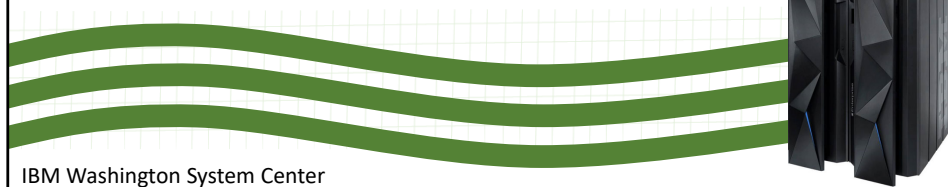


Securing and Encrypting Network Traffic with z/OS Communications Server and Policy Agent

Security Workshop

Protecting Traffic with IPsec



IBM Washington System Center
IBM Technical Sales Support

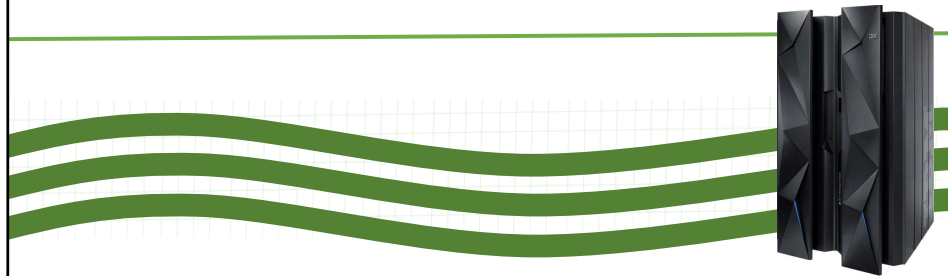
Trademarks

- **The following are Registered Trademarks of the International Business Machines Corporation in the United States and/or other countries.**
 - IBM
 - z/OS
- **The following are trademarks or registered trademarks of other companies.**
 - Microsoft is a registered trademark of Microsoft Corporation in the United States and other countries.
- All other products may be trademarks or registered trademarks of their respective companies.
- Refer to www.ibm.com/legal for further legal information.
- OSA-Express Features
- There are many different types of OSA-Express features. In this document where OSA is mentioned it refers to an OSA port on any of the OSA-Express features unless a specific OSA-Express feature is mentioned.

Agenda

- Overview of IPsec Protocol
- IPsec Tunnels
- Asymmetric and Symmetric Encryption
- Transport Mode Versus Tunnel Mode
- z/OS IPsec Implementation
- Network Security Services
- Sysplex Wide Security Associations

Overview of IPsec Protocol

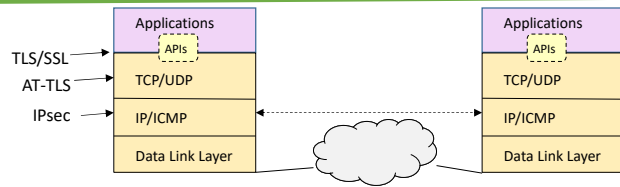


010_ZCS301_IPsec

© Copyright IBM Corporation 2023

Page 4

IPsec Protocol Overview



- Open standard network layer security protocol defined by IETF in RFCs
 - Provides authentication, integrity, and data privacy
 - RFC 5996
- IPsec security protocols
 - Authentication Header (AH) - provides authentication / integrity
 - Encapsulating Security Protocol (ESP) - provides data privacy with optional authentication/integrity
- Implemented at IP layer
 - Requires no application change
 - Secures traffic between any two IP resources
 - Security Associations (SA)
- Management of crypto keys and security associations can be
 - Manual
 - Automated via key management protocol (IKE)

010_ZCS301_IPsec

© Copyright IBM Corporation 2023

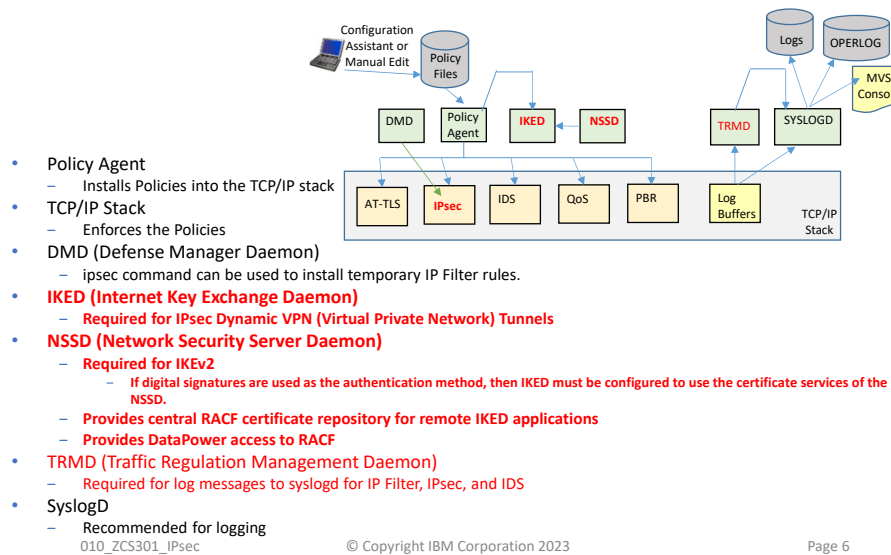
Page 5

IPsec protocol supports all upper level protocols.

- TCP and UDP

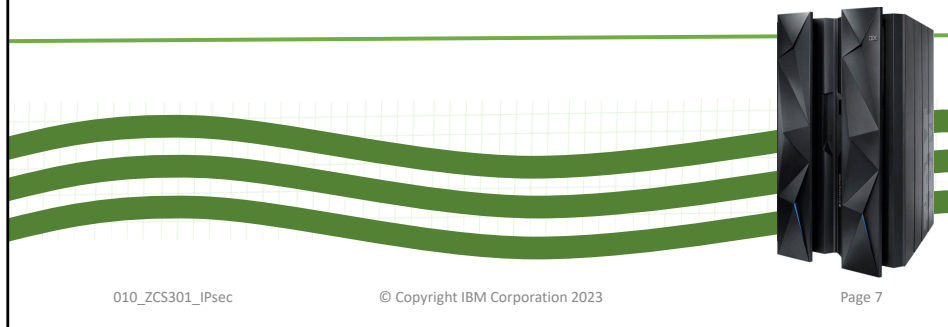
A Security Association (SA) includes the security algorithms decided upon in the negotiation between both sides.

Lots of Different Policy Types and Started Tasks



Configuration Assistant can really help getting the whole environment setup.

IPsec Tunnels



010_ZCS301_IPsec

© Copyright IBM Corporation 2023

Page 7

IPsec Header

IP Header	IPsec Header (ESP) and/or (AH)	TCP Header	Data
-----------	---	------------	------

- Data (original IP packets) passing through a tunnel can be:
 - authenticated (AH)
 - encrypted (ESP; authentication is optional)
 - Commonly used alone since it can perform both functions (encrypt + authenticate)
 - both: encrypted and then authenticated (ESP and AH)
- AH Protocol:
 - Protocol #51
 - Defined in RFC 2402 (supersedes RFC 1826)
 - Provides integrity and authentication
 - Includes selected fields of the IP header
 - Requires authentication algorithms:
 - HMAC-MD5-96 (RFC 2403)
 - HMAC-SHA-1-96 (RFC 2404)
 - Provides optional replay protection
 - May be used in combination with ESP
- Encapsulating Security Protocol
 - Protocol #50
 - Defined in RFC 2406 (supersedes RFC 1827)
 - Provides integrity, authentication, and encryption
 - Does not include fields of the IP header
 - Required authentication algorithms:
 - HMAC-MD5-96
 - HMAC-SHA-1-96
 - Null Authentication (i.e. none)
 - Required encryption algorithms:
 - DES_CBC (RFC 2405)
 - NULL (RFC 2410)
 - 3DES (RFC 2451)
 - Provides optional replay protection
 - May be used in combination with AH

010_ZCS301_IPsec

© Copyright IBM Corporation 2023

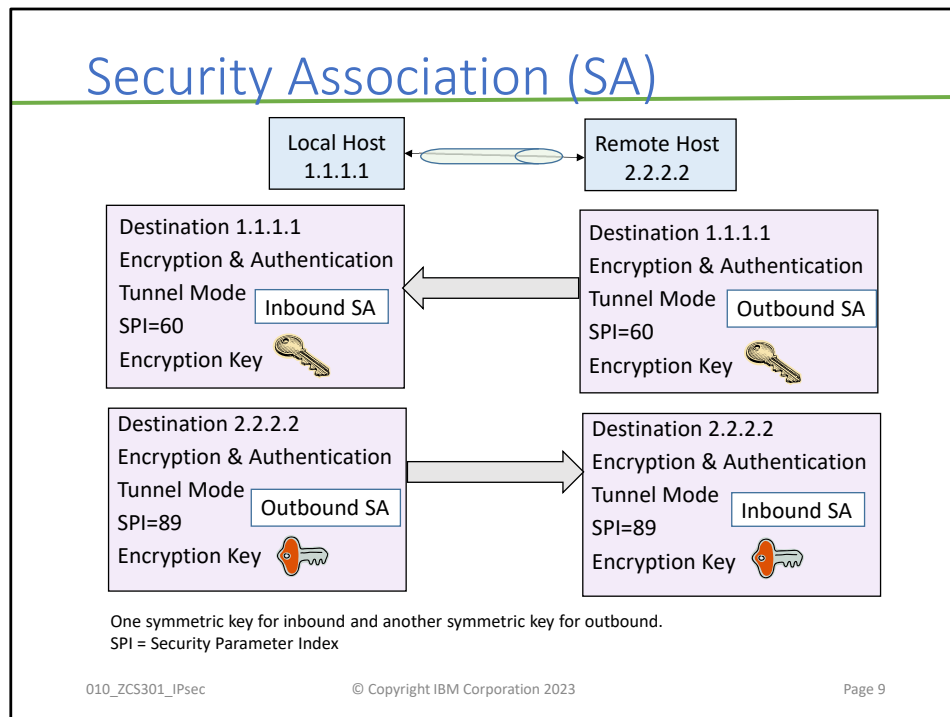
Page 8

The tunnel policy determines the level of protection and should be based on the customer's security requirements.

Authentication Header (AH Protocol -- Protocol #51):

- This protocol adds integrity to an IP packet by adding an Integrity Check Value (ICV).
- An ICV is a cryptographic checksum. Part of an IPSec SA contains the information needed to create and validate the ICV. The inclusion of the ICV makes it computationally infeasible for someone to modify the IPSec packet.
- The source address contained in the IP header is included as part of the ICV calculation. This means that the origin IP address can be authenticated.
- Since the ICV is cryptographically generated the ability to properly process the ICV provides another method to authenticate the data origin (data had to come from someone that processed the proper cryptographic key.)
- A sequence number is included in the AH header and the AH header is also part of the ICV calculation. The sequence number provides a mechanism to detect replays. The sender of an IPSec packet is required to send this field, but the receiver is not required to process it. To that extent the replay protection is considered optional.
- HMAC-MD5 computes the checksum by combining a 128 bit key, the Hash-based Message Authentication Code (HMAC) authentication algorithm and the MD5 hash algorithm
- HMAC-SHA computes the checksum by combining a 160 bit key, the HMAC authentication algorithm and the Secure Hash Algorithm (SHA).

It is quite common to use only the ESP header (protocol #50), since it is capable of performing both encryption and authentication. In this case, the authentication header is unnecessary.



An SA tells who is on the tunnel end points and what security services they will use: confidentiality, authentication and/or data integrity. The SA also states which cryptography and protocols the transactions will use.

Information contained in the SA includes;

- IP addresses of the communicating parties
- Algorithms to be used for authentication and/or encryption (cryptographic algorithms, hashing algorithms, MAC, etc.)
- Authentication and encryption keys and the key lifetimes
- A unique identifier known as the Security Parameter Index (SPI)
- Encapsulation mode (tunnel or transport)
- IPsec SAs are unidirectional - need two (inbound, outbound) to support bidirectional traffic

An IPsec SA identifies the information needed by one device to construct an IPsec protected packet and the information needed by the other device to deconstruct the IPsec protected packet.

IPsec SAs are unidirectional, meaning that if two devices are going to send and receive IPsec packets, then 2 SAs are needed (inbound and outbound).

Initiator's view of the Security Associations::

- 2 SAs, each with its own - SPI
- encryption key
- authentication key

Responder's view:

- 2 SAs, each with its own - SPI
- encryption key
- authentication key

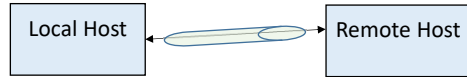
Each IPsec SA is specific to the security service (protocol) it provides.

- If only authentication is being applied to an IP packet, then each side must maintain an inbound and an outbound Security Association; both of these are for Authentication.
- If only encryption is being applied to an IP packet, then each side must maintain an inbound and outbound Security Association; both of these are for Encryption.
- If using ESP protocol performing both authentication and encryption, then you need an inbound and an outbound SA on each end. These SAs perform both authentication and encryption. This is the most commonly used model for IPsec.

But the story about SAs can get more complicated:

- If using protocol AH and protocol ESP together, then each protocol needs an inbound and an outbound, for 4 SAs on each end.

Manual versus Dynamic Tunnels



- Two different types of IPsec Virtual Private Network (VPN) Tunnels
 - Manually
 - Dynamically using the Internet Key Exchange (IKE) protocol
- Manual Tunnels
 - All attributes are manually configured
 - Attributes of the Security Associations must match
- Dynamic Tunnels
 - IKE protocol is implemented by IKE Daemon (IKED)
 - IKE securely negotiates IPsec SAs
 - Uses a special "IKE SA" to protect the IKE protocol messages
 - Much more flexible and scalable than manual tunnels, but requires more infrastructure
 - Additional security definitions for authentication and encryption

Manual (Static) Tunnel



- **Manual VPN Flows:**

- IPsec Manual VPNs use Symmetric Cryptography.
 - There is no partner key exchange or verification at VPN setup time.
 - All partner information - including a shared secret key for authentication -- is manually configured ahead of time and shared without threat of compromise
- Security Associations:
 - Security Association for ESP, including Authentication and Encryption
 - Most commonly used
 - Authentication Security Associations for Inbound & Outbound
 - Encryption Security Associations for Inbound & Outbound
- All right for testing, but
 - Open to intrusion and security threats

010_ZCS301_IPsec

© Copyright IBM Corporation 2023

Page 11

Manual IPsec VPNs:

- An IPsec tunnel whose security parameters and encryption keys are statically configured must be manually managed by a security administrator.
 - The two administrators negotiate or agree upon the values that will be used to form the encryption and authentication keys and also the algorithms to be used. This is sometimes called "out-of-band" negotiation since it does not take place over a networking connection.
- TCP/IP stack connects to partner implementing predefined SA information.
- Predefined / shared secret keys are created identically on both sides of the connection.
- Since there is a security association per IPsec protocol, you might find, per direction, separate SAs for Authentication and for Encryption or you might find, per direction, a single one for both Authentication and Encryption, as would be the case with the ESP protocol.
- In manual tunnels:
 - The key values are static, remain the same for the life of the tunnel and must be manually updated
 - Only one endpoint can automatically generate a key -- automatically generated Keys need to agree on both endpoints
 - You have the widest choice of header and encryption options
 - You must have mutual agreement between tunnel endpoint administrators
 - Manual VPNs must be shutdown to refresh keys

Sample Manual Tunnel Definition

```

IpManVpnAction      IPSecGoldStatic~0
{
  Active                               Yes
  LocalSecurityEndpointAddr            Any4
  RemoteSecurityEndpointAddr Any4
  HowToAuth                           ESP Hmac_Sha
  HowToEncrypt                         3DES
  HowToEncap                           Transport
  AuthOutboundSa                       300 0x0123456789ABCDEF0123456789ABCDEF01234567
  AuthInboundSa                       301 0x9876543219876543219876543219876
  EncryptOutboundSa                   300 0x0123456789ABCDEF0123456789ABCDEF0123456789ABCDEF
  EncryptInboundSa                    301 0x987654321987654321987654321987654321987
}
## Connectivity Rule  MyStaticVPNTweenOSAs combines the following items:
## Local data endpoint All4
## Remote data endpoint All4
## Topology            Host to Host
## Requirement Map     LPAR1toOtherLPARs
## EE                  => IPSecGoldStatic

IpFilterRule          MyStaticVPNTweenOSAs~1
{
  IpSourceAddr         All4
  IpDestAddr           All4
  IpServiceRef         EE
  IpGenericFilterActionRef IpSec~LogYes
  IpManVpnActionRef     IPSecGoldStatic~0
}
010_ZCS301_IPsec

```

© Copyright IBM Corporation 2023

Page 12

This is extracted from a Manual Tunnel Definition for Enterprise Extender and all its ports.

The keys for authentication and encryption are static and cannot be refreshed dynamically.

We recommend that you use Dynamic Tunnels to protect EE traffic. If you cannot use dynamic tunnels with RSA Signature Mode (using x.509 certificates), then deploy dynamic tunnels with Preshared Key.

- Preshared key is different from the static key model of Manual tunnels.
- The keys in manual tunnels must be provided to the peers so that the peer knows how to decrypt the data or to perform the authentication..
- The preshared key of dynamic tunnels is only a "seed value" used to compute symmetric keys .

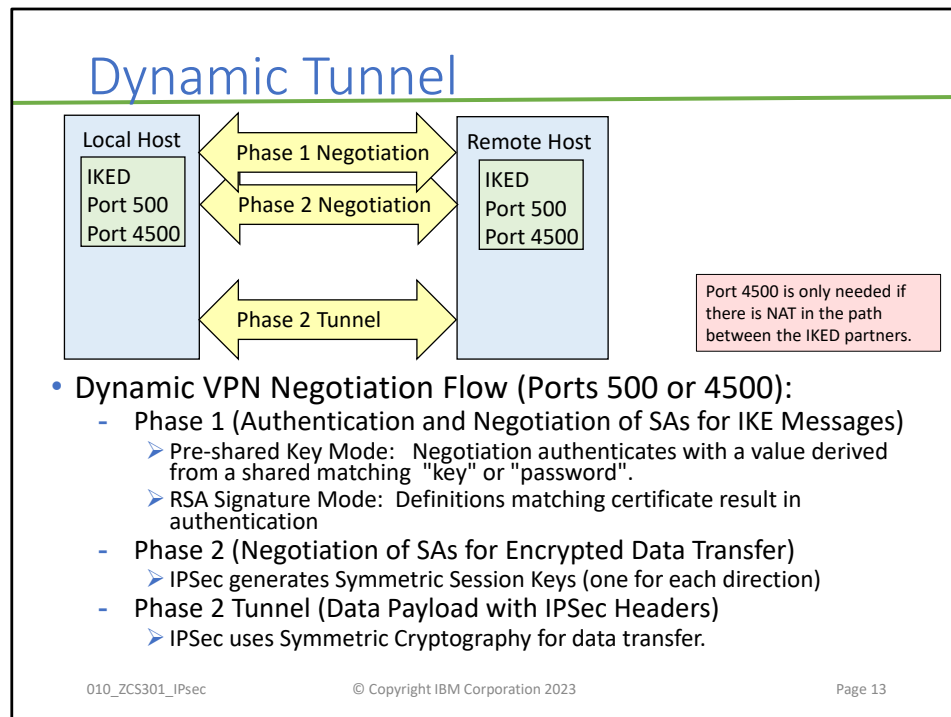
AuthOutboundSa

- Specifies the SA parameters for authentication traffic transmitted outbound to the remote security endpoint.
- spi Specifies the remote Security Parameter Index. Valid values for n are in the range 1 - 4 294 967 294. The set of SPI values in the range 1 - 255 are reserved to the Internet Assigned Numbers Authority (IANA) for future use.
- key Specifies the authentication key. The key must be specified in hexadecimal prefixed with '0x'. Each byte of the key represents a value in the range 00 - FF. The length of the key is determined by the associated algorithm. The key length (in bytes) for each algorithm type are: HMAC_MD5 (16), HMAC_SHA (20).

EncryptOutboundSa

- Specifies the SA parameters for encryption traffic transmitted outbound to the remote security endpoint.
- spi Specifies the remote Security Parameter Index. Valid values for n are in the range 1 - 4 294 967 294. The set of SPI values in the range 1 - 255 are reserved to the Internet Assigned Numbers Authority (IANA) for future use.
- key Specifies the encryption key. The key must be specified in hexadecimal prefixed with '0x'. Each byte of the key represents a value 00-FF. The length of the key is determined by the associated algorithm. The key length (in bytes) for each algorithm type are: DES (8), 3DES_CBC (24), and AES_CBC (16).

If ESP authentication is being used with encryption, the SPI values on the EncryptInboundSa and AuthInboundSa parameters must be the same value. Also, the SPI values on EncryptOutboundSa and AuthOutboundSa parameters must be the same value.



Dynamic IPsec VPNs:

- IKED performs IKE protocol negotiations.
- TCP/IP stack connects to partner implementing negotiated SA information.
- IKED also refreshes keys dynamically.

The authentication mode is different for each type of Dynamic Tunnel. Each of these authentication methods provides a way for hosts to verify the identity of the other, and while the pre-shared key mechanism is easier to configure, the RSA signature method is more versatile, secure, and scalable.

Pre-Shared Key: A matching "identifier" at both ends of the connection authenticates the peer during Phase I negotiations.

- Both sides agree on an arbitrary secret value called the pre-shared key.
- The pre-shared key is only used to authenticate, not protect
- Dynamic calculation creates a secret encryption key

RSA Signature Mode: x.509 certificate contents that correlate with definitions on the peer authenticate the peers to each other.

- Both sides have a certificate
- IBM IKE implementations obtain a peer's certificate as part of the IKE exchange

Unlike SSL/TLS or AT-TLS, both sides **MUST** authenticate to each other.

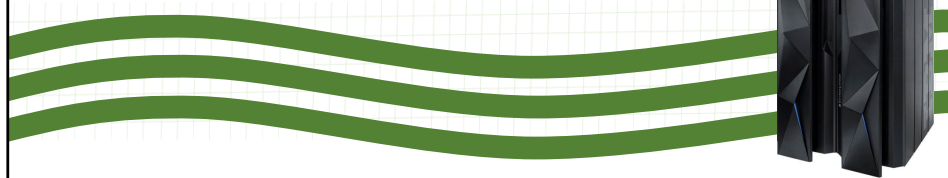
Phase I of the Dynamic Negotiation process performs these activities:

- An encryption algorithm to be used, such as the Data Encryption Standard (DES) for IKE messages.
- A hash algorithm (MD5 or SHA, as used by AH or ESP) to be used for IKE messages.
- An authentication method, such as authentication using previously shared keys or using RSA Signature Mode..
- A Diffie-Hellman group that defines how the symmetric session key will be kept private through the interaction of a Public and a Private key used to encrypt the session key.

Phase 2 of the Dynamic Negotiation performs these activities:

- Transmission of the IP Data Offer Symmetric Key (to encrypt the data payload)

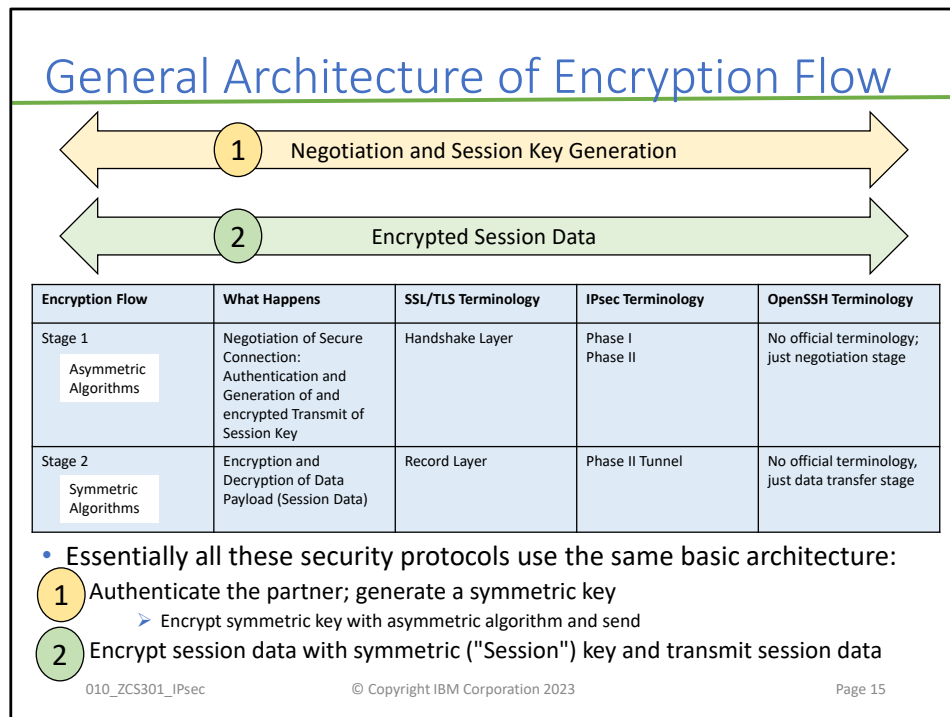
Asymmetric and Symmetric Encryption



010_ZCS301_IPsec

© Copyright IBM Corporation 2023

Page 14



From the z/OS Security Server RACF Security Administrator's Guide, Chapter 21 "RACF and Digital Certificates":

"Each party, both client and server, has its own certificate, a matching private key, and a list of trusted certificate-authority (CA) certificates. When the client needs to authenticate itself to the server to be able to perform a transaction, both the server and client need to verify one another. The protocol for a secure handshake for mutual verification begins with the parties exchanging certificates. Each party then separately validates the other's certificate to make sure that its signature is valid, that the subject name in the certificate is correct, and that the certificate originated from a trusted certificate authority. If successful, each party must prove to the other that it owns the private key that matches its public key certificate. This step establishes proof of possession and can be accomplished by having each party sign a known unique value, such as a hash of the message traffic between the two parties. If each signature can be validated using the associated public key, the proofs are successful. The final step in this handshake is for one of the parties to generate a random symmetric key, encrypt it using the other party's public key, and send it to the other party. This random symmetric key may then be used to encrypt the data for the remainder of the session. Once the secure handshake is complete, secure transactions can be safely handled in the z/OS environment between this client and server."

SSL is a layered protocol. At each layer, messages may include fields for length, description, content.

There are 2 distinct bounds of communication between the client and server called the handshake layer and the record layer.

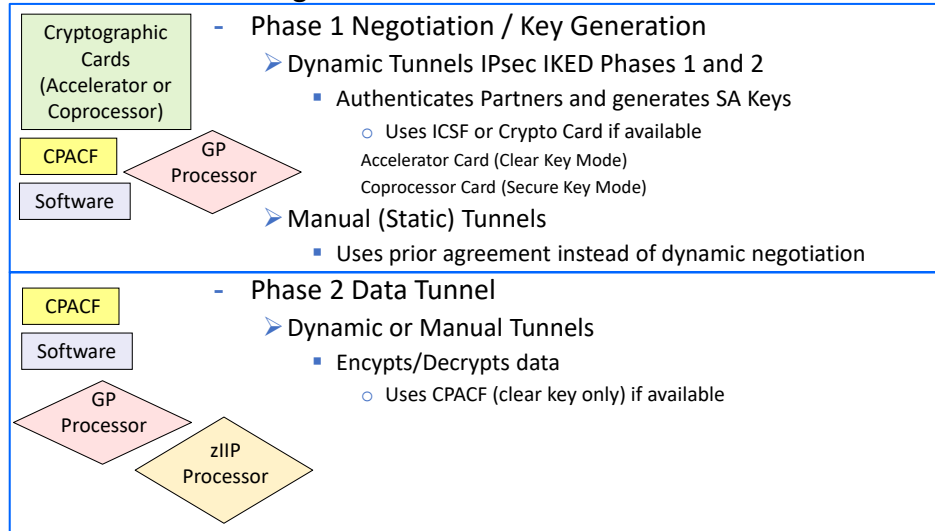
In the Handshake Layer the client contacts the server and information is exchanged to determine the capabilities of each and to come to an agreement on possible information required later.

The session (symmetric) key is used to encrypt the Record Layer flows.

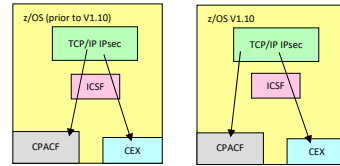
In the Record Layer the client and server exchange data, based on the functions selected in the Handshake Layer. Data protected is:

- the userid
- the password
- the actual data payload

• Two Stages

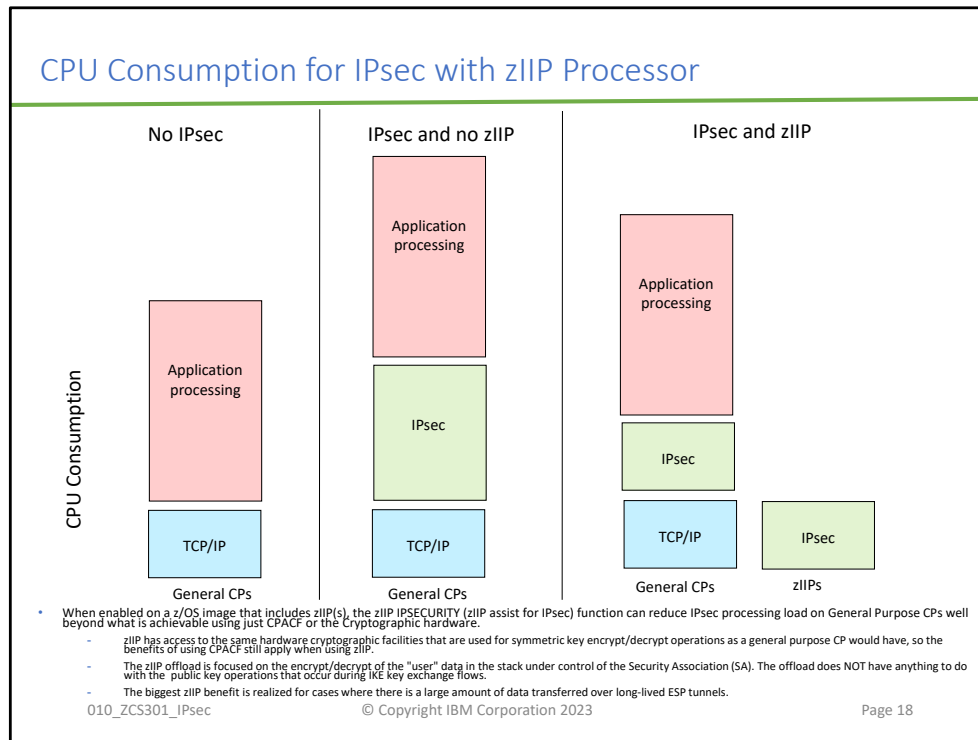


Cryptography Hardware Support



- IPsec work utilizes System z cryptography hardware if the hardware is enabled, and if the required cryptographic algorithm is supported by the hardware.
 - See the "Using Hardware Cryptographic Features with System SSL" table in the z/OS Cryptographic Services System Secure Sockets Layer Programming, manual for details about which cryptographic algorithms are supported by the System z cryptographic hardware.
- Prior to z/OS V1.10, IPsec uses (Integrated Cryptographic Service Facility) ICSF for all hardware crypto functions.
- Starting in z/OS V1.10, IPsec will save cycles by using the CPACF (CP Assist for Cryptographic Function) functions directly, for the crypto functions that are supported by the CPACF.
 - Throughput improvements are greatest for especially short datagrams (< 1K).
 - Anything else that currently is directed to ICSF will continue to be directed to ICSF.
 - Workload for Crypto Express will continue to flow through ICSF.
 - Note: Plans for the z/OS Communications Server are subject to change prior to general availability.

See the "Using Hardware Cryptographic Features with System SSL" table in the z/OS Cryptographic Services System Secure Sockets Layer Programming, manual for details about which cryptographic algorithms are supported by the System z cryptographic hardware.



The performance document is at <http://www-03.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP100988>

The zIIP assisted IPsec function is designed to move most of the IPsec processing from the general purpose processors to the zIIPs. CPACF is available on zIIPs as well as general CPs, so when the IPsec processing moves to a zIIP, it will use CPACF on that zIIP to do its symmetric encrypt/decrypt as well as SHA hashing operations.

z/OS CS TCP/IP recognizes IPsec packets and routes a portion of them to an independent enclave SRB.

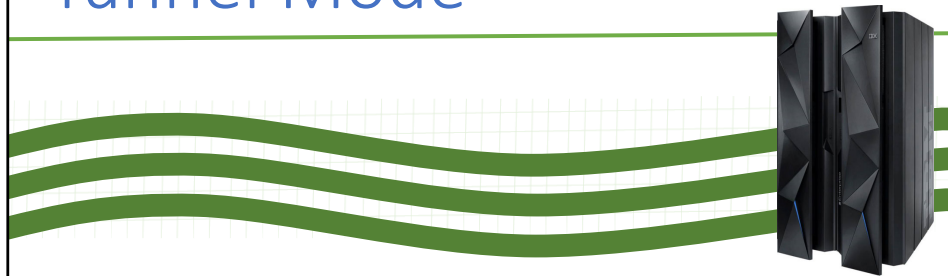
- This workload is eligible for the zIIP

The zIIP IPSECURITY design allows Communication Server to interact with z/OS Workload Manager to have a portion of its enclave Service Request Block (SRB) work directed to zIIP. Within CommServer starting with z/OS V1R8, much of the processing related to security routines (Encryption and Authentication algorithms) runs in Enclave SRBs. In z/OS V1R8, this Enclave SRB workload can be directed to available zIIPs. A single configuration statement within the TCP/IP profile triggers CommServer to request z/OS to direct this IPsec Enclave SRB processing to available zIIPs.

- GLOBALCONFIG ZIIP IPSECURITY

It is possible for IPsec workload performance (response time and/or aggregate throughput) to be improved when zIIPs are added to the configuration. Such Response Time/Throughput improvement is likely if your network performance is currently being constrained by high CPU utilization, and addition of zIIP(s) relieves this constraint.

Transport Mode Versus Tunnel Mode



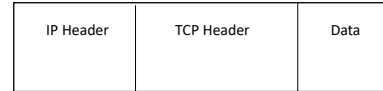
010_ZCS301_IPsec

© Copyright IBM Corporation 2023

Page 19

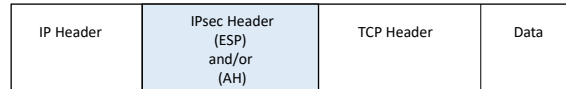
Transport Mode or Tunnel Mode

- Original Packet



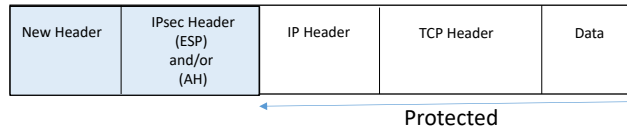
- Transport Mode

- Original data is protected but certain header fields are not.
- Typically used when Security Endpoint and Data Endpoint are the same IP Address.



- Tunnel Mode

- Protects the entire IP packet
- A new IP header and IPsec header are placed in front of the original IP packet
- Always used when Security Endpoint and Data Endpoint are different IP Addresses on either or both ends.
- Optionally can be used to protect all contents of Original Packet where the Security Endpoint and Data Endpoints are the same IP Address.



010_ZCS301_IPsec

© Copyright IBM Corporation 2023

Page 20

In Transport Mode the IP header is separated from the original packet. The IPsec protocol is then applied to the packet and the original header is then placed back in front of the IPsec header.

- Separate IP header and data portion of the packet
- Apply IPsec to the data creating an IPsec packet
- Attach original IP header to IPsec packet
- Adjust length, protocol, checksum fields in the original IP header

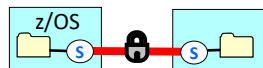
Transport is used by hosts, not by gateways.

In Tunnel Mode a new IP packet is created and the original IP datagram (headers and data) is then the data of the new IP packet. Tunnel Mode must be used whenever the Security Endpoint and the Data Endpoint are different at one or both ends of a peer relationship.

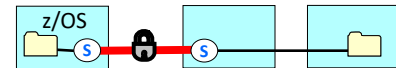
Tunnel Mode may be used when the Security Endpoint and the Data Endpoint are the same, even though you could have used Transport Mode in this case.

- In this instance, although the IP Header contains the same addresses, you could use tunnel mode to hide the original IP header and other protocol Headers as a whole as the packet traverses the network. Another reason to implement tunnel mode in such an instance would be if the remote system might prefer tunnel mode or could possibly not even support transport mode (as might be the case with IKEv2).

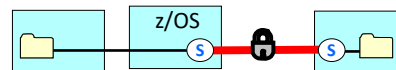
IPsec Network Topology



- Host-to-Host: z/OS node is data host and security endpoint both (host), connecting to a data host and security endpoint remote node (host).



- Host-to-Gateway: z/OS node is a data host and security endpoint both (host), connecting to a security endpoint (gateway) in front of a remote data host.



- Gateway-to-Host: Data host in front of z/OS node security endpoint (gateway), which connects to a data host and security endpoint remote node (host).



- Gateway-to-Gateway: Data host in front of z/OS node security endpoint (gateway), which connects to a security endpoint (gateway) in front of a remote data host.

- Security Endpoints are the endpoints of the IKE (phase 1) SA.
 - The negotiating IKEDs are located at the security endpoints.
 - The data is protected between the security endpoints.
- Data Endpoints are the endpoints of the Dynamic (phase 2) SA.
 - The data is not protected between the data endpoints and the security endpoints.
- When IPsec is configured, z/OS is considered a "Host" if it is the security endpoint (IKE/phase 1) and data endpoint (dynamic/phase 2), otherwise it is considered a "Gateway".
- z/OS may be a data endpoint behind a security endpoint.
 - In this case z/OS does not require any IKE or IPsec customization.

010_ZCS301_IPsec

© Copyright IBM Corporation 2023

Page 21

With IPsec the data is protected only between the IPsec endpoints (i.e., between the two peers).

There are four configurations and variations therefore that define which part of the data path is protected.

Host-to-Host:

- The Data Endpoints and the Security Endpoints are the same.
- The IP Header can contain the address of the data endpoints because the Security Endpoints have the same IP Addresses.

Host-to-Gateway:

- The Data Endpoints and the Security Endpoints are the same on the left but not on the right of the visual.
- We need an IP Header to identify the Security Endpoints.
- We need a different IP Header to identify the Data Endpoints.
- In this way we can TUNNEL the Data Endpoint IP Header inside the Security Endpoint IP Header.

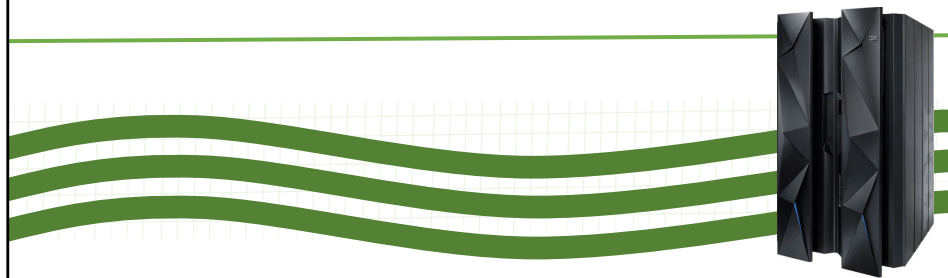
Gateway to Gateway:

- The Data and Security Endpoints are different.
- We need an IP Header to identify the Security Endpoints.
- We need a different IP Header to identify the Data Endpoints.
- In this way we can TUNNEL the Data Endpoint IP Header inside the Security Endpoint IP Header.

Gateway-to-Host:

- The Data and Security Endpoints on the left are different.
- The Data and Security Endpoints on the right are the same.
- We need an IP Header to identify the Security Endpoints.
- We need a different IP Header to identify the Data Endpoints.
- In this way we can TUNNEL the Data Endpoint IP Header inside the Security Endpoint IP Header.

z/OS IPsec Implementation



010_ZCS301_IPsec

© Copyright IBM Corporation 2023

Page 22

Policy Definition

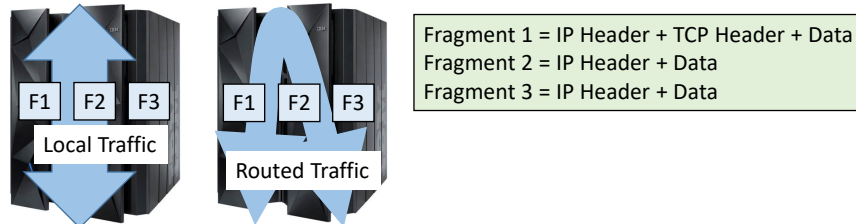
Criteria	Description
Packet	
Source address	Source IP Address in IP header of packet
Destination address	Destination IP Address in IP header of packet
Protocol	Protocol in the IP header of packet
Source port	Source Port in TCP or UDP transport header of packet
Destination port	Destination Port in TCP or UDP transport header of packet
ICMP type and code	ICMP type and code in ICMP header of packet
OSPF type	OSPF type in OSPF header of packet
Network Attributes	
Direction	Direction of packet
Routing	Traffic is Local if source or destination IP address exists on local host, otherwise traffic is Routed
Link security class	Class that allows you to group interfaces with similar security requirements. Non-VIPA addresses can be assigned a security class. Packets inherit the security class of the interface over which packet is sent/received.
Time Condition	
Time, Day, Week, Month	When filter rule is active.
Action	
Permit, Deny, or IPsec	Permit, Deny, or use IPsec

010_ZCS301_IPsec

© Copyright IBM Corporation 2023

Page 23

RFC4301



- Routed Traffic
 - First fragmented packet contains protocol header, while later packets do not.
- Local Traffic
 - Fragmented packets are put back together before being inspected.
- RFC4301 does not allow policy rules to be created for routed traffic that define those items that exist in the protocol header:
 - Port Numbers
 - ICMP(v6) Code Types
 - OSPF Types
- z/OS Network Configuration Assistant will prevent the creation of a rule that does not adhere to RFC4301.
- z/OS Policy Agent will not load a rule that does not adhere to RFC4301.

010_ZCS301_IPsec

© Copyright IBM Corporation 2023

Page 24

IPsec Dynamic Tunnel IKED Identity Choices

- X.509 Certificate

```
Label: IKED1
...
Issuer's Name:OU=MVSNM1 Certificate Authority,O=IBM,C=US
1 Subject's Name:CN=Server 1,T=IKED1,OU=MVS,O=IBM
Subject's AltNames:
2 IP:172.17.0.71
3 EMail: mvsnm1 at washington.ibm.com
4 Domain: washington.ibm.com
URI: mvsnm1.washington.ibm.com
...
```

A choice of four different Certificate fields. The Policy may define any one of the four fields. You can see how the optional AltNames fields are usually easier to define in the policy.

One AltName field may be used for to identify one side and a different AltName field may be used to identify the other side.

- Policy Configuration

```
Policy 2 "IP Address" = 172.17.0.71
Policy 4 "FQDN" = washington.ibm.com
Policy 3 "Userid@FQDN" = mvsnm1@washington.ibm.com
Policy 1 "X.509 Distinguished Name" = CN=Server 1,T=IKED1,OU=MVS,O=IBM
Policy 5 "Key ID" (ASCII, EBCDIC, or Hexadecimal) = gaithersburg
```

← Preshared Key Mode rather than RSA Signature Mode Authentication

- Sequence of Elements must be accurate!

010_ZCS301_IPsec

© Copyright IBM Corporation 2023

Page 25

If implementing Dynamic Tunnels with RSA Signature Mode, the initial authentication of the peers requires a match between fields in the exchanged certificate and fields in the IPsec policy.

Guideline: Do not use the same identity on more than one TCP/IP stack or z/OS system image. IKED assumes that an identity is not used by any other stack, and this assumption can lead to a disruption of IPsec service for other stacks if identities are shared between stacks. Identity choice in Policy Configuration must match information in X.509 Certificate. The IKE Identity that is used for the IPsec protocol exchanges **MUST BE UNIQUE!!** Otherwise the negotiation will fail!

If Local IKED Identity "IP Address" is configured in Policy Configuration, then it must match "IP" defined in the certificate. "IP" can be named in an x.509 certificate V3 extension field called the "Alternate Name" or the "Altname".

NOTE: IKED Identity is also known as "RSA Identity" or "Security Endpoint Identity".

If Local IKED Identity "FQDN" is configured in Policy Configuration, then it must match "Domain" defined in the certificate. "Domain" can be named in an x.509 certificate V3 extension field called the "Alternate Name" or the "Altname".

If Local IKED Identity "Userid@FQDN" is configured in Policy Configuration, then it must match "EMail" defined in the certificate. "EMail" can be named in an x.509 certificate V3 extension field called the "Alternate Name" or the "Altname".

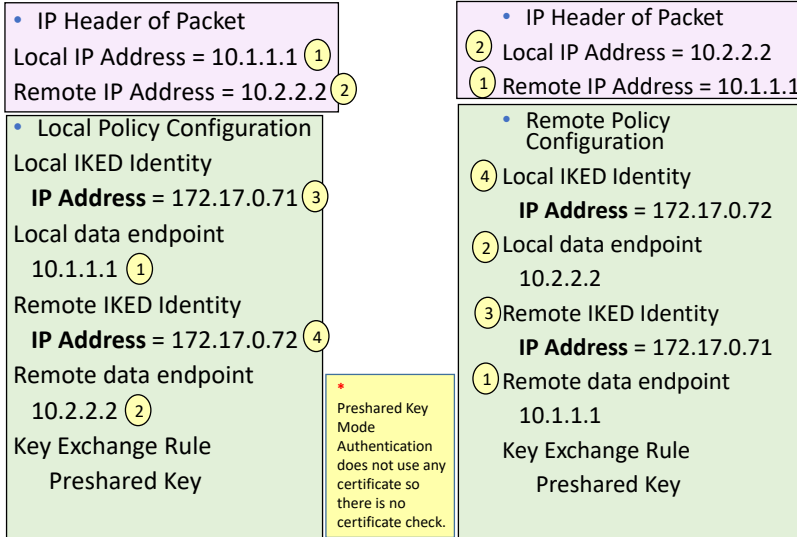
If Local IKED Identity "X.509 Distinguished Name" is configured in Policy Configuration, then it must match all the elements of "Subject's Name" defined in the certificate and in the same order as how RACF has stored them. (This may not be the same order as that in which you defined the elements. Use "racdcert list" command to verify the correct sequence of the Distinguished Name elements.)

The identity choices for the Remote IKED Identity (RSA Identity) can be any of the choices named above, but whichever Identity is chosen for the protocol exchange **MUST BE UNIQUE**. That is, you cannot have two peers that are presenting the same IKED Identity.

If implementing Dynamic Tunnels with Pre-shared Key Mode, the x.509 certificate does not play a role. Instead, matching is based strictly on the strings in the IPsec policy.

Again, the IKED identities must be unique at all Peers. Otherwise IKED negotiation will fail.

Data Endpoint and IKED Identity Match



010_ZCS301_IPsec

© Copyright IBM Corporation 2023

Page 26

If implementing Dynamic Tunnels with Pre-shared Key Mode, the x.509 certificate does not play a role.

Instead, matching is based strictly on the strings in the IPSec policy.

Note in the example above, where we show you two IKED peers, how the two policies and their entries correlate with each other.

Local IKED Identity must match the entry for Remote IKED Identity.

Local data endpoint must match the entry for Remote data endpoint.

The key exchange at both sides must include the same key, specified in ASCII, EBCDIC, or Hexadecimal.

Also note how, with the exception of the data endpoint, the entries themselves are not validated by any other configuration option you may have chosen.

The Local IKE Identity can be anything you like -- even a completely fictitious entry -- even a subnet value so that a group of like IKED peers can use the same definition if desirable.

IKED Identity = Distinguished Name

<ul style="list-style-type: none"> IP Header of Packet Local IP Address = 10.1.1.1 (1) Remote IP Address = 10.2.2.2 (2) Local X.509 Certificate Subject's Name:CN=Server 1.T=... (3) Local Policy Configuration Local IKED Identity X.509 Dist Name = CN=Server 1,T=... (3) Local data endpoint 10.1.1.1 (1) Remote IKED Identity X.509 Dist Name = CN=Server 2,T=... (4) Remote data endpoint 10.2.2.2 (2) Key Exchange Rule RSA Signature 	<ul style="list-style-type: none"> IP Header of Packet Local IP Address = 10.2.2.2 (2) Remote IP Address = 10.1.1.1 (1) Remote X.509 Certificate Subject's Name:CN=Server 2.T=IKED2... (4) Remote Policy Configuration Local IKED Identity X.509 Dist Name = CN=Server 2,T=IKED2... (4) Local data endpoint 10.2.2.2 (2) Remote IKED Identity X.509 Dist Name = CN=Server 1,T=IKED1... (3) Remote data endpoint 10.1.1.1 (1) Key Exchange Rule RSA Signature
--	---

010_ZCS301_IPsec

© Copyright IBM Corporation 2023

Page 27

Notice in Local IKED's certificate that its identity for purposes of RSA Signature Mode Authentication is "172.17.0.71."

When Local IKED presents its certificate during Phase I IKE negotiations, this field needs to match the value for "Remote IKED Identity" in the Remote IKED policy file.

The policy also requires that you configure a Local Data Endpoint and a Remote Data Endpoint. These must correlate with the Peer's policy.

Note in the example above, where we show you two IKED peers, how the two policies and their entries correlate with each other.

Local IKED Identity must match the entry for Remote IKED Identity.

Local data endpoint must match the entry for Remote data endpoint.

Also note how, with the exception of the data endpoint, the entries themselves are not validated by any other configuration option you may have chosen.

The Local IKE Identity can be anything you like -- even a completely fictitious entry -- even a subnet value so that a group of like IKED peers can use the same definition if desirable.

IKED Identity AltName IP Address

<ul style="list-style-type: none"> IP Header of Packet Local IP Address = 10.1.1.1 (1) Remote IP Address = 10.2.2.2 (2) 	<ul style="list-style-type: none"> IP Header of Packet (2) Local IP Address = 10.2.2.2 (1) Remote IP Address = 10.1.1.1
<ul style="list-style-type: none"> Local X.509 Certificate Subject's AltNames: IP: 172.17.0.71 (3) 	<ul style="list-style-type: none"> Remote X.509 Certificate Subject's AltNames: (4) IP: 172.17.0.72
<ul style="list-style-type: none"> Local Policy Configuration Local IKED Identity IP Address = 172.17.0.71 (3) Local data endpoint 10.1.1.1 (1) Remote IKED Identity IP Address = 172.17.0.72 (4) Remote data endpoint 10.2.2.2 (2) Key Exchange Rule RSA Signature 	<ul style="list-style-type: none"> Remote Policy Configuration Local IKED Identity (4) IP Address = 172.17.0.72 Local data endpoint (2) 10.2.2.2 Remote IKED Identity (3) IP Address = 172.17.0.71 Remote data endpoint (1) 10.1.1.1 Key Exchange Rule RSA Signature

If Host Topology and IKED Identity is defined as IP address, IP Address in certificate is checked against IP Address, unless "BypassIpValidation" is defined. See KeyExchangeAction.

Notice in Local IKED's certificate that its identity for purposes of RSA Signature Mode Authentication is "172.17.0.71."

When Local IKED presents its certificate during Phase I IKE negotiations, this field needs to match the value for "Remote IKED Identity" in the Remote IKED policy file.

The policy also requires that you configure a Local Data Endpoint and a Remote Data Endpoint. These must correlate with the Peer's policy.

Note in the example above, where we show you two IKED peers, how the two policies and their entries correlate with each other.

Local IKED Identity must match the entry for Remote IKED Identity.

Local data endpoint must match the entry for Remote data endpoint.

The key exchange at both sides must include the same key, specified in ASCII, EBCDIC, or Hexadecimal.

Also note how, with the exception of the data endpoint, the entries themselves are not validated by any other configuration option you may have chosen.

The Local IKE Identity can be anything you like -- even a completely fictitious entry -- even a subnet value so that a group of like IKED peers can use the same definition if desirable.

The Local IKE Identity can be an FQDN while the Remote IKE Identity could be another choice.

IKED Identity AltName Email

<ul style="list-style-type: none"> IP Header of Packet Local IP Address = 10.1.1.1 (1) Remote IP Address = 10.2.2.2 (2) 	<ul style="list-style-type: none"> IP Header of Packet (2) Local IP Address = 10.2.2.2 (1) Remote IP Address = 10.1.1.1
<ul style="list-style-type: none"> Local X.509 Certificate Subject's AltNames: (3) Email: mvsnm1 at washington.ibm.com 	<ul style="list-style-type: none"> Remote X.509 Certificate Subject's AltNames: (4) Email: mvsnm2 at washington.ibm.com
<ul style="list-style-type: none"> Local Policy Configuration Local IKED Identity Userid@FQDN = mvsnm1@washington.ibm.com (3) Local data endpoint 10.1.1.1 (1) Remote IKED Identity Userid@FQDN = mvsnm2@washington.ibm.com (4) Remote data endpoint 10.2.2.2 (2) Key Exchange Rule RSA Signature 	<ul style="list-style-type: none"> Remote Policy Configuration Local IKED Identity Userid@FQDN = mvsnm2@washington.ibm.com (4) Local data endpoint (2) 10.2.2.2 Remote IKED Identity Userid@FQDN = (3) mvsnm1@washington.ibm.com Remote data endpoint (1) 10.1.1.1 Key Exchange Rule RSA Signature

010_ZCS301_IPsec

© Copyright IBM Corporation 2023

Page 29

Notice in Local IKED's certificate that its identity for purposes of RSA Signature Mode Authentication is "172.17.0.71."

When Local IKED presents its certificate during Phase I IKE negotiations, this field needs to match the value for "Remote IKED Identity" in the Remote IKED policy file.

The policy also requires that you configure a Local Data Endpoint and a Remote Data Endpoint. These must correlate with the Peer's policy.

Note in the example above, where we show you two IKED peers, how the two policies and their entries correlate with each other.

Local IKED Identity must match the entry for Remote IKED Identity.

Local data endpoint must match the entry for Remote data endpoint.

The key exchange at both sides must include the same key, specified in ASCII, EBCDIC, or Hexadecimal.

Also note how, with the exception of the data endpoint, the entries themselves are not validated by any other configuration option you may have chosen.

The Local IKE Identity can be anything you like -- even a completely fictitious entry -- even a subnet value so that a group of like IKED peers can use the same definition if desirable.

The Local IKE Identity can be an FQDN while the Remote IKE Identity could be another choice.

IKED Identity AltName Domain

<ul style="list-style-type: none"> IP Header of Packet Local IP Address = 10.1.1.1 (1) Remote IP Address = 10.2.2.2 (2) 	<ul style="list-style-type: none"> IP Header of Packet Local IP Address = 10.2.2.2 (2) Remote IP Address = 10.1.1.1 (1)
<ul style="list-style-type: none"> Local X.509 Certificate Subject's AltNames: Domain: washington.ibm.com (3) 	<ul style="list-style-type: none"> Remote X.509 Certificate Subject's AltNames: Domain: gbg.lab.ibm.com (4)
<ul style="list-style-type: none"> Local Policy Configuration Local IKED Identity FQDN = washington.ibm.com (3) Local data endpoint 10.1.1.1 (1) Remote IKED Identity FQDN = gbg.lab.ibm.com (4) Remote data endpoint 10.2.2.2 (2) Key Exchange Rule RSA Signature 	<ul style="list-style-type: none"> Remote Policy Configuration Local IKED Identity FQDN = gbg.lab.ibm.com (4) Local data endpoint 10.2.2.2 (2) Remote IKED Identity FQDN = washington.ibm.com (3) Remote data endpoint 10.1.1.1 (1) Key Exchange Rule RSA Signature

010_ZCS301_IPsec

© Copyright IBM Corporation 2023

Page 30

Notice in Local IKED's certificate that its identity for purposes of RSA Signature Mode Authentication is "172.17.0.71."

When Local IKED presents its certificate during Phase I IKE negotiations, this field needs to match the value for "Remote IKED Identity" in the Remote IKED policy file.

The policy also requires that you configure a Local Data Endpoint and a Remote Data Endpoint. These must correlate with the Peer's policy.

Note in the example above, where we show you two IKED peers, how the two policies and their entries correlate with each other.

Local IKED Identity must match the entry for Remote IKED Identity.

Local data endpoint must match the entry for Remote data endpoint.

The key exchange at both sides must include the same key, specified in ASCII, EBCDIC, or Hexadecimal.

Also note how, with the exception of the data endpoint, the entries themselves are not validated by any other configuration option you may have chosen.

The Local IKE Identity can be anything you like -- even a completely fictitious entry -- even a subnet value so that a group of like IKED peers can use the same definition if desirable.

The Local IKE Identity can be an FQDN while the Remote IKE Identity could be another choice.

[illegible]

IPsec Steps

- Provide x.509 certificates if deploying IPsec with RSA Signature Mode.
- Build IPsec policies with IBM Configuration Assistant
- Implement PAGENT, SYSLOGD, and TRMD on z/OS
- Implement IKED for dynamic VPNs
- Enable IPSECURITY in the TCP/IP Stack:
 - Set IPCONFIG IPSECURITY in PROFILE.TCPIP.
 - Optionally establish IPSECRULEs in TCP/IP Profile to override the Default Implicit "denyall" rule that is in effect until a set of PAGENT IPsec policy rules is activated.
- IKE daemon configuration file search order
 - The MVS data set or z/OS UNIX file specified by IKED_FILE environment variable
 - /etc/security/iked.conf
- Sample IKE daemon configuration file
 - /usr/lpp/tcpip/samples/iked.conf
- Reserve ports for IKED in the TCP/IP profile
 - PORT
 - 500 UDP IKED
 - 4500 UDP IKED
- Open Firewalls if necessary:
 - Ports
 - Protocols (UDP; ESP=50; AH=51; etc)
- IKED sample proc
 - TCPIP.SEZAINST(IKED)

010_ZCS301_IPsec

© Copyright IBM Corporation 2023

Page 32

Consult the following manuals for guidance:

- z/OS Communications Server IP Configuration Guide (SC31-8775)
- Communications Server for z/OS TCP/IP Implementation Volume IV: Security and Policy-based Networking

Once you code IPCONFIG IPSECURITY in a stack, you still want to ensure that you can telnet into that stack for maintenance. This means that you will want to ensure that your Policy Agent rules are working the way you want them to. If PAGENT does not initialize, you will not be able to access the TCP/IP stack unless you have IPSECRULEs inside the TCP/IP Profile that allow at least the administrator to telnet into the stack.

Of course, you should always have a backup plan to be able to modify non-working TCP/IP profiles.

Examples: an alternate TCP/IP profile without IPCONFIG IPSECURITY. (Note that the OBEYFILE technique will not work for removing IPCONFIG IPSECURITY from a TCP/IP Profile.)

Example: an SNA path into the VTAM of the MVS image.

Example: an alternate path to the DASD that holds the non-functioning TCP/IP profile so that you may edit the necessary definitions to provide access.

Activate IPsec VPN

Advanced Connectivity Rule Settings

Dynamic Tunnels: How to Activate

- Use this panel to indicate how each dynamic tunnel may be activated
- Indicate Yes or No in each activation column.

When using Command or Auto Active, edit the Handle column to enter a required handle; see Help for details

Traffic Descriptor	Protocol	Local Port	Remote Port	Connect Direction	IPsec Security Level	Allow Remote Activation	Allow On Demand Activation	Auto Activate	IPsec Command Activation
<input type="checkbox"/> FTP-Client	TCP	All Ephemeral	21	Outbound	IPSec_Gold	Yes	Yes	No	No
<input type="checkbox"/> FTP-Client	TCP	All Ephemeral	20	Inbound	IPSec_Gold	Yes	No	No	No
<input type="checkbox"/> FTP-Client	TCP	All Ephemeral	50000-50020	Outbound	IPSec_Gold	Yes	Yes	No	No
<input type="checkbox"/> FTP-Server	TCP	21	All Ephemeral	Inbound	IPSec_Gold	Yes	No	No	No
<input type="checkbox"/> FTP-Server	TCP	20	All Ephemeral	Outbound	IPSec_Gold	Yes	Yes	No	No
<input type="checkbox"/> FTP-Server	TCP	50000-50020	All Ephemeral	Inbound	IPSec_Gold	Yes	No	No	No

- Specify when IPsec VPN will be activated:
 - Remote end causes activation
 - Activate this end when a filter applies (on-demand)(default)
 - Activate automatically at IPsec Policy initialization
 - Activate manually with 'ipsec' command

010_ZCS301_IPsec © Copyright IBM Corporation 2023 Page 33

By default, tunnels will activate when a condition requires a tunnel to be built (on-demand) or when the remote side requests activation of this side of the tunnel.
However, you may override the defaults and cause any activation type you like.

FIPS 140

- Configure the IKED, the NSSD, and the TCP/IP stack components to operate in FIPS 140 mode.
- Configure FIPS 140 mode in System SSL
 - Places restrictions on the cryptographic algorithms and key lengths that can be used for IP security.
 - See z/OS Cryptographic Services System Secure Sockets Layer Programming, SC14-7495, for the latest support.
- Recommendation: Initialize ICSF in FIPS 140 mode.

010_ZCS301_IPsec

© Copyright IBM Corporation 2023

Page 34

Please consult the IP Configuration Guide for more detail and references on the subject of FIP 140 enablement. FIPS 140 places some restrictions on the use of cryptographic algorithms and modules.

Some examples of the restrictions are:

- Cryptographic algorithms and keys must be contained within a cryptographic module and accessed through a well defined cryptographic boundary.
- Use of weaker cryptographic algorithms (for example, DES and MD5) is not allowed.
- Use of weaker asymmetric key lengths (for example, RSA digital signature operations using key lengths less than 1024 bits) is not allowed.
- Use of Diffie-Hellman groups with weaker key lengths (key lengths less than 2048 bits) is not allowed. This restriction applies to groups 1, 2, and 5.

See the National Institute of Standards and Technology (NIST) Web site at <http://csrc.nist.gov/publications/PubsFIPS.html> for the most recent FIPS 140 publication, and other related publications.

On z/OS systems, Integrated Cryptographic Services Facility (ICSF) and System SSL provide cryptographic services. z/OS Communications Server uses ICSF and System SSL in addition to its own cryptographic algorithms in some of its networking security functions, such as AT-TLS and IP security.

You can configure ICSF, System SSL, and the z/OS Communications Server networking security functions in FIPS 140 mode, in which they will enforce FIPS 140 restrictions.

ICSF in FIPS 140 mode is not always required; the requirement depends on the encryption algorithms in use. However, for ease of implementation we recommend initializing ICSF.

Enterprise Extender (EE) IPsec Performance

- z/OS V1R11 Improved performance for EE over IPsec
 - The “bursty” nature of HPR traffic can cause significant performance degradation when it is carried over IPsec tunnels.
 - Smaller bursts frequently get sent before larger bursts. This results in out-of-order segments that are dropped, forcing retransmits.
 - Primarily observed for streaming traffic over EE.
 - Very large volumes of SNA interactive traffic (such as tens of thousands of sessions that may appear similar to bulk data traffic) are likely to see issues too.
 - V1R11 breaks large bursts into batches of smaller bursts.
- z/OS V1R11 support for EE over IPsec offloaded to a zIIP
 - Support for offloading outbound EE over IPsec traffic to a zIIP processor.
 - Previously only inbound EE over IPsec traffic was processed on the zIIP.
- Protecting both interactive and streaming EE workload with IPsec is now fully recommended.

ipsec Command

- ipsec command SERVAUTH profile
 - EZB.IPSECCMD.sysname.stackname.command_type
 - SETROPTS GENERIC(SERVAUTH)
 - RDEFINE SERVAUTH EZB.IPSECCMD.sysname.tcpprocname.* UACC(NONE)
 - PERMIT EZB.IPSECCMD.sysname.tcpprocname.* CLASS(SERVAUTH)
 - ID(userid) ACCESS(READ)
 - SETROPTS GENERIC(SERVAUTH) REFRESH
- ipsec command options
 - -f for IP Filter
 - -F for Defensive Filter
 - -m for Manual Tunnel
 - -k for IKE Tunnel
 - -y for Dynamic Tunnel
 - -i for Interface
 - -t for IP Traffic Test
 - -o for NATT Port Translation
 - -w for IKED Network Security
 - -x for Network Security Server
 - -?

010_ZCS301_IPsec

© Copyright IBM Corporation 2023

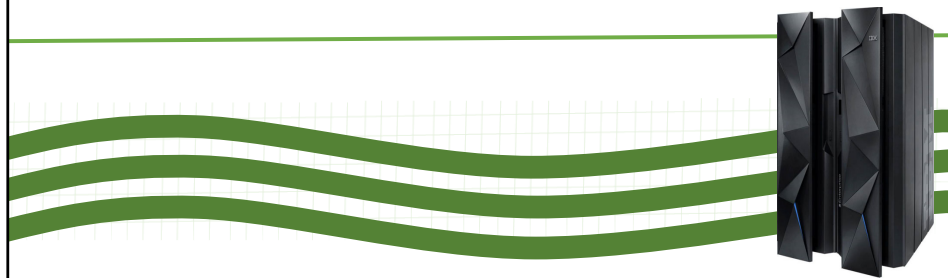
Page 36

You must have IPSEC command authorization in the SERVAUTH class to be able to execute the IPSEC command:

- On our systems, the USER and the SYS1 groups have access to the SERVAUTH class named EZB.IPSECCMD.*.*.

The ipsec command is an APF-authorized application. Users of the ipsec command must also be authorized through the security access facility (SAF). This information assumes that the SAF is RACF. Authorization is managed with the SERVAUTH profile. For the ipsec -f default and ipsec -f reload command, file system access is also required. You do not need root authority to use the ipsec command, but for filter rule set control on a local stack, the administrator must provide you with some file access capability

Network Security Services

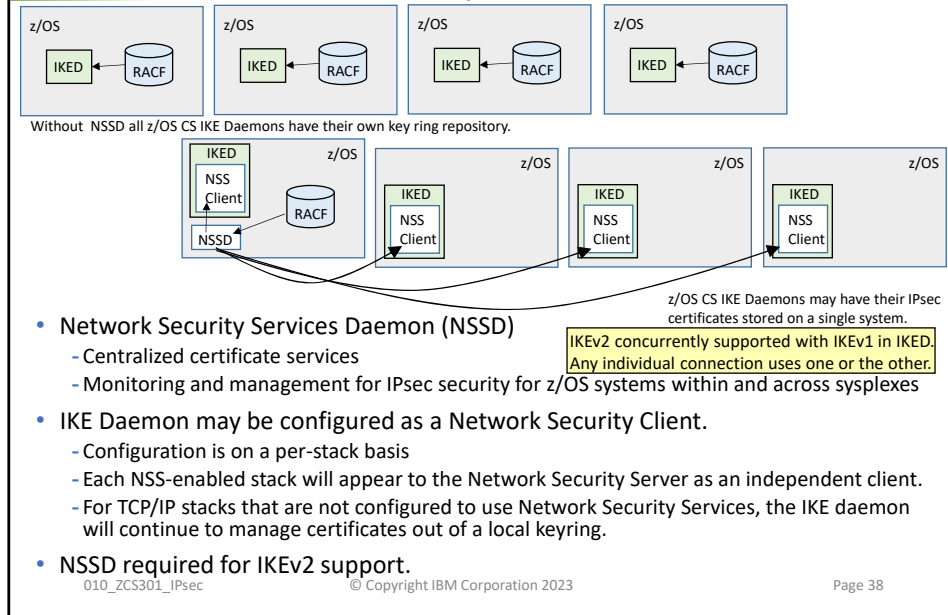


010_ZCS301_IPsec

© Copyright IBM Corporation 2023

Page 37

Network Security Services



Without NSSD all z/OS CS IKE Daemons have their own Key Ring repository for RSA Signature Mode.

With NSSD z/OS CS IKE Daemons may have their IPsec certificates stored on a single system.

NSSD centralizes the sensitive key ring material that would otherwise need to reside in less secure zones of the network onto a single location in the most secure zone of the network. In addition, NSSD allows for centralized configuration and administration of certificates. It provides a central, SAF-enabled repository for RSA certificates along with signature services within the most trusted zones.

We list here several other advantages inherent in the NSS solution, which allows you to:

Eliminate the need to distribute certificates to security endpoints

- Centralize and reduce configuration and deployment complexity, especially when used with Centralized Policy Services
- Offload digital signature operations from the IKE daemon (the NSS client)
- Enable monitoring and management of remote IPsec endpoints through the ipsec command and a network management programming interface.

IKEv2 Enhancements

- Reduced bandwidth used by control messages
 - This leaves increased bandwidth available for data transmission.
- Extensible Authentication Protocol (EAP) authentication
 - In addition to IKEv1 pre-shared key and certificate authentication
 - EAP is a protocol for wireless networks that expands on authentication methods used by the Point-to-Point Protocol (PPP), a protocol often used when connecting a computer to the Internet. EAP can support multiple authentication mechanisms, such as token cards, smart cards, certificates, one-time passwords, and public key encryption authentication.
- MOBIKE support
 - Required for mobile platforms like phones and by users with multi-homed setups. MOBIKE allows a host that has multiple simultaneous points of attachment to a network to change which interface is forwarding traffic while maintaining a VPN session.
- Network Address Translation (NAT) traversal support
 - NAT in routers between the two endpoints is supported.
- Liveness check
 - Detects whether the tunnel is still alive or not. If the tunnel is down, IKEv2 is able to re-establish the connection automatically.

010_ZCS301_IPsec

© Copyright IBM Corporation 2023

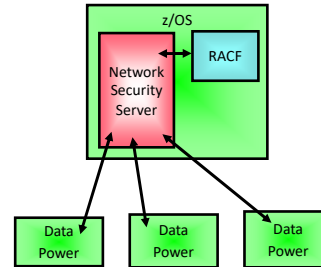
Page 39

IKE version 1.0 (IKEv1) is defined by RFC 2409, *The Internet Key Exchange (IKE)*, and related RFCs. This is the version that has been supported by z/OS

Communications Server for a number of years.

IKE version 2.0 (IKEv2) is defined by RFC 5996, *Internet Key Exchange Protocol: IKEv2*, and related RFCs. Support for IKEv2 was introduced with z/OS V1R12.

DataPower NSSD Support



- Network Security Services Daemon z/OS XMLAppliance Discipline
 - Data Power are non-System z XML appliances that may receive Web Service requests, parse and transform the XML messages, and forward them on to WebSphere Application Servers (WAS).
 - <https://www.ibm.com/products/datapower-gateway>
 - NSSD API provides an interface for the remote Data Power boxes to SAF access services.
 - When WAS is on z/OS, zLinux, or even non-System z.
 - nssctl command for monitoring

010_ZCS301_IPsec

© Copyright IBM Corporation 2023

Page 40

NSSD is used by Data Power appliances to authenticate users of the Data Power platform.
NSSD is used to house x.509 certificates for Data Power appliances.

RACF for NSSD

- NSSD userid requires UID 0.
- Define IRR.DIGTCERT FACILITY class resources in RACF if they do not already exist and PERMIT NSSD administrator userid to IRR.DIGTCERT FACILITY class resources.
- Define NSS client userid and password.
- Optionally define an NSSD profile in the APPL class with UACC(NONE) and issue PERMIT to authorize each NSS client.
- Define SERVAUTH profiles to authorize NSS clients.
 - Create SERVAUTH resource profile for NSS Service, and PERMIT READ access for the NSS clients.
 - IPsec certificate service EZB.NSS.sysname.clientname.IPSEC.CERT
 - IPsec certificate service EZB.NSS.sysname.clientname.IPSEC.NETMGMT
 - IPsec certificate service EZB.NSS.sysname.clientname.XMLAPPLIANCE.CERT
 - IPsec certificate service EZB.NSS.sysname.clientname.XMLAPPLIANCE.PRIVKEY
 - IPsec certificate service EZB.NSS.sysname.clientname.XMLAPPLIANCE.SAFACCESS
 - Create SERVAUTH resource profile for each NSS client certificate and CA cert added to NSSD key ring, and PERMIT NSS client to profile for their own certificates.
 - EZB.NSSCERT.sysname.mappedlabelname.HOST
 - EZB.NSSCERT.sysname.mappedlabelname.CERTAUTH
 - Create SERVAUTH resource profile for each **XMLAppliance** client certificate (HOST or CERTAUTH as documented above) and private key, and PERMIT XMLAppliance clients to profile for their own certificates.
 - EZB.NSSCERT.sysname.mappedlabelname.PRIVKEY
 - Create SERVAUTH resource profile for remote monitor (IPSEC.DISPLAY) and manage (IPSEC.CONTROL) NSS clients.
 - EZB.NETMGMT.sysname.clientname.IPSEC.DISPLAY
 - EZB.NETMGMT.sysname.clientname.IPSEC.CONTROL

010_ZCS301_IPsec

© Copyright IBM Corporation 2023

Page 41

See IP Configuration Guide.

```
ADDUSER NSSD DFLTGRP(OMVSGRP) NOPASSWORD OMVS(UID(0) HOME('/'))
RDEFINE STARTED NSSD.* STDATA(USER(NSSD))
SETROPTS RACLIST(STARTED) REFRESH
SETROPTS GENERIC(STARTED) REFRESH
PERMIT SYS1.PARMLIB ID(NSSD) ACCESS(READ)
IRR.DIGTCERT command (ADD, ADDRING, CONNECT, GENCERT, and GENREQ) authority:
RDEFINE FACILITY IRR.DIGTCERT.command UACC(NONE)
PERMIT IRR.DIGTCERT.ADD CLASS(FACILITY) ID(userid) ACC(CONTROL)
PERMIT IRR.DIGTCERT.ADDRING CLASS(FACILITY) ID(userid) ACC(UPDATE)
PERMIT IRR.DIGTCERT.CONNECT CLASS(FACILITY) ID(userid) ACC(CONTROL)
PERMIT IRR.DIGTCERT.GENCERT CLASS(FACILITY) ID(userid) ACC(CONTROL)
PERMIT IRR.DIGTCERT.GENREQ CLASS(FACILITY) ID(userid) ACC(CONTROL)
PERMIT IRR.DIGTCERT.LIST CLASS(FACILITY) ID(userid) ACC(CONTROL)
PERMIT IRR.DIGTCERT.LISTRING CLASS(FACILITY) ID(userid) ACC(UPDATE)
PERMIT IRR.DIGTCERT.LIST CLASS(FACILITY) ID(NSSD) ACC(READ)
PERMIT IRR.DIGTCERT.LISTRING CLASS(FACILITY) ID(NSSD) ACC(READ)
SETROPTS RACLIST(FACILITY) REFRESH
PERMIT BPX.DAEMON CLASS(FACILITY) ID(userid) ACCESS(READ)
ADDUSER userid DFLTGRP(OMVSGRP) OMVS(UID(x))
PERMIT NSSD CLASS(APPL) ID(userid) ACC(READ)
SETROPTS RACLIST(APPL) REFRESH
RDEFINE SERVAUTH profile_name UACC(NONE)
PERMIT profile_name (SERVAUTH) ID(nssclient) ACCESS(READ)
RDEFINE SERVAUTH EZB.NSSCERT.sysname.mappedlabelname.HOST UACC(NONE)
PERMIT EZB.NSSCERT.sysname.mappedlabelname.HOST CLASS(SERVAUTH) ID(userid) ACCESS(READ)
RDEFINE SERVAUTH EZB.NSSCERT.sysname.mappedlabelname.CERTAUTH UACC(NONE)
PERMIT EZB.NSSCERT.sysname.mappedlabelname.CERTAUTH CLASS(SERVAUTH) ID(userid) ACCESS(READ)
RDEFINE SERVAUTH EZB.NSSCERT.sysname.mappedlabelname.PRIVKEY UACC(NONE)
PERMIT EZB.NSSCERT.sysname.mappedlabelname.PRIVKEY CLASS(SERVAUTH) ID(userid) ACCESS(READ)
SETROPTS GENERIC(SERVAUTH) REFRESH
SETROPTS RACLIST(SERVAUTH) REFRESH
NSSD can support passticket in addition to password.
Multiple NSS clients can use a single user ID. However, each NSS client must have a unique client name.
```

RACF for NSSD (cont.)

- If using **XMLAppliance** with ICSF-protected private keys, authorize NSS server to ICSF (CSFSERV).
 - ICSF is required for RSA operations in private key service.
 - When using cryptographic coprocessor, the callable ICSF service names are:
 - CSNDDSG
 - CSNDPKD
- If **XMLAppliance** clients using the SAF access service are using certificates for access checks, enable RACF certificate name filtering.
 - RACF certificate name filtering maps an X.500 distinguished name to a userid when performing SAF access checks.
- NSSD uses ICSF callable services for ECDSA digital signature support.
 - Services it uses are the PKCS11 private key sign service and the PKCS11 public key verify service.
 - Use the CSFSERV general resource class, and the CSF1PKS and CSF1PKV profiles.
 - PERMIT NSSD userid READ access to the profiles.

010_ZCS301_IPsec

© Copyright IBM Corporation 2023

Page 42

RACF commands:

```
RDEFINE service-name CLASS(CSFSERV) UACC(NONE)
```

```
PERMIT service-name CLASS(CSFSERV) ID(server-name) ACCESS(READ)
```

```
SETROPTS CLASSACT(CSFSERV)
```

```
SETROPTS RACLIST(CSFSERV) REFRESH
```

DIGTNMAP class must be active to perform certificate name filtering.

Activate the DIGTNMAP class with the following commands:

```
SETOPS CLASSACT(DIGTNMAP)
```

```
SETROPTS RACLIST(DIGTNMAP) REFRESH
```

Create a certificate name filter for each mapping of an X.500 distinguished name to a RACF ID using the following commands:

```
RACDCERT ID(userid) MAP SDNFILTER('x500dn')
```

```
SETROPTS RACLIST(DIGTNMAP) REFRESH
```

```
PERMIT CSF1PKS CLASS(CSFSERV) ID(NSSD) ACCESS(READ)
```

```
PERMIT CSF1PKV CLASS(CSFSERV) ID(NSSD) ACCESS(READ)
```

```
SETROPTS RACLIST(CSFSERV) REFRESH
```

Configure and Modify NSSD

- Create NSSD configuration file.
 - Use z/OS Configuration Assistant
 - Or use sample /usr/lpp/tcpip/samples/nssd.conf
 - Search Order
 - Environment variable NSSD_FILE
 - /etc/security/nssd.conf
- Create NSSD key ring.
- Define NSSD port to PROFILE.TCPIP.
 - Default is TCP port 4159.
- Create AT-TLS policies to protect NSSD traffic to NSS clients.
- Create NSSD JCL procedure or start from Unix.
 - Sample proc in SEZAINST(NSSD)
 - Only one NSSD per z/OS.
- Modify
 - Flush all cached URLs and reread the NSS server configuration file.
 - MODIFY procname,REFRESH
 - MODIFY NSSD,REFRESH,FILE='/etc/security/nssd.conf2'
 - Display the configuration file parameters.
 - MODIFY procname,DISPLAY
 - Display the contents of the URL cache.
 - MODIFY procname,DISPLAY,URLCACHE

010_ZCS301_IPsec

© Copyright IBM Corporation 2023

Page 43

See IP Configuration Guide and IP Configuration Reference.

The NSS server configuration file allows the URL of a certificate or certificate bundle that is on an HTTP web server to be associated with the label of a certificate on the key ring of the network security server. See IP Config Guide.

Sample profile default IP Filter rules:

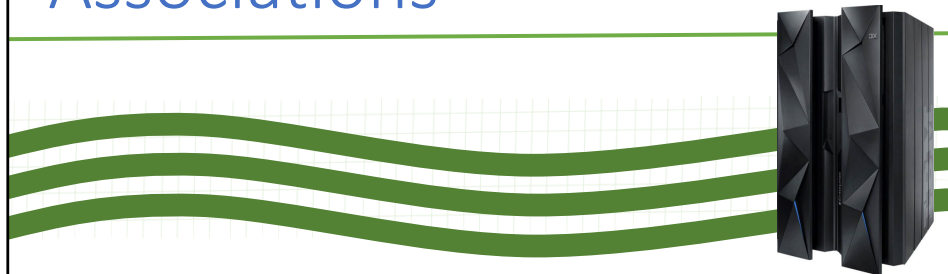
```
IPSEC LOGENable
; Rule SrcAddr DstAddr Logging Protocol SrcPort DestPort Routing Secclass
; OSPF protocol used by Omproute
IPSECRule * * NOLOG PROTO OSPF
; IGMP protocol used by Omproute
IPSECRule * * NOLOG PROTO 2
; DNS queries to UDP port 53
IPSECRule * * NOLOG PROTO UDP SRCPort * DESTport 53
; Administrative access
IPSECRule * 9.1.1.2 LOG SECCLASS 100
; Network security services (NSS) server access to the NSS client
IPSECRule * * LOG TCP SRCPort 4159 DESTport *
; Network security services (NSS) server access to the NSS client
IPSEC6Rule * * LOG TCP SRCPort 4159 DESTport *
ENDIPSEC
```

NSSD Display Commands

- Display NSS IPsec client connection information.
 - `ipsec -x display`
- Display connected NSS IPsec clients.
 - `nssctl -d`
- Filter ipsec command by client.
 - `ipsec -y display -z client4`

See the IP System Administrator's Commands manual.

Sysplex Wide Security Associations

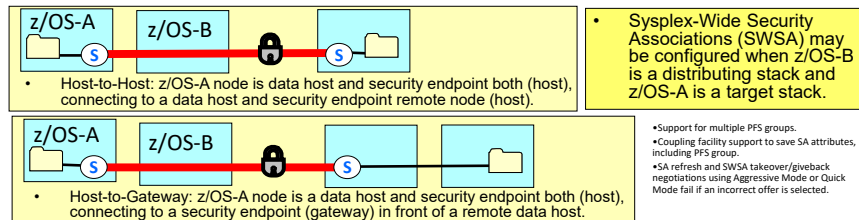


010_ZCS301_IPsec

© Copyright IBM Corporation 2023

Page 45

IPsec for Distributed DVIPA (Sysplex Distributor)



- Existing connections to target systems will be preserved if distributing stack fails and backup stack takes over distribution.
 - When a DVIPA is moved during DVIPA takeover (planned or unplanned), SWSA automatically reestablishes new IPsec SAs with the same security service characteristics as the SAs that existed on the host that previously owned the DVIPA. The SA reestablishment is transparent to the client that owns the other end of the SA. That is, the SA reestablishment looks like a normal SA refresh.
- The distributing stack(s) require: IPCONFIG IPSECURITY, IKED, IPCONFIG IPSECURITY, the filter/VPN rules for the IPsec traffic to/from the distributed DVIPA.
- Because the outbound traffic does not necessarily pass through the distributing stack SWSA has additional requirements on the target stacks: identical filter/VPN rules for the IPsec traffic on the target stacks as what is configured on the distributing stack(s). Stacks that are only distributed DVIPA targets do not need IKE, key exchange rules, or DVIPSEC.
- SWSA also requires the use of a coupling facility structure with a name in the form EZBDVIPAvvt, where vv is the 2-digit VTAM group ID suffix specified on the XCFGRPID start option, and tt is the TCP group ID suffix specified on the GLOBALCONFIG statement in the TCP/IP profile.

010_ZCS301_IPsec

© Copyright IBM Corporation 2023

Page 46

Sysplex-Wide Security Associations (SWSA) may be configured when z/OS-B is a distributing stack and z/OS-A is a target stack. Existing connections to target systems will be preserved if distributing stack fails and backup stack takes over distribution. When a DVIPA is moved during DVIPA takeover (planned or unplanned), SWSA automatically reestablishes new IPsec SAs with the same security service characteristics as the SAs that existed on the host that previously owned the DVIPA. The SA reestablishment is transparent to the client that owns the other end of the SA. That is, the SA reestablishment looks like a normal SA refresh. The distributing stack(s) require: IPCONFIG IPSECURITY, IKED, , the filter VPN rules for the IPsec traffic to/from the distributed DVIPA. Because the outbound traffic does not necessarily pass through the distributing stack SWSA has additional requirements on the target stacks: identical filter VPN rules for the IPsec traffic on the target stacks as what is configured on the distributing stack(s). Stacks that are only distributed DVIPA targets do not need IKE, key exchange rules, or DVIPSEC. SWSA also requires the use of a coupling facility structure with a name in the form EZBDVIPAvvt, where vv is the 2-digit VTAM group ID suffix specified on the XCFGRPID start option, and tt is the TCP group ID suffix specified on the GLOBALCONFIG statement in the TCP/IP profile. Please consult the IP Configuration Guide for restrictions placed on SWSA when implemented with FIPS 140.

End of Topic



010_ZCS301_IPsec

© Copyright IBM Corporation 2023

Page 47

End of Topic



010_ZCS301_IPsec

© Copyright IBM Corporation 2023

Page 48