Deploying secure software configuration management

IBM Rational ClearCase security features overview

Steve Bertrand
December 22, 2015

IBM® Rational® ClearCase provides robust support for all four dimensions of computer security: authentication, authorization, encryption, and event-logging. In this overview of security features, learn about deploying software configuration management.

Collectively, computer security breaches have entailed enormous financial damages for the organizations that have been attacked. Although all organizations must protect their assets from attacks, the protective measures that they take vary in type and extent according to the information security requirements of each organization. Investments in computer security vary accordingly, but as the Gordon-Loeb Model proposes, every organization should invest about one-third of the value of the loss that would be incurred by a breach. Part of that investment must be allocated to ensure that the tools used to produce the assets are themselves secure.

Learn more. Develop more. Connect more.

The new developerWorks Premium membership program provides an all-access pass to powerful development tools and resources, including 500 top technical titles (dozens specifically for web developers) through Safari Books Online, deep discounts on premier developer events, video replays of recent O'Reilly conferences, and more. Sign up today.

ClearCase security

IBM® Rational® ClearCase® is a closed-source software configuration management (SCM) system that supports different security solutions for different requirements. It is an enterprise-level system that provides more rigorous security mechanisms than are offered by other SCM systems. When evaluating the security of SCM systems, consider:

- The risk inherent in open-source solutions: their vulnerabilities are discoverable by the public.
- Your security requirements: Implement the simplest effective measures. As the complexity of administration increases, so do the security risks entailed by mis-administration.
- Your organizational requirements: ClearCase scales comfortably to large, globally distributed enterprises, supporting both WAN- and replica-based distributed development. It provides security features that support both of these distribution models.

This article describes how ClearCase addresses the following aspects of computer security:
• Authentication: the verification of a security principal's identity
• Authorization: the granting of access to assets to an authenticated principal
• Encryption: the encoding of information such that it is usable only by an authorized principal
• Event logging: the creation of auditable records of the actions of principals on objects

**ClearCase authentication**

Figure 1 shows the two basic client types that ClearCase supports:

• The LAN-dependent client, referred to as the ClearCase Local Client (CCLC)
• The WAN-friendly client, referred to as the ClearCase Remote Client (CCRC)

This article characterizes the client types as *basic* to the extent that they are highly configurable as to workspace type (the ClearCase *view* type) and other parameters.

**Figure 1. Client authentication mechanisms**

ClearCase Local Client authentication

The OS identity of the ClearCase Local Client (CCLC) user, as well as that of the user’s computer, are registered within a security domain whose controller is dedicated to vouching for the identities of ClearCase principals. CCLC users initiate ClearCase operations by presenting their network identities (supplied by their OS login's) to ClearCase authorization mechanisms. The authenticated identities persist until users explicitly log out of the domain.
ClearCase Remote Client authentication

While CCLC users are assumed to be trusted principals, ClearCase Remote Client (CCRC) users are regarded as untrustworthy. Therefore, authentication is appropriately explicit (as well as more burdensome) than CCLC authentication. CCRC users can reside:

- Outside the domain
- On the LAN of which the domain is a part
- On a private WAN
- On the Internet

**Note:** A CCRC user on a LAN may be a member of the domain as well, but CCRC clients must be explicitly authenticated regardless of their users' domain credentials.

Although CCRC users on the LAN can be assumed to be more trustworthy than clients on the Internet, all clients that are not members of the domain are regarded as equally untrustworthy.

The identities of CCRC principals are first established as OS-level identities on the CCRC WAN server. This server resides within the domain and is authenticated by the domain controller. After the CCRC WAN server has verified the identities of CCRC users connected via http(s) using username/password authentication, users can initiate ClearCase operations through requests to the CCRC WAN server, which formulates the corresponding requests to ClearCase servers on their behalf. Unlike CCLC users, CCRC users must log into the CCRC WAN server for each session, with the session timing-out according to the server setting.

ClearCase MultiSite

With ClearCase MultiSite replication, you can configure bidirectional synchronization of ClearCase deployments that reside in distributed domains. Such a configuration constitutes multiple "hubs of trust" for development projects that require them. You can also configure unidirectional synchronization when the criterion of mutual trust among participating replicas does not apply: For example, the replica for a proprietary development project receives updates from the replica for an open-source project, but synchronizes none of its changes to the open-source replica.

Authorization

Non-access-controlled authorization

An authorization policy should first address access controls that are not dependent on an explicit authorization mechanism. Such controls greatly reduce the possibility of authorization violations because they preclude user access to sensitive operations and data.

ClearCase Remote Client

CCRC supports non-access-controlled authorization by virtue of its isolation from sensitive operations, as shown in Figure 2. Whereas CCLC supports all ClearCase functionality—about 270 operations encompassing development, administration, and build activities—CCRC supports about 50 operations, all of them confined to development activities except for a few session-management
commands. CCRC operations are mediated by the CCRC WAN server: CCRC users have no direct access to ClearCase servers. CCRC is the appropriate interface for users who don't need access to features that are only available in CCLC (for example, audited builds and administrative operations).

**Figure 2. ClearCase operations by client type and level of indirection to ClearCase servers**

![Diagram showing ClearCase operations by client type and level of indirection to ClearCase servers](image)

**Deploying clients on virtual machines**

When user isolation from SCM data is warranted, you can deploy ClearCase clients on virtual machines (VMs) behind a firewall. In this case, use a Remote-Desktop-like solution to access the VM. Thus, ClearCase view data does not reside on developer machines, but only on the firewall-protected VM.

**Access-controlled authorization**

A discussion of ClearCase authorization mechanisms needs to begin at the level of file system objects. ClearCase uses UNIX permissions (modes) on UNIX/Linux® platforms and UNIX-style permissions on Windows. User identifiers (UIDs) on UNIX/Linux and session IDs (SIDs) on Windows® can be assigned permissions individually and also organized into OS groups. The OS groups are named sets of UIDs/SIDs that are granted permissions assigned to the groups. An OS-authenticated principal can belong to multiple groups, through which its permissions accumulate and vary according to the protections on the object to which access is requested.

**Versioned object bases**

ClearCase versioned files and directories reside in a database called a versioned object base (VOB). A VOB is the primary repository for protected objects and most access control operations apply to VOBs and their resident objects. The VOB itself is protected by a built-in ClearCase identity that is referred to as the VOB owner and is the identity of the creator of the VOB. The VOB owner (or another highly privileged identity, such as root on UNIX or Linux) can edit VOB...
protections to change the owner and principal group, and add or remove secondary groups. The VOB owner can also deny a highly privileged user access to the server host if that user is not logged in locally.

**Elements**

ClearCase versioned objects are called *elements*. Each element has an owner (a UID or SID), a group, and a protection mode. A VOB can contain elements associated with multiple groups, by which authorization can be administered.

**Constraints on filesystem-level authorization administration**

Administration at the file system level is effective as long as the number of objects is not too large. It can become unmanageable as the scale of a deployment grows. Consider the case of a deployment where a number of VOBs containing many versioned objects serve collectively as the repository for the development of a product. An individual developer might need access to only a subset of the objects and so is named as a member of the few groups that provide that access. The builder, on the other hand, needs access to all objects and therefore requires membership in the many groups that permit comprehensive access.

However, administration at the file system level can entail some problems:

- A large number of groups can be difficult to administer, potentially leading to authorization violations occasioned by mis-administration.
- The number of groups to which a user needs to belong might exceed the limits supported by OS and ONC-RPC protocols.
- Unix-style permissions allow only a single group per element.

Access control lists (ACLs) can address these problems.

**Access control lists**

With VOB ClearCase ACLs, identities and protection modes are administered through policies and rolemaps. A policy defines the permissions that are granted to specified roles and a rolemap maps the roles to principals (groups and users).

Each entry in a policy lists a metatype object, followed by a list of principals and the principals' permissions:

```plaintext
[element]
Role:Manager read
Role:Developer change
Role:Integrator change
```

The rolemap specifies the principals that assume the roles listed in the policy:

```plaintext
Role:Manager --> Group:DOMAIN/mgrs
Role:Developer --> User:DOMAIN/savitri
Role:Integrator --> Group:DOMAIN/integs
```
By combining the rolemap and its policy, and substituting a mapping in the rolemap for a role in the policy entry, an effective ACL that specifies no indirect principals is computed:

<table>
<thead>
<tr>
<th>Group:DOMAIN/mgrs read</th>
</tr>
</thead>
<tbody>
<tr>
<td>User:DOMAIN/savitri change</td>
</tr>
<tr>
<td>Group:DOMAIN/integs change</td>
</tr>
</tbody>
</table>

**Example**

The following example illustrates this scenario:

- The core and security components of a product reside in VOB A
- Each of these components has its corresponding team of developers
- The GUI component resides in VOB B; a single user, Savitri, is responsible for GUI code
- The release engineer, Jo, builds the product

A single policy is established for both VOBs; it is created once in each VOB:

```plaintext
#Policy for VOBs A and B
[element]
Role:maintainer change
Role:consumer read
Role:builder read, write-dir
#write-dir allows view-private file creation (build output)
```

Three rolemaps correspond to the three components:

```plaintext
#Rolemap for core component
Role:maintainer --> Group:DOMAIN/core_dev
Role:consumer --> User:DOMAIN/savitri
Role:builder --> User:DOMAIN/jo

#Rolemap for security component
Role:maintainer --> Group:DOMAIN/security_dev
Role:builder --> User:DOMAIN/jo

#Rolemap for GUI component
Role:maintainer --> User:DOMAIN/savitri
Role:consumer --> Group:DOMAIN/core_dev
Role:builder --> User:DOMAIN/jo
```

The policy and rolemaps yield the following effective ACLs:

```plaintext
#EACLs for core component
Group:DOMAIN/core_dev change
User:DOMAIN/savitri read
User:DOMAIN/jo read, write-dir

#EACLs for security component
Group:DOMAIN/security_dev change
User:DOMAIN/jo read, write-dir

#EACLs for gui component:
User:DOMAIN/savitri change
Group:DOMAIN/core_dev read
User:DOMAIN/jo read, write-dir
```
Identity and permissions administration of VOB replicas

A ClearCase VOB replica can update other replicas not only with changes to versioned objects, but also with information about the identities and permissions attached to those objects. Replicas can be configured to preserve:

- Both identities and permissions; these are referred to **fully-preserving replicas**
- Permissions only
- Neither identities nor permissions

Ideally, fully-preserving replicas (which share ACLs) are deployed in environments where information security is an important consideration. At such replicas, policies and rolemaps can be administered by a single trusted principal (the administrator) rather than independently at each non-preserving replica by its administrator. A disadvantage of independent replica administration is that it requires some coordination, failures of which can result in authorization violations. As a further security measure, the administrator of a fully preserving replica can set ACLs that deny requests for mastership of rolemaps and policies from other replicas.

Encryption

There are three aspects to encryption in the context of distributed development:

- Data in motion within a domain
- Data in motion over a WAN
- Data at rest

Because CCLC is designed to be deployed in a trusted domain, encryption of data in motion within the domain is not employed. But assume the unlikely scenario of a CCLC user attempting an attack on data to which the user's access is denied. The attacker needs to map the object ID (OID) representation of NFS/SMB pathnames to human-readable pathnames. That is possible, but then there's the needle-in-the-haystack problem of voluble RPC chatter to contend with: Which data are valuable to the attacker?

Data in motion over a WAN presents a significant risk due to the presence of untrusted principals. ClearCase employs the Secure Sockets (SSL/TLS) cryptographic protocol for these communications.

Stored, unencrypted data presents a risk that should be countered when appropriate. ClearCase view and VOB storage can be encrypted with whole-disk or whole-partition encryption schemes such as BitLocker, PGP Desktop, and LUKS. Storage devices in the domain should be housed in secure facilities.

Event logging

Event logs address two aspects of security:

- Audits of event logs enable the identities of violators to be determined
• Awareness of event logging can be an effective deterrent to attacks

ClearCase logs about 80 types of events affecting approximately 20 kinds of object, including the types of which the objects are instances. The output of the ClearCase history function can be input to a script that scans for unusual patterns of activities. A ClearCase administrator can also consult the ClearCase access_log, which is dedicated to recording failed access attempts.

When comparing SCM systems, evaluate objects whose histories are not maintained and event types that are not logged. Depending on your security requirements, such omissions can entail risks.

Managing threats

Firewalls, anti-malware, the secure housing of devices, and employee-access badges are assumed to be the minimum outer layer of protection for any organization whose information is at risk. Although all of the protective mechanisms described in this article can be specified by an organization’s security policy, for certain assets of incalculable value—such as those held by an intelligence agency—they are obviously insufficient. An information security analyst can recommend the appropriate protection against the risks to your information assets.

Acknowledgments

Many thanks to my IBM colleagues Peter Hack, ClearCase Architect, and John Kohl, Security Architect, for their conscientious reviews of this article (through all eight drafts!). Thanks also to Robin Wood at developerWorks for her suggestions and help publishing this article.
Related topics

- For more information about Rational ClearCase:
  - ClearCase product pages
  - Support portal
  - Knowledge center
- Download a free trial version of IBM Rational software.
- Try IBM Cloud for free. Build your next app quickly and easily.

© Copyright IBM Corporation 2015
Trademarks
(www.ibm.com/developerworks/ibm/trademarks/)