します。

```
TCpipservice   : HTTPSSL
GROup          : JULESWEB
DEScription  ==> CICS WEB TCPIPSERVICE WITH AT-TLS SSL SUPPORT
POrtnumber   ==> 25008
STatus       ==> Open
PROtocol     ==> Http
SSl          ==> NO|ATTLSAWARE
CErtificate  ==>
CIphers      ==>
AUthenticate ==> Basic
```

SSL が NO に設定されている場合、CICS は、AT-TLS がインバウンド・クライアント接続を保護しているかどうかを検査しません。

SSL が ATTLSAWARE に設定されている場合、CICS は、AT-TLS がインバウンド・クライアント接続を保護しているかどうかを検査します。クライアント接続が AT-TLS によって保護されない場合は、HTTP 403 エラーで拒否され、メッセージ DFHWB0365 が CICS ログに書き込まれます。

また、SSL が ATTLSAWARE に設定されている場合、CICS はクライアント証明書が存在するかどうか検査します。前述の例の AT-TLS 構成では、クライアント証明書の使用はサポートされていません。したがって、クライアント証明書を必要とする AUTHENTICATE オプションが TCPIPSERVICE 定義で指定されていないことを確認してください。前述の例の TCPIPSERVICE リソースでは、AUTHENTICATE(BASIC) が指定されていますが、この場合は、クライアント証明書を必要としません。

AT-TLS ポリシーがアクティブで、TCPIPSERVICE リソースの再定義により SSL 属性が削除される場合は、関連する SSL SIT パラメーターをすべて削除することもできます。ただし、まず CICS 領域内のどの項目も、これらのパラメーターに依存していないことを確認してください。

CICS-SSL システムが **NISTSP800131A=CHECK** を使って開始される場合、CICS は **MINTLSLEVEL=TLS12** を設定し、さらに FIPS140 をオンに設定します。AT-TLS POLICY の構成例でこれらの設定を反映するには、以下のように変更します。

```
TTLSGroupAction CICSGroupAct1
{
TTLSEnabled On
FIPS140 on
}

TTLSEnvironmentAdvancedParms CICSEnvAdvParms1
{
SSLv3 Off
```

```
TLSV1 Off
TLSV1.1 Off
TLSV1.2 On
CertificateLabel CICS-2048-certificate
}
```

例 2: TLS/SSL クライアント認証の AT-TLS ポリシー・ルール

CICS を使用してインバウンド HTTP 接続を保護するための構成例では、TCPIPSERVICE リソースでのクライアント認証を使用することもできます (SSL(CLIENTAUTH))。この構成では、クライアント証明書がサポートされます。以下の例は、クライアント認証のための CICS-TLS 環境を確立するのに必要な CICS 構成ステートメントを示しています。

CICS 始動パラメーターは以下のとおりです。

```
MINTLSLEVEL=TLS10  (or its deprecated equivalent ENCRYPTION=STRONG)
KEYRING=CICSKeyRing (includes the certificate named CICS-2048-certificate)
SSLDELAY=600
MAXSSLTCBS=8
SSLCACHE=CICS
NISTSP800131A=NOCHECK
```

SSL 関連の TCPIPSERVICE リソース属性は、以下のとおりです。

```
TCpipservice   : CLAUTH
GROup          : JULESWEB
DEScription  ==> CICS Web TCPIPSERVICE with SSL CLIENTAUTH support
POrtnumber   ==> 25009
STatus       ==> Open
PROtocol     ==> Http
SSl          ==> Clientauth
CErtificate  ==> CICS-2048-certificate
CIphers      ==> 35363738392F303132330A1613100D15120F0C
AUthenticate ==> Certificate
```

CICS ではなく AT-TLS を使用してインバウンド HTTP 接続を保護する場合は、次の AT-TLS ポリシーを使用して、TCPIPSERVICE リソース定義を更新して SSL(ATTLSAWARE) を使用することもできます。

注: 以下の AT-TLS ポリシーの例では、TLSV1.2 オプションを使用します。このオプションは、z/OS 2.1 以降でサポートされています。 TLSV1.2 オプションを使用すると、パフォーマンスの最適化に役立ちます。さらに、NIST SP800-131A に準拠する必要がある場合は、そのオプションを使用することが前提条件です。
NIST 標準の詳細については、NIST Computer Security Resource Center (crc.nist.gov)を参照してください。

CLAUTH という名前の TCPIPSERVICE 用の CICS 環境を複製する AT-TLS クライアント認証構成は、以下のとおりです。

```
TTLSRule CLIENTAUTHCICS
{
LocalPortRange 25009
Direction Inbound
Priority 256
TTLSGroupActionRef CICSGroupAct2
TTLSEnvironmentActionRef CICSEnvironmentAct2
}
TTLSGroupAction CICSGroupAct2
{
TTLSEnabled On
```

```
FIPS140 off
}
TTLSEnvironmentAction CICSEnvironmentAct2
{
HandShakeRole ServerWithClientAuth
TTLSKeyRingParmsRef CICSKeyRingParms2
TTLSCipherParmsRef CICSCipherParms2
TTLSEnvironmentAdvancedParmsRef CICSEnvAdvParms2
TTLSGskAdvancedParmsRef CICSGskAdvParms2
}
TTLSKeyRingParms CICSKeyRingParms2
{
Keyring CICSKeyRing
}
TTLSCipherParms CICSCipherParms2
{
V3CipherSuites TLS_RSA_WITH_AES_256_CBC_SHA
V3CipherSuites TLS_DH_DSS_WITH_AES_256_CBC_SHA
V3CipherSuites TLS_DH_RSA_WITH_AES_256_CBC_SHA
V3CipherSuites TLS_DHE_DSS_WITH_AES_256_CBC_SHA
V3CipherSuites TLS_DHE_RSA_WITH_AES_256_CBC_SHA
V3CipherSuites TLS_RSA_WITH_AES_128_CBC_SHA
V3CipherSuites TLS_DH_DSS_WITH_AES_128_CBC_SHA
V3CipherSuites TLS_DH_RSA_WITH_AES_128_CBC_SHA
V3CipherSuites TLS_DHE_DSS_WITH_AES_128_CBC_SHA
V3CipherSuites TLS_DHE_RSA_WITH_AES_128_CBC_SHA
V3CipherSuites TLS_RSA_WITH_3DES_EDE_CBC_SHA
V3CipherSuites TLS_DHE_RSA_WITH_3DES_EDE_CBC_SHA
V3CipherSuites TLS_DHE_DSS_WITH_3DES_EDE_CBC_SHA
V3CipherSuites TLS_DH_RSA_WITH_3DES_EDE_CBC_SHA
V3CipherSuites TLS_DH_DSS_WITH_3DES_EDE_CBC_SHA
V3CipherSuites TLS_DHE_RSA_WITH_DES_CBC_SHA
V3CipherSuites TLS_DHE_DSS_WITH_DES_CBC_SHA
V3CipherSuites TLS_DH_RSA_WITH_DES_CBC_SHA
V3CipherSuites TLS_DH_DSS_WITH_DES_CBC_SHA
}
TTLSEnvironmentAdvancedParms CICSEnvAdvParms2
{
SSLv3 Off
TLSV1 On
TLSV1.1 On
TLSV1.2 On
CertificateLabel CICS-2048-certificate
ClientAuthType Full
}
TTLSGskAdvancedParms CICSGskAdvParms2
{
GSK_SYSPLEX_SIDCACHE off
GSK_V3_SESSION_TIMEOUT 600
}
```

この AT-TLS ポリシーの例をアクティブ化する前に、CICS TCPIPSERVICE リソース定義を以下のように変更します。

```
TCpipservice   : CLAUTH
GROup          : JULESWEB
DEScription  ==> CICS Web TCPIPSERVICE with SSL CLIENTAUTH support
POrtnumber   ==> 25009
STatus       ==> Open
PROtocol     ==> Http
SSl          ==> ATTLSAWARE
CErtificate  ==>
CIphers      ==>
AUthenticate ==> Certificate
```

この例では、SSL を ATTLSAWARE に設定して、CICS で AT-TLS からクライアント証明書を取り出す必要があります。これは、AUTHENTICATE が CERTIFICATE (クライアント証明書が必要) に設定されているためです。クライアント接続が AT-TLS によって保護されない場合は、HTTP 403 エラーで拒否され、メッセージ DFHWB0365 が CICS ログに書き込まれます。

SSL(ATTLSAWARE) が使われている場合、CICS はクライアント証明書を検査します。この検査により RACF USERID にマップされる場合、CICS はこの USERID の下で Web ユーザー・トランザクションを実行します。

前述の例の AT-TLS ポリシーは、ClientAuthType Full で定義されています。この ClientAuthType は、SSL 環境と、CICS で SSL を使用する場合に発生するハンドシェーク動作を複製します。しかし、CICS では、ClientAuthType Required と ClientAuthType SAFCheck もサポートされます。

CICS は、ClientAuthType PassThru の使用をサポートしていません。ClientAuthType PassThru を使用して TCPIPSERVICE ポートが構成され、SSL(ATTLSAWARE) を使用して TCPIPSERVICE リソースが定義されている場合、最初のクライアント接続の到着時に、CICS はサポートされない構成を検出します。次に CICS は、TCPIPSERVICE を閉じてメッセージ DFHSO0149 を発行します。

AT-TLS ポリシーがアクティブで、TCPIPSERVICE リソースの再定義により SSL 属性が削除される場合は、関連する SSL SIT パラメーターをすべて削除することもできます。ただし、まず CICS 領域内のどの項目も、これらのパラメーターに依存していないことを確認してください。

CICS-SSL システムが **NISTSP800131A=CHECK** を使って開始される場合、CICS は **MINTLSLEVEL=TLS12** を設定し、さらに FIPS140 をオンに設定します。AT-TLS POLICY の構成例でこれらの設定を反映するには、以下のように変更します。

```
TTLSGroupAction CICSGroupAct2
{
TTLSEnabled On
FIPS140 on
}

TTLSEnvironmentAdvancedParms CICSEnvAdvParms2
{
SSLv3 Off
TLSV1 Off
TLSV1.1 Off
TLSV1.2 On
CertificateLabel CICS-2048-certificate
ClientAuthType Full
}
```

Atom フィードのセキュリティー

CICS Web サポートは、Atom コレクションおよび (必要な場合に) Atom フィードへの Web クライアントのアクセスを制御するための適切なセキュリティー・プロトコルおよび認証方式を提供します。ユーザーは、CICS リソースおよびコマンド・セキュリティーを使用して、Atom フィードまたはコレクションを配信するために使用するリソースを保護できます。

RFC 5023 では、認証を使用して Atom コレクションを保護することが推奨されています。 Atom フィードのデータを編集可能なコレクションとして使用可能にすると、Web クライアントが新しいエントリーの挿入、既存のエントリーの変更、またはエントリーの削除を行えるようになります。 そのため、Web クライアントの ID を検証し、信頼できるクライアントにのみコレクションへのアクセスを許可する必要があります。特に、コレクションにビジネス・データを含めている場合は注意してください。 Web クライアントが編集できない通常の Atom フィードは、通常、セキュリティーの制限なく任意の加入者が使用できるようになります。ただし、機密のビジネス・データが含まれている、または特定のユーザーのみを対象にした Atom フィードについては、アクセスを制限する必要があります。

RFC 4287 および RFC 5023 には、Atom 文書でのデジタル署名および暗号化の使用に関する説明があります。 CICS では Atom 文書のデジタル署名および暗号化はサポートされていませんが、RFC 4287 と準拠するため、署名が含まれる Atom 文書が拒否されることはありません。

CICS Web サポートには、Atom フィードまたはコレクションを無許可のアクセスや更新から保護するために使用できる以下のセキュリティー機能があります。

SSL または **TLS** セキュリティー・プロトコル
: RFC 5023 では、コレクションに対する最小レベルのセキュリティー・プロトコルとして、 Transport Layer Security (TLS) 1.0 を使用することが推奨されています。 CICS でサポートされるセキュリティー・プロトコルのリストについては、セキュリティー・プロトコルのサポートを参照してください。

HTTP 基本認証
: RFC 5023 では、コレクションに関する最小レベルの認証として、HTTP 基本認証を使用することが推奨されています。

クライアント証明書認証
: クライアント証明書認証はより安全なクライアント認証方式で、信頼のおける第三者機関 (または認証局) が発行し、 SSL 暗号化を使用して送信されるクライアント証明書が使用されます。 このメカニズムについては、SSL 認証を参照してください。

CICS Web サポートのこれらの機能をセットアップする際、CICS が Atom フィードまたはコレクションの Web クライアント要求を受け取るポートの TCPIPSERVICE 定義の属性を使用して、Atom フィードまたはコレクションにそれらの機能を適用することができます。 CICS Web サポート用に SSL サポートをセットアップする方法については、Configuring CICS to use SSLを参照してください。

第 13 章 Web サービスを保護するためのサポート

CICS Transaction Server for z/OS では、SOAP および JSON メッセージを保護するために使用できる多数の関連テクノロジーがサポートされます。

これらのテクノロジーのいくつかは HTTP プロトコルに含まれており、SOAP と JSON の両方に等しく適用されます。 Web Services Security (WSS): SOAP Message Security 1.0 仕様を使用する、SOAP でのみ使用可能なものもあります。共用される TCP/IP および HTTP セキュリティー・オプションについては、Security for TCP/IP clientsおよびCICS Web サポートのセキュリティーを参照してください。

SAML アサーションの使用については、SAML 用の CICS の構成を参照してください。

SOAP Web サービスのセキュリティー

Web Services Security (WSS): SOAP Message Security 1.0 は、SOAP メッセージを保護し認証するためのセキュリティー・トークン およびデジタル署名 の使用について記述しています。詳しくは、Web Services Security: SOAP Message Security 1.0 仕様を参照してください。 Web Services Security: SOAP Message Security 1.0

Web Services Security は、メッセージが不正に開示されたり、不正または気付かれずに変更されたりしないようにすることによって、SOAP メッセージのプライバシー と保全性 を保護します。 WSS は、メッセージ内の XML エレメントをデジタル署名および暗号化することでこのような保護を提供します。 保護できるエレメントは、本体または本体やヘッダー内のエレメントです。 SOAP メッセージ内の異なるエレメントに対して異なるレベルの保護を適用することができます。

Web Services Trust Language (WS-Trust) 仕様は、セキュリティー・トークンを要求して発行するためのフレームワークを提供し、Web サービス・リクエスターと Web サービス・プロバイダー間の信頼関係を管理することによって、Web Services Security をさらに拡張します。 SOAP メッセージの認証をこのように拡張することによって、Web サービスは、信頼のおける第三者機関を使用して、さまざまなタイプのセキュリティー・トークンを検証および交換することができます。 この第三者機関は、*Security Token Service* (STS) と呼ばれます。 Web Services Trust Language の詳細については、Web Services Trust Language 仕様を参照してください。

CICS Transaction Server for z/OS では、パイプライン内で CICS 提供のセキュリティー・ハンドラーを使用することにより、これらの仕様がサポートされます。

- アウトバウンド・メッセージでは、CICS は SOAP 本体全体にデジタル署名して暗号化するためのサポートを提供します。また CICS は、STS を使用して、さまざまなタイプのセキュリティー・トークンでユーザー名トークンを交換することができます。

© Copyright IBM Corp. 1989, 2018

- インバウンド・メッセージでは、CICS は、本体または本体やヘッダーのエレメントが暗号化されているかデジタル署名されているメッセージをサポートします。また CICS は、STS を使用してセキュリティー・トークンを交換して検証することもできます。

CICS は、個別の Trust クライアント・インターフェースを提供して、CICS セキュリティー・ハンドラーを使用せずに STS とデータをやり取りすることもできます。

注**:** Web Services Security は SP800-131A に準拠しない可能性があります。 Web Services Security はパイプライン内にハンドラーを追加することで構成され、CICS はカスタム作成ハンドラーでの処理の制御を行うことができません。デジタル署名を使用する場合、アルゴリズム `dsa-sha1` および `rsa-sha1` だけを指定できます。これらのアルゴリズムは SP800-131A に準拠していません。 SOAP 本体の暗号化に使用できる、2 つの鍵を使った Triple-DES 暗号化アルゴリズムもまた準拠していません。

Web Services Security を実装するための前提条件

Web Services Security を実装するには、次のような更新を CICS 領域に適用する必要があります。IBM XML Toolkit for z/OS v1.10 をインストールし、APAR OA14956 を適用し、3 つのライブラリーを DFHRPL 連結に追加します。

このタスクについて

Web Services Security を実装する前に、以下の手順を完了してください。

手順

1. 無料の IBM XML Toolkit for z/OS v1.10 をインストールします。 これは、サイト http://www.ibm.com/servers/eserver/zseries/software/xml/ からダウンロードできます。 バージョン 1.10 をインストールする必要があります。それより後のバージョンは、CICS の Web Services Security サポートでは機能しません。
2. ICSF APAR OA14956 が z/OS にまだインストールされていない場合は、これを適用します。
3. 次のライブラリーを DFHRPL 連結に追加します。
   - *hlq*.SIXMLOD1。*hlq* は、XML ツールキットの高位修飾子です。
   - *hlq*.SCEERUN。*hlq* は、言語環境の高位修飾子です。
   - *hlq*.SDFHWSLD。*hlq* は、 CICS インストール済み環境の高位修飾子 (例えば、CICSTS55) です。

   最初の 2 つのライブラリーには、実行時にセキュリティー・ハンドラーで必要な DLL が含まれています。 IXM4C57 は XML ツールキットによって提供され、*hlq*.SIXMLOD1 にあります。C128N は言語環境ランタイムによって提供され、*hlq*.SCEERUN にあります。

   hlq.SDFHWSLD ライブラリーを使用すると、CICS は DFHWSSE1 と DFHWSXXX の Web Services Security モジュールを検索できるようになります。

4. **EDSALIM** システム初期化パラメーターの値を増やさなければならないことがあります。 ロードされる 3 つの DLL は、約 15 MB の EDSA ストレージを必要とします。

タスクの結果

ライブラリーを指定していない場合は、次のメッセージが表示されます。

```
CEE3501S The module module_name was not found.
(CEE3501S モジュール module_name が見つかりませんでした。)
```

module_name は、欠落しているライブラリーによって異なります。

SOAP Web サービスの保護の計画

Web サービスを保護するための最良の方法を判別します。 CICS は、構成可能なセキュリティー・メッセージ・ハンドラーや、個別の Trust クライアント・インターフェースなど、多くのオプションをサポートしています。

このタスクについて

CICS は、Web サービスごとではなくパイプライン・レベルで、Web Services Security (WS-Security または WSS) を実装します。 以下の質問に答えて、セキュリティーを実装する最良の方法を判別してください。

手順

1. パイプライン処理のパフォーマンスは重要ですか? WSS を使って Web サービスを保護する場合、パフォーマンスがかなり影響を受けます。

   WSS を実装する主な利点は、SOAP メッセージの一部を暗号化することによって、一連の中間ノードを介してメッセージを送信できることです。これらの中間ノードはすべて、ルーティングまたは処理の決定を行うために SOAP ヘッダーを調べることができますが、メッセージの内容を表示することはできません。機密にすべきセクションだけを暗号化することには、以下のような利点があります。

   - 一連の中間プロセス内のすべてのノードで暗号化および暗号化解除が行われることによるオーバーヘッドが生じません。
   - データの最終的な受信側からの理解を得られる場合に限り、信頼できないノードの公衆網を介して機密メッセージを送付することができます。

   WSS の使用に代わる方法として、SSL を使用してデータ・ストリーム全体を暗号化することができます。
2. WSS を使用する場合、どのレベルのセキュリティーが必要ですか? このオプションは、メッセージ・ヘッダーがユーザー名およびパスワードを含む基本認証から、メッセージでのデジタル署名と暗号化の組み合わせまで、多岐にわたります。 CICS セキュリティー・ハンドラーがサポートするオプションについては、 500 ページの『SOAP メッセージを保護するためのオプション』で説明しています。
3. CICS 提供のセキュリティー・ハンドラーは、要件を満たしていますか? より高度なセキュリティー処理を実行する場合は、独自にカスタムのセキュリティー・

ハンドラーを作成する必要があります。 このハンドラーは、RACF によって直接か、または Security Token Service を使用して、メッセージの必要な認証を実行し、デジタル証明書および暗号化されたエレメントの処理を行う必要があります。 詳しくは、 517 ページの『カスタムのセキュリティー・ハンドラーの作成』を参照してください。

4. パイプラインに MTOM ハンドラーが含まれていますか? パイプライン構成ファイルで、MTOM ハンドラーとセキュリティー・ハンドラーの両方を使用できるようにする計画の場合、MIME Multipart または Related メッセージはすべて、互換モードで処理されるため、セキュリティー・ハンドラーはメッセージ本文の XOP エレメントを構文解析できません。 この処理は、パイプライン処理のパフォーマンスに、さらに影響を与える恐れがあります。

SOAP メッセージを保護するためのオプション

CICS では、SOAP メッセージの署名と暗号化の両方がサポートされるため、SOAP メッセージで送受信するデータに最適なセキュリティー・レベルを選択することができます。

プロバイダー・モードの Axis2 Web サービス Java アプリケーション、または Axis2 の MessageContext を使用してパイプラインに接続するプロバイダー Web サービスでは、SOAP メッセージの署名および暗号化はサポートされていません。

以下のオプションの中から選択できます。

トラステッド認証
: サービス・プロバイダー・パイプラインでは、CICS は、SOAP メッセージ・ヘッダーのユーザー名トークンを、信頼できるものとして受け入れることができます。 この通常ユーザー名とパスワードを含むタイプのセキュリティー・トークンですが、この場合、パスワードは不要です。 CICS は提供されたユーザー名を信頼し、それを DFHWS-USERID コンテナーに置きます。メッセージはパイプラインで処理されます。

  サービス・リクエスターのパイプラインでは、CICS は、SOAP メッセージ・ヘッダーにパスワードがないユーザー名トークンを、サービス・プロバイダーに送信することができます。

基本認証
: サービス・プロバイダー・モードでは、CICS は、インバウンド SOAP メッセージでの認証のために、SOAP メッセージ・ヘッダーのユーザー名トークンを受け入れることができます。このユーザー名とパスワードを含むタイプのセキュリティー・トークンです。 CICS は、RACF などの外部セキュリティー・マネージャーを使用して、ユーザー名トークンを検証します。成功すると、ユーザー名はコンテナー DFHWS-USERID に置かれ、SOAP メッセージがパイプラインで処理されます。 CICS がユーザー名トークンを検証できない場合は、SOAP 障害メッセージがサービス・リクエスターに戻されます。

  パスワードを含むユーザー名トークンは、サービス・リクエスター・モードや、アウトバウンド SOAP メッセージではサポートされません。

HTTP 基本認証
: サービス・プロバイダー・モードでは、CICS は、HTTP プロトコルでの基

本認証情報を受け入れることができます。 サービス・リクエスターは URIMAP 定義を使用して資格情報 (ユーザー識別情報) がグローバル・ユーザー出口 XWBAUTH によって収集されることを指定します。 XWBAUTH は要求に応じてこの情報を CICS に受け渡し、CICS は HTTP 許可ヘッダーの情報をサービス・プロバイダーに送信します。

拡張認証
: サービス・プロバイダーおよびリクエスター・パイプラインでは、認証の目的で、Security Token Service (STS) によってセキュリティー・トークンを検証または交換できます。 この認証により、CICS は、メッセージ・ヘッダーにセキュリティー・トークンのある、通常はサポートされないメッセージ (Kerberos トークンや SAML アサーションなど) の送受信を行うことができるようになります。

  インバウンド・メッセージの場合は、セキュリティー・トークンの検証または交換を選択できます。 要求が、セキュリティー・トークンの交換である場合、CICS は、STS からユーザー名トークンを受け取る必要があります。 アウトバウンド・メッセージの場合は、セキュリティー・トークンに関するユーザー名トークンの交換だけが可能です。

X.509 証明書による署名
: サービス・プロバイダー・モードとサービス・リクエスター・モードでは、SOAP メッセージ・ヘッダーで X.509 証明書を提供して、認証のために SOAP メッセージの本文に署名することができます。 このバイナリー・セキュリティー・トークン として知られるタイプのセキュリティー・トークンです。 インバウンド SOAP メッセージからのバイナリー・セキュリティー・トークンを受け入れるには、証明書に関連付けられた公開鍵を RACF などの外部セキュリティー・マネージャーにインポートして、**KEYRING** システム初期化パラメーターで指定された鍵リングに関連付ける必要があります。アウトバウンド SOAP メッセージでは、公開鍵を生成して、意図した受信側に発行されます。 公開鍵の生成には、Integrated Cryptographic Service Facility (ICSF) が使用されます。

  X.509 デジタル証明書に関連付けられたラベルを指定する場合は、次の文字は使用しないでください。

  ```
  < > : ! =
  ```

  また、ヘッダーに 2 番目の X.509 証明書を含めて、最初の証明書を使用して署名することができます。 この 2 番目の証明書により、2 番目の X.509 証明書に関連付けられたユーザー ID を使用して CICS で作業を実行できるようになります。 SOAP メッセージに署名するために使用する証明書は、トラステッド・ユーザー ID に関連付けられている必要があります。また、異なる ID (宣言 *ID*) に関連付けられたパスワードをトラステッド・ユーザー ID が持たなくても、この ID で作業を実行することを表明するために代理権限も必要です。

暗号化
: サービス・プロバイダー・モードおよびサービス・リクエスター・モードでは、Triple-DES や AES などの対称アルゴリズムを使用して、SOAP メッセージの本文を暗号化することができます。 対称アルゴリズムでは、データの暗号化と暗号化解除に同じ鍵が使用されます。 この鍵は、対称鍵 として知られています。この鍵はメッセージに含められ、意図した受信側の公開

鍵と非対称鍵暗号化アルゴリズム RSA 1.5 との組み合わせを使用して暗号化されます。 非対称アルゴリズムは複雑で、対称鍵を暗号化解除するのは困難であるため、この暗号化によってセキュリティーが強化されます。 また一方、SOAP メッセージの大部分は、より迅速に暗号化解除できる対称アルゴリズムで暗号化されるため、パフォーマンスが向上します。

インバウンド SOAP メッセージでは、SOAP 本体のエレメントを暗号化してから、SOAP 本体を全体として暗号化することができます。 暗号化の種類は特に、機密データを含むエレメントに適していることがあります。 CICS が 2 つのレベルで暗号化された SOAP メッセージを受信すると、CICS は両方のレベルを自動的に暗号化解除します。暗号化の種類は、アウトバウンド SOAP メッセージではサポートされていません。

CICS では、メッセージ・ヘッダーのみに暗号化されたエレメントを含み、SOAP 本体のエレメントは暗号化されていないインバウンド SOAP メッセージはサポートされません。

署名および暗号化
: サービス・プロバイダー・モードおよびサービス・リクエスター・モードでは、SOAP メッセージの署名と暗号化の両方を選択することができます。 CICS は必ず、最初に SOAP メッセージの本文に署名してから、暗号化します。 この方法の利点は、メッセージの機密性と保全性の両方を確保できる点です。

ICRX ベースの **ID** 伝搬
: サービス・プロバイダー・モードでは、非認証 WS-Security ユーザー ID トークンを使用する場合と同じ状況で、非認証 ICRX (Extended Identity Context Reference) ID トークンを使用できます。 ICRX ID トークンは、ユーザー ID へマップされる z/OS ID です。 CICS は ICRX ID トークンをユーザー ID に解決し、DFHWS-ICRX コンテナーにコピーを置きます。 また、CICS は DFHWS-USERID コンテナーにデータを入れます。 ICRX ID トークンについて詳しくは、Identity propagation and distributed securityを参照してください。

Security Token Service を使用した認証

CICS は、Tivoli Federated Identity Manager などの Security Token Service (STS) と相互運用して、Web サービスのより高度な認証を提供することができます。

STS は、信頼のおける第三者機関として働き、Web サービス・リクエスターと Web サービス・プロバイダー間の信頼関係を仲介する Web サービスです。 SSL ハンドシェークでの認証局と同様の方法で、STS は、メッセージが示す資格情報をリクエスターとプロバイダーが「信頼」できることを保証します。 この信頼関係は、セキュリティー・トークンの交換によって表されます。 STS は、これらのセキュリティー・トークンを発行、交換、および検証して信頼関係を確立し、さまざまな信頼ドメインからの Web サービス同士が正常に対話できるようにします。詳細については、Web Services Trust Language の仕様を参照してください。

CICS は Trust クライアントとして働き、2 つのタイプの Web サービス要求を STS に送信できます。 要求の 1 つ目のタイプは、WS-Security メッセージ・ヘッ

ダーのセキュリティー・トークンを検証することであり、要求の 2 つ目のタイプは、セキュリティー・トークンを別のタイプに交換することです。 これらの要求により、多岐にわたる信頼ドメインからのさまざまなセキュリティー・トークン (SAML アサーションや Kerberos トークンなど) を含むメッセージを、CICS で送受信できるようになります。

CICS セキュリティー・ハンドラーを構成して、CICS が STS とデータをやり取りする方法を定義するか、独自のメッセージ・ハンドラーを作成して、個別に提供される Trust クライアント・インターフェースを使用することができます。 選択したメソッドに関わらず、SSL を使用して CICS と STS の間の接続を保護します。

セキュリティー・ハンドラーで STS を呼び出す方法

CICS セキュリティー・ハンドラーは、パイプライン構成ファイルの情報を使用して、Web サービス要求を Security Token Service (STS) に送信します。 送信される要求のタイプは、STS で実行するアクションによって異なります。

サービス・プロバイダー・パイプラインの場合
: サービス・プロバイダー・パイプラインでは、セキュリティー・ハンドラーの構成方法に応じて、次のような 2 種類のアクションがセキュリティー・ハンドラーによってサポートされます。

  - インバウンド・メッセージの WS-Security ヘッダーにある、セキュリティー・トークンの最初のインスタンスまたは特定タイプの最初のセキュリティー・トークンを検証するために、STS へ要求を送信します。
  - インバウンド・メッセージの WS-Security ヘッダーにある、セキュリティー・トークンの最初のインスタンスまたは特定タイプの最初のセキュリティー・トークンを、CICS が理解できるセキュリティー・トークンに交換するために、STS へ要求を送信します。

  セキュリティー・ハンドラーは、動的にパイプラインを作成し、Web サービス要求を STS に送信します。 このパイプラインは、STS からの応答が受信されるまで存在し、その後に削除されます。 要求が成功した場合、STS は、ID トークンまたはトークンの妥当性の状況を戻します。 セキュリティー・ハンドラーは、トークンから派生した RACF ID を DFHWS-USERID コンテナーに配置します。

  STS でエラーが発生した場合、STS は SOAP 障害をセキュリティー・ハンドラーに戻します。 その後セキュリティー・ハンドラーは、Web サービス・リクエスターに障害を戻します。

サービス・リクエスター・パイプラインの場合
: サービス・リクエスター・パイプラインでは、セキュリティー・ハンドラーが要求できるのは、STS によってトークンを交換することのみです。 パイプライン構成ファイルは、STS がセキュリティー・ハンドラーに対して発行する必要があるトークンのタイプを定義します。

  要求が成功した場合、RACF ID は DFHWS-USERID コンテナーに配置され、トークンはアウトバウンド・メッセージ・ヘッダーに組み込まれます。 STS でエラーが発生した場合、STS は SOAP 障害をセキュリティー・ハンドラーに返します。その後セキュリティー・ハンドラーは、パイプラインを介して、障害を Web サービス・リクエスター・アプリケーションに戻します。

セキュリティー・ハンドラーは、パイプラインに関する 1 つのタイプのアクションだけを STS に要求できます。 また、アウトバウンド要求メッセージに関する 1 つのタイプのトークンだけを交換でき、WS-Security メッセージ・ヘッダー内の最初のトークン (最初のインスタンス、または特定のタイプの最初のインスタンス) だけを限定的に処理します。 これらのオプションは、STS の使用に関する一般的なシナリオのほとんどをカバーしますが、インバウンドおよびアウトバウンド・メッセージを扱う際に必要となる処理を提供しない場合もあります。

より限定的な処理を準備してインバウンド・メッセージ・ヘッダーで多くのトークンを扱うようにする場合、または複数のタイプのトークンをアウトバウンド・メッセージと交換する場合は、Trust クライアント・インターフェースを使用します。このインターフェースを使用すると、カスタムのメッセージ・ハンドラーを作成して、独自の Web サービス要求を STS に送信できます。

Trust クライアント・インターフェース

Trust クライアント・インターフェースによって、セキュリティー・ハンドラーを使用せずに、直接 Security Token Service (STS) と対話することができます。 この方法を使用すると、セキュリティー・ハンドラーを使った処理よりも高度な処理をトークンに対して柔軟に実行できます。

Trust クライアント・インターフェースは、CICS 提供のプログラム DFHPIRT を拡張したものです。 このプログラムは通常、CICS Web サービス・アシスタントを使用して Web サービス・リクエスター・アプリケーションが配置されていない場合に、パイプラインを開始するために使用されます。ただし、STS に対する Trust クライアント・インターフェースとして機能することもできます。

Trust クライアント・インターフェースを起動するには、メッセージ・ハンドラーまたはヘッダー処理プログラムから DFHPIRT にリンクして、DFHWSTC-V1 と呼ばれるチャネルおよびセキュリティー・コンテナー一式を渡します。 これらのコンテナーを使用することで、柔軟性が高められ、STS の検証または実行アクションのいずれかを要求し、交換するトークンのタイプを選択し、メッセージ・ヘッダーの該当するトークンを渡すことができます。 DFHPIRT は動的にパイプラインを作成し、セキュリティー・コンテナーからの Web サービス要求を構成し、それを STS に送信します。

DFHPIRT は、STS から応答が戻るのを待機し、それを DFHWS-RESTOKEN コンテナーに入れてメッセージ・ハンドラーに渡します。 STS でエラーが発生した場合、STS は SOAP 障害を戻します。 DFHPIRT は、その障害を DFHWS-STSFAULT コンテナーに入れ、パイプライン内のリンクしているプログラムに戻します。

サービス・プロバイダーおよびサービス・リクエスター・パイプラインでセキュリティー・ハンドラーを使用できるようにしなくても、Trust クライアント・インターフェースを使用できます。あるいは、セキュリティー・ハンドラーに追加して、Trust クライアント・インターフェースを使用することもできます。

SOAP メッセージへの署名

インバウンド・メッセージでは、CICS は SOAP 本体内のエレメントおよび SOAP ヘッダー・ブロックのデジタル署名をサポートしています。 アウトバウンド・メッセージでは、CICS は SOAP 本体内のすべてのエレメントに署名します。

SOAP メッセージは、<Envelope> エレメントから構成される XML 文書です。この <Envelope> エレメントには、オプションの <Header> エレメントと必須の <Body> エレメントが格納されています。

WSS: SOAP Message Security 仕様では、<Header> と <Body> の内容にエレメント・レベルで署名することができます。 つまり、あるメッセージでは、署名するエレメントと署名しないエレメントがあり、別の署名または別のアルゴリズムを使用して署名することができます。 例えば、オンライン購入アプリケーションで使用される SOAP メッセージでは、受注を確認するエレメントは法的状況を持つことがあるため、これらのエレメントには署名するのが適しています。 しかし、メッセージ全体への署名にかかるオーバーヘッドを避けるために、一部の情報は署名しないでおくことも可能であるかもしれません。

インバウンド・メッセージでは、セキュリティー・メッセージ・ハンドラーが、SOAP <Header> および <Body> 内の個々のエレメントのデジタル署名を検証することができます。

- <Header> で検出される署名済みエレメント。
- SOAP <Body> 内の署名済みエレメント。 署名済みの本体を受け入れるようハンドラーが構成されている場合、CICS は本体が署名されていない SOAP メッセージをすべて拒否して、SOAP 障害を発行します。

アウトバウンド・メッセージでは、セキュリティー・メッセージ・ハンドラーが署名できるのは、<Body> だけで、<Header> には署名しません。 アルゴリズム、および本体への署名に使用される鍵は、ハンドラーの構成情報で指定されます。

署名アルゴリズム

CICS は、XML Signature 仕様で必要な署名アルゴリズムをサポートします。 それぞれのアルゴリズムは、汎用リソース ID (URI) で識別されます。

アルゴリズム	**URI**
Digital Signature Algorithm と Secure Hash Algorithm 1 (DSA と SHA1) インバウンド SOAP メッセージでのみサポートされます。	http://www.w3.org/2000/09/xmldsig#dsa-sha1
Rivest-Shamir-Adleman アルゴリズムと Secure Hash Algorithm 1 (RSA と SHA1)	http://www.w3.org/2000/09/xmldsig#rsa-sha1

署名された SOAP メッセージの例

これは、CICS によって署名された SOAP メッセージを示した例です。

```
<?xml version="1.0" encoding="UTF8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Header>
 <wsse:Security xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
                xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
                xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
                xmlns:xenc="http://www.w3.org/2001/04/xmlenc#" SOAP-ENV:mustUnderstand="1">
 <wsse:BinarySecurityToken 1
              EncodingType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0#Base64Binary"
              ValueType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-x509-token-profile-1.0#X509"
              wsu:Id="x509cert00">MIIChDCCAe2gAwIBAgIBADANBgkqhkiG9w0BAQUFADAwMQswCQYDVQQGEwJHQjEMMAoGA1UEChMD
                                  SUJNMRMwEQYDVQQDEwpXaWxsIFlhdGVzMB4XDTA2MDEzMTAwMDAwMFoXDTA3MDEzMTIzNTk1OVow
                                  MDELMAkGA1UEBhMCR0IxDDAKBgNVBAoTA0lCTTETMBEGA1UEAxMKV2lsbCBZYXRlczCBnzANBgkq
                                  hkiG9w0BAQEFAAOBjQAwgYkCgYEArsRj/n+3RN75+jaxuOMBWSHvZCB0egv8qu2UwLWEeiogePsR
                                  6Ku4SuHbBwJtWNr0xBTAAS9lEa70yhVdppxOnJBOCiERg7S0HUdP7a8JXPFzA+BqV63JqRgJyxN6
                                  msfTAvEMR07LIXmZAte62nwcFrvCKNPCFIJ5mkaJ9v1p7jkCAwEAAaOBrTCBqjA/BglghkgBhvhC
                                  AQ0EMhMwR2VuZXJhdGVkIGJ5IHRoZSBTZWN1cml0eSBTZXJ2ZXIgZm9yIHovT1MgKFJBQ0YpMDgG
                                  ZQVRFU0BVSy5JQk0uQ09ggdJQk0uQ09NhgtXV1cuSUJNLkNPTYcECRRlBjAO
 </wsse:BinarySecurityToken>
 <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
  <ds:SignedInfo xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
                 xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
                 xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
                 xmlns:xenc="http://www.w3.org/2001/04/xmlenc#">
   <ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#">
    <c14n:InclusiveNamespaces xmlns:c14n="http://www.w3.org/2001/10/xml-exc-c14n#" PrefixList="ds wsu xenc SOAP-ENV "/>
   </ds:CanonicalizationMethod>
   <ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
   <ds:Reference URI="#TheBody">
    <ds:Transforms>
     <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#">
       <c14n:InclusiveNamespaces xmlns:c14n="http://www.w3.org/2001/10/xml-exc-c14n#" PrefixList="wsu SOAP-ENV "/>
     </ds:Transform>
    </ds:Transforms>
    <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/> 2
    <ds:DigestValue>QORZEA+gpafluShspHxhrjaFlXE=</ds:DigestValue> 3
   </ds:Reference>
  </ds:SignedInfo>
  <ds:SignatureValue>drDH0XESiyN6YJm27mfK1ZMG4Q4IsZqQ9N9V6kEnw2lk7aM3if77XNFnyKS4deglbC3ga11kkaFJ 4
                     p4jLOmYRqqycDPpqPm+UEu7mzfHRQGe7H0EnFqZpikNqZK5FF6fvYlv2JgTDPwrOSYXmhzwegUDT
                     lTVjOvuUgXYrFyaO3pw=</ds:SignatureValue>
   <ds:KeyInfo>
    <wsse:SecurityTokenReference>
      <wsse:Reference URI="#x509cert00"
                      ValueType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-x509-token-profile-1.0#X509"/> 5
    </wsse:SecurityTokenReference>
   </ds:KeyInfo>
  </ds:Signature>
 </wsse:Security>
</SOAP-ENV:Header>
<SOAP-ENV:Body xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" wsu:Id="TheBody">
 <getVersion xmlns="http://msgsec.wssecfvt.ws.ibm.com"/>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

1. バイナリー・セキュリティー・トークンには、base64binary エンコードの X.509 証明書が含まれています。 このエンコードには、SOAP メッセージの意図した受信側が署名を検証するために使用する公開鍵が格納されています。
2. メッセージ・ダイジェストを生成するためにハッシュ・プロセス中に使用されるアルゴリズム。
3. メッセージ・ダイジェストの値。
4. その後、ダイジェスト値はユーザーの秘密鍵で暗号化され、署名値としてここに含められます。
5. 署名を検証するために使用される公開鍵を含むバイナリー・セキュリティー・トークンを参照します。

暗号化された SOAP メッセージの CICS サポート

インバウンド・メッセージでは、CICS は、SOAP 本体内の暗号化済みエレメント、および本体も暗号化された暗号化済みの SOAP ヘッダー・ブロックを暗号化解除することができます。アウトバウンド・メッセージでは、CICS は SOAP 本体全体を暗号化します。

SOAP メッセージは、<Envelope> エレメントから構成される XML 文書です。この <Envelope> エレメントには、オプションの <Header> エレメントと必須の <Body> エレメントが格納されています。

WSS: SOAP Message Security 仕様では、<Header> エレメントの内容の一部と <Body> エレメントのすべての内容をエレメント・レベルで暗号化することができます。つまり、あるメッセージでは、個々のエレメントを異なるレベルで暗号化したり、別のアルゴリズムを使用して暗号化したりすることができます。 例えば、オンライン購入アプリケーションで使用される SOAP メッセージでは、個々のクレジット・カードの詳細が機密のままになるように、詳細を暗号化するのが適しています。 しかし、メッセージ全体の暗号化にかかるオーバーヘッドを避けるために、一部の情報は安全性の低い (しかし高速な) アルゴリズムを使用して暗号化し、一部の情報は暗号化しないでおくことも可能であるかもしれません。

インバウンド・メッセージでは、CICS 提供のセキュリティー・メッセージ・ハンドラーが、SOAP <Body> 内の個々のエレメントを暗号化解除して、SOAP 本体も暗号化されている場合は SOAP <Header> 内のエレメントを暗号化解除することができます。 セキュリティー・メッセージ・ハンドラーは、以下のエレメントを常に暗号化解除します。

- <Header> エレメント内に検出される各エレメント (見つかった順序で)。
- SOAP <Body> エレメント内のエレメント。 暗号化された <Body> を含まない SOAP メッセージを拒否する場合は、<expect_encrypted_body> エレメントを使用して暗号化された本体を要求するようハンドラーを構成します。

アウトバウンド・メッセージでは、セキュリティー・メッセージ・ハンドラーがサポートするのは、SOAP <Body> の内容の暗号化だけです。<Header> エレメント内のエレメントは暗号化されません。 セキュリティー・メッセージ・ハンドラーが <Body> エレメントを暗号化すると、本体内のすべてのエレメントが、同じアルゴリズムと同じ鍵を使用して暗号化されます。アルゴリズム、および鍵に関する情報は、ハンドラーの構成情報で指定されます。

暗号化アルゴリズム

CICS は、XML 暗号化仕様で必要な暗号化アルゴリズムをサポートします。 それぞれのアルゴリズムは、汎用リソース ID (URI) で識別されます。

アルゴリズム	**URI**
Triple Data Encryption Standard algorithm (Triple DES)	http://www.w3.org/2001/04/xmlenc#tripledes-cbc
Advanced Encryption Standard (AES) アルゴリズム (鍵の長さは 128 ビット)	http://www.w3.org/2001/04/xmlenc#aes128-cbc
Advanced Encryption Standard (AES) アルゴリズム (鍵の長さは 192 ビット)	http://www.w3.org/2001/04/xmlenc#aes192-cbc
Advanced Encryption Standard (AES) アルゴリズム (鍵の長さは 256 ビット)	http://www.w3.org/2001/04/xmlenc#aes256-cbc

暗号化された SOAP メッセージの例

これは、CICS によって暗号化された SOAP メッセージの例です。

```
<?xml version="1.0" encoding="UTF8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Header>
 <wsse:Security xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
                xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
                xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
                xmlns:xenc="http://www.w3.org/2001/04/xmlenc#" SOAP-ENV:mustUnderstand="1">

  <wsse:BinarySecurityToken
                EncodingType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0#Base64Binary" 1
                ValueType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-x509-token-profile-1.0#X509"
                wsu:Id="x509cert00">MIIChDCCAe2gAwIBAgIBADANBgkqhkiG9w0BAQUFADAwMQswCQYDVQQGEwJHQjEMMAoGA1UEChMD
                                    SUJNMRMwEQYDVQQDEwpXaWxsIFlhdGVzMB4XDTA2MDEzMTAwMDAwMFoXDTA3MDEzMTIzNTk1OVow
                                    MDELMAkGA1UEBhMCR0IxDDAKBgNVBAoTA0lCTTETMBEGA1UEAxMKV2lsbCBZYXRlczCBnzANBgkq
                                    hkiG9w0BAQEFAAOBjQAwgYkCgYEArsRj/n+3RN75+jaxuOMBWSHvZCB0egv8qu2UwLWEeiogePsR
                                    6Ku4SuHbBwJtWNr0xBTAAS9lEa70yhVdppxOnJBOCiERg7S0HUdP7a8JXPFzA+BqV63JqRgJyxN6
                                    msfTAvEMR07LIXmZAte62nwcFrvCKNPCFIJ5mkaJ9v1p7jkCAwEAAaOBrTCBqjA/BglghkgBhvhC
                                    AQ0EMhMwR2VuZXJhdGVkIGJ5IHRoZSBTZWN1cml0eSBTZXJ2ZXIgZm9yIHovT1MgKFJBQ0YpMDgG
                                    A1UdEQQxMC+BEVdZQVRFU0BVSy5JQk0uQ09NggdJQk0uQ09NhgtXV1cuSUJNLkNPTYcECRRlBjAO
                                    BgNVHQ8BAf8EBAMCAfYwHQYDVR0OBBYEFMiPX6VZKP5+mSOY1TLNQGVvJzu+MA0GCSqGSIb3DQEB
                                    BQUAA4GBAHdrS409Jhoe67pHL2gs7x4SpV/NOuJnn/w25sjjop3RLgJ2bKtK6RiEevhCDim6tnYW
                                    NyjBL1VdN7u5M6kTfd+HutR/HnIrQ3qPkXZK4ipgC0RWDJ+8APLySCxtFL+J0LN9Eo6yjiHL68mq
                                    uZbTH2LvzFMy4PqEbmVKbmA87alF
  </wsse:BinarySecurityToken>
  <xenc:EncryptedKey xmlns:xenc="http://www.w3.org/2001/04/xmlenc#">
   <xenc:EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#rsa-1_5"/> 2
   <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
    <wsse:SecurityTokenReference>
     <wsse:Reference URI="#x509cert00"
                    ValueType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-x509-token-profile-1.0#X509"/> 3
    </wsse:SecurityTokenReference>
   </ds:KeyInfo>
   <xenc:CipherData>
    <xenc:CipherValue>M6bDQtJrvX0pEjAEIcf6bq6MP3ySmB4TQOa/B5UlQj1vWjD56V+GRJbF7ZCES5ojwCJHRVKW1ZB5 4
                      Mb+aUzSWlsoHzHQixc1JchgwCiyIn+E2TbG3R9m0zHD3XQsKTyVaOTlR7VPoMBd1ZLNDIomxjZn2
                      p7JfxywXkObcSLhdZnc=</xenc:CipherValue>
   </xenc:CipherData>
   <xenc:ReferenceList>
    <xenc:DataReference URI="#Enc1"/>
   </xenc:ReferenceList>
  </xenc:EncryptedKey>
 </wsse:Security>
</SOAP-ENV:Header>
<SOAP-ENV:Body>
 <xenc:EncryptedData xmlns:xenc="http://www.w3.org/2001/04/xmlenc#" Id="Enc1" Type="http://www.w3.org/2001/04/xmlenc#Content">
  <xenc:EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#tripledes-cbc"/> 5
  <xenc:CipherData>
   <xenc:CipherValue>kgvqKnMcgIUn7rl1vkFXF0g4SodEd3dxAJo/mVN6ef211B1MZelg7OyjEHf4ZXwlCdtOFebIdlnK 6
                     rrksql1Mpw6So7ID8zav+KPQUKGm4+E=</xenc:CipherValue>
  </xenc:CipherData>
 </xenc:EncryptedData>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

1. バイナリー・セキュリティー・トークンには、base64binary エンコードの X.509 証明書が含まれています。 このエンコードには、対称鍵の暗号化に使用された公開鍵が格納されています。
2. 対称鍵の暗号化に使用されたアルゴリズムを提示します。
3. 対称鍵の暗号化に使用された公開鍵を含むバイナリー・セキュリティー・トークンを参照します。
4. メッセージの暗号化に使用された暗号化済みの対称鍵。
5. メッセージの暗号化に使用された暗号化アルゴリズム。
6. 暗号化されたメッセージ。

Web Services Security に合わせた RACF の構成

アウトバウンド SOAP メッセージに署名して暗号化するための公開鍵と秘密鍵のペアおよび X.509 証明書を作成して、署名および暗号化されたインバウンド SOAP メッセージを認証して暗号化解除するには、RACF などの外部セキュリティー・マネージャーを構成する必要があります。

始める前に

この作業を実行するには、その前に、CICS で作業するよう RACF をセットアップしておく必要があります。 Web サービス・パイプラインを含む CICS 領域の **DFLTUSER**、**KEYRING**、および **SEC=YES** の各システム初期化パラメーターを指定します。

注**:** 同じ **KEYRING** 上の同じ識別名を持つ複数の証明書はサポートされていません。

手順

1. 署名されたインバウンド SOAP メッセージを認証するには、次のようにします。
   a. X.509 証明書を ICSF 鍵として RACF にインポートします。
   b. **RACDCERT** コマンドを使用して、**KEYRING** システム初期化パラメーターで指定した鍵リングに証明書を添付します。

      ```
      RACDCERT ID(userid1)
      CONNECT(ID(userid2) LABEL('label-name') RING(ring-name)
      ```

      ここで、

      - *userid1* は、鍵リングのデフォルトのユーザー ID であるか、他のユーザー ID の鍵リングに証明書を添付する権限を持っています。
      - *userid2* は、証明書に関連付けるユーザー ID です。
      - *label-name* は、証明書の名前です。
      - *ring-name* は、**KEYRING** システム初期化パラメーターで指定された鍵リングの名前です。

   c. オプション: 宣言 ID を使用する場合は、証明書に関連付けられたユーザー ID が、作業を他のユーザー ID の下で実行できる代理権限を持っていることを確認します。 また、SOAP メッセージ・ヘッダーに含まれる追加の証明書も忘れずに RACF にインポートします。

   SOAP メッセージのヘッダーには、証明書または証明書への参照のいずれかが入ったバイナリー・セキュリティー・トークンを含めることができます。 この参照は、KEYNAME (RACF での証明書ラベル)、ISSUER と SERIAL 番号の組み合わせ、または SubjectKeyIdentifier です。 SubjectKeyIdentifier が RACF における証明書の定義で属性として指定された場合、CICS が認識できるのは SubjectKeyIdentifier だけです。
2. アウトバウンド SOAP メッセージに署名するには、次のようにします。
   a. 次の **RACDCERT** コマンドを使用して、X.509 証明書および公開鍵と秘密鍵のペアを作成します。

      ```
      RACDCERT ID(userid2) GENCERT
      SUBJECTSDN(CN('common-name')
                 T('title')
                 OU('organizational-unit')
                 O('organization')
                 L('locality')
                 SP('state-or-province')
                 C('country'))
      WITHLABEL('label-name')
      ```

ここで、*userid2* は、証明書に関連付けるユーザー ID です。 証明書の *label-name* 値を指定する場合は、次の文字は使用しないでください。

```
< > : ! =
```

b. **KEYRING** システム初期化パラメーターで指定した鍵リングに証明書を添付します。 **RACDCERT** コマンドを使用します。

c. 証明書をエクスポートして、SOAP メッセージの意図した受信側に発行します。

CICS が、署名を検証するために、意図した受信側の SOAP メッセージ・ヘッダーのバイナリー・セキュリティー・トークンに X.509 証明書を自動的に含めるように、パイプライン構成ファイルを編集することができます。

3. 暗号化されたインバウンド SOAP メッセージを暗号化解除するには、SOAP メッセージに、鍵ペアの一部である公開鍵が含まれている必要があります。この場合、秘密鍵は CICS で定義されます。

a. 暗号化のために、RACF で公開鍵と秘密鍵のペアおよび証明書を生成します。 鍵ペアと証明書は、ICSF を使用して生成する必要があります。

b. **KEYRING** システム初期化パラメーターで指定した鍵リングに証明書を添付します。 **RACDCERT** コマンドを使用します。

c. 証明書をエクスポートして、暗号化解除する SOAP メッセージの生成プログラムに発行します。

その後、SOAP メッセージの生成プログラムは、公開鍵を含む証明書をインポートして、これを使用して SOAP メッセージを暗号化することができます。 SOAP メッセージのヘッダーには、公開鍵またはこの公開鍵への参照のいずれかが入ったバイナリー・セキュリティー・トークンを含めることができます。 この参照は、KEYNAME、ISSUER と SERIAL 番号の組み合わせ、または SubjectKeyIdentifier です。 SubjectKeyIdentifier が RACF における公開鍵の定義で属性として指定された場合、CICS が認識できるのは SubjectKeyIdentifier だけです。

4. アウトバウンド SOAP メッセージを暗号化するには、次のようにします。

a. 暗号化に使用する公開鍵を含む証明書を ICSF 鍵として RACF にインポートします。 意図した受信側が SOAP メッセージを暗号化解除するには、公開鍵に関連付けられた秘密鍵が必要です。

b. **KEYRING** システム初期化パラメーターで指定した鍵リングに、公開鍵を含む証明書を添付します。 **RACDCERT** コマンドを使用します。

CICS は、証明書の公開鍵を使用して、SOAP 本体を暗号化し、SOAP メッセージ・ヘッダー内にバイナリー・セキュリティー・トークンとして公開鍵を含む証明書を送信します。 公開鍵は、パイプライン構成ファイルで定義されます。

次のタスク

アウトバウンド・メッセージに署名して暗号化するこの構成では、使用される証明書が CICS 領域のユーザー ID によって所有されている必要があります。 RACF では、証明書の所有者だけが秘密鍵 (署名または暗号化のプロセスで使用される) を抽出できるため、証明書は CICS 領域ユーザー ID によって所有される必要があります。

CICS が所有していない証明書を使用してメッセージの署名または暗号化を行う必要がある場合、Using an existing certificate that is not owned by the CICS region user IDの説明に従って、単一の証明書を複数の CICS システム間で共用できます。

ID 伝搬のためのプロバイダー・モードの Web サービスの構成

Web サービス要求での ID 伝搬は、トラスト・ベースの構成 (WebSphere DataPower からのクライアント認証 SSL 接続の使用など) に依存します。このタスクでは、トラステッド・クライアントから送信される ICRX ID トークンを WS-Security ヘッダーで受け取るように、PIPELINE リソースを構成します。

始める前に

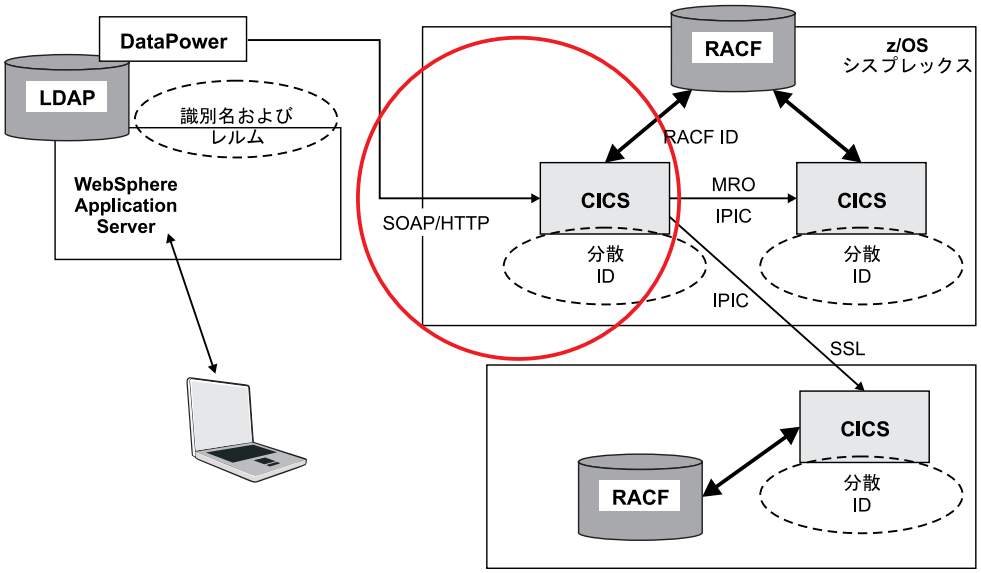
Web サービス接続を構成する前に、RACF RACMAP 設定を構成する必要があります。そうしないと、RACF へ送信されるマップされていない要求ごとに、RACF ICH408I メッセージを受け取ることになります。 RACF **RACMAP** コマンドの構成について詳しくは、ID 伝搬のための RACF の構成を参照してください。

WebSphere DataPower アプライアンスと CICS の間で信頼 関係を構成する必要があります (例えば、WebSphere DataPower と CICS の間で SSL クライアント認証を使用するなど)。 WebSphere DataPower の認証のために使用するデジタル証明書をユーザー ID に関連付け、そのユーザー ID に宣言 ID への代理権限を付与する必要があります。 代理権限について詳しくは、代理ユーザー・セキュリティーを参照してください。

このタスクについて

このタスクでは、CICS と WebSphere DataPower アプライアンスを使用して、安全かつ堅固な方法で分散 ID を伝搬できる Web サービス構成を提供する方法について説明します。 図の中の円は、このタスクが CICS 固有の構成について説明していることを示しています。

図 *31. WebSphere DataPower* からの *ICRX ID* トークンを受け入れるように *CICS* を構成する作業



WebSphere DataPower は CICS と他のアプリケーションの間の中間ノードの役割を果たします。 リモート Web サービス・リクエスター・アプリケーションは、SOAP プロトコルを使用して WebSphere DataPower アプライアンスに接続します。 WebSphere DataPower は、リモート・クライアントによって提供される資格情報を認証し、ユーザーの分散 ID を識別する z/OS ICRX ID トークンへの資格情報のマッピングを行います。 その後 SOAP メッセージは、WS-Security ヘッダーの ICRX ID トークンを使用し、トラステッド SSL 接続を介して CICS に転送されます。 ICRX ID トークンについて詳しくは、z/OS Security Server RACF Data Areas を参照してください。

CICS は、WebSphere DataPower から SOAP メッセージを受け取ります。 PIPELINE 構成ファイルは、ブラインド・トラストを指定します。これは、考えられるクライアントが WebSphere DataPower アプライアンスだけであり、WebSphere DataPower はセキュア SSL 接続を介して CICS と通信するためです。 そのため、PIPELINE 構成ファイルで追加の認証を指定する必要はありません。 WS-Security ハンドラー・プログラムは、WS-Security ヘッダーにある最初の ICRX を探し、その ICRX を使用して、ユーザーを識別します。

手順

1. PIPELINE リソースを作成するか、または既存の PIPELINE リソースを編集して、basic-ICRX モードを指定します。これを使用することにより、PIPELINE で ICRX を受け取ることができます。 ブラインド・トラストと basic-ICRX モードは、最も一般的な組み合わせです。 PIPELINE リソース・エレメントについて詳しくは、<authentication> エレメントを参照してください。

   以下は、ブラインド・トラストと basic-ICRX モードを示す PIPELINE 構成ファイルの例です。

```
<?xml version="1.0" encoding="EBCDIC-CP-US"?>
<provider_pipeline xmlns="http://www.ibm.com/software/htp/cics/pipeline">
   <service>
```

```
    <service_handler_list>
      <wsse_handler>
        <dfhwsse_configuration version="1">
          <authentication trust="blind" mode="basic-ICRX"/>
        </dfhwsse_configuration>
      </wsse_handler>
    </service_handler_list>
    <terminal_handler>
      <cics_soap_1.2_handler/>
    </terminal_handler>
  </service>
  <apphandler>DFHPITP</apphandler>
</provider_pipeline>
```

以下は、ブラインド・トラストを使用した、ICRX ID を持つ SOAP メッセージの例です。

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header>
    <wsse:Security
        xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
        SOAP-ENV:mustUnderstand="1">

      <wsse:BinarySecurityToken EncodingType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss
-soap-message-security-1.0#Base64Binary"
        wsu:Id="ICRX"
        xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wsswssecurity-utility-1.0.xsd"
        ValueType="http://www.IBM.com/xmlns/prod/zos/saf#ICRXV1">

        ICRX IS HERE

      </wsse:BinarySecurityToken>

    </wsse:Security>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>

   APPLICATION SPECIFIC XML IS HERE

  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

2. WebSphere DataPower が ICRX 情報を送信できるように構成されていることを確認します。 ID 伝搬を使用するためのネットワーク・トポロジーのサンプル (Sample network topologies for using identity propagation) を参照してください。

タスクの結果

クライアント認証 SSL 接続を介して接続し、WS-Security ヘッダーの ICRX ID トークンを使用して WebSphere DataPower から出される Web サービス要求が、正常に作動するようになります。

Web Services Security に合わせたパイプラインの構成

Web Services Security (WSS) をサポートするようにパイプラインを構成するには、パイプライン構成ファイルにセキュリティー・ハンドラーを追加する必要があります。 説明に従って CICS 付属のセキュリティー・ハンドラーを使用することも、独自に作成することもできます。

始める前に

CICS 提供のセキュリティー・ハンドラーを定義する前に、WSS に関する構成情報の追加先となるパイプライン構成ファイルを指定または作成する必要があります。

手順

1. <wsse_handler> エレメントをパイプラインに追加します。 サービス・プロバイダー・パイプラインまたはサービス・リクエスター・パイプライン内の <service_handler_list> エレメントにハンドラーを含める必要があります。 次のエレメントをコーディングします。

   ```
   <wsse_handler>
     <dfhwsse_configuration version="1">

     </dfhwsse_configuration>
   </wsse_handler>
   ```

   <dfhwsse_configuration> エレメントは、構成内の他のエレメントのコンテナーです。

2. オプション: <authentication> エレメントをコーディングします。
   - サービス・リクエスター・パイプラインでは、<authentication> エレメントが、アウトバウンド SOAP メッセージのセキュリティー・ヘッダーで使用する必要がある認証のタイプを指定します。
   - サービス・プロバイダー・パイプラインでは、このエレメントが、CICS がインバウンド SOAP メッセージでセキュリティー・トークンを使用して、処理が行われるユーザー ID を決定するかどうかを指定します。

   a. **trust** 属性をコーディングして、宣言 ID を使用するかどうか、およびサービス・プロバイダーとサービス・リクエスター間の信頼関係の性質を指定します。 **trust** 属性について詳しくは、<authentication> エレメントを参照してください。

   b. オプション: **trust=none** を指定した場合は、**mode** 属性をコーディングして、メッセージで見つかった資格情報の処理方法を指定します。 **mode** 属性について詳しくは、<authentication> エレメントを参照してください。

   c. 以下のエレメントを <authentication> エレメント内にコーディングします。

      1) オプションの、空の <suppress/> エレメント。

         このエレメントがサービス・プロバイダー・パイプラインに指定される場合、ハンドラーは、作業が行われるユーザー ID を決定するメッセージ内のどのセキュリティー・トークンの使用も試みません。

         このエレメントがサービス・リクエスター・パイプラインに指定される場合、ハンドラーは、アウトバウンド SOAP メッセージに、認証に必要な、どのセキュリティー・トークンの追加も試みません。

      2) リクエスター・パイプラインでは、SOAP メッセージの本文の署名に使用されるアルゴリズムの URI を指定する、オプションの <algorithm> エレメント。trust 属性と mode 属性の値の組み合わせが、メッセージが署名されていることを示している場合は、このエレメントを指定する必要があります。 このエレメントでは、SHA1 を使用

する RSA アルゴリズムだけを指定できます。 URI は http://www.w3.org/2000/09/xmldsig#rsa-sha1 です。

3) オプションとして、<certificate_label> エレメントを組み込めます。このエレメントでは、RACF にインストールされている X.509 デジタル証明書に関連付けるラベルを指定できます。このエレメントがサービス・リクエスター・パイプラインに指定され、<suppress> エレメントが指定されない場合は、証明書が SOAP メッセージのセキュリティー・ヘッダーに追加されます。 <certificate_label> エレメントを指定しない場合は、 CICS が RACF 鍵リングでデフォルトの証明書を使用します。

このエレメントはサービス・プロバイダー・パイプラインでは無視されます。

3. オプション: <authentication> エレメントの代わりに <sts_authentication> エレメントをコーディングします。 両方をパイプライン構成ファイルにコーディングすることはできません。 このエレメントは、Security Token Service (STS) が認証に使用されることを指定し、送信される要求のタイプを決定します。
   a. オプション: サービス・プロバイダー・モードの場合のみ、**action** 属性をコーディングして、STS がセキュリティー・トークンを検証または交換するかどうかを指定します。 **action** 属性について詳しくは、<sts_authentication> エレメントを参照してください。
   b. 以下のエレメントを <sts_authentication> エレメント内にコーディングします。
      1) <auth_token_type> エレメント。 このエレメントは、サービス・リクエスター・パイプラインで <sts_authentication> エレメントを指定する場合は必須で、サービス・プロバイダー・パイプラインではオプションです。詳しくは、<auth_token_type> エレメントを参照してください。
         - サービス・リクエスター・パイプラインでは、<auth_token_type> エレメントは、CICS が DFHWS-USERID コンテナーに含まれるユーザー ID を STS に送信したときに、STS が発行するトークンのタイプを示します。 CICS が STS から受け取るトークンは、アウトバウンド・メッセージのヘッダーに置かれます。
         - サービス・プロバイダー・パイプラインでは、<auth_token_type> エレメントは、CICS がメッセージ・ヘッダーから取得して、交換または妥当性検査のために STS に送信する識別トークンを判別するために使用されます。 CICS は最初に、メッセージ・ヘッダーで指定されたタイプの識別トークンを使用します。 このエレメントを指定しない場合、CICS は、メッセージ・ヘッダーで見つけた最初の識別トークンを使用します。 CICS は、次のものは ID トークンと見なしません。
           - wsu:Timestamp
           - xenc:ReferenceList
           - xenc:EncryptedKey
           - ds:Signature

2) サービス・プロバイダー・パイプラインの場合に限り、オプションの空の <suppress/> エレメント。このエレメントが指定される場合、ハンドラーは、操作の実行に使われるユーザー ID を判別する目的でメッセージ内のどのセキュリティー・トークンの使用も試みません。<suppress/> エレメントは、STS によって戻される ID トークンが含まれます。

4. オプション: <sts_endpoint> エレメントをコーディングします。 このエレメントは、<sts_authentication> エレメントを指定した場合にのみ使用してください。 <sts_endpoint> エレメント内で、以下のエレメントをコーディングします。
   - <endpoint> エレメント。 このエレメントには、ネットワーク上の Security Token Service (STS) の場所を指し示す URI が含まれます。 STS への接続を安全に保つには、HTTP ではなく、SSL または TLS を使用することをお勧めします。

     SAML サポートを使用するには、エンドポイントを cics://PROGRAM/DFHSAML に設定します。

     また、WebSphere MQ エンドポイントは、JMS フォーマットの URI を使用して指定することもできます。
   - オプションの <jvmserver> エレメント。このエレメントは、SAML トークン・サービスを実行するよう構成された JVM サーバーを識別します。このエレメントが含まれていない場合、デフォルトのサンプル・リソース JVM サーバー DFHXSTS が想定されます。このエレメントは、SAML を使用する場合にのみ有効です。その他の状況でこれを使用した場合、エラーが発生します。
5. オプション: インバウンド SOAP メッセージにデジタル署名する必要がある場合は、空の <expect_signed_body/> エレメントをコーディングします。

   <expect_signed_body/> エレメントは、インバウンド・メッセージの <body> に署名が必要であることを示します。 インバウンド・メッセージの本文が正しく署名されていない場合、CICS はセキュリティー障害でメッセージを拒否します。
6. オプション: デジタル署名されたインバウンド SOAP メッセージを拒否する場合は、空の <reject_signature/> エレメントをコーディングします。
7. オプション: インバウンド SOAP メッセージを暗号化する必要がある場合は、空の <expect_encrypted_body/> エレメントをコーディングします。

   <expect_encrypted_body/> > エレメントは、インバウンド・メッセージの <body> を暗号化する必要があることを示します。インバウンド・メッセージの本文が正しく暗号化されていない場合、CICS はセキュリティー障害でメッセージを拒否します。
8. 部分的に、または完全に暗号化されたインバウンド SOAP メッセージを拒否する場合は、空の <reject_encryption/> エレメントをコーディングします。
9. オプション: アウトバウンド SOAP メッセージに署名する必要がある場合は、<sign_body> エレメントをコーディングします。

a. <sign_body> エレメント内にある <algorithm> エレメントをコーディングします。

b. <algorithm> エレメントの後にある <certificate_label> エレメントをコーディングします。

これは、完成した <sign_body> エレメントの例です。

```
<sign_body>
  <algorithm>http://www.w3.org/2000/09/xmldsig#rsa-sha1</algorithm>
  <certificate_label>SIGCERT01</certificate_label>
</sign_body>
```

10. オプション: アウトバウンド SOAP メッセージを暗号化する必要がある場合は、<encrypt_body> エレメントをコーディングします。

a. <encrypt_body> エレメント内にある <algorithm> エレメントをコーディングします。

b. <algorithm> エレメントの後にある <certificate_label> エレメントをコーディングします。

これは、完成した <encrypt_body> エレメントの例です。

```
<encrypt_body>
  <algorithm>http://www.w3.org/2001/04/xmlenc#tripledes-cbc</algorithm>
  <certificate_label>ENCCERT02</certificate_label>
</encrypt_body>
```

例

次の例は、ほとんどのオプション・エレメントが存在する、完成したセキュリティー・ハンドラーを示しています。

```
<wsse_handler>
    <dfhwsse_configuration version="1">
      <authentication trust="signature" mode="basic">
        <suppress/>
        <certificate_label>AUTHCERT03</certificate_label>
      </authentication>
      <expect_signed_body/>
      <expect_encrypted_body/>
      <sign_body>
        <algorithm>http://www.w3.org/2000/09/xmldsig#rsa-sha1</algorithm>
        <certificate_label>SIGCERT01</certificate_label>
      </sign_body>
      <encrypt_body>
        <algorithm>http://www.w3.org/2001/04/xmlenc#tripledes-cbc</algorithm>
        <certificate_label>ENCCERT02</certificate_label>
      </encrypt_body>
    </dfhwsse_configuration>
</wsse_handler>
```

カスタムのセキュリティー・ハンドラーの作成

独自のセキュリティー手順および処理を使用する場合は、カスタムのメッセージ・ハンドラーを作成して、セキュアな SOAP メッセージをパイプラインで処理することができます。

始める前に

ご使用のセキュリティー・ハンドラーでサポートするセキュリティーのレベルを決定し、サポートされないセキュリティーをメッセージが含んでいる場合には、必ず

該当する SOAP 障害を戻すようにする必要があります。

このタスクについて

メッセージ・ハンドラーは、インバウンドおよびアウトバウンド・メッセージでのセキュリティーも処理できる必要があります。

手順

1. **EXEC CICS GET CONTAINER** コマンドを使用して、DFHREQUEST または DFHRESPONSE コンテナーを取り出します。
2. XML を構文解析して、WS-Security メッセージ・ヘッダーにあるセキュリティー・トークンを検出します。 ヘッダーの先頭は、<wsse:Security> エレメントです。セキュリティー・トークンは、ユーザー名およびパスワード、デジタル証明書、または暗号鍵である可能性があります。 メッセージは、セキュリティー・ヘッダーに多くのトークンを持つことがあるので、ハンドラーは、処理対象であるトークンを正しく識別する必要があります。
3. メッセージに実装されているセキュリティーに応じて、適切な処理を実行します。
   a. Kerberos トークンの基本認証を実行するには **EXEC CICS VERIFY TOKEN** コマンドを発行します。 このコマンドは、提供された Kerberos トークンが有効であることを検査します。 コマンドが成功した場合は、DFHWS-USERID コンテナーを **EXEC CICS PUT CONTAINER** で更新します。その他の場合は、**EXEC CICS SOAPFAULT CREATE** コマンドを実行します。
   b. パスワードまたはパスワード・フレーズの基本認証を実行するには **EXEC CICS VERIFY PHRASE** コマンドを発行します。 このコマンドは、メッセージのセキュリティー・ヘッダーにあるユーザー名およびパスワードを検査します。 コマンドが成功した場合は、DFHWS-USERID コンテナーを **EXEC CICS PUT CONTAINER** で更新します。 その他の場合は、**EXEC CICS SOAPFAULT CREATE** コマンドを実行します。
   c. Security Token Service によってトークンの範囲を交換するか検証することにより、拡張認証を実行する場合は、Trust クライアント・インターフェースを使用します。 詳しくは、 519 ページの『メッセージ・ハンドラーからの Trust クライアントの起動』を参照してください。
   d. メッセージが署名済みの場合は、デジタル証明書の資格情報を検証します。
   e. メッセージの部分が暗号化されている場合は、セキュリティー・ヘッダーの情報を使用して、メッセージを暗号化解除します。 CICS が Web Services Security 仕様に準拠する仕組みの仕様では、これを行う方法が説明されています。

タスクの結果

CICS でセキュリティー・ハンドラー・プログラムを定義し、パイプライン構成ファイルを更新して、そのプログラムが XML に正しく配置されるようにします。サービス・リクエスターのパイプライン構成ファイルでは、パイプラインの最後で実行されるように、セキュリティー・ハンドラーを構成する必要があります。サービス・プロバイダーのパイプライン構成ファイルでは、パイプラインの最初で実行されるように、セキュリティー・ハンドラーを構成する必要があります。

次のタスク

カスタムのメッセージ・ハンドラーの作成方法に関する一般情報については、IBM Redbooks: Application Development for IBM CICS Web Services を参照してください。

メッセージ・ハンドラーからの Trust クライアントの起動

CICS は、独自のメッセージ・ハンドラーを作成して Security Token Service (STS) を起動できるようにするインターフェースを備えています。 このインターフェースを使用すると、CICS 提供のセキュリティー・ハンドラーよりも高度な処理を実行することができます。

このタスクについて

セキュリティー・ハンドラーの代わりに、またはそれに追加して Trust クライアントを使用できます。 Trust クライアント・インターフェースを使用するには、次のようにします。

手順

1. 正しいトークンを、インバウンドまたはアウトバウンド・メッセージのセキュリティー・メッセージ・ヘッダーから取り出します。
2. チャネル DFHWSTC-V1 および以下の必要なコンテナーを渡す、プログラム DFHPIRT にリンクします。
   - DFHWS-STSURI。ネットワーク上の STS の位置を格納しています。
   - DFHWS-STSACTION。STS が実行する必要がある要求のタイプの URI を格納しています。 サポートされている 2 つのアクションは `issue` および `validate` です。
   - DFHWS-IDTOKEN。STS によって検証または交換される必要があるトークンを格納しています。
   - DFHWS-TOKENTYPE。STS が応答で返す必要があるトークンのタイプを格納しています。
   - DFHWS-SERVICEURI。呼び出されている Web サービス操作の URI を格納しています。

   オプションで DFHWS-XMLNS コンテナーを組み込んで、セキュリティー・トークンを格納する SOAP メッセージのネームスペースを提供することができます。 このコンテナーについては、ヘッダー処理プログラム・インターフェースで詳しく説明しています。
3. DFHPIRT は、STS からの応答によって戻ります。 成功した応答は、DFHWS-RESTOKEN コンテナーに格納されます。

   STS で要求に関する問題が発生した場合、STS は SOAP 障害を戻します。 DFHPIRT は、その SOAP 障害を DFHWS-STSFAULT コンテナーに入れます。 STS が、SOAP 障害を発行した理由を提供した場合、理由は DFHWS-STSREASON コンテナーに入れられます。

   異常終了が発生した場合、処理エラーの詳細を格納している DFHERROR コンテナーが戻されます。

メッセージ・ハンドラーは、これらの応答を処理し、エラーが発生した際には適切な処理を実行する必要があります。例えば、メッセージ・ハンドラーは、適切な SOAP 障害を Web サービス・リクエスターに戻す場合があります。

4. 必要に応じて、応答を処理します。 プロバイダー・モードでは、パイプライン処理は、メッセージがアプリケーション・ハンドラーに到達するまでに、CICS が理解できるユーザー名を DFHWS-USERID コンテナーに配置する必要があります。リクエスター・モードでは、メッセージ・ハンドラーは、正しいトークンをアウトバウンド・メッセージのセキュリティー・ヘッダーに追加する必要があります。

次のタスク

メッセージ・ハンドラーを作成した場合、CICS でそのプログラムを配置し、該当するパイプライン構成ファイルを更新します。 サービス・リクエスターのパイプラインで、パイプライン処理の最後 (ただし CICS 提供のセキュリティー・ハンドラーの前) に発生するように、メッセージ・ハンドラーを定義します。 サービス・プロバイダーのパイプラインで、パイプラインの最初 (ただし CICS 提供のセキュリティー・ハンドラーの後) に発生するように、メッセージ・ハンドラーを定義します。

z/OS Connect のセキュリティー

z/OS Connect は、WebSphere Liberty アプリケーションであり、その構成と考慮事項は、他の WebSphere Liberty アプリケーションと同じです。さらに、z/OS Connect for CICS 1.0 および z/OS Connect Enterprise Edition には、追加のセキュリティー要件があります。

z/OS Connect for CICS 1.0 および z/OS Connect Enterprise Edition は、動的サービス・ディスカバリーを可能にする RESTful 管理インターフェースを備えています。このインターフェースは、個々の JSON サービスと同じホスト名およびポート番号でホストされます。このインターフェースを保護するための Transport Layer Security (TLS) と個々の JSON サービスを使用することが推奨されます。

デフォルトでは、z/OS Connect for CICS 1.0 および z/OS Connect Enterprise Edition へのすべてのクライアント接続は、HTTPS プロトコルを使用する必要があります。デフォルトの動作では、CICS へのクライアント認証 TLS 接続が必要です。このデフォルトをそのまま使用する場合は、クライアント証明書を SAF ユーザー ID に関連付ける必要があります。アプリケーションは、この証明書から得られた ID を使用して実行されます。

z/OS Connect for CICS 1.0 と z/OS Connect Enterprise Edition は両方とも、HTTP 基本認証プロトコルをサポートするように構成できます。このプロトコルを使用すると、クライアントは TLS と SAF ユーザー ID/パスワードとを組み合わせて使用して接続できるようになります。HTTP 基本認証をサポートできるようにするには、Liberty サーバー構成ファイル (`server.xml`) に「`<webAppSecurity allowFailOverToBasicAuth="true"/>`」という行を追加します。

z/OS Connect for CICS 1.0 と z/OS Connect Enterprise Edition のユーザーは、`zos.connect.access.roles.zosConnectAccess` EJBROLE のメンバーでなければなりません。詳細については、SAF ロール・マッピングを使用した許可

を参照してください。

Liberty についてはLiberty でのアプリケーションの許可の構成、z/OS Connect については z/OS Connect に関するセキュリティーの構成、z/OS Connect EE については z/OS Connect Enterprise Edition V3.0 製品資料内の『z/OS Connect EE に関するセキュリティーの構成』を参照してください。

詳しくは、SAF ロール・マッピングを使用した許可、およびLiberty JVM サーバーのセキュリティーの構成 (Configuring security for a Liberty JVM server)を参照してください。

z/OS Connect サービスおよび API の許可の構成

CICS セキュリティー・モデルでは、z/OS Connect for CICS 1.0 および z/OS Connect Enterprise Edition を使用してサービスおよび API の許可を構成するという点で、いくつかの追加のアクションが必要です。

このタスクについて

z/OS Connect を使用して CICS に作業を組み込む場合、処理のさまざまな段階で次の 2 つの ID が作業に関連付けられます。

- 作業の接続プロセスで、初期の一時 ID が割り振られます。
- その後、認証済みの ID が、作業の残りの部分を実行するために使用されます。

これらの ID は、設定およびシステム環境に応じて、いくつかの方法で構成できます。

手順

1. オプション: z/OS Connect の代替の初期ユーザー ID を作成します。

   デフォルトでは、初期 ID はデフォルトの CICS ユーザー ID ですが、別のユーザー ID を割り当てて、デフォルトの CICS ユーザー ID にトランザクション CPIH またはそれと同等の作業を実行する許可を与えないようにすることもできます。

   a. 代替の初期ユーザー ID に、トランザクション CPIH および z/OS Connect を介して開始されるその他のトランザクションを実行することを許可します。

      初期ユーザー ID には、サービスまたは API のターゲット・トランザクションを実行するための許可が必要です。
2. デフォルトの初期ユーザー ID を割り当てます。以下の方法のいずれかを選択することも、両方を選択することもできます。
   - z/OS Connect をホストする JVMSERVER リソースの JVM プロファイルでユーザー ID のオーバーライド値を設定します。

     以下にオーバーライドの例を示します。ここで、`ZOSCUSER` はデフォルトの初期ユーザー ID の `-Dcom.ibm.cics.jvmserver.http.userid=ZOSCUSER` です。

     注: JVM プロファイルでデフォルトの初期ユーザー ID を設定する場合は、各 URIMAP に USERID 値を指定する必要はありません。ただし、

URIMAP に USERID を指定し、かつ JVM プロファイルでオーバーライド値を指定した場合、その URIMAP に指定した USERID が優先されます。

- z/OS Connect をターゲットとする特定の URIMAP リソースに対して USERID フィールドを設定します。

  HTTP 要求が z/OS Connect で受信されると、CICS はその要求を、インストールされている URIMAP リソースと照合します。検出された URIMAP が USERID 属性を指定する場合、JVM サーバーのデフォルトの初期ユーザー ID の代わりに、そのユーザー ID が初期ユーザー ID として使用されます。

  以下に、ZOSCDEFT という名前の URIMAP リソースの構成例を示します。ここで、JVMSERVER は USAGE 値です。総称値は PATH 属性に対して設定されます。CPIH はターゲット・トランザクションです。ZOSCUSER はデフォルトの初期ユーザー ID です。

```
NAME: ZOSCDEFT
USAGE: JVMSERVER
SCHEME: HTTP
PORT: NO
HOST: *
PATH: /zosConnect/*
TRANSACTION: CPIH
USERID: ZOSCUSER
```

注: PIPELINE SCAN メカニズムを使用することによってインストールされる URIMAP リソースは、デフォルト・ユーザー ID を使用して構成されない可能性があります。このような場合、JVMSERVER でユーザー ID のオーバーライド値を指定することを考慮できます。

注: 初期ユーザー ID は WSBind ファイルに保管できます。DFHLS2JS または DFHJS2LS のユーザーは、**USERID** 入力パラメーターの値を提供できます。**USERID** パラメーターが使用されている場合、PIPELINE SCAN 中に生成されるすべての URIMAP には要求された初期ユーザー ID が含まれます。

タスクの結果

これで、CICS がサービスおよび API の URI を認識し、ターゲット・トランザクションが接続されている場合は、使用する初期ユーザー ID を関連付けるように環境が構成されました。

第 14 章 BTS のセキュリティー

CICS ビジネス・トランザクション・サービスのセキュリティー上の考慮事項となるのは、BTS リソースにアクセスするための権限、プロセスとそれを構成するアクティビティーを実行するためのユーザー ID、プロセスとそれを構成するアクティビティーを接続するための権限、および BTS システム・プログラミング・コマンドを使用するための権限です。

CICS セキュリティーについては、保護の概要で詳細に説明されています。リソース・アクセス管理機能 (RACF) 以外の外部セキュリティー・マネージャー (ESM) のユーザーは、ご使用の ESM の資料と共にこの情報をお読みください。

BTS のリソース・セキュリティー

プロセス、アクティビティー、コンテナーなどの BTS リソースを、CICS ファイル制御コマンドによってアクセスされるリソースと同じように保護することができます。つまり、プロセス、そのアクティビティー、およびそれらのコンテナーのリソース・レベルのセキュリティーは CICS ファイル定義に基づいており、その CICS ファイル定義で、このタイプのプロセスのレコードを書き込むリポジトリー・データ・セットが指定されます。

特定のプロセス・タイプのプロセスやアクティビティーを定義または獲得するプログラムを実行するユーザーは、対応する CICS ファイルへの UPDATE アクセス権限が必要です。

注: タスクが ACQUIRE コマンドを発行すると、CICS では、要求に関連付けられたユーザー ID に READ アクセス権限しかない場合でも、BTS リポジトリーから適切なレコードを読み取ることができます。ただし、タスクが同期点を発行すると、レコードはデータ・セットに書き戻され、その際にユーザー ID に UPDATE アクセス権限がない場合はタスクが異常終了します。

特定のプロセス・タイプのプロセスやアクティビティーを照会またはブラウズするユーザーは、少なくとも、対応する CICS ファイルへの READ アクセス権限が必要です。

プロセスおよびアクティビティーのユーザー ID

RUN または LINK コマンドを使用して、プロセスやアクティビティーをアクティブ化することができます。使用するコマンドによって、プロセスまたはアクティビティーの実行に使用されるユーザー ID が影響を受けます。

RUN コマンドによってアクティブ化されたアクティビティーのユーザー ID
プロセスまたはアクティビティーが RUN コマンドによってアクティブ化された場合、そのプロセスまたはアクティビティーは、RUN を発行するトランザクションとは別のユーザー ID を使用して実行される可能性があります。

© Copyright IBM Corp. 1989, 2018

アプリケーション・プログラマーは、プロセスまたはアクティビティーが RUN コマンドによってアクティブ化されるときに、どのユーザーの権限のもとでプロセスまたはアクティビティーを実行するかを指定できます。それには、DEFINE PROCESS または DEFINE ACTIVITY コマンドのユーザー ID オプションをコーディングします。ユーザー ID オプションを省略すると、その値は、DEFINE コマンドを発行するトランザクションのユーザー ID にデフォルトで設定されます。

DEFINE コマンドから取得されたユーザー ID は、**定義済みプロセス・ユーザー ID** または**定義済みアクティビティー・ユーザー ID** と呼ばれます。このセクションの残りの部分では、定義済みプロセス・ユーザー ID または定義済みアクティビティー・ユーザー ID のいずれかを表す場合に「定義済みユーザー ID」という語を使用します。

DEFINE PROCESS または ACTIVITY のユーザー ID オプションが指定された場合、CICS は (定義時に) 代理セキュリティー検査を実行して、DEFINE コマンドを発行したトランザクションのユーザー ID が定義済みユーザー ID の使用を許可されているかどうかを確認します。BTS プロセスまたはアクティビティーの代理検査に使用される RACF プロファイルは、SURROGAT クラスの userid.DFHSTART です。

次の RACF コマンドの例は、ユーザーを定義済みプロセス・ユーザー ID および定義済みアクティビティー・ユーザー ID の代理ユーザーとして許可します。

```
RDEFINE SURROGAT defined_process_userid.DFHSTART UACC(NONE)
         OWNER(defined_process_userid)

PERMIT defined_process_userid.DFHSTART CLASS(SURROGAT)
         ID(define_process_command_userid) ACCESS(READ)

RDEFINE SURROGAT defined_activity_userid.DFHSTART UACC(NONE)
         OWNER(defined_activity_userid)

PERMIT defined_activity_userid.DFHSTART CLASS(SURROGAT)
         ID(define_activity_command_userid) ACCESS(READ)
```

LINK コマンドによってアクティブ化されたアクティビティーのユーザー **ID**
LINK コマンドによってプロセスまたはアクティビティーがアクティブ化された場合、そのプロセスまたはアクティビティーは、LINK を発行するトランザクションのユーザー ID のもとで実行されます。

プロセスまたはアクティビティー内 でのリソース・レベルのセキュリティー検査は、プロセスまたはアクティビティーを実行する際に権限が使用されるユーザー ID (定義済みユーザー ID、または LINK コマンドを発行するトランザクションのユーザー ID) に基づきます。このユーザー ID は、プロセス・タイプに対応する CICS ファイルへの UPDATE アクセス権限を持っていなければなりません。

プロセスおよびアクティビティーの接続時セキュリティー

接続時セキュリティーを使用すると、トランザクションがプロセスまたはアクティビティーを接続 (アクティブ化) する権限を持っているかどうかを検査できます。接続時セキュリティーは、プロセスまたはアクティビティーが RUN コマンドによってアクティブ化された場合のみ適用されます。LINK によってアクティブ化された場合は適用されません。

プロセスに接続時セキュリティーが必要な場合は、定義済みユーザー ID (DEFINE PROCESS コマンドから取得したユーザー ID) に対し、プロセスの詳細とそれを構成するアクティビティーが保管された BTS データ・セットに対応する CICS ファイルへの UPDATE アクセス権限を付与する必要があります。

BTS のコマンド・セキュリティー

CICS のコマンド・レベルのセキュリティーを使用して、BTS システム・プログラミング・コマンド **EXEC CICS CREATE PROCESSTYPE**、**DISCARD PROCESSTYPE**、**INQUIRE PROCESSTYPE**、および **SET PROCESSTYPE** を保護することができます。

第 15 章 CICS-MQ アダプターのセキュリティー

CICS-MQ アダプターは、IBM MQ に対して、特に IBM MQ セキュリティーで使用される情報を提供します。

提供される情報は、次のとおりです。

- CICS リソース・レベル・セキュリティーはこのトランザクションに対してアクティブかどうか。詳しくは、リソース定義のセキュリティーを参照してください。
- ユーザー ID。
  - ユーザーがサインオンしていない端末タスクの場合、ユーザー ID は端末に関連した CICS ユーザー ID で、以下のいずれかです。
    - CICS **DFLTUSER** システム初期設定パラメーターで指定されたデフォルトの CICS ユーザー ID。
    - 端末の定義に事前設定されているセキュリティー・ユーザー ID。
  - 非端末タスクの場合、アダプターはユーザー・ドメインの呼び出しを使用してユーザー ID を取得します。

CICS-MQ アダプター・トランザクションのセキュリティーの実装

CICS-MQ アダプターをユーザーに管理させる場合は、適切な CICS トランザクションに対する権限をそのユーザーに付与する必要があります。

必要に応じて、アダプターの特定の機能に対するアクセスを制限できます。例えば、ユーザーにアダプターの現在の状況を表示することは許可するが、それ以外のものは許可したくない場合、そのユーザーには、CKQC、CKBM、CKRT、および CKDP のアクセス権のみを与えます。

CICS に対してこれらのトランザクションを RESSEC(NO) と CMDSEC(NO) で定義します。詳しくは、リソース定義のセキュリティーおよびCICS command securityを参照してください。

トランザクション	機能
CKAM	アラート・モニター
CKBM	アダプター機能を制御する
CKCN	接続
CKDL	行モード表示
CKDP	フルスクリーン表示
CKQC	アダプター機能を制御する
CKRS	統計
CKRT	アダプター機能を制御する
CKSD	切断

© Copyright IBM Corp. 1989, 2018

トランザクション	機能
CKSQ	CKTI 開始/停止
CKTI	トリガー・モニター

管理者と同様に、IBM MQ に接続するユーザー ID、**PLTPIUSR** システム初期設定パラメーターで設定されたユーザー ID、および CICSPlex SM MAS エージェント・ユーザー ID にも CKTI トランザクションおよび CKAM トランザクションを実行する権限が必要です。

MQCONN コマンドおよび MQMONITOR コマンドのセキュリティー

MQCONN リソース定義および MQMONITOR リソース定義に対してユーザーが SPI コマンドを発行する機能を制御するには、CICS コマンド・セキュリティーを使用します。例えば、それを使用して、どのユーザーが CICS 領域の MQCONN リソース定義に対して CREATE コマンドおよび DISCARD コマンドを発行できるかを制御できます。

コマンド・セキュリティーがトランザクションに対して有効にされている場合、外部セキュリティー・マネージャーは、そのトランザクションに関連付けられているユーザー ID が必要に応じて MQCONN リソースまたは MQMONITOR リソースでコマンドの使用を許可されているかどうかを検査します。リソース・セキュリティーは、MQCONN リソースおよび MQMONITOR リソースには使用できません。

CICS コマンド・コマンド・セキュリティーは、**EXEC CICS CREATE MQCONN**、**DISCARD MQCONN**、**SET MQCONN**、**INQUIRE MQCONN**、**CREATE MQMONITOR**、**DISCARD MQMONITOR**、**SET MQMONITOR**、および **INQUIRE MQMONITOR** の各コマンドを対象としています。コマンド・セキュリティーの説明および CICS 領域に対してコマンド・セキュリティーをセットアップする手順については、CICS command securityを参照してください。各コマンドに必要な権限レベルのリストについては、Resource and command check cross-referenceを参照してください。

コマンド・セキュリティーがアクティブである場合、**EXEC CICS SET MQCONN** コマンドを使用して IBM MQ への接続を開始または停止するには、ユーザーが **EXEC CICS SET MQCONN** コマンドを使用する権限、インストールされている MQMONITOR リソースに関連付けられたトランザクションに対して **EXEC CICS START** コマンドを発行する権限、および **EXEC CICS EXTRACT EXIT** コマンドを使用する権限を持っている必要があります。ユーザーが **EXEC CICS EXTRACT EXIT** コマンドを使用する権限がないときに接続を開始または停止しようとする場合、CICS はメッセージ DFHXS1111 および DFHMQ0302 を発行します。MQMONITOR に関連付けられたトランザクションを指定して **EXEC CICS START** コマンドを発行する権限がユーザーにない場合、CICS はメッセージ DFHMQ0390 を発行します。ユーザー作成の MQMONITOR プログラムには、**EXEC CICS SET MQMONITOR** コマンドに対する権限が必要です。

CICS-MQ アダプター・ユーザー ID

CICS-MQ アダプターに関連するユーザー ID は、IBM MQ に用意されているタスク・イニシエーター・トランザクション CKTI のユーザー ID です。

MQMONITOR によって開始された CKTI のユーザー ID

CKTI のインスタンスが MQMONITOR によって開始される場合、CKTI トランザクションに関連付けられるユーザー ID は以下のように取得されます。

- セキュリティー検査がアクティブの場合 (つまり、**SEC** システム初期設定パラメーターが YES に設定されている場合)、ユーザー ID は MQMONITOR リソースの **MONUSERID** 属性から取得されます。
- セキュリティー検査が無効の場合 (つまり、**SEC** が NO に設定されている場合)、ユーザー ID は、MQMONITOR 状態が開始に設定されているトランザクションのユーザー ID です。

PLTPI フェーズと PLTSD フェーズで実行される IBM MQ リソースに関するユーザー ID 検査

IBM MQ リソースが CICS PLTPI フェーズでアクセスされる場合、 IBM MQ に渡されるユーザー ID は、PLTPIUSR システム初期設定パラメーターで指定されているユーザー ID です。 CICS の PLTSD フェーズで IBM MQ リソースにアクセスするときに IBM MQ に渡されるユーザー ID は、シャットダウン・トランザクションに関連するユーザー ID です。

CICS-MQ アダプターの接続セキュリティー

アプリケーション・プログラムが **MQCONN** 要求または **MQCONNX** 要求を発行してキュー・マネージャーに接続しようとする場合、またはチャネル・イニシエーターあるいは CICS-MQ アダプターが接続要求を発行する場合に、IBM MQ は接続セキュリティー検査を実行します。

キュー・マネージャー・レベルのセキュリティーを使用している場合は、特定のキュー・マネージャーに対する接続セキュリティー検査をオフにすることができますが、そのようにすると、どのユーザーでもそのキュー・マネージャーに接続できます。

接続セキュリティー検査には、CICS アドレス・スペース・ユーザー ID のみ使用され、個々の CICS 端末ユーザー ID は使用されません。

IBM MQ の接続セキュリティー検査は、キュー・マネージャー・レベルまたはキュー共用グループ・レベルのいずれかでオン/オフにすることができます。

第 16 章 CICS-MQ ブリッジのセキュリティー

CICS-MQ ブリッジを開始する際、認証のレベルを指定できます。ブリッジ・モニターは、 IBM MQ の要求メッセージから抽出したユーザー ID とパスワードの検査が要求されていれば、その検査を実行してから、その要求メッセージで指定されている CICS プログラムを実行します。

CICS-MQ ブリッジ・モニター・トランザクション (例えば、CKBR またはユーザーのトランザクション名) を実行する際に、**AUTH** パラメーターを指定して以下のいずれかの認証レベルを選択できます。

LOCAL

このレベルがデフォルトです。ブリッジ・モニターは CICS のデフォルト・ユーザー ID を使用してブリッジ・タスクを開始します。ブリッジ・タスクが実行する CICS ユーザー・プログラムには、このユーザー ID に関連付けられた権限があります。メッセージ内のユーザー ID またはパスワードは無視されるため、IBM MQ 要求メッセージは、より高い権限を要求できません。ブリッジ・タスクが保護リソースへのアクセスを試行する CICS プログラムを実行する場合、CICS プログラムは失敗する可能性があります。

IDENTIFY

要求メッセージ内のメッセージ記述子 (MQMD) がユーザー ID を指定する場合、ブリッジ・モニターはそのユーザー ID を使用してブリッジ・タスクを開始します。ブリッジ・タスクが実行する CICS ユーザー・プログラムには、そのユーザー ID に関連付けられた権限があります。ユーザー ID は信頼できると見なされます。つまり、ブリッジ・モニターはパスワードまたはパスチケット情報の使用による ID の認証を行いません。 MQMD がユーザー ID を指定しない場合、ブリッジ・モニターは CICS のデフォルト・ユーザー ID を使用して、LOCAL オプションと同じ方法でブリッジ・タスクを開始します。

VERIFY_UOW

以下のすべての条件が該当する場合、ブリッジ・モニターはパスワードまたはパスチケットを使用してユーザー ID を認証します。

- 要求メッセージ内のメッセージ記述子 (MQMD) はユーザー ID を指定します。
- 要求メッセージには IBM MQ CICS 情報ヘッダー (MQCIH) が含まれています。
- MQCIH の「認証子 (Authenticator)」フィールドには、パスワードまたはパスチケットが含まれています。

認証が成功すると、ブリッジ・モニターはそのユーザー ID を使用してブリッジ・タスクを開始します。認証が失敗すると、ブリッジ・モニターは要求に対処できず、MQCRC_SECURITY_ERROR 戻りコードが出されます。

前にリストされた条件のうちのいずれか 1 つが満たされない場合、ブリッジ・モニターは CICS のデフォルト・ユーザー ID を使用して、LOCAL

© Copyright IBM Corp. 1989, 2018

オプションと同じ方法でブリッジ・タスクを開始します。作業単位内の最初の要求メッセージのみが検査されます。同じ作業単位を構成する後続のメッセージのユーザー ID とパスワードまたはパスチケットの情報は無視されます。

VERIFY_ALL
このレベルは VERIFY_UOW と同じです。ただし、ブリッジ・タスクは、ユーザー ID が同じ作業単位内のすべての要求メッセージで同一であることも確認し、要求メッセージに含まれるパスワードまたはパスチケットを使用して、要求メッセージごとにユーザー ID を再認証する点は異なります。

さまざまなアプリケーション用に別々の認証レベルが必要な場合は、複数のブリッジ・モニターを別々のトランザクション ID で使用します。CICS 代理セキュリティーを使用して、ブリッジ・モニター・トランザクションとユーザー ID が開始できるトランザクションとユーザー ID の組み合わせを制限できます。

表 52 は、ブリッジ・モニターの開始ユーザー ID を示しています。ユーザー ID は、ブリッジ・モニター・トランザクションの実行に使用するメソッド (通常は CKBR) によって異なります。

表 *52. CICS-MQ* ブリッジ・モニターのセキュリティー

ブリッジ・モニターの開始メソッド	サインオンした端末かどうか	ブリッジ・モニターのユーザー **ID**
端末、またはプログラム内の **EXEC CICS LINK** から	はい	サインオン・ユーザー ID
端末、またはプログラム内の **EXEC CICS LINK** から	いいえ	CICS デフォルト・ユーザー ID
ユーザー ID が指定された **EXEC CICS START**	–	START のユーザー ID
ユーザー ID が指定されていない **EXEC CICS START**	–	START のユーザー ID
CICS-MQ トリガー・モニター CKTI	–	START のユーザー ID

表 52. CICS-MQ ブリッジ・モニターのセキュリティー (続き)

ブリッジ・モニターの開始メソッド	サインオンした端末かどうか	ブリッジ・モニターのユーザー ID
CICS MQ モニター (MQMONITOR)	–	• セキュリティー検査が CICS 領域に対してアクティブの場合 (つまり、**SEC** システム初期設定パラメーターが YES に設定されている場合)、MQMONITOR リソースの **MONUSERID** 属性 • セキュリティー検査が CICS 領域に対して無効の場合 (つまり、**SEC** が NO に設定されている場合)、MQMONITOR リソースを開始したユーザー ID

要求メッセージ内のユーザー ID とパスワード

IDENTIFY、VERIFY_UOW、または VERIFY_ALL の認証オプションを使用する場合、ブリッジ・タスクおよびそれが実行する CICS プログラムは、要求メッセージ内のメッセージ記述子 (MQMD) で指定されたユーザー ID で開始されます。VERIFY_UOW オプションおよび VERIFY_ALL オプションを使用して、ブリッジ・モニターは要求メッセージ内の IBM MQ CICS 情報ヘッダー (MQCIH) で指定されたパスワードも検査します。

これらの認証レベルを使用するには、 IBM MQ アプリケーションは MQMD でユーザー ID を提供し、さらに MQCIH (パスワードを含む) を提供する必要があります。これらのユーザー ID を RACF に対して定義する必要があります。使用されるユーザー ID を制御するには、IBM MQ アプリケーションは、MQOO_SET_IDENTITY_CONTEXT が含まれるオープン・オプションを使用して、ブリッジ・モニターの要求キューを開く必要があります。また、アプリケーションはメッセージの書き込みオプションに MQPMO_SET_IDENTITY_CONTEXT の値を含める必要があります。

ブリッジ・モニターが要求メッセージ内のユーザー ID またはパスワードに問題があることを検出した場合、以下のように処理されます。

- 認証レベルが IDENTIFY の場合、取り消されたユーザー ID がメッセージに含まれると、異常終了 AICO が発生する可能性があります。このエラー応答には、戻りコード MQCRC_BRIDGE_ERROR と理由 MQFB_CICS_BRIDGE_FAILURE があります。

- 認証レベルが VERIFY_UOW または VERIFY_ALL の場合、ユーザー ID またはパスワードが無効であると、要求は失敗し、戻りコード MQCRC_SECURITY_ERROR が出されます。
- 要求メッセージでユーザー ID またはパスワードのいずれかが省略される場合、ブリッジ・モニターが他のいずれかの認証オプションを使用して開始した場合でも、ブリッジ・モニターは LOCAL 認証レベルでブリッジ・タスクを実行します。このような状況では、ブリッジ・タスクによって開始された CICS プログラムは、ブリッジ・モニターの開始ユーザー ID で実行されます。

パスワードの代わりにパスチケットを使用して、パスワードがメッセージ内をフローする必要がないようにすることができます。

- IBM MQ アプリケーションは、要求メッセージ内の MQCIH でパスチケットを提供する必要があります。
- パスチケットを生成するには、アプリケーション ID が必要です。デフォルトのアプリケーション ID は CICS APPLID です。 CICS-MQ ブリッジ・モニター・トランザクション (例えば、CKBR またはユーザーのトランザクション名) を開始する際に、**PASSTKTA** パラメーターを使用して、代替アプリケーション ID を指定できます。
- 同じ要求キューに複数のブリッジ・モニターを使用する場合、**PASSTKTA** パラメーターを使用して、ブリッジ・モニターごとに同じアプリケーション ID を指定する必要があります。

CICS サービスはユーザーが APPLID を指定することを許可しないため、パスチケットの検証は、**EXEC CICS VERIFY** ではなく、IBM MQ サービスを使用して実行されます。パスチケットの詳細については、セキュア・サインオンのためのパスチケットの生成と使用および「IBM MQ 製品資料内の『z/OS でのセキュリティーのセットアップ』」を参照してください。

権限

CICS-MQ ブリッジで使用するユーザー ID に以下の権限を与える必要があります。ユーザー ID には、ブリッジ・モニター・トランザクションの開始ユーザー ID (532 ページの表 52 にリストされている) と、IBM MQ アプリケーションが要求メッセージ内で指定するユーザー ID が含まれます。

- ブリッジ・モニター・トランザクションの開始ユーザー ID には、CICS DPL プログラムの CKBP トランザクションと CKBC トランザクション、および IBM MQ アプリケーションが要求メッセージの MQCIH 構造の「TransactionId」フィールドで指定する代替のトランザクションを開始するための権限が必要です。
- IBM MQ アプリケーションが要求メッセージでユーザー ID を指定している場合、ブリッジ・モニター・トランザクションの開始ユーザー ID を、要求メッセージで使用されるすべてのユーザー ID の代理として RACF に対して定義する必要があります。代理ユーザーとは、他のユーザーのパスワードを知らなくてもそのユーザーに代わって作業を開始する権限を持っているユーザーのことです。代理ユーザー・セキュリティーの詳細については、代理ユーザー・セキュリティーを参照してください。
- ブリッジ・モニターのユーザー ID およびすべてのブリッジ・タスクのユーザー ID には、要求キューからメッセージを取得するための権限が必要です。

- ブリッジ・タスクのユーザー ID には、その応答先キューにメッセージを書き込むための権限が必要です。
- エラー応答が受信されるようにするために、ブリッジ・モニター・トランザクションの開始ユーザー ID には、メッセージをすべての応答先キューに書き込むための権限が必要です。
- ブリッジ・タスクのユーザー ID には、送達不能キューにメッセージを書き込むための権限が必要です。
- ブリッジ・モニター・トランザクションの開始ユーザー ID には、メッセージを送達不能キューに書き込むための権限が必要です (エラーが発生した場合にブリッジを停止させたくない場合)。
- ブリッジ・モニターのユーザー ID およびすべてのブリッジ・タスクのユーザー ID には、バックアウト・リキュー・キュー (定義されている場合) にメッセージを書き込むための権限が必要です。

第 17 章 Kerberos のサポート

CICS Transaction Server for z/OS は、Kerberos のサポートを提供します。

CICS は、外部セキュリティー・マネージャー (ESM) を使用して Kerberos をサポートします。サポートのレベルは、ESM によって提供されるサポートによって異なります。ESM が RACF である場合、サポートは Kerberos バージョン 5 および Generic Security Services (GSS) に基づいています。

CICS は、サービス・プロバイダー・パイプラインを構成するか、API コマンド **VERIFY TOKEN** を使用することにより、Kerberos トークンを検証できます。

Kerberos サービスは、**KERBEROSUSER** システム初期設定パラメーターを設定することにより、領域内で使用可能になります。 **KERBEROSUSER** を指定する場合は、CICS 領域の Kerberos サービス・プリンシパルに関連付ける無保護ユーザー ID を使用します。

Kerberos 用の RACF の構成

Kerberos のサポートを有効にするには、RACF などの外部セキュリティー・マネージャーを構成する必要があります。

注: 以下の手順はローカル・プリンシパルの使用にのみ適用されます。

始める前に

- z/OS Integrated Security Services ネットワーク認証サービスを使用して、z/OS Security Server RACF 用に Kerberos 環境をセットアップする必要があります。 RACF、Kerberos および z/OS Integrated Security Services ネットワーク認証サービスについては、RACF および z/OS Integrated Security Services ネットワーク認証サービスを参照してください。
- 構成 `/etc/skrb/krb5.conf` は、LPAR またはシスプレックスのローカル・レルムを定義します。
- SKRBKDC 開始タスクが実行されている必要があります。詳しくは、「z/OS Network File System Guide and Reference」の『Setting up a Kerberos Key Distribution Center』を参照してください。

手順

1. Kerberos に対する RACF 保護を有効にします。 RACF **SETROPTS** コマンドを使用して、KERBLINK クラスに対する RACF 保護をアクティブにします。

   ```
   SETROPTS CLASSACT(KERBLINK)
   ```

   SETROPTS コマンドの詳細については、z/OS Security Server RACF コマンド言語解説書を参照してください。
2. Kerberos を使用するように CICS 領域をセットアップするには、以下のように、サービス・プリンシパル名を定義し、それをユーザー ID に関連付けます。

© Copyright IBM Corp. 1989, 2018

a. **KERBEROSUSER** システム初期設定パラメーターを指定します。 このパラメーターは、領域内の Kerberos サービスのサポートを有効にし、Kerberos サービス・プリンシパルに関連付けるユーザー ID を指定します。無保護ユーザー ID を使用する必要があります。

b. **ALTUSER** コマンドを使用して、**KERBEROSUSER** で指定したユーザー ID にサービス・プリンシパル名を関連付けます。

```
ALTUSER user_id KERB(KERBNAME(service_principal))
```

3. **ALTUSER** コマンドを使用して、RACF ユーザー ID をクライアント・プリンシパルに関連付けます。

```
ALTUSER userid PASSWORD(password) NOEXPIRED KERB(KERBNAME(client_principal))
```

あるいは、以下のコマンドを使用して、関連付けられていないすべてのプリンシパルにデフォルト・ユーザー ID を関連付けることもできます。

```
RDEFINE KERBLINK /.../realm APPLDATA('userid')
```

ここで、*userid* は、レルム *realm* のマップされていないすべてのプリンシパルに関連付けるローカル・ユーザー ID です。

4. これをアクティブ化するには、Kerberos キーを作成する必要があります。これは、ユーザーが次回パスワードを変更したときに自動的に行われます。

Kerberos 用の CICS Web サービスの構成

Kerberos 認証を実装するようにプロバイダー・パイプラインを構成するには、パイプライン構成ファイルにセキュリティー・ハンドラーを追加する必要があります。

始める前に

Kerberos のサポートを有効にするには、RACF などの外部セキュリティー・マネージャーを構成する必要があります。詳しくは、Configuring RACF for Kerberosを参照してください。

Kerberos の構成情報を追加するパイプライン構成ファイルを識別するか、または作成する必要があります。

手順

1. <wsse_handler> エレメントをパイプラインに追加します。このハンドラーを<service_handler_list>エレメントに組み込む必要があります。次のエレメントをコーディングします。

```
<wsse_handler>
    <dfhwsse_configuration version="1">
    </dfhwsse_configuration>
</wsse_handler>
```

<dfhwsse_configuration> エレメントは、構成内の他のエレメントのコンテナーです。

2. <authentication> エレメントをコーディングします。

a. **trust** 属性が trust="basic" を指定するようにコーディングします。

b. **mode** 属性が mode="basic-kerberos" を指定するようにコーディングします。

c. オプション: 空の <suppress/> エレメントをコーディングします。 このエレメントを指定しない場合、処理はトークンに関連したユーザー ID を使用して実行されます。

タスクの結果

CICS プロバイダー・パイプラインは Kerberos 認証用に構成されます。パイプラインから受信するインバウンド Web サービス要求には、有効な Kerberos トークンが含まれている必要があります。含まれていない場合、要求はリジェクトされて、該当する SOAP 障害が返されます。パイプライン構成オプションにしたがって、ターゲット・アプリケーションはトークンに関連したユーザー ID を使用して実行されます。

例

以下の例は、プロバイダー・パイプラインを構成する方法を示しています。

```
<provider_pipeline xmlns="http://www.ibm.com/software/htp/cics/pipeline">
    <service>
        <service_handler_list>
            <wsse_handler>
                <dfhwsse_configuration version="1">
                   <authentication trust="basic" mode="basic-kerberos"/>
                </dfhwsse_configuration>
            </wsse_handler>
        </service_handler_list>
        <terminal_handler>
            <cics_soap_1.1_handler/>
        </terminal_handler>
    </service>
    <apphandler>DFHPITP</apphandler>
</provider_pipeline>
```

Kerberos アプリケーションの開発

Kerberos トークンを検証するためのカスタム・セキュリティー・ハンドラーを作成することも、Kerberos トークンを検証するための独自のアプリケーションを作成することもできます。

始める前に

以下のステップでは、**EXEC CICS VERIFY TOKEN** コマンドの ISUSERID オプションを使用して、Kerberos プリンシパルの RACF ユーザー ID を抽出できます。そのためには、まずユーザー ID とプリンシパルの間の関連を定義しておく必要があります。関連をセットアップするには、RACF **RACLINK** コマンドを使用します。詳しくは、『z/OS Security Server RACF Security Administrator's Guide』の『ユーザー ID アソシエーションの定義』を参照してください。

手順

要件に応じて、以下のいずれかの手法を使用してください。

- カスタムのセキュリティー・ハンドラーの作成の説明に従って、**VERIFY TOKEN** コマンドを使用するセキュリティー・ハンドラーを作成します。トークンに関連付けられた Kerberos プリンシパルのユーザー ID で実行する場合は、**VERIFY TOKEN** コマンドの ISUSERID オプションを使用します。

- 独自のフロントエンド・セキュリティー・プログラムを作成します。このようなプログラムでは、HTTP ヘッダーまたは IBM MQ メッセージから Kerberos トークンを抽出した後に、**VERIFY TOKEN** コマンドを発行することができます。トークンに関連付けられた Kerberos プリンシパルのユーザー ID で実行する場合は、**VERIFY TOKEN** コマンドの ISUSERID オプションを使用してユーザー ID を取得します。その後、そのユーザー ID で新しい要求を開始できます。

3270 エミュレーター・サインオンでの Kerberos セキュリティー・トークンの使用

Kerberos を使用すると、セキュリティーが強化されます。これはパスワードがネットワーク経由でフローする必要がないためです。

プロセスは次のとおりです。

1. クライアント端末エミュレーターが Kerberos 認証サーバーに適用され、Kerberos トークンを取得します。
2. Kerberos トークンはクライアント端末エミュレーターに返され、内容は Base64 形式でエンコードされます。
3. 次に、トークンは CICS サーバーにメッセージで転送されます。ここで、サインオン・トランザクションは Base64 でエンコードされた Kerberos トークンを受け取り、**SIGNON TOKEN** コマンドを発行します。
4. RACF Kerberos レジストリーは Kerberos トークンを検証し、関連付けられた RACF USERID を CICS に返します。この USERID は、後続のタスクの端末セッションに関連付けられます。

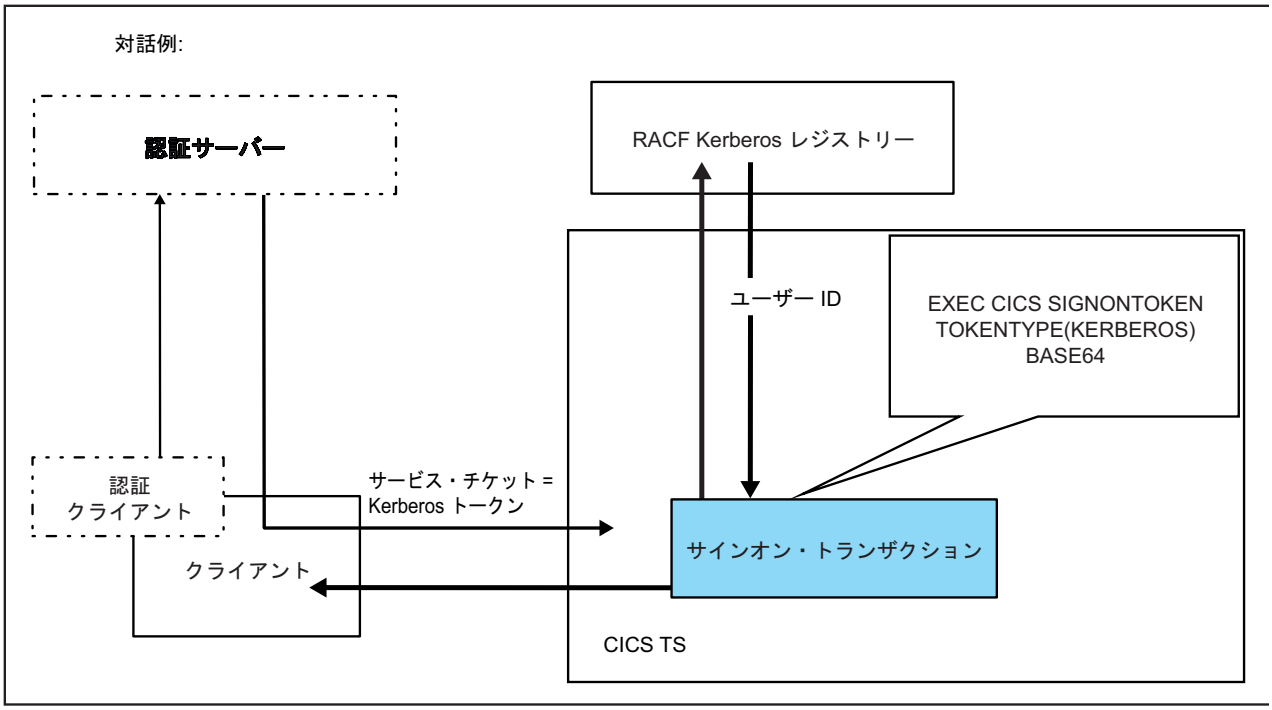


図 32. クライアント端末エミュレーター、認証サーバー、および *CICS TS* の間の要求のフロー

注: ログオン・データは 255 文字に制限されているため、Kerberos トークンの送信に使用することはできません。

第 18 章 RACF を使用した Multi-Factor Authentication のサポート

CICS Transaction Server for z/OS は、RACF を使用した Multi-Factor Authentication (MFA) のサポートを提供します。

RACF ユーザーである場合は、MFA の概要とこの機能の前提条件について、「z/OS Security Server RACF セキュリティー管理者のガイド」の『Multi-Factor Authentication for z/OS』 を参照してください。

他のセキュリティー製品を使用している場合は、サポートおよび前提条件の詳細について、ご使用の ESM の資料を参照してください。

以下の情報は、RACF および IBM Multi-Factor Authentication for z/OS の例に基づいて、MFA を CICS に実装する方法を示しています。

CICS は、インバンド MFA トークンをサポートしています。 z/OS アウト・オブ・バンド認証を使用する場合は、ワンタイム使用トークンを生成可能で、これは CICS でサポートされます。

MFA トークンは、以下のセッション・ベースのログオン・インターフェースでサポートされます。

表 *53. MFA* をサポートしているセッション・ベースのログオン・インターフェース

Interface (インターフェース**)**	**CICS** レベルの要件
CICS Explorer®	CICS TS V5.4 (APAR PI87691 適用済み) 以降
CESN および CESL	• CICS TS V4.2 (APAR PI21865 適用済み) • CICS TS V5.1 (APAR PI21866 適用済み) • CICS TS V5.2 (APAR PI21866 適用済み) • CICS TS V5.3 以降
CICSPlex SM Web ユーザー・インターフェース	
EXEC CICS SIGNON を使用したユーザー作成サインオン・プログラム	

MFA トークンは、資格情報をキャッシュに入れるステートレス要求ではサポートされません。

長さに応じて、MFA トークンは、フェーズ・フィールドまたはパスワード・フィールドに入力する必要があります。

詳しくは、IBM Multi-Factor Authentication for z/OS ユーザーズ・ガイドを参照してください。

© Copyright IBM Corp. 1989, 2018

第 19 章 SAML 用の CICS の構成

SAML を使用するように CICS を構成するには、最初にサンプル JVM サーバー・プロファイルをカスタマイズおよびインストールすることによって JVM サーバーを構成してから、適切な CICS 領域に CSD グループをインストールします。

始める前に

CICS セキュリティー・トークン・サービス (STS) をデプロイする領域を識別する必要があります。アプリケーション・コードの存在しない領域に STS をインストールします。SAML トークンを検証する領域にアプリケーション・コードが存在する場合、リモートで STS を定義します。Java コードを実行する領域を他の領域から分離する場合も、リモートで領域を定義することを選択できます。STS に別個の領域を使用する別の理由は、その領域を独自の鍵リングで定義できることです。この鍵リングには、署名の検証および SAML トークンの署名に必要な証明書のみが含まれています。

このタスクについて

CICS は、DFHSAML という名前のリンク可能インターフェースを提供します。このインターフェースによって、CICS Web サービス・パイプラインおよびアプリケーションは、SAML アサーションからの情報を検証し抽出できます。CICS における SAML のサポートでは、ご使用のシステムにインストールおよび構成された JVM サーバーが必要です。

手順

1. JVM サーバー用に JVM サーバー・プロファイルを作成します。

   適切な提供プロファイル (DFHJVMST) をインストール・ディレクトリーから、**JVMPROFILEDIR** システム初期設定パラメーターで指定されたディレクトリーにコピーすることができます。

2. 以下のように、選択した構成で CSD グループ DFHSAML をインストールします。
   a. STS を実行するために選択した領域に DFHSAML をインストールします。
   b. SAML をリモートで使用する場合、STS を実行する領域を指し示す DFHSAML のリモート・プログラム定義を定義します。

   注**:** 独自の JVM サーバー定義を使用する場合、DFHSAML をコピーし、このグループをカスタマイズし、DFHSAML グループの代わりにカスタマイズしたグループをインストールします。新しいグループは、独自の JVM サーバー定義をポイントする必要があります。セキュリティー・トークン拡張サポートを呼び出すすべてのプログラムは、JVM サーバーの名前を使用して DFHSAML JVMSERVER コンテナーを作成する必要があります。

タスクの結果

CICS が SAML 用に構成されました。

© Copyright IBM Corp. 1989, 2018

次のタスク

『SAML 用の CICS の構成の検証』で説明されているように、構成を検証できます。

SAML 用の CICS の構成の検証

CICS が SAML に対して正常に構成されていることを検証するために使用できるサンプルが提供されています。コンパイルしてから、トランザクションを介して呼び出すことができる 2 つのプログラムが提供されています。

始める前に

545 ページの『第 19 章 SAML 用の CICS の構成』で説明されているように、JVM サーバーを構成する必要があります。

このタスクについて

サンプルは、サンプル・プログラム、トランザクション、およびテンプレートのプログラム定義が含まれる CSD グループ DFH$SAML で提供されます。このサンプルを使用して構成を検証できます。サンプル・アプリケーションをコンパイルおよびデプロイすると、CICS セキュリティー・トークン拡張によって処理される SAML トークン・アサーションのサンプルが提供されます。アプリケーションは、CICS トランザクションによって開始されます。

手順

1. オプション: DFHXSTS 以外の名前で JVM サーバーをカスタマイズしてインストールした場合、プログラム DFH0XST2 を更新して新しいサーバー名を反映します。
2. サンプル・ライブラリー SDFHSAMP にあるプログラム DFH0XST1 および DFH0XST2 をコンパイルします。COBOL プログラムのコンパイルについては、COBOL プログラムのバッチ・コンパイルを参照してください。
3. DFHSAML プログラムを呼び出す領域にグループ DFH$SAML をインストールします。
4. トランザクション XST1 を実行します。

タスクの結果

サンプル・トランザクション XST1 が正常に実行される場合、SAML サポートは正しく構成されています。

サンプルは、構文解析されたコンテナーを TSQ DFH0XSTO に出力します。

これらのコンテナーを確認するために、**CEBR DFH0XSTO** を使用します。

インストール検証が失敗した場合、DFHSAML-RESPONSE コンテナーに、理由を示す戻りコードが保管されます。コンテナー応答コードについて詳しくは、SAML サポート・コンテナーを参照してください。

異常終了コードが返された場合、詳しくは、サンプルを参照してください。

次のタスク

- サンプル SAML トークンを独自のトークンに置き換えることができます。SAML トークンが含まれるファイルの名前を指定する DOCTEMPLATE リソース定義を作成して、インストールします。サンプルの実行時に、トランザクション ID の後にこの DOCTEMPLATE の 48 バイトの TEMPLATENAME を指定します。

  ```
  XST1 templatename
  ```

  templatename が指定されていない場合、デフォルトの TEMPLATENAME である DFH0XSTI が使用されます。
- シグニチャー検証を使用する場合、プログラム DFH0XST2 を更新します。詳しくは、そのプログラム内のコメントを参照してください。

SAML トークンを使用するためのプロバイダー・パイプラインの構成

SAML トークン・アサーションの検証と抽出を使用できるようにプロバイダー・パイプラインを構成し、それらを CICS アプリケーションで使用できるようにします。

始める前に

SAML サポートを既存のパイプラインに追加している場合、パイプライン構成ファイルを識別します。新規のパイプラインを作成している場合、新規のパイプライン構成ファイルを作成します。CICS は、2 つのサンプル構成ファイル、`samlprovider.xml` と `propagatesamlprovider.xml` を提供します。後者には **`tran_channel`** 属性が含まれています。

手順

1. パイプライン構成ファイルに `<sts_authentication>` エレメントをコーディングします。 このエレメントは、セキュリティー・トークン・サービス (STS) が認証に使用されることを指定し、送信される要求のタイプを決定します。`<authentication>` エレメントはコーディングしないでください。
   a. **`action="validate"`** 属性をコーディングして、STS がセキュリティー・トークンを検証することを指定します。 この属性を指定しない場合、CICS は、アクションが ID トークンを要求することであると想定します。
   b. オプション: CICS が署名を無視することを指定するには、**`token_signature="ignored"`** 属性をコーディングします。 この属性を指定しない場合、デフォルト値は `required` です。この値は、有効なシグニチャーを指定する必要があることを意味します。
   c. オプション: SAML トークンのエレメントをコンテナーに入れないことを指定するには、**`extract="no"`** 属性をコーディングします。 この属性を指定しない場合、デフォルト値は `yes` です。これは、CICS が SAML トークンの主要なエレメントを含むコンテナーを作成することを意味します。これらのコンテナーの詳細については、SAML サポート・コンテナーを参照してください。
   d. オプション: CICS アプリケーションによる SAML 情報の伝搬を許可するために、SAML アサーションが DFHTRANSACTION チャネルのコンテナーにコピーされることを指定するには、**`tran_channel="yes"`** 属性をコーデ

ィングします。 この属性を指定しない場合、デフォルト値は no です。これは、パイプラインから渡されるチャネルのコンテナーでアサーションが使用可能になることを意味します。

e. <sts_authentication> エレメント内に、<auth_token_type> エレメントをコーディングします。 <auth_token_type> エレメント内に、以下のエレメントをコーディングします。

<namespace>
使用する SAML のバージョンに応じて、このエレメントの内容を urn:oasis:names:tc:SAML:1.0:assertion または urn:oasis:names:tc:SAML:2.0:assertion のいずれかに設定します。

<element>
このエレメントの内容を Assertion に設定します。

2. <sts_endpoint> エレメントをコーディングします。

a. <sts_endpoint> エレメントに、<endpoint> エレメントをコーディングします。 このエレメントには、ネットワーク上のセキュリティー・トークン・サービス (STS) の場所を指し示す URI が含まれます。

CICS セキュリティー・トークン・サービスを使用して SAML トークンを処理するには、エンドポイントを cics://PROGRAM/DFHSAML に設定します。CICS セキュリティー・トークン・サービスがセキュリティー・ハンドラーによって呼び出されます。<sts_authentication> エレメントで **extract="yes"** が構成されている場合、「DFHSAML」が先頭に付くコンテナーがパイプライン・チャネルにコピーされ、後続のパイプライン・ハンドラーおよびターゲット・アプリケーションで使用できるようになります。

b. オプション: <sts_endpoint> エレメントに、<jvmserver> エレメントをコーディングします。 このエレメントには実行するサーバーを指定します。このエレメントが含まれていない場合、デフォルトのサンプル・リソース JVM サーバー DFHXSTS が想定されます。

例

```
<?xml version="1.0" encoding="EBCDIC-CP-US"?>
<provider_pipeline xmlns="http://www.ibm.com/software/htp/cics/pipeline">
  <service>
    <service_handler_list>
      <wsse_handler>
        <dfhwsse_configuration version="1">
          <sts_authentication action="validate" token_signature="required"
           extract="yes" tran_channel="yes">
            <auth_token_type>
              <namespace>urn:oasis:names:tc:SAML:2.0:assertion</namespace>
              <element>Assertion</element>
            </auth_token_type>
          </sts_authentication>
          <sts_endpoint>
            <endpoint>cics://PROGRAM/DFHSAML</endpoint>
            <jvmserver>DFHXSTS</jvmserver>
          </sts_endpoint>
        </dfhwsse_configuration>
      </wsse_handler>
    </service_handler_list>
    <terminal_handler>
      <cics_soap_1.1_handler/>
```

```
    </terminal_handler>
  </service>
  <apphandler>DFHPITP</apphandler>
</provider_pipeline>
```

SAML トークンを使用するためのリクエスター・パイプラインの構成

アウトバウンド・メッセージの SOAP ヘッダーに SAML トークンを挿入するようにリクエスター・パイプラインを構成します。

始める前に

SAML サポートを既存のパイプラインに追加している場合、パイプライン構成ファイルを識別します。新規のパイプラインを作成している場合、新規のパイプライン構成ファイルを作成します。CICS は、2 つのサンプル構成ファイル、`samlrequester.xml` と `propagatesamlrequester.xml` を提供します。後者には **`tran_channel`** 属性が含まれています。

このタスクについて

このタスクでは、SAML トークンをアウトバウンド SOAP メッセージに自動的に付加するようにリクエスター・パイプラインを構成します。CICS は、以前に検証されたトークンのみが SOAP メッセージに送信されることを許可します。検証済みのトークンは DFHSAML-OUTTOKEN コンテナーに保管され、SOAP ヘッダーに挿入されたトークンはこのコンテナーから取得されます。

手順

1. パイプライン構成ファイルに `<sts_authentication>` エレメントをコーディングします。 このエレメントは、セキュリティー・トークン・サービス (STS) が認証に使用されることを指定し、送信される要求のタイプを決定します。 `<authentication>` エレメントはコーディングしないでください。
   a. オプション: DFHTRANSACTION チャネルの DFHSAML-OUTTOKEN コンテナーの内容が要求の SAML トークンとして使用されることを指定するには、**`tran_channel="yes"`** 属性をコーディングします。 この属性を指定しない場合、デフォルト値は `no` です。これは、SAML トークンが SOAP パイプラインに渡されるチャネルの DFHSAML-OUTTOKEN コンテナーから取得されることを意味します。
   b. `<sts_authentication>` エレメント内に、`<auth_token_type>` エレメントをコーディングします。 `<auth_token_type>` エレメント内に、以下のエレメントをコーディングします。

      <namespace>
      使用する SAML のバージョンに応じて、このエレメントの内容を `urn:oasis:names:tc:SAML:1.0:assertion` または `urn:oasis:names:tc:SAML:2.0:assertion` のいずれかに設定します。

      <element>
      このエレメントの内容を `Assertion` に設定します。
2. `<sts_endpoint>` エレメントをコーディングします。

a. <sts_endpoint> エレメントに、<endpoint> エレメントをコーディングします。

   エンドポイントの値を cics://PROGRAM/DFHSAML に設定します。

b. オプション: <sts_endpoint> エレメントに、<jvmserver> エレメントをコーディングします。 このエレメントには実行するサーバーを指定します。このエレメントが含まれていない場合、デフォルトのサンプル・リソース JVM サーバー DFHXSTS が想定されます。

例

```
<?xml version="1.0" encoding="EBCDIC-CP-US"?>
<requester_pipeline xmlns="http://www.ibm.com/software/htp/cics/pipeline">
  <service>
    <service_handler_list>
      <cics_soap_1.1_handler/>
      <wsse_handler>
        <dfhwsse_configuration version="1">
          <sts_authentication tran_channel="yes">
            <auth_token_type>
              <namespace>urn:oasis:names:tc:SAML:2.0:assertion</namespace>
              <element>Assertion</element>
            </auth_token_type>
          </sts_authentication>
          <sts_endpoint>
            <endpoint>cics://PROGRAM/DFHSAML</endpoint>
            <jvmserver>DFHXSTS</jvmserver>
          </sts_endpoint>
        </dfhwsse_configuration>
      </wsse_handler>
    </service_handler_list>
  </service>
  <service_parameter_list/>
</requester_pipeline>
```

CICS セキュリティー・トークン・サービスの構成

CICS STS の動作を制御するには、STS 構成ファイルを更新し、可能であれば java.policy ファイルとシステム初期設定パラメーターを更新します。

このタスクについて

CICS は *STS* 構成ファイル と呼ばれるファイルを使用して、CICS STS の動作を制御します。デフォルトでは、CICS は **JVMPROFILEDIR** SIT パラメーターによって識別されるディレクトリー内の sts.xml という名前のファイルを使用します。そのファイルが存在しない場合、CICS は JVM サーバー・プロファイル内の JVM プロパティー **com.ibm.cics.sts.config** によって指定されるファイルを使用します。例えば、-Dcom.ibm.cics.sts.config=/var/security/sts/sts-config.xml です。サンプル・ファイル sts-config.xml が用意されています。

手順

1. 要件に従って、STS 構成ファイルを更新します。
   a. 属性をトークンに追加するときに SAML トークンの発行者を更新する場合、発行者 を指定します。
      1) STS 構成ファイルに <issuer> エレメントをコーディングします。

2) <issuer> エレメント内に、<format> エレメントをコーディングします。

3) <issuer> エレメント内に、<uri> エレメントをコーディングします。

b. さまざまなコンピューター上のシステム・クロックの時差を許可するには、許容値またはクロック・スキュー を指定できます。 SAML トークンが定期的に「有効期限切れ」または「まだ有効でない」として拒否される場合、発行側システムと受信側システムの間のクロック時間の不一致が原因である可能性があります。スキュー時間を設定すると、SAML トークンが発行される前に到着しているように見えたり、実際に発行されて到着したときに既に有効期限切れになっていたりすることがなくなります。しかし、設定する値が非常に大きい場合は、実際に有効期限が切れたトークンを受け入れることもできます。

1) STS 構成ファイルに <clock_skew> エレメントをコーディングします。値はミリ秒単位で設定されるため、90 秒の時間を設定するには、`<clock_skew>90000</clock_skew>` とコーディングします。

c. SAML トークンに再署名する予定がある場合は、証明書ラベルを指定します。

1) STS 構成ファイルに <signature> エレメントをコーディングします。使用するハッシュ・アルゴリズムに応じて、`hash_algorithm` 属性を `sha-1` または `sha-2` に設定します。

注**:** ハッシュ・アルゴリズム `sha-3` はサポートされていません。

2) オプション: <signature> エレメント内に <certificate> エレメントをコーディングします。

3) オプション: <certificate> エレメント内に <label> エレメントをコーディングします。それを RACF 証明書ラベルの値に設定します。

d. 署名の検証または SAML トークンの再署名を行う場合は、鍵ストア・タイプを指定できます。

1) STS 構成ファイルに <keystore> エレメントをコーディングします。

2) <keystore> エレメント内に <type> エレメントをコーディングします。署名の検証に暗号化ハードウェアを使用する場合、値 `JCECCARACFKS` を指定します。使用しない場合、値 `JCERACFKS` (デフォルト値) を使用します。

2. オプション: 署名の検証または SAML トークンの再署名を行う予定がある場合は、`java.policy` ファイルを更新して、**`KEYRING`** SIT パラメーターを設定します。

a. `java.policy` ファイルをアップデートして、Java セキュリティー・プロバイダーのリストを変更する権限を、CICS ユーザーに付与します。 サンプル・ファイル `sts-java.policy` が提供されています。

1) サンプル・ファイルを使用するには、サンプル・ファイルを編集して `/&USSHOME/` をインストール済み環境に適切な値に変更し、そのパスおよびファイル名が指定されるように JVM サーバー・プロファイルの JVM プロパティー **`java.security.policy`** を設定します。

2) 既存の `java.policy` ファイルを使用するには、以下の規則を含むようそのファイルをアップデートします。

```
// All permissions granted to CICS codesource protection domain
grant codeBase "file:///&USSHOME///-" {
permission java.security.AllPermission;
};
```

ここで、*USSHOME* は、z/OS UNIX 上の CICS Transaction Server ファイルのルート・ディレクトリーの名前とパスです。

b. SIT パラメーター **KEYRING** を設定して、使用する証明書のセットを含む RACF 鍵リングを指定します。

STS 構成ファイル

STS 構成ファイルは、CICS セキュリティー・トークン・サービス (STS) のさまざまな局面を指定します。

CICS は、**JVMPROFILEDIR** SIT パラメーターによって識別されたディレクトリーにある sts.xml というファイルを使用します。そのファイルが存在しない場合、CICS は JVM サーバー・プロファイル内の JVM プロパティー **com.ibm.cics.sts.config** によって指定されるファイルを使用します。例えば、-Dcom.ibm.cics.sts.config=/var/security/sts/sts-config.xml です。サンプルの STS 構成ファイル、sts-config.xml が参照用に用意されています。

XML スキーマ・ファイル、sts.xsd は /usr/lpp/cicsts/cicsts52/schemas/sts ディレクトリーに用意されています。

STS 構成ファイルには以下のエレメントが含まれます。

<keystore>
RACF 鍵ストア・タイプを定義します。可能な値は JCERACFKS および JCECCARACFKS です。デフォルト値は JCERACFKS です。

<issuer>
STS をアサーティング・パーティーとして定義します。このエレメントには以下のエレメントが含まれます。

<format>
任意のストリングが含まれます。デフォルト値はありません。

<uri>
任意のストリングが含まれます。デフォルト値は http://cics です。

<signature>
ハッシュ・アルゴリズムおよび証明書ラベルを指定します。このエレメントには以下の属性があります。

hash_algorithm
可能な値は sha-1 と sha-2 です。デフォルト値は sha-2 です。

このエレメントには、次のエレメントが含まれます。

<certificate>
このエレメントには、次のエレメントが含まれます。

<label>
RACF 証明書ラベルの値。デフォルト値は CICSCERT です。

<clock_skew>
クロック・スキュー時間 (ミリ秒単位)。デフォルト値は 180000 ミリ秒 (3 分) です。

クロック・スキューは、異なるコンピューター上のシステム・クロックの時差を許可します。これは、SAML トークンのすべてのタイミング条件に適用されます。

次の図は、すべてのエレメントが設定された STS 構成ファイルの例を示しています。

```
<sts_configuration xmlns="http://www.ibm.com/xmlns/prod/cics/JVMSERVER/stsconfig">

 <keystore>
  <type>JCECCARACFKS</type>
 </keystore>

 <issuer>
  <format>urn:oasis:names:tc:SAML:2.0:nameid-format:entity</format>
  <uri>http://cics</uri>
 </issuer>

 <signature hash_algorithm="sha-1">
  <certificate>
   <label>CICSCERT</label>
  </certificate>
 </signature>

 <clock_skew>90000</clock_skew>

</sts_configuration>
```

SAML 対応プログラムを開発するパターン

SAML 対応プログラムは、一般的なパターンに準拠する場合があります。そのようなパターンの 1 つは、アプリケーションのいくつかの部分へのアクセスを制御する初期プログラムです。もう 1 つは、ユーザーに関する情報をログに記録することです。

パターン: 検証済みトークンの再利用

SAML トークンを検証し、後で同じトランザクションで、リクエスター・プログラムから Web サービスを呼び出して、同じトークンを使用できます。検証された SAML トークンは DFHSAML-OUTTOKEN コンテナーに保持されます。このコンテナーは読み取り専用であるため、チャネル間で移動できません。検証要求が再発行されないようにしてパフォーマンスを向上させるには、トランザクション・チャネル、DFHTRANSACTION を使用できます。

着信 Web サービスから SAML トークンを検証するとき、プロバイダー・パイプラインの構成ファイル内の `<sts_authentication>` エレメントに含まれる **tran_channel="yes"** 属性をコーディングします。この属性は、SAML アサーションが出力コンテナーから DFHTRANSACTION チャネルのコンテナーにコピーされることを指定します。

検証された SAML トークンを Web サービスで再利用するには、Web サービスによって使用されるリクエスター・パイプラインの構成ファイル内の

<sts_authentication> エレメントに **tran_channel="yes"** 属性をコーディングします。

第 20 章 SAML アプリケーションの開発

CICS で提供される SAML 機能を使用するアプリケーションを開発できます。

このタスクについて

SAML トークンを処理するアプリケーションの開発方法をいくつかの方法の中から選択することができます。例:

- SAML トークンを着信メッセージから抽出するプログラムを作成する
- SAML トークンをアウトバウンド要求に配置するアプリケーションを作成する
- 属性を SAML トークンに追加して、そのトークンに再署名するアプリケーションを作成する
- SAML トークンを作成するアプリケーションを作成する

SAML 対応の初期プログラムの開発

SAML トークンを抽出し、そのトークンを処理するため DFHSAML にリンクするプログラムを作成します。

このタスクについて

Web サービスの使用 (SAML トークンを使用するためのプロバイダー・パイプラインの構成を参照してください) に代わる方法として、独自の SAML 対応初期プログラムをセキュリティー・フロントエンドとして作成し、SAML トークンを検証することができます。このようなプログラムは、HTTP、IBM MQ、または SOAP メッセージ以外の他のプロトコルを使用するメッセージを使用している場合に役立ちます。

SAML 対応の初期プログラムは、SAML トークンをメッセージから抽出します。それから、そのトークンをチャネルにある文字コンテナー DFHSAML-TOKEN に入れます。その後、プログラム DFHSAML をチャネルとリンクし、トークンの検証およびコンテナーの抽出を実行します。コンテナーの詳細については、SAML サポート・コンテナーを参照してください。

ユーザー定義のチャネルまたはトランザクション・チャネル (DFHTRANSACTION) のいずれかを使用できます。ユーザー定義のチャネルを使用する場合、コンテナーは、そのチャネルを明示的に渡す LINK 要求で渡します。トランザクション・チャネルを使用する場合、コンテナーは、そのトランザクション全体で使用できます。

アプリケーションは、**GET CONTAINER** コマンドを使用することによって、コンテナーに含まれている情報 (SAML の属性など) を使用できます。

この情報の使用法を示す具体例については、 556 ページの『SAML 対応の初期プログラム開発のパターン』を参照してください。

© Copyright IBM Corp. 1989, 2018

SAML 対応の初期プログラム開発のパターン

SAML 対応プログラムは、一般的なパターンに準拠する場合があります。そのようなパターンの 1 つは、アプリケーションのいくつかの部分へのアクセスを制御する初期プログラムです。

アプリケーションの代表的なパターンとして、初期プログラムを SAML 対応にすることが挙げられます。このパターンでは、アプリケーションの該当する部分を実行する前に、プログラムが SAML アサーション内の情報を使用して決定することを意味します。

SAML アサーションの情報は、読み取り専用のコンテナーに存在します。これらのコンテナーは、SOAP パイプラインからプログラムに渡されるチャネル内に存在するか、またはプログラムが独自のメッセージ処理および SAML 検証を実行する場合、DFHSAML プログラムから返されます。

SAML 対応プログラムが実行する処理の一例は、属性コンテナーから情報を取得する処理です。プログラムは属性名コンテナー (例えば、DFHSAML-ATTRN001) やそのコンテナー内の属性値 (例えば、DFHSAML-A001V001) を検索します。この属性は、ユーザーの役割や権限を示すためにプログラムによって使用される場合があります。また、この属性よって、アプリケーションのどの部分を呼び出し元が使用できるかをアプリケーションで選択できます。

アプリケーションは、情報をアプリケーションの他のプログラムに渡す必要が生じる場合があります。コンテナーは読み取り専用なので、LINK、XCTL、RETURN、または START コマンドで、チャネルを次のプログラムまたは CHANNEL インターフェースを使用するトランザクションに渡すことによって、情報を安全に渡すことができます。

また、アプリケーションは、トランザクションを元の SAML トークンと (したがって、ユーザーに) 関連付けることができるように、要求を監査する必要がある場合もあります。この関連付けを実行するには、検証された SAML トークン (コンテナー DFHSAML-OUTTOKEN 内) または選択されたコンテナーをジャーナルに書き込むカスタマー・ロギング・プログラムを作成します。

アウトバウンド要求で検証済み SAML トークンを使用するプログラムの開発

検証済み SAML トークンをアウトバウンド要求に追加するように Web サービスを構成します。

このタスクについて

アプリケーションが SAML トークンをアウトバウンド要求で使用する場合があります。 SAML トークンをアウトバウンド要求に追加するように Web サービスを構成できます。手順については、SAML トークンを使用するためのリクエスター・パイプラインの構成を参照してください。

アプリケーションによって SAML トークンを Web サービス要求の SOAP メッセージに挿入するには、そのトークンがまず検証済みになっている必要があります。あるいは、アプリケーションでそのトークンを検証してから、そのトークンに属性

を追加し、その後 Web サービスを呼び出すこともできます。詳しくは、『SAML トークンを作成したり拡張したりするプログラムの開発』を参照してください。

SAML トークンは、読み取り専用コンテナー DFHSAML-OUTTOKEN に格納されます。

SAML トークンを作成したり拡張したりするプログラムの開発

SAML トークンを作成したり、SAML トークンに属性を追加してそのトークンに再署名したりするための CICS アプリケーションや Web サービス・アプリケーションを開発できます。

このタスクについて

CICS SAML サポートを使用して、SAML トークンに属性を追加したり、CICS STS 構成ファイルで指定されている証明書によって要求に再署名したりすることができます。SAML トークンは、外部の送信側から受け取ることも、テンプレートから作成することもできます。開発するアプリケーションは、CICS アプリケーションまたは Web サービス・アプリケーションのいずれかにすることができます。

重要**:** トークンの拡張に関与するすべてのアプリケーション・コードがフェデレーションの他のメンバーによって信頼されている領域でのみ、トークンの作成、拡張、再署名を実行してください。

手順

1. 属性の追加は、元の SAML トークンの検証で使用したのと同じチャネルで以下のコンテナーを作成することによって実行します。

   注**:** トークンを変更するには、そのトークンが検証済みになっている必要があります。検証済みのトークンは、DFHSAML-OUTTOKEN コンテナーに入っています。

   a. 属性の名前をコンテナー DFHSAML-ATTRN*aaa* に入れます (*aaa* は 3 文字の大文字英数字です)。 以下に例を示します。

      ```
      EXEC CICS PUT CONTAINER('DFHSAML-ATTRNORG')
      CHANNEL('SAML-CHANNEL') FROM('title')
      ```

   b. オプション: 属性の名前空間をコンテナー DFHSAML-ATTRS*aaa* に入れます (*aaa* は属性名のコンテナーで使用したのと同じ文字です)。 SAML バージョン 2.0 の場合は、この手順を実行する必要はありません。

   c. オプション: 属性のフレンドリー名をコンテナー DFHSAML-ATTRY*aaa* に入れます (*aaa* は属性名のコンテナーで使用したのと同じ文字です)。 以下に例を示します。

      ```
      EXEC CICS PUT CONTAINER('DFHSAML-ATTRYORG')
      CHANNEL('SAML-CHANNEL') FROM('eduPersonAffiliation')
      ```

   d. オプション: 属性の形式をコンテナー DFHSAML-ATTRF*aaa* に入れます (*aaa* は属性名のコンテナーで使用したのと同じ文字)。 以下に例を示します。

```
EXEC CICS PUT CONTAINER('DFHSAML-ATTRNORG')
CHANNEL('SAML-CHANNEL')
FROM('urn:oasis:names:tc:SAML:2.0:attrname-format:uri')
```

e. オプション: 1 つ以上の属性値をコンテナー DFHSAML-A*aaa*N*bbb* に入れます (*aaa* は属性名のコンテナーで使用したのと同じ文字、*bbb* は 3 文字の大文字英数字です)。 以下に例を示します。

```
EXEC CICS PUT CONTAINER('DFHSAML-AORGV001')
CHANNEL('SAML-CHANNEL') FROM('staff')
EXEC CICS PUT CONTAINER('DFHSAML-AORGV002') CHANNEL('SAML-CHANNEL')
FROM('employee')
```

2. 以下のいずれかの方法でトークンを作成します。
   - SAML-ISSUE 値を DFHSAML-FUNCTION コンテナーに入れて、リンク可能なインターフェース DFHSAML にリンクすると、新しいトークンが作成されます。デフォルトでは、STS 構成ファイルで指定されている署名オプションを使用することにより、このトークンの再署名がなされます。署名が不要な場合は、アプリケーションで、SAML-IGNORED オプションが指定されている DFHSAML-SIGNED コンテナーを作成してから、DFHSAML を呼び出すこともできます。STS 構成ファイルで <issuer> が指定されている場合は、その値が新しい SAML トークンで使用されます。
   - Web サービスを呼び出します。その Web サービスに関連付けられているリクエスター・パイプラインが SAML に対応するように構成されている場合は、元のトークンに自動的に属性が追加され、新しいトークンが作成されます。デフォルトでは、STS 構成ファイルで指定されている署名オプションを使用することにより、パイプラインによってこの SAML トークンの再署名がなされます。署名が不要な場合は、リクエスター・パイプラインの構成オプション token_signature を no に設定してください。STS 構成ファイルで <issuer> が指定されている場合は、その値が新しい SAML トークンで使用されます。リクエスター・パイプラインは、新しい SAML トークンを作成する処理に加えて、その SAML トークンをアウトバウンド Web サービス要求に挿入する処理も実行します。

特記事項

本書は米国 IBM が提供する製品およびサービスについて作成したものです。この資料の他の言語版を IBM から入手できる場合があります。ただし、これを入手するには、本製品または当該言語版製品を所有している必要がある場合があります。

本書に記載の製品、サービス、または機能が日本においては提供されていない場合があります。日本で利用可能な製品、サービス、および機能については、日本 IBM の営業担当員にお尋ねください。本書で IBM 製品、プログラム、またはサービスに言及していても、その IBM 製品、プログラム、またはサービスのみが使用可能であることを意味するものではありません。 IBM 製品、プログラムまたはサービスに代えて、IBM の知的所有権を侵害することのない機能的に同等のプログラムまたは製品を使用することができます。ただし、IBM 以外の製品とプログラムの操作またはサービスの評価および検証は、お客様の責任で行っていただきます。

IBM は、本書に記載されている内容に関して特許権 (特許出願中のものを含む) を保有している場合があります。本書の提供は、お客様にこれらの特許権について実施権を許諾することを意味するものではありません。実施権についてのお問い合わせは、書面にて下記宛先にお送りください。

〒*103-8510*
東京都中央区日本橋箱崎町*19*番*21*号
日本アイ・ビー・エム株式会社
法務・知的財産
知的財産権ライセンス渉外

IBM およびその直接または間接の子会社は、本書を特定物として現存するままの状態で提供し、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任を負わないものとします。国または地域によっては、法律の強行規定により、保証責任の制限が禁じられる場合、強行規定の制限を受けるものとします。

この情報には、技術的に不適切な記述や誤植を含む場合があります。本書は定期的に見直され、必要な変更は本書の次版に組み込まれます。 IBM は予告なしに、随時、この文書に記載されている製品またはプログラムに対して、改良または変更を行うことがあります。

本書において IBM 以外の Web サイトに言及している場合がありますが、便宜のため記載しただけであり、決してそれらの Web サイトを推奨するものではありません。それらの Web サイトにある資料は、この IBM 製品の資料の一部ではありません。それらの Web サイトは、お客様自身の責任でご使用ください。

IBM は、お客様が提供するいかなる情報も、お客様に対してなんら義務も負うことのない、自ら適切と信ずる方法で、使用もしくは配布することができるものとします。

© Copyright IBM Corp. 1989, 2018

本プログラムのライセンス保持者で、(i) 独自に作成したプログラムとその他のプログラム (本プログラムを含む) との間での情報交換、および (ii) 交換された情報の相互利用を可能にすることを目的として、本プログラムに関する情報を必要とする方は、下記に連絡してください。

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119 Armonk,
NY 10504-1785
United States of America

本プログラムに関する上記の情報は、適切な使用条件の下で使用することができますが、有償の場合もあります。

本書で説明されているライセンス・プログラムまたはその他のライセンス資料は、IBM 所定のプログラム契約の契約条項、IBM プログラムのご使用条件、またはそれと同等の条項に基づいて、IBM より提供されます。

IBM 以外の製品に関する情報は、その製品の供給者、出版物、もしくはその他の公に利用可能なソースから入手したものです。IBM は、それらの製品のテストは行っておりません。したがって、他社製品に関する実行性、互換性、またはその他の要求については確証できません。 IBM 以外の製品の性能に関する質問は、それらの製品の供給者にお願いします。

本書には、日常の業務処理で用いられるデータや報告書の例が含まれています。より具体性を与えるために、それらの例には、個人、企業、ブランド、あるいは製品などの名前が含まれている場合があります。これらの名前はすべて架空のものであり、類似する個人や企業が実在しているとしても、それは偶然にすぎません。

著作権使用許諾:

本書には、様々なオペレーティング・プラットフォームでのプログラミング手法を例示するサンプル・アプリケーション・プログラムがソース言語で掲載されています。お客様は、サンプル・プログラムが書かれているオペレーティング・プラットフォームのアプリケーション・プログラミング・インターフェースに準拠したアプリケーション・プログラムの開発、使用、販売、配布を目的として、いかなる形式においても、IBM に対価を支払うことなくこれを複製し、改変し、配布することができます。このサンプル・プログラムは、あらゆる条件下における完全なテストを経ていません。従って IBM は、これらのサンプル・プログラムについて信頼性、利便性もしくは機能性があることをほのめかしたり、保証することはできません。これらのサンプル・プログラムは特定物として現存するままの状態で提供されるものであり、いかなる保証も提供されません。 IBM は、お客様の当該サンプル・プログラムの使用から生ずるいかなる損害に対しても一切の責任を負いません。

プログラミング・インターフェース情報

CICS には、プログラミング・インターフェースと見なすことのできる資料と、プログラミング・インターフェースと見なすことのできない資料があります。

オンライン製品資料の以下のセクションには、CICS Transaction Server for z/OS, バージョン 5 リリース 5 のサービスを取得するプログラムをお客様が作成するためのプログラミング・インターフェースが含まれています。

- アプリケーションの開発
- Developing system programs
- 保護の概要
- 外部インターフェースに向けた開発
- リファレンス: アプリケーション開発h
- リファレンス: システム・プログラミング
- リファレンス: 接続

オンライン製品資料の以下のセクションには、CICS Transaction Server for z/OS, バージョン 5 リリース 5 のプログラミング・インターフェースとして意図されていない (プログラミング・インターフェースと誤解される可能性のある) 情報が含まれています。

- Troubleshooting and support
- リファレンス: 診断

PDF 形式のマニュアルで CICS 資料にアクセスする場合は、CICS Transaction Server for z/OS, バージョン 5 リリース 5 のサービスを取得するプログラムをお客様が作成するためのプログラミング・インターフェースが以下のマニュアルに含まれています。

- アプリケーション・プログラミング・ガイドおよびアプリケーション・プログラミング・リファレンス
- Business Transaction Services
- Customization Guide
- C++ OO Class Libraries
- Debugging Tools Interfaces Reference
- Distributed Transaction Programming Guide
- External Interfaces Guide
- Front End Programming Interface Guide
- IMS Database Control Guide
- インストール・ガイド
- セキュリティー・ガイド
- Supplied Transactions
- CICSPlex SM Managing Workloads
- CICSPlex SM Managing Resource Usage
- CICSPlex SM アプリケーション・プログラミング・ガイドおよび CICSPlex SM アプリケーション・プログラミング・リファレンス
- Java Applications in CICS

PDF 形式のマニュアルで CICS 資料にアクセスする場合は、CICS Transaction Server for z/OS, バージョン 5 リリース 5 のプログラミング・インターフェース

として意図されていない (プログラミング・インターフェースと誤解される可能性のある) 情報が以下のマニュアルに含まれています。

- Data Areas
- Diagnosis Reference
- Problem Determination Guide
- CICSPlex SM Problem Determination Guide

商標

IBM、IBM ロゴおよび ibm.com® は、世界の多くの国で登録された International Business Machines Corporation の商標または登録商標です。他の製品名およびサービス名等は、それぞれ IBM または各社の商標である場合があります。現時点での IBM の商標リストについては、 http://www.ibm.com/legal/copytrade.shtml をご覧ください。

Adobe、Adobe ロゴ、PostScript、PostScript ロゴは、Adobe Systems Incorporated の米国およびその他の国における登録商標または商標です。

インテル、Intel、Intel ロゴ、Intel Inside、Intel Inside ロゴ、Intel Centrino、Intel Centrino ロゴ、Celeron、Intel Xeon、Intel SpeedStep、Itanium、および Pentium は、Intel Corporation または子会社の米国およびその他の国における商標または登録商標です。

Java およびすべての Java 関連の商標およびロゴは Oracle やその関連会社の米国およびその他の国における商標または登録商標です。

Linux は、Linus Torvalds の米国およびその他の国における登録商標です。

Microsoft、Windows、Windows NT および Windows ロゴは、Microsoft Corporation の米国およびその他の国における商標です。

UNIX は The Open Group の米国およびその他の国における登録商標です。

製品資料に関するご使用条件

これらの資料は、以下のご使用条件に同意していただける場合に限りご使用いただけます。

適用範囲
: IBM Web サイトの「ご利用条件」に加えて、以下のご使用条件が適用されます。

個人使用
: これらの資料は、すべての著作権表示その他の所有権表示をしていただくことを条件に、非商業的な個人による使用目的に限り複製することができます。ただし、IBM の明示的な承諾をえずに、これらの資料またはその一部について、二次的著作物を作成したり、配布 (頒布、送信を含む) または表示 (上映を含む) することはできません。

商用使用
: これらの資料は、すべての著作権表示その他の所有権表示をしていただくことを条件に、お客様の企業内に限り、複製、配布、および表示することがで

きます。ただし、IBM の明示的な承諾をえずにこれらの資料の二次的著作物を作成したり、お客様の企業外で資料またはその一部を複製、配布、または表示することはできません。

権利　ここで明示的に許可されているもの以外に、資料や資料内に含まれる情報、データ、ソフトウェア、またはその他の知的所有権に対するいかなる許可、ライセンス、または権利を明示的にも黙示的にも付与するものではありません。

資料の使用が IBM の利益を損なうと判断された場合や、上記の条件が適切に守られていないと判断された場合、IBM はいつでも自らの判断により、ここで与えた許可を撤回できるものとさせていただきます。

お客様がこの情報をダウンロード、輸出、または再輸出する際には、米国のすべての輸出入関連法規を含む、すべての関連法規を遵守するものとします。

IBM これらの資料の内容についていかなる保証もしません。これらの資料は、特定物として現存するままの状態で提供され、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任なしで提供されます。

IBM オンラインでのプライバシー・ステートメント

サービス・ソリューションとしてのソフトウェアも含めた IBM ソフトウェア製品 (「ソフトウェア・オファリング」) では、製品の使用に関する情報の収集、エンド・ユーザーの使用感の向上、エンド・ユーザーとの対話またはその他の目的のために、Cookie はじめさまざまなテクノロジーを使用することがあります。多くの場合、ソフトウェア・オファリングにより個人情報が収集されることはありません。IBM の「ソフトウェア・オファリング」の一部には、個人情報を収集できる機能を持つものがあります。ご使用の「ソフトウェア・オファリング」が、これらの Cookie およびそれに類するテクノロジーを通じてお客様による個人情報の収集を可能にする場合、以下の具体的事項をご確認ください。

CICSPlex SM Web ユーザー・インターフェース **(**メイン・インターフェース**)** の場合**:**　このソフトウェア・オファリングは、展開される構成に応じて、セッション管理、認証、お客様の利便性の向上、または利用の追跡または機能上の目的のために、それぞれのお客様のユーザー名、およびその他の個人情報を、セッションごとの Cookie および持続的な Cookie を使用して収集する場合があります。これらの Cookie を無効にすることはできません。

CICSPlex SM Web ユーザー・インターフェース **(**データ・インターフェース**)** の場合**:**　このソフトウェア・オファリングは、展開される構成に応じて、セッション管理、認証、または利用の追跡または機能上の目的のために、それぞれのお客様のユーザー名またはその他の個人情報を、セッションごとの Cookie を使用して収集する場合があります。これらの Cookie を無効にすることはできません。

CICSPlex SM Web ユーザー・インターフェース **(**「**Hello World**」ページ**)** の場合**:**　このソフトウェア・オファリングは、展開される構成に応じて、個人情報を収集しないセッションごとの Cookie を使用する場合があります。これらの Cookie を無効にすることはできません。

CICS Explorer の場合**:**

このソフトウェア・オファリングは、展開される構成に応じて、セッション管理、お客様の利便性の向上、または利用の追跡または機能上の目的のために、それぞれのお客様のユーザー名、およびその他の個人情報を、セッションごとの設定および持続的な設定を使用して収集する場合があります。これらの設定を無効にすることはできませんが、ユーザー・パスワードの暗号化形式でのディスクへの保管は、サインオン中にチェック・ボックスにチェック・マークを付けることによるユーザーの明示的な操作によってのみ有効化することができます。

この「ソフトウェア・オファリング」が Cookie およびさまざまなテクノロジーを使用してエンド・ユーザーから個人を特定できる情報を収集する機能を提供する場合、 お客様は、このような情報を収集するにあたって適用される法律、ガイドライン等を遵守する必要があります。これには、エンドユーザーへの通知や同意の要求も含まれますがそれらには限られません。

このような目的での Cookie を含む様々なテクノロジーの使用の詳細については、『IBM オンラインでのプライバシー・ステートメント』(http://www.ibm.com/privacy/details/jp/ja/) の『クッキー、ウェブ・ビーコン、その他のテクノロジー』および「IBM Software Products and Software-as-a-Service Privacy Statement」(http://www.ibm.com/software/info/product-privacy) を参照してください。

索引

日本語, 数字, 英字, 特殊文字の順に配列されています。なお, 濁音と半濁音は清音と同等に扱われています。

[ア行]

アカウンティング 417
アクセス許可レベル 113, 128
アクセス制御
  HFS ファイル 125
  z/OS UNIX ファイル 124
アクセス・リスト
  作成のための PERMIT コマンド 34
  条件付き、トランザクション・プロファイルの 98
  UACC(READ) で回避 98
アクティビティー
  セキュリティー 523, 525
アクティブ化、RACF クラスの 35
アプリケーション
  セキュリティー 228, 246, 265
アプリケーション・プログラム・セキュリティー
  アクセス許可レベル 122
  リソース・クラスの定義 121
  MCICSPPT 一般リソース・クラス 121
  NCICSPPT 一般リソース・クラス 121
  QUERY SECURITY コマンド 8, 174
アルゴリズム 505, 507
  暗号化 371
暗号化 369
  公開鍵 369
  データ・セット 425
暗号化アルゴリズム 371
暗号スイート 371, 389
  どれを使用するかの識別 389
  CIPHERS 属性による制限 371, 389
  z/OS および CICS によってサポートされる 371
暗号スイート仕様ファイル 371
一時記憶 107, 122, 136
  アクセス許可レベル 123
  名前付きカウンター・サーバーへのアクセスの許可 56
  名前付きカウンター・プールへのアクセスの許可 55
  SCICSTST 一般リソース・クラス 122
一時記憶 (続き)
  TS サーバーへのアクセスを許可する 54
  TS プールへのアクセスを許可する 53
  UCICSTST 一般リソース・クラス 122
一時記憶域
  リソース・クラスの定義 122
一時データ
  アクセス許可レベル 111
  セキュリティーの考慮事項 110
  CICS 必須の一時データ・キュー・リソース定義 112
一時データのトリガー・レベル・トランザクション 157
一般リソース・クラス
  システム・リソースを保護するための 29
  ユーザー定義 182
  リソース保護のための 28
  ACICSPCT 115, 119
  APPCLU 29
    プロファイルの定義 277
  APPL 29
  BCICSPCT 115, 119
  CCICSCMD 171
  CICS リソースを保護するための 27
  CONSOLE 29
  CPSMOBJ 28
  CPSMXMP 28
  DIGTCERT 29
  FACILITY 30, 336
  FIELD 20, 29
  JCICSJCT 114
  JESSPOOL 29, 62
  KCICSJCT 114
  LOGSTRM 29
  MCICSPPT 121
  NCICSPPT 121
  OPERCMDS 29, 87
  PCICSPSB 128
  PROGRAM 30
  PROPCNTL 30, 61
  PTKTDATA 30
  QCICSPSB 128
  SCICSTST 122
  STARTED 30
  SUBSYSNM 30
  SURROGAT 30, 61
  TERMINAL 30
  UCICSTST 122
  VCICSCMD 171
一般リソース・クラス (続き)
  VTAMAPPL 30, 60
一般リソース・プロファイル 15
  最新表示 254
委任、RACF 管理責任の 13
インストール定義クラス 71
  トランザクションの例 71
  ファイルの例 71
  ユーザー定義リソースの例 182
  PSB の例 71
オープン中のバックアップ (BWO) (backup while open (BWO)) 53
オペレーター、端末
  サインオン時のデータ 90
  デフォルト・ユーザーのデータ 90
オペレーター、CICS 端末
  データの取得 90
  RACF に対する定義の例 92

[カ行]

開始されたトランザクションのセキュリティー 115, 119
開始ジョブ
  CICS 領域ユーザー ID の定義 42
開始済みトランザクションのセキュリティー 120
開始タスク
  および RACF ユーザー ID 16
  CICS プロシージャーの許可 42
開発
  アプリケーション
    Kerberos 539
外部 CICS インターフェース (EXCI)
  セキュリティー 429
  CALL インターフェース (CALL interface)
    DFHAPPL プロファイル定義 430
外部 CICS インターフェース (EXCI) および代理検査 158
外部セキュリティー・マネージャー (ESM) 207
  別の呼び出し
    インターフェースの概要 260
    RACROUTE マクロ 261
  RACF 使用 211
    プロファイルの作成 216, 228
    CICS セキュリティーの制御 251
外部呼び出しインターフェース 348
鍵リング 378
  ビルド 378

© Copyright IBM Corp. 1989, 2018

鍵リングの作成 378
拡張総称命名
  データ・セット・プロファイル名 25
  SETROPTS EGN コマンド 25
カスタムのセキュリティー・ハンドラー 517
カタログ式プロシージャー
  CICS を開始タスクとして許可する 42
各国語サポート 22, 90
カップリング・ファシリティー・データ・テーブル・セキュリティー 354
カップリング・ファシリティー・データ・テーブル・プール 355
監査
  許可要求で CICS によって要求される 109
  バインド・セキュリティー障害 279
  ログ・データを書き込むための RACF への 2 番目の要求 98
  SMF タイプ 80 ログ・レコード 98
管理 12
  セキュリティー 523
起動、Trust クライアント 519
機能シップ
  ミラー・トランザクション 293, 300, 346
  DEFINE TRANSACTION の RESSEC オペランド 291, 300, 344
基本認証 469, 471, 472
共用データ・テーブル
  サーバー許可セキュリティー検査 351
  セキュリティー検査 351
  バインド・セキュリティー 352
  ファイル・セキュリティー 353
  CONNECT セキュリティー検査 352
許可 ID、1 次 415
許可 ID、2 次 418
許可、RACF への CICS ユーザーの 91
許可障害
  エラー・メッセージ 98
  コマンド・セキュリティー 173
  CICS リソース 109
  ICH408I、RACF メッセージ 98
許可する、CICS 領域ユーザー ID を代理ユーザーとして 62
区画内一時データ・リソース 157
グッドナイト・トランザクション 78
クライアント証明書 469, 471
クライアント・プログラム
  MRO ログオン・セキュリティーおよびバインド時のセキュリティー 429
クラス、リソース 15
クラス、リソース・グループ 15
クラス記述子テーブル (CDT) (class descriptor table (CDT)) 181
グループ ID 72
グループ SPECIAL属性 13
グループ・クラス、リソース 15
グループ・プロファイル 15, 23
グローバル・セキュリティー・パラメーター 254
クロック・スキュー
  設定 550
  変更 550
言語セグメント
  システム・デフォルト 22
  ユーザー・プロファイル 22
検証 546
  署名 550
公開鍵暗号化 369
構成 546
構成、パイプラインの 514
構成ファイル
  Security Token Service 552
個別プロファイル 15
コマンド権限
  Db2 420
コマンド・セキュリティー 162
  許可障害 173
  指定 171
  対象となる CICS リソース 164
  CCICSCMD 一般リソース・クラス 171
  CEMT の考慮事項 173
  IPIC 接続用の 332, 333
  ISC over TCP/IP 接続に対する 333
  QUERY SECURITY コマンド 174
  VCICSCMD 一般リソース・クラス 171
  XCMD パラメーター 66, 106, 135
  XUSER パラメーター 85
コマンド・レベルのセキュリティー 527

［サ行］

サーバー・プログラム
  セキュリティーの考慮事項 429
サインオフ
  持続セッションの再始動後 18
  プロセス 77
  ロギング・アクティビティー 83
  XRF テークオーバー後 18
サインオフ・プロセス 77
サインオン
  持続セッションの再始動後 18
  失敗したサインオン、サンプル・フロー 311
  端末ユーザーのユーザー・データ 90
  ロギング・アクティビティー 83
  XRF テークオーバー後 18
サインオン定義の有効範囲 76
サインオン出口ルーチン 418
サインオン・セキュリティー 10, 74
サインオン・トランザクション・プログラム用の SNA サービス・トランザクション・プログラム名 318
サインオン・リクエスターのトランザクション
  許可されるユーザー ID とパスワードの長さ 315
  新規パスワード ID 315
  データが最大バッファー・サイズを超える 316
  ユーザー ID およびパスワードの EBCDIC 315
  ATTACH セキュリティー・フィールド 315
  PIP データ・オプション 316
  PROFILE オプション 315
  X'06F3F0F1'、トランザクション ID 315
サインオン・リクエスター・トランザクション
  同期レベル 0 315
  CICS PEM サーバーによって必要とされる入力データ 318
  SNA サービス・トランザクション・プログラム名 318
サンプルのサインオン出口ルーチン (DSN3SSGN) 418
時刻サブフィールド、フォーマット 320
システム SPECIAL 属性 13
システム管理機能 (SMF) (System Management Facility (SMF)) 12, 83
システム間連絡 (ISC) セキュリティー
  コーディング、ATTACHSEC の 284, 341
システム初期設定パラメーター、CICS 69
  リソース・セキュリティー 66
  CICS リソース名の接頭部の付加 64
  CMDSEC 65, 171
  DFLTUSER 65
  ESMEXITS 65
  PLTPISEC 65
  PLTPIUSR 65
  PSBCHK 65, 129, 174
  QUERY SECURITY による SEC 174
  RESSEC 66
  SEC 63
  SECPRFX 64
  SECPRFX と QUERY SECURITY 174
  SNSCOPE 66
  XAPPC 66, 69, 105, 134, 276
  XCMD 66, 106, 135, 171
  XDB2 66, 106, 135
  XDCT 66, 106, 110, 135
  XFCT 66, 106, 113, 135
  XHFS 66, 69, 106, 124, 125, 135

システム初期設定パラメーター、CICS (続き)
XJCT 67, 106, 114, 135
Xname パラメーター 174
XPCT 67, 106, 115, 119, 120, 135
XPPT 67, 106, 121, 135
XPSB 67, 107, 129, 136
XRES 67, 101, 107, 131, 136, 481
XTRAN 97
XTST 67, 107, 123, 136
XUSER 67, 69
システム・セキュリティー
CICS インストール要件 39
システム・データ・セット
アクセスの許可 49
アクセスのレベル 49
必要な総称プロファイル 50
保護 41
システム・ネットワーク体系 (SNA) セッション・セキュリティー 272
システム・リソース
保護 29
事前設定セキュリティー・セッション 87
事前設定端末 NATLANG 94
事前設定端末セキュリティー 7, 83, 155
自動インストール・モデル 87
その他の考慮事項 87
端末でのタスクの開始 117
端末に関連付けられていないトランザクション 99
端末ルーティング 292
定義およびインストールの制御 85
CEDA LOCK コマンド 85
CEDA トランザクション 85
CSD へのバッチ・アクセスの制限 85
MVS システム・コンソールを CICS 端末として使用する 87
SURROGAT トランザクション 85
持続検査 (PV)
サインオン元リスト 309
持続検査 (PV) (persistent verification (PV))
サインオン 313
サインオン先リスト 309
失敗したサインオン 314
失敗したサインオン、PV を使用した 314
正常なサインオン、サンプル・フロー 310
正常なサインオン・フロー 312
ATTACHSEC-PERSISTENT 310
CONNECTION 309
LU6.2 セキュリティーの実装時 283
持続セッション
XRFSOFF オペランド 18
持続セッションの再始動
サインオフ 18
サインオン状態を維持 18
自動インストール・モデル 87
ジャーナルとログ・ストリーム
ジャーナルのアクセス許可レベル 114
ジャーナル・セキュリティー
アクセス許可レベル 114, 119
リソース・クラスの定義 114
XJCT パラメーター 67, 106, 114, 135
条件付きアクセス処理 82
条件付きアクセス・リスト 98
証明書 361, 370
証明書、新規作成 381
証明書失効リスト 394
証明書を Untrusted としてマーク付けする 385
初期設定後の処理、代理セキュリティー 155
ジョブ実行依頼、代理 61
署名の検証
SAML 550
診断メッセージ 487
スキュー時間
設定 550
変更 550
ストレージ内のプロファイル
および XCMD リソース・クラス 171
最新表示 23
GTERMINL プロファイル 80
ストレージ内プロファイル
必要性の削減 28
スプール・ファイル、セキュリティー・トークン 62
正常なサインオン
失敗したサインオン 311
新規パスワード 323
正常なサインオン 310
正しいサインオン・データに対する応答 324
間違ったデータ・フォーマットに対する応答 326
PEM クライアントから CICS PEM サーバーへの 310
PV を使用した失敗したサインオン 314
PV を使用した正常なサインオン 312
生成と使用、RACF パスチケットの 10
セキュア・サインオン 435
セキュア・ハッシュ
暗号化アルゴリズム 371
セキュリティー 221, 429, 433, 469
アクティビティーの
接続時 525
定義済みアクティビティー・ユーザー ID 523
セキュリティー (続き)
アクティビティーの (続き)
リソース・レベル 523
アプリケーションが生成する応答 478
インバウンド・ポート 476
基本認証 469, 471, 472
コマンド・セキュリティー 401
サインオン 10, 74
識別 469, 471
代理ユーザー検査 406
端末以外の 99
認証 469, 471, 475
パスチケットの使用 10, 74
パスワード有効期限管理 472
パスワード・フレーズ有効期限管理 472
プロキシー認証 471
プロセスの
接続時 525
定義済みプロセス・ユーザー ID 523
リソース・レベル 523
文書テンプレート 475, 477
別名トランザクション 478
ポート番号
セキュリティー 476
リソース 100, 134
リソース・セキュリティー 402
リソース・レベル 475
AUTHTYPE 406
BTS コマンドの 527
CICS システム 477
CICSESM インターフェース 207
CKTI 529
COVA 221
COVC 221
COVE 221
COVG 221
COVP 221
COVU 221
DB2TRAN リソース・セキュリティー 403
DFHHTML 223
EYUCOVE 223
EYUCOVI 223
EYULOG 223
EYUWREP 223
EYUWUI 223
RACF 399
RACF クラス、DSNR 412
RACF プロファイルの定義 403
SAML 545
SASS 412
SSL 482
WUI リソースへのアクセス 225

セキュリティー (続き)
z/OS UNIX システム・サービス 475, 477
z/OS UNIX ファイル 475, 477
セキュリティー、代理 406
セキュリティー、DBCTL
CICS による PSB 許可検査 424
セキュリティー検査
項目の免除 251
パラメーター 253
評価シーケンス 255
別の ESM を使用 260
CICS の制御 251
ESM インターフェース 260
RACF による 211
セキュリティーの再作成 24, 280
セキュリティー分類、データおよびユーザーの 36
セキュリティー・イベントのロギング
許可要求で CICS によって要求される 109
サインオンおよびサインオフ・アクティビティー 83
QUERY SECURITY 185
SMF での RACF 監査メッセージ 109
セキュリティー・カテゴリー 36
セキュリティー・タスク、例 262
セキュリティー・トークン、JES スプール・ファイルの 62
セキュリティー・トークン・サービス構成ファイル 552
セキュリティー・パラメーターの活動化 253
セキュリティー・ハンドラー
作成、独自の 517
セキュリティー・プロファイル
最新表示 254
セキュリティー・プロファイル、RACF
作成 216
保護されたビュー 230
セキュリティー・プロファイルによって保護されたビュー 230
セキュリティー・ポリシーの適用 465
セキュリティー・ポリシーの有効化 465
セキュリティー・マネージャー
セキュリティー・ポリシーの適用 465
セキュリティー・ポリシーの有効化 465
セキュリティー・ラベル 36
セキュリティー・レベル 36
セグメント (segment)
端末ユーザーのデータ 19
CICS 17
LANGUAGE 22
RACF 16
セッション鍵 276
セッション・セキュリティー (session security) 276
セッション・セグメント 277
接続許可 411
接続セキュリティー 529
接続出口ルーチン 413
接続出口ルーチン (DSN3SATH) の例 413
接頭部の付加
SECPRFX による 64
ソース・ライブラリー、保護 40
総称プロファイル 15
SETROPTS GENERIC 14, 25, 38, 80
SETROPTS コマンド 35
総称リソース・プロファイル 130, 136

[タ行]

対称暗号化 369
代理権限の照会、ユーザーの 185
代理ジョブ実行依頼
JES 内部読み取りプログラムへの 61
代理セキュリティー検査 523
代理端末 (surrogate terminal) 292
代理ユーザー
CICS 領域ユーザー ID を許可する 62
代理ユーザー・セキュリティー 154
検査 154
初期設定後の処理 155
RACF 定義 159
RACF 定義の例 161
大/小文字混合
password 93
タスク、例
セキュリティー 262
タスクの例
セキュリティー 262
単一アドレス・スペース (SASS) 412
端末
エミュレーター 540
端末、個々のプロファイルの定義 80
端末、TCT に定義された 87
端末セキュリティー
アクセスの制御 79
サインオン 75
事前設定 7, 83
自動インストール・モデル 87
端末プロファイル 80
汎用アクセス権限 81
未定義端末 81
ユーザーの CICS 関連データの取得 90
ユーザーの識別 72
CEDA LOCK コマンド 85
MVS システム・コンソールを CICS 端末として使用する 87
端末セキュリティー (続き)
RACF に対するユーザーの定義の例 92
SETROPTS TERMINAL のオーバーライド 81
TCT 内の端末 87
user 7
XTRAN 67
端末ユーザー・セキュリティー 7
データ、デフォルト・ユーザーの 90
データ暗号化規格
暗号化アルゴリズム 371
データ・セット
暗号化 425
データ・セット・セキュリティー
アプリケーション ID パラメーター 50
ユーザー・データ・セットへのアクセス 52
CICS インストール要件 39
CICS システム 41
CICS データ・セットへのアクセス 49
MVS ライブラリー・ルックアサイド (LLA) 機能 52
データ・セット・プロファイル 15
拡張総称命名 25
SETROPTS EGN コマンド 25
データ・テーブル
カップリング・ファシリティー 354
サーバー許可セキュリティー検査 351
セキュリティー検査 351
バインド・セキュリティー 352
ファイル・セキュリティー 353
CONNECT セキュリティー検査 352
定義、保護
リソースへのアクセスの制御 228
定義済みアクティビティー・ユーザー ID 523
定義済みプロセス・ユーザー ID 523
出口
ESM、CICS 関連情報へのアクセス 208
RACF ユーザー出口パラメーター・リスト 208
デフォルト・ユーザー 155
デフォルト・ユーザー、CICS
定義 46
DFLTUSER パラメーター 65
SIT で指定する 65
デプロイ 545
伝搬の制御、ユーザー ID の 60
動的解析検証ルーチン 39
特定の接続
MRO ログオン・セキュリティー検査 429
ドメイン 487

トランザクション 218, 220
CMAS 内の 218
RACF に対する定義 218
Kerberos 540
MAS における 220
RACF に対する定義 220
トランザクションおよびリソースの保護 274
トランザクション開始 289, 300, 332, 344
トランザクション接続セキュリティー
制御する CICS パラメーター 94
SEC=YES および XTRAN=YES の場合の処理 97
トランザクションの開始
開始済みトランザクション 155
トランザクション・セキュリティー 274
アクセス許可レベル 119
開始済みトランザクション 94, 115, 119, 120
条件付きアクセス・リスト 98
端末なしで開始されるトランザクション 117
トランザクション接続セキュリティー 94
リソース定義 290, 300, 333, 344
CRTE 290, 292, 333, 346
IPIC 接続用の 332
ISC over TCP/IP 接続に対する 332
RACF へのプロファイルの定義 97
XJCT パラメーター 115, 119, 120
XPCT 検査されるトランザクション 115, 119
XPCT 検査トランザクション 120
XPCT パラメーター 67, 106, 135
XTRAN システム初期設定パラメーター 97
トランザクション・ルーティングおよび QUERY SECURITY 174
トリガー・レベル・トランザクション
セキュリティーの指定 99, 157
デフォルト・セキュリティー 48, 157
トレース・ポイント 487

[ナ行]

認証 469, 471, 472
認証、RPC 433
認証局 361, 370, 371
ヌル認証 433

[ハ行]

パイプライン
プロバイダー
SAML トークン用の構成 547
パイプライン (続き)
リクエスター
SAML 用の構成 549
バインド時のセキュリティー 429
バインド時のセキュリティー (bind-time security) 276, 298, 336
概要 272
IPCONN 用の 327
IPIC 接続用の 327
MRO リンク 336
パスチケット 435
パスチケット、サインオン・セキュリティー用 10, 74
パスワード
更新 304
8 文字 315
APPC PEM によって提供される情報 304
ESM ユーザー・プロファイル 320
パスワードおよびパスワード・フレーズ 76
パスワード有効期限管理プログラム DFHWBPW 469, 472
パスワード・フレーズ
大/小文字混合 93
パスワード・フレーズ (password phrase) 76
パターン
SAML プログラム 553, 556
バッチ呼び出しインターフェース 348
バッチ・アクセス、CSD への、制限 85
パラメーター
セキュリティー
活動化 253
グローバル 254
検査 253
汎用アクセス権限 98
汎用接続
セキュリティー検査を実行しない場合の注意事項 429
汎用リソース名 (z/OS Communications Server)
z/OS Communications Server 汎用リソース 58
非端末開始トランザクションのユーザー ID 117
非端末セキュリティー
端末に関連付けられていないトランザクション 8
日付サブフィールド、フォーマット 320
ビュー
セキュリティーの考慮事項 225
評価シーケンス、セキュリティー 255
ファイル、CICS によって処理される 112
ファイル・セキュリティー
アクセス許可レベル 113
ファイル・セキュリティー (続き)
総称データ・セット・プロファイル 25
データ・セット・プロファイル 24
リソース・クラスの定義 112
FCICSFCT 一般リソース・クラス 113
HCICSFCT 一般リソース・クラス 113
XFCT パラメーター 66, 106, 113, 135
ファイル・リソース・セキュリティー検査 357
付加時セキュリティー 525
複数のボリューム 53
複数領域操作 (MRO) 8
ログオン・セキュリティーおよびバインド時のセキュリティー 429
プラットフォーム
セキュリティー 228, 246, 265
フロー、サンプル 310
プロキシー認証 471
プログラム初期設定パラメーター (PIP) データ 316
プログラム・セキュリティー
XPPT パラメーター 67, 106, 121, 135
プログラム・プロパティー・テーブル (PPT)、MVS 40
プロセス
セキュリティー 523, 525
プロバイダー・パイプライン
構成
SAML トークン用 547
プロファイル 14
拡張総称命名 25
削除のための RDELETE コマンド 34
作成のための RDEFINE コマンド 33
主記憶域内のリフレッシュ 26
総称 35
総称データ・セット 25
端末 (PoE)、定義 79
データ・セット 24
定義
APPCLU 一般リソース・クラス 277
トランザクション、RACF への定義 97
トランザクションおよび条件付きアクセス・リスト 98
変更のための RALTER コマンド 34
リソース、総称の定義 130, 136
リソースおよび WARNING オプション 110
ACICSPCT 一般リソース・クラス 115, 119
BCICSPCT 一般リソース・クラス 115, 119
CCICSCMD 一般リソース・クラス 171, 182

プロファイル (続き)
CICS JOB ステートメントの USER パラメーター 44
DCICSDCT 一般リソース・クラス 110
ECICSDCT 一般リソース・クラス 110
FCICSFCT 一般リソース・クラス 113
GCICSTRN 一般リソース・クラス 64, 95, 116
HCICSFCT 一般リソース・クラス 113
JCICSJCT 一般リソース・クラス 114
JESSPOOL 62
KCICSJCT 一般リソース・クラス 114
MCICSPPT 一般リソース・クラス 121
NCICSPPT 一般リソース・クラス 121
PCICSPSB 一般リソース・クラス 128
PROPCNTL 61
QCICSPSB 一般リソース・クラス 128
RCICSRES リソース・クラス 101, 131
SCICSTST 一般リソース・クラス 122
SETROPTS EGN コマンド 25
SETROPTS コマンド 35, 79
SURROGAT 一般リソース・クラス 61, 159
TCICSTRN 一般リソース・クラス 64, 95, 116
UCICSTST 一般リソース・クラス 122
VCICSCMD 一般リソース・クラス 171
VTAMAPPL 60
WCICSRES グループ化クラス 101, 131
プロファイル、一時データ・キューの 111
プロファイル、一般リソース 15
プロファイル、グループ 15
プロファイル、個別 15
プロファイル、総称 15
プロファイル、データ・セット 15
プロファイル、ユーザー 2, 14
プロファイル、リソース・グループ 15
プロファイルの定義
APPCLU 一般リソース・クラス 277
分散プログラム・リンク (DPL)
LU6.2 での 295
MRO を使用する 348
文書テンプレート
セキュリティー 475, 477
文書テンプレート・セキュリティー 481
リソース・クラスの定義 101, 131
RCICSRES リソース・クラス 101, 131
WCICSRES グループ化クラス 101, 131
文書テンプレート・セキュリティー (続き)
XRES パラメーター 101, 107, 131, 136, 481
分類、データおよびユーザーの 36
保護
システム・リソース 29
リソース 28
CICS リソース 27

[マ行]

マップ
セキュリティーの考慮事項 225
ミラー・トランザクション
可用性 191
機能シップ 293, 300, 346
LU6.2 上での DPL の 295
MRO 上での DPL の 348
明示的サインオン 76
メッセージ
許可障害 98
DFHSNxxxx 83
ICH408I メッセージの宛先 98
ICH408I、RACF 98
メッセージ認証コード (MAC) 369
メッセージ・ダイジェスト
暗号化アルゴリズム 371
メッセージ・ハンドラー
起動、Trust クライアント 519
メンバー、グループ
除去する DELMEM オペランド 81
追加する ADDMEM オペランド 81
問題判別
許可障害のエラー・メッセージ 98
サインオン失敗 310
サインオン失敗の理由 310
サインオン要求フォーマット設定エラー 322
新規パスワード ID 315
データが最大バッファー・サイズを超える 316
同期レベル 315
トランザクション ID 315
パスワードが EBCDIC でない 315
間違ったデータ・フォーマットに対する応答 326
ユーザー IDが EBCDIC ではない 315
8 文字を超えるユーザー ID とパスワード 315
ATTACH セキュリティー・フィールド 315
CICS 領域のユーザー ID の判別 196
GDS FREE コマンドを受信 316
ICH408I、RACF メッセージ 98
PIP データ・オプション 316
問題判別 (続き)
PROFILE オプション 315

[ヤ行]

ユーザー ID
セキュリティー 475, 477, 478
セキュリティー・トークンとしての CICS 領域の 62
代理ジョブ実行依頼 61
デフォルト 65
デフォルト CICS ユーザー ID を追加するための ADDUSER 47
伝搬の制御 60
非端末開始トランザクション 117
CICS デフォルト・ユーザーの定義 46
CICS の定義 43
CRTE を使用したセキュリティー検査 290, 292, 333, 346
DFLTUSER パラメーター 65
ユーザー ID セキュリティー
CKTI 529
ユーザー ID、URIMAP リソース定義の 159
ユーザー、CICS 2
ユーザー定義 RACF クラスのアクティブ化 184
ユーザー定義クラス 182
ユーザー出口
ICHRTX00 MVS ルーター出口 196
RACF パラメーター・リスト 208
ユーザー・セキュリティー 283, 341, 431
トランザクション・ルーティング 291, 345
ユーザー・プロファイル 15
CICS デフォルト・ユーザー 21
IPIC 接続用の 330
ユーザー・データ
フォーマット 316
付加 FMH5 およびデータ 317
GDS LL の長さ 317
SFL1 と SFL2 の長さ 317
TP LL の長さ 317
ユーザー・プロファイル 2, 14, 320
最新表示 254
ESM を使用した 320
RACF 15

[ラ行]

ライブラリー、CICSPlex SM
保護、RACF 使用 211
ラベル、RACF セキュリティー 36

リクエスター・パイプライン
構成
SAML 用 549
リソース
保護 28
リソースおよびコマンドの検査の相互参照 137
リソース定義
トランザクション・セキュリティー 290, 300, 333, 344
ユーザー・セキュリティー、リンク定義での 284, 341
リソース・セキュリティー 291, 300, 344
LU6.2 (APPC) セッション・セキュリティー 279
SECURITYNAME オプション 279
リソース定義パラメーター
CMDSEC 171, 172
RESSEC 107, 129
リソース名、CICSPlex SM 228
リソース・アクセス管理機能 (RACF)
セキュリティー検査からの項目の免除 251
トランザクションの定義 218, 220
リソースへのアクセスの制御 228
リソース・アクセス制御機能 (RACF) 429
リソース・クラス 15
APPCLU
プロファイルの定義 277
リソース・クラス、CICSPlex SM
アクセスの制御 228
リソース・グループ
除去する DELMEM オペランド 81
リソース・グループ・クラス 15
GCPSMOBJ 28
GTERMINL 30
リソース・グループ・プロファイル 15
リソース・セキュリティー 100, 134, 291, 300, 344
アクセス許可レベル、ファイル 113
アクセス許可レベル、z/OS UNIX ファイル 128
アプリケーション・プログラム 121
一時記憶 122
一時データ・キュー 110
一般リソース・プロファイル 25
監査 109
許可障害 109
ジャーナルとログ・ストリーム 114
主記憶域内のプロファイルのリフレッシュ 26
総称プロファイルの定義 130, 136
独自のリソース・クラス名の定義 36
トランザクション・ルーティング 291, 345
リソース・セキュリティー (続き)
必要なアクセス・レベル 130
ファイル 112
プログラム仕様ブロック 128
プロファイルおよび WARNING オプション 110
文書テンプレート 101, 131, 481
リソース定義 291, 300, 344
ACICSPCT 一般リソース・クラス 115, 119
BCICSPCT 一般リソース・クラス 115, 119
CCICSCMD 一般リソース・クラス 182
CICS SIT パラメーター 66
CICS および RACF による一般検査 105, 134
CICS クラスのアクティブ化 26
DCICSDCT 一般リソース・クラス 110
ECICSDCT 一般リソース・クラス 110
FCICSFCT 一般リソース・クラス 113
FIELD 一般リソース・クラス 20
GCICSTRN 一般リソース・クラス 64, 95, 116
HCICSFCT 一般リソース・クラス 113
HFS ファイル 125
IPIC 接続用の 332, 333
ISC over TCP/IP 接続に対する 333
JCICSJCT 一般リソース・クラス 114
KCICSJCT 一般リソース・クラス 114
MCICSPPT 一般リソース・クラス 121
NCICSPPT 一般リソース・クラス 121
PCICSPSB 一般リソース・クラス 128
QCICSPSB 一般リソース・クラス 128
QUERY SECURITY RESCLASS 181
QUERY SECURITY コマンド 8, 174
RCICSRES リソース・クラス 101, 131
RESSEC システム初期設定パラメーター 108
RESSEC トランザクション・リソース・セキュリティー・パラメーター 107
SCICSTST 一般リソース・クラス 122
SMF への RACF 監査メッセージのロギング 109
TCICSTRN 一般リソース・クラス 64, 95, 116
TD キューのプロファイルの定義 111
UCICSTST 一般リソース・クラス 122
WCICSRES グループ化クラス 101, 131
XAPPC パラメーター 105, 134
XCMD パラメーター 106, 135
リソース・セキュリティー (続き)
XDB2 パラメーター 106, 135
XDCT パラメーター 106, 135
XFCT パラメーター 106, 113, 135
XHFS パラメーター 66, 106, 124, 125, 135
XJCT パラメーター 106, 114, 135
XPCT パラメーター 106, 115, 119, 120, 135
XPPT パラメーター 106, 121, 135
XPSB パラメーター 107, 129, 136
XRES パラメーター 67, 101, 107, 131, 136, 481
XTST パラメーター 107, 123, 136
z/OS UNIX ファイル 124
リソース・チェッカー (CICS ONC RPC)
作成 436
リソース・プロファイル
削除のための RDELETE コマンド 34
作成のための RDEFINE コマンド 33
変更のための RALTER コマンド 34
リソース・レベルのセキュリティー 523
リモート・オペレーター 283, 341
リモート・ユーザー 283, 341
リモート・ユーザーの検査 287
リモート・ユーザーの識別 286, 331, 342
リモート・ユーザー・サインオフ 286, 332, 342
リンク・セキュリティー 281, 298
概要 273
IPCONN 用の 328
ルーティング・トランザクション、CRTE 290, 292, 333, 346
例
代理セキュリティー検査 523
付加時セキュリティー 525
RACF コマンド 523, 525
レベル、RACF セキュリティー 36
ロード・ライブラリー、保護 40
ログオン・セキュリティー 429
ログ・ストリーム
アクセスの許可 48
LOGSTRM 一般リソース・クラス 48
ログ・レコード
SMF タイプ 80 98, 109

A

ACICSPCT 一般リソース・クラス 115, 119
ADDMEM オペランド 405
ADDUSER command
defining the userid for CICS to RACF 43
Advanced Encryption Standard
暗号化アルゴリズム 371

ALLOCATE_PIPE コマンド
セキュリティー検査の失敗 429
APPC PEM (パスワード有効期限管理)
許可されるユーザー ID とパスワードの長さ 315
サインオン状況 304
サインオン要求、フォーマット設定エラー 322
サンプル構成 304, 305
持続検査 (PV) での使用 309
処理の概要 308
パスワードに関する情報 304
バッファー・サイズ 316
ユーザー ID およびパスワードの EBCDIC 315
利点 304
APPC PEM (パスワード有効期限管理) 303
ATTACH セキュリティー・フィールド 315
CICS PEM サーバーに送信されるサインオン・データ 318
CICS PEM サーバーによって実行される処理 309
CICS アクティビティー 308
CICS から PEM クライアントへのデータ 319
PEM クライアントによって送信されるサインオン入力データ 318
PEM クライアントによって要求される処理 309
PEM クライアントのセットアップ 315
PROFILE オプション 315
PV を使用した失敗したサインオン 314
APPCLU 一般リソース・クラス 29
プロファイルの定義 277
APPL 一般リソース・クラス 29
CICS 領域へのアクセスの制御 58
Atom コレクション
セキュリティー 496
Atom フィード
セキュリティー 496
ATTACHSEC オペランド 284, 299
IDENTIFY パラメーター 285
LOCAL パラメーター 285
MIXIDPE パラメーター 286
PERSISTENT パラメーター 285
VERIFY パラメーター 285
ATTACHSEC 属性 341
ATTLS 487, 489
ATTLS 基本 487, 489
ATTLS 制御 487, 489
ATTLS 認識 487, 489
ATTLSAWARE 483
AT-TLS 483
Application Transparent Transport Layer Security 483, 487, 489
PROTOCOL(HTTP) 483, 487, 489
SSL(ATTLSAWARE) 483, 487, 489
SSL(NO) 483, 487, 489
AT-TLS 基本 483
AT-TLS 制御 483
AUTHENTICATE(AUTOREGISTER) 489
AUTHENTICATE(CERTIFICATE) 489
AUTHID
代理セキュリティー 159
AUTHTYPE 値、セキュリティー用
許可 ID 410
グループ ID 410
端末 ID 410
トランザクション ID 410
ユーザー ID 410
DB2CONN からのサイン ID 410
AUTHTYPE セキュリティー 406

B

BCICSPCT 一般リソース・クラス 115, 119
BINDSECURITY オプション 279
BMS コマンド 107
BUILD ATTACH コマンド 342
BWO (オープン中のバックアップ) 53

C

CCICSCMD 一般リソース・クラス 171, 182
CCRL
端末からの実行 396
START コマンドからの実行 396
CDT (クラス記述子テーブル) 181
インストール先定義クラスのセットアップ 37, 183
リソースの長さ 181
IBM 提供のデフォルト・クラス 37
CEDA LOCK コマンド 85
CEDA トランザクション 85
CEDF 129
CEDF トランザクション 129, 172
CEDX 129
CEDX トランザクション 129
CEMT、マスター端末トランザクションおよび CRTE 290, 292, 333, 346
コマンド・セキュリティーの考慮事項 173
システム・プログラミング・コマンド 162
CESL トランザクション 76
CESN トランザクション 76
CFDT サーバーの許可 355
CFRM ポリシー 355
CICS Business Transaction Services
管理
セキュリティー 523
セキュリティー
コマンド・レベル 527
接続時 525
リソース・レベル 523
CICS command security 162, 401
VCICSCMD 一般リソース・クラス 405
CICS JOB ステートメント、PASSWORD パラメーター 44
CICS JOB ステートメント、USER パラメーター 44
CICS SIT パラメーター
セキュリティー関連 253
CICS Web サポート
セキュリティー 469
CICS コマンドおよびリソース
RACF を使用したセキュリティー・プロファイルの作成 228
リソースへのアクセスの制御 228
CICS システム初期設定パラメーター
SEC 433
XCMD 434
XPPT 434
XUSER 434
CICS システム定義ファイル (CSD)、バッチ・アクセスの制限 85
CICS シミュレーション・セキュリティー 251
CICS セキュリティー 400
CICS セキュリティー、制御 251
CICS セグメント (CICS segment) 17
CICS ソース・ライブラリー、保護 40
CICS 提供トランザクションのセキュリティー 187
CICS 提供の RACF 動的解析検証ルーチン 39
CICS で使用されるユーザー ID 2
CICS ユーザー 2
CICS ユーザー再始動プログラム、PLTPI 99
CICS リソース
保護 27
CICS リソース・セキュリティー
XCICSDB2 一般リソース・クラス 403
CICS 領域
アクセス 58
セキュリティー・トークンとしてのユーザー ID 62
リモート 59

CICS 領域 (続き)
APPL クラス・プロファイルへのアクセス 59
CICS 領域のユーザー ID 41
開始ジョブの 42
CICS ロード・ライブラリー、保護 40
CICSPlex SM
許可
プロシージャー 218
ライブラリー 216
保護
別の ESM を使用 260
RACF による 211
リソース名 228
CICSPlex SM セキュリティー・プロファイル
作成 216
CICSPlex SM 定義、保護
SAF リソース・クラスの追加 228
CICSPlex SM リソース・クラス
アクセスの制御 228
CICS-RACF インターフェースのカスタマイズ
インストール・データ・パラメーター・リスト 209
CICS 領域のユーザー ID の判別 196
ESMEXITS パラメーター 65
RACF ユーザー出口パラメーター・リスト 208
CICS-RACF セキュリティー・インターフェース
インストール・データ・パラメーター・リスト 209
ESM 出口プログラムによる CICS 関連情報へのアクセス方法 208
RACF ユーザー出口パラメーター・リスト 208
System Authorization Facility (SAF) 207, 260
CIPHERS 属性
CORBASERVER リソース 371, 389
TCPIPSERVICE リソース 371, 389
URIMAP リソース 371, 389
CKBM セキュリティー 527
CKCN セキュリティー 527
CKDL セキュリティー 527
CKDP セキュリティー 527
CKQC トランザクション
セキュリティー 527
CKRS セキュリティー 527
CKRT セキュリティー 527
CKSD セキュリティー 527
CKSQ セキュリティー 527
CKTI トランザクション
セキュリティー 527
CLAUTH (クラス権限) 属性
インストール定義クラス 36, 184
ユーザー・プロファイル内の 14
CICS 関連の一般リソース・クラス内 13
ClientAuthType 483, 487
clock_skew 550
CLOUD.APPLICATION セキュリティー・プロファイル 246, 265
CLOUD.DEF セキュリティー・プロファイル 246, 265
CLOUD.PLATFORM セキュリティー・プロファイル 246, 265
CLS4 トランザクション
XTRANID X'06F3F0F1' 315
CMDSEC 434
CMDSEC システム初期設定パラメーター 171
CMDSEC、コマンド・セキュリティー・パラメーター 171
COMAUTHID
代理セキュリティー 159
CONSOLE 一般リソース・クラス 29, 82
説明 82
COVA 221
COVC 221
COVE 221
COVG 221
COVP 221
COVU 221
CPLT トランザクション 155
CPSMOBJ 一般リソース・クラス 28
CPSMXMP 一般リソース・クラス 28
CRL (証明書失効リスト) 394
CRTE、ルーティング・トランザクション 290, 292, 333, 346
CSCS 一時データ宛先 83
CSD (CICS システム定義ファイル)、バッチ・アクセスの制限 85
CSD 定義、ロック 86
CSMI (CICS 提供のミラー・トランザクション)
セキュリティー 431
リンク・ユーザー ID の許可 431

D

Db2 セキュリティー
アカウンティング 417
許可 ID の確立 415, 418
計画の実行許可 421
セキュリティー・メカニズム 401
1 次許可 ID 411
2 次許可 ID 411
Db2 の AUTHID パラメーターおよび COMAUTHID パラメーターに対するユーザー ID 159
DB2ENTRY リソース・クラス 32
DCICSDCT 一般リソース・クラス 110
プロファイルの定義 110
DEFINE CONNECTION
ATTACHSEC オペランド 284, 299
ATTACHSEC 属性 341
BINDSECURITY オペランド 279
SECURITYNAME オプション 279
DEFINE TRANSACTION
CMDSEC オペランド 333
RESSEC オペランド 291, 300, 333, 344
DEFINE TRANSACTION の CMDSEC オペランド 333
DEFINE TRANSACTION の RESSEC オペランド 291, 300, 333, 344
DELMEM オペランド 81
DES
暗号化アルゴリズム 371
DES (Data Encryption Standard) 369
DES 認証 433
DFHAPPL FACILITY クラス・プロファイル、定義 430
DFHEXCI 代理プロファイル 160
DFHHTML 223
DFHINSTL 代理プロファイル 160
DFHIRP (領域間通信プログラム)
実行されるセキュリティー検査 429
DFHSNxxxx メッセージ 83
DFHSTART 代理プロファイル 160
DFHWBPW、パスワード有効期限管理プログラム 469, 472
DFHXCIS 348
DFHXCOPT、EXCI オプション・テーブル 158
DFH$RACF 37, 71
DFLTUSER SIT パラメーター
セキュリティー環境の作成 216
DFLTUSER パラメーター
定義 21
ユーザー・データの取得 90
DFLTUSER、システム初期設定パラメーター 65
DIGTCERT 一般リソース・クラス 29
DSN3SATH の例 413
DSN3SSGN のサンプル 418

E

EBCDIC、PEM ユーザー ID およびパスワードの 315
ECICSDCT 一般リソース・クラス 110
プロファイルの定義 110

ENCRYPTION パラメーター
および SP800-131A 391
ESDS、VSAM へのアクセス 53
ESM インターフェース
概要 260
RACROUTE マクロ 261
ESM (外部セキュリティー・マネージャー) 207
サンプル構成 305
パスチケット 10, 74
別の呼び出し
インターフェースの概要 260
RACROUTE マクロ 261
ユーザー ID およびパスワードの EBCDIC 315
ユーザー・プロファイル 320
CICS から PEM クライアントへのサインオン・データ 319
RACF 使用 211
プロファイルの作成 216, 228
CICS セキュリティーの制御 251
ESMEXITS、システム初期設定パラメーター 65
EXCI セキュリティー 348
EXCI 呼び出しでパラメーターとして渡されるユーザー ID 158
EXEC CICS LINK コマンド
セキュリティー検査 430, 431
DFHAPPL プロファイル定義 430
EXEC CICS QUERY SECURITY 434
EXEC CICS SET TRQUEUE ATIUSERID 158
EXEC CICS START USERID 433
EXEC CICS VERIFY PASSWORD 434
EXEC CICS コマンド
QUERY SECURITY 8
QUERY SECURITY コマンド 174
EXTRACT CERTIFICATE コマンド 469
EYUCOVE 223
EYUCOVI 223
EYULOG 223
EYUWREP 223
EYUWUI 223

F

FACILITY 一般リソース・クラス 30
FACILITY クラス・プロファイル、定義 430
FCICSFCT 一般リソース・クラス 113
FEPIRESOURCE リソース名 162
FIELD 一般リソース・クラス 20, 29
FIPS 準拠 391
FMH (機能管理ヘッダー)
後に続くユーザー・データ 316
生成する 318
FMH (機能管理ヘッダー) (続き)
付加 FMH5 およびデータ 317
FMH5 付加ヘッダー 316

G

GCICSTRN 一般リソース・クラス 64, 95, 116
GCPSMOBJ リソース・グループ・クラス 28
GDS (汎用データ・ストリーム)
データを渡す変数 317
GDS LL の長さ 317
GRANT コマンド 420
GROUP special コマンド
SEARCH コマンドの警告 35
GTERMINAL 定義 80
GTERMINL リソース・グループ・クラス 30

H

handshakerole=serverwithclientauth 489
HCICSFCT 一般リソース・クラス 113
HFS ファイル・セキュリティー
XHFS パラメーター 106, 125, 135
HTTP クライアントとしての CICS
セキュリティー 471
SSL 482
HTTP クライアント要求
セキュリティー 471
HTTP サーバーとしての CICS
セキュリティー 469, 472, 475

I

IBM 提供のクラス
トランザクションの例 69, 70
ファイルの例 69, 70
PSB の例 69, 70
ICH408I、RACF メッセージ 98
ICHERCDE マクロ 14, 183
ICHRFR01 (RACF ルーター・テーブル) 183
ICHRFRTB マクロ 183
ICHRIN03 開始プロシージャー・テーブル 218
ICHRIN03、RACF 開始タスク・テーブル 42
ICHRRCDE (インストール定義のクラス記述子テーブル) 183
ICHRTX00、MVS ルーター出口 196
IDENTIFY パラメーター、ATTACHSEC オペランド 285
IPCONN
リンク・セキュリティー 328
IPIC
バインド時のセキュリティー (bind-time security) 327
IPIC セキュリティー
概要 8
CRTE 290, 292, 333, 346
IPIC 接続
コマンド・セキュリティー 332, 333
セキュリティーの実装方法 326
トランザクション・セキュリティー 332
ユーザー・セキュリティー 330
リソース・セキュリティー 332, 333
IRR.DIGTCERT.ADD プロファイル 383
ISC over SNA セキュリティー
APPC (LU6.2) セッション・セキュリティー (session security) 8
ISC over TCP/IP 接続
コマンド・セキュリティー 333
トランザクション・セキュリティー 332
リソース・セキュリティー 333

J

Java セキュリティー・マネージャー 465
java.security.policy 465
JCICSJCT 一般リソース・クラス 114
JES スプールの保護 62
JESSPOOL 一般リソース・クラス 29, 62

K

KCICSJCT 一般リソース・クラス 114
Kerberos 537
アプリケーションの開発 539
Web サービスの構成 538
KEYRING 483, 487

L

LDAP サーバー
構成 394
LOCAL パラメーター、ATTACHSEC オペランド 285
LOCK コマンド、CEDA 85
LOGSTRM 一般リソース・クラス 29
LU6.1 セキュリティー 298
LU6.1 リンク 298
LU6.2 (APPC) セッション・セキュリティー
概要 8, 276

LU6.2 (APPC) セッション・セキュリティー (続き)
XAPPC パラメーター 66, 69, 105, 134, 276
XDB2 パラメーター 106, 135
LU6.2 セキュリティー
CRTE 290, 292, 333, 346

M

MAC (メッセージ認証コード) 369
MCICSPPT 一般リソース・クラス 121
MD5
暗号化アルゴリズム 371
MINTLSLEVEL/ENCRYPTION 483, 487
MIXIDPE パラメーター、ATTACHSEC オペランド 286
MQCONN
セキュリティー 528
MQMONITOR
セキュリティー 528
MRO (複数領域操作) セキュリティー
CRTE 290, 292, 333, 346
MRO ログオンおよび接続 338
MVS
パスワードおよび RACF 許可検査 40
プログラム・プロパティー・テーブル (PPT) 40
ライブラリー・ルックアサイド (LLA) 機能 52

N

NATLANG および非端末トランザクション 94
NCICSPPT 一般リソース・クラス 121
NETNAME 端末定義 80
NIST 規格適合 391

O

OIDCARD (オペレーター ID カード) 7
OPCLASS 17
OPERCMDS 一般リソース・クラス 29
OPIDENT 17
OPPRTY 17
OSGi セキュリティー 465

P

parameter
CICS データ・セットの保護 50
CICS 領域へのアクセスの許可 58
password
大/小文字混合 93
PCICSPSB 一般リソース・クラス 128
PEM サーバー、CICS
返されるエラー状況 310
最大バッファー・サイズを超えたデータ 316
同期レベル 0 315
日時サブフィールドのフォーマット 320
ユーザー ID およびパスワードの EBCDIC 315
PROFILE オプション 315
PEM リクエスター
会話タイプ 315
定義 304
ユーザー・データのフォーマット 316
CICS によって返されるサインオン完了状況値 321
PERMIT コマンド 34, 184
WHEN オペランド 82
PERSISTENT パラメーター、ATTACHSEC オペランド 285
PIP (プログラム初期設定パラメーター) データ 316
PKCS (Public Key Cryptography Standard) 369
PLT
初期設定後の処理 155
PLT プログラム 99
PLTPI 99
PLTPISEC、システム初期設定パラメーター 65
PLTPIUSR システム初期設定パラメーター 65, 155
PLTSD 99
POSIT 番号
インストール先定義の一般リソース・クラス 26, 36, 37
PREFIX 属性定義 123
PRODCFT1 355
PROGRAM 一般リソース・クラス 30
PROPCNTL 一般リソース・クラス 30, 61
プロファイルの定義 61
PSB セキュリティー
アクセス許可レベル 129
リソース・クラスの定義 128
PCICSPSB 一般リソース・クラス 128
QCICSPSB 一般リソース・クラス 128
XPSB パラメーター 67, 107, 129, 136
PSBCHK パラメーター 129, 174
PSBCHK、システム初期設定パラメーター 65
pthreads 374
PTKTDATA 一般リソース・クラス 30
PVDELAY システム初期設定パラメーター 286

Q

QCICSPSB 一般リソース・クラス 128
QUERY SECURITY コマンド 8
およびリソース・クラス 174
コマンドの動作 174
説明 174
トランザクション・ルーティング 174
ユーザー定義リソースの指定 182
ロギング 185
RESCLASS 181
RESTYPE 176
RESTYPE、返される値 180
SEC パラメーターの効果 174
SECPRFX パラメーターの効果 174

R

RACDCERT コマンド 378, 383
RACF 399
外部セキュリティー・マネージャー 399
コマンドの例 523, 525
プロファイル 14
RACF definitions for surrogate user checking 159
RACF SPECIAL 権限 383
RACF クラス
DSNR 412
「RACF グループ・リスト」オプション 419
RACF コマンド
ADDGROUP、例 24
ADDUSER、デフォルト CICS ユーザー ID の例 47
ALTUSER コマンドの例 14
CONNECT コマンド (グループ SPECIAL) の例 14
CONNECT、例 24
DELMEM オペランド 81
PERMIT 34
RALTER 34, 81
RDEFINE 33
RDELETE 34
REMOVE、例 24
SEARCH 警告 35
SETROPTS 25, 35
RACF セキュア・サインオン 435
RACF に対する定義
グループ 91
ユーザー 91
ユーザー、例 92
RACF の構成 509

RACF のデフォルト・リソース・プロファイル
VCICSCMD 一般リソース・クラス 405
RACF パスチケット 10
RACF プロファイル
最新表示 254
RACF プロファイルのリフレッシュ 254
RACF (リソース・アクセス管理機能)
一般リソース・プロファイル 25
エントリー・ポート・プロファイルの定義 79
管理 12
クラス記述子テーブル、ICHRRCDE 183
グループ・プロファイル 23
言語セグメント 22
セキュリティー検査からの項目の免除 251
セキュリティー・レベル 36
総称データ・セット・プロファイル 25
総称リソース・プロファイル 130, 136
端末プロファイル 80
データ・セット・プロファイル 24
デフォルトの CICS ユーザー ID の定義 46
独自のリソース・クラス名の定義 36
トランザクションの定義 218, 220
複数の MVS イメージがある 39
未定義端末 81
ユーザー・プロファイル 15
リソースへのアクセスの制御 228
ルーター・テーブル、ICHRFR01 183
CICS インストール要件 39
CICS クラスのアクティブ化 26
CICS セグメント (CICS segment) 17
CICS デフォルト・ユーザー 21
CICS ユーザーの許可 91
FIELD 一般リソース・クラス 20
RACF セグメント 16
SETROPTS TERMINAL のオーバーライド 81
RACF (リソース・アクセス制御機能)
主記憶域内のリソース・プロファイルのリフレッシュ 26
セキュリティー・ラベル 36
RACFVARS プロファイル 161
RACLIST 184
RACROUTE マクロ、セキュリティー用の 261
RALTER コマンド 34
RC2
暗号化アルゴリズム 371
RC4
暗号化アルゴリズム 371
RCICSRES 37
RCICSRES リソース・クラス 101, 131
RDELETE コマンド 34
RDO
トランザクションの使用制限 85
RESSEC 434
RESSEC リソース・セキュリティー・パラメーター 107
RESSEC、システム初期設定パラメーター 66
Rivest
暗号化アルゴリズム 371
RRCDTE サンプル・ジョブ 403

S

SAF (system authorization facility)
および MVS ルーター 207, 260
RACF への要求のルーティング 1
SAML
アプリケーションの開発 555
CICS の構成 545
SAML 対応プログラム 553, 555, 556
アウトバウンド要求 556
トークンの拡張 557
SAML トークン
期限切れ 550
再署名 550
署名 550
変更 550
まだ有効でない 550
SASS (単一アドレス・スペース) 412
SCICSTST 一般リソース・クラス 122
SEC システム初期設定パラメーター 433
SECPRFX、システム初期設定パラメーター 64
Security Token Extensions 546
プロバイダー・パイプラインの構成 547
リクエスター・パイプラインの構成 549
Security Token Service
Trust クライアント・インターフェース 504
SECURITYPREFIXID 356
SEC、システム初期設定パラメーター 63
SETROPTS GENERICOWNER コマンド 14
SETROPTS コマンド 25, 35
総称端末プロファイル 80
総称データ・セット・プロファイル 25
総称ユーザー・プロファイル 38
CLASSACT オプション 184
GENERIC オプション 184
RACLIST オプション 184
REFRESH オプション 184
SHA
暗号化アルゴリズム 371
SIOCTTLSCTL ioctl 483, 487
SIT パラメーター、CICS
セキュリティー関連 253
SMF (システム管理機能) 12, 83
SNA LU
SNA ACB アクセス 60
SNSCOPE サインオン・オペランド 66
SOAP メッセージ
暗号化 507
署名 505
SP800-131A 391
SPOOLOPEN コマンド 62
SQL
静的 422
動的 422
SSL 482
クライアント証明書認証 469, 471
HTTP クライアントとしての CICS の場合 482
SSL 接続の改善 374
SSL プール 374
SSL(ATTLSAWARE) 489
STARTED 一般リソース・クラス 30
STS 構成ファイル 552
sts-config.xml 552
sts.xml 552
sts.xsd 552
SUBSYSNM 一般リソース・クラス 30
SURROGAT 一般リソース・クラス 30, 61, 85, 159
SURROGAT トランザクション 85
SURROGCHK パラメーター 158
SYS1.PARMLIB ライブラリーの許可データ・セット 216

T

TCICSTRN 一般リソース・クラス 64, 95, 116
TCPIPSERVICE 489
TCPIPSERVICE リソース定義
セキュリティー 476
SSL
URIMAP リソース定義 476
TCP/IP 接続
バインド時のセキュリティー (bind-time security) 327
TERMINAL 一般リソース・クラス 30
TERMINAL 定義 80
TIMEOUT 17
TLS12 391
Triple DES
暗号化アルゴリズム 371

Trust クライアント
インターフェース 504
起動 519
TSO コマンド
TSO コマンドおよびセキュリティー処理 184
TSO コマンドを使用したリフレッシュ 26

U

UACC 98
UCICSTST 一般リソース・クラス 122
UNIX 認証 433
Untrusted 証明書、マーク付けする 385
URIMAP 定義
代理セキュリティー 159
URP_DISASTER 応答
リソース・チェッカーに対する 438
URP_EXCEPTION 応答
リソース・チェッカーに対する 438
URP_INVALID 応答
リソース・チェッカーに対する 438
URP_OK 応答
リソース・チェッカーに対する 438
USER パラメーター、CICS JOB ステートメントの 44
USRDELAY、システム初期設定パラメーター 342

V

validate 546
VCICSCMD 一般リソース・クラス 171, 405
VERIFY パラメーター、ATTACHSEC オペランド 285
VSAM ESDS へのアクセス 53
VSAM データ・セットおよび BWO 53
VSAM データ・セットおよび動的ボリューム・カウント 53
VTAMAPPL 一般リソース・クラス 30, 60
プロファイルの定義 60

W

WARNING オプション 110
WCICSRES 37
WCICSRES グループ化クラス 101, 131
Web Services Security (WSS) 497, 509, 514
Web サービス
構成
Kerberos 538
Web サービスのためのセキュリティー 497
WHEN オペランド、PERMIT の WHEN オペランド 82

X

XAPPC、システム初期設定パラメーター 66, 69, 105, 134, 276
XCICSDB2 一般リソース・クラス 403
XCICSDB2 メンバー・クラス 403
XCMD システム初期設定パラメーター 434
XCMD、システム初期設定パラメーター 66, 106, 135, 171
XDB2、システム初期設定パラメーター 66, 106, 135
XDCT、システム初期設定パラメーター 66, 106, 110, 135
トリガーされるトランザクションについての考慮事項 112
XFCT、システム初期設定パラメーター 66, 106, 113, 135
XHFS、システム初期化パラメーター 66, 69
XHFS、システム初期設定パラメーター 106, 124, 125, 135
XJCT、システム初期設定パラメーター 67, 106, 114, 135
XPCT 検査されるトランザクションのセキュリティー 115, 119
XPCT 検査トランザクションのセキュリティー 120
XPCT、システム初期設定パラメーター 67, 106, 115, 119, 120, 135
XPPT システム初期設定パラメーター 434
XPPT、システム初期設定パラメーター 67, 106, 121, 135
XPSB、システム初期設定パラメーター 67, 107, 129, 136
XRES、システム初期設定パラメーター 67, 101, 107, 131, 136, 481
XRF (拡張回復機能)
テークオーバー後もサインオン状態を維持 18
XRFSOFF オペランド 18
XRF (拡張リカバリー機能)
テークオーバー後のサインオフ 18
FORCE オペランド 18
NOFORCE オペランド 18
XTRAN、システム初期設定パラメーター 67
XTST、システム初期設定パラメーター 67, 107, 123, 136
XUSER システム初期設定パラメーター 434
XUSER、システム初期設定パラメーター 67, 69, 85
システム初期設定パラメーター、CICS XUSER 107, 136
リソース・セキュリティー
XUSER パラメーター 107, 136

Z

ZCICSDB2 グループ化クラス 403
z/OS Communications Server 60
汎用リソース名 58
z/OS Connect
セキュリティー 520
z/OS UNIX システム・サービス
セキュリティー 475, 477
z/OS UNIX ファイル
セキュリティー 475, 477
z/OS UNIX ファイル・セキュリティー
アクセス許可レベル 128
XHFS パラメーター 124

［特殊文字］

* 25
** (二重アスタリスク)
データ・セット・プロファイル名の 25
% 25

日本アイ・ビー・エム株式会社
〒103-8510 東京都中央区日本橋箱崎町19-21