IBM Education Assistance for z/OS V2R2

Item: UNIX Search Authority
Element/Component: RACF

Material current as of May 2015

Agenda

- Trademarks
- Presentation Objectives
- Overview
- Usage & Invocation
- Migration & Coexistence Considerations
- Presentation Summary
- Appendix

Trademarks

- See url http://www.ibm.com/legal/copytrade.shtml for a list of trademarks.

Presentation Objectives

- This item introduces two new controls over z/OS UNIX System Services authorization. Both are implemented in the ck_access callable service (IRRSKA00).
  -
  - Allow directory search (DIRSRCH)
  -
  - Deny file execution (FSEXEC)

Overview – Directory Search

Problem Statement / Need Addressed

- To make best use of SUPERUSER.FILESYS.CHANGEPERMS and CHOWN to delegate UNIX security administration, it is necessary to grant READ and SEARCH to all directories or grant a higher-than-desired authority such as AUDITOR or SUPERUSER.FILESYS

- Solution
  - Define a new UNIXPRIV resource to control read/search access to all directories.

- Benefit / Value
  - Provides a more granular mechanism to delegate UNIX security administration, avoiding over-authorization.

Usage & Invocation – Directory Search

- Define a new UNIXPRIV profile SUPERUSER.FILESYS.DIRSRCH
  - READ (or higher) access grants user read and search permission to UNIX directories
  - Generics allowed
- Example:
-

```
RDEFINE UNIXPRIV SUPERUSER.FILESYS.DIRSRCH UACC(NONE)

PERMIT SUPERUSER.FILESYS.DIRSRCH CLASS(UNIXPRIV)
      ID(appropriate-groups-and-users) ACCESS(READ)

SETROPTS RACLIST(UNIXPRIV) REFRESH
```

- DIRSRCH authority does NOT grant read, write, or execute permission to ordinary UNIX files.
- DIRSRCH authority does NOT grant write permission to UNIX directories.

Overview – File Execution

Problem Statement / Need Addressed

- Need to prevent the execution of all files in a file system, similar to a 'NOEXEC' mount option. Recommended for directories like /tmp, where any user can write files.

Solution

- Define RACF profile(s) in the new FSEXEC class the denies file execute access to the specific file system(s).

Benefit / Value

- Provides a RACF control over file execution, complementary to mounting the file system with 'SETUID NO'.
- Provides straight-forward compliance/audit verification.

Usage & Invocation – File Execution

- Define a profile in the new FSEXEC class.
  - Profile name must match the FILESYSTEM name specified on the MOUNT statement.
  - Profile name is case sensitive. Generic names are allowed.
  - Update (or higher) access makes the user eligible for file execution, subject to other access checks.
- Example:
-

```
RDEFINE FSEXEC /tmp UACC(NONE)
      or
RDEFINE FSEXEC OMVS.ZFS.ADMIN.** UACC(NONE)

PERMIT OMVS.ZFS.ADMIN.** CLASS(FSEXEC) ID(USER019 GROUPADM)
ACCESS(UPDATE)

SETROPTS CLASSACT(FSEXEC) RACLIST(FSEXEC)
```

- Superuser or auditor privilege does not override FSEXEC denial of access.
- On denial, ICH408I message includes 'ACCESS ALLOWED (FSEXEC ---)'.
- FSEXEC is supported for ZFS and TFS type file systems.
- FSEXEC does not apply to file systems mounted with the '-s nosecurity' option.

Migration & Coexistence Considerations

- None. Lower-level systems sharing the RACF database will not look for DIRSRCH or FSEXEC profiles.

Presentation Summary

- UNIX Search Authority can reduce the number of administrators requiring superuser or auditor authorization.

- FSEXEC can lessen the risk of malicious or unauthorized code execution.

Appendix

- *z/OS Security Server RACF Security Administrator's Guide* (SA23-2289)