CA Top Secret® for z/OS

Audit Guide

r15

technologies

This Documentation, which includes embedded help systems and electronically distributed materials (hereinafter referred to as the “Documentation”), is for your informational purposes only and is subject to change or withdrawal by CA at any time. This Documentation is proprietary information of CA and may not be copied, transferred, reproduced, disclosed, modified or duplicated, in whole or in part, without the prior written consent of CA.

If you are a licensed user of the software product(s) addressed in the Documentation, you may print or otherwise make available a reasonable number of copies of the Documentation for internal use by you and your employees in connection with that software, provided that all CA copyright notices and legends are affixed to each reproduced copy.

The right to print or otherwise make available copies of the Documentation is limited to the period during which the applicable license for such software remains in full force and effect. Should the license terminate for any reason, it is your responsibility to certify in writing to CA that all copies and partial copies of the Documentation have been returned to CA or destroyed.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CA PROVIDES THIS DOCUMENTATION “AS IS” WITHOUT WARRANTY OF ANY KIND, INCLUDING WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT. IN NO EVENT WILL CA BE LIABLE TO YOU OR ANY THIRD PARTY FOR ANY LOSS OR DAMAGE, DIRECT OR INDIRECT, FROM THE USE OF THIS DOCUMENTATION, INCLUDING WITHOUT LIMITATION, LOST PROFITS, LOST INVESTMENT, BUSINESS INTERRUPTION, GOODWILL, OR LOST DATA, EVEN IF CA IS EXPRESSLY ADVISED IN ADVANCE OF THE POSSIBILITY OF SUCH LOSS OR DAMAGE.

The use of any software product referenced in the Documentation is governed by the applicable license agreement and such license agreement is not modified in any way by the terms of this notice.

The manufacturer of this Documentation is CA.

Provided with “Restricted Rights.” Use, duplication or disclosure by the United States Government is subject to the restrictions set forth in FAR Sections 12.212, 52.227-14, and 52.227-19(c)(1) - (2) and DFARS Section 252.227-7014(b)(3), as applicable, or their successors.

CA Technologies Product References

This documentation set references the following CA products:

- CA ACF2™ for z/OS (CA ACF2)
- CA Common Services for z/OS (CA Common Services)
- CA Distributed Security Integration Server for z/OS (CA DSI Server)
- CA LDAP Server for z/OS (CA LDAP Server)
- CA Top Secret® for z/OS (CA Top Secret)

Contact CA Technologies

Contact CA Support

For your convenience, CA Technologies provides one site where you can access the information that you need for your Home Office, Small Business, and Enterprise CA Technologies products. At http://ca.com/support, you can access the following resources:

- Online and telephone contact information for technical assistance and customer services
- Information about user communities and forums
- Product and documentation downloads
- CA Support policies and guidelines
- Other helpful resources appropriate for your product

Providing Feedback About Product Documentation

If you have comments or questions about CA Technologies product documentation, you can send a message to techpubs@ca.com.

To provide feedback about CA Technologies product documentation, complete our short customer survey which is available on the CA Support website at http://ca.com/docs.

Contents

Chapter 1: Issues for the Auditor 7

Chapter 2: Misuse of CA Top Secret 11

Chapter 3: Auditing, Reporting, and Surveillance 21

Chapter 4: Threats and Exposures 29

Chapter 1: Issues for the Auditor

This section contains the following topics:

Introduction

This guide provides the auditor with the information to maximize the use of CA Top Secret features. This guide identifies:

- Some of the exposures and security deficiencies within IBM z/OS and other vendor products
- Problems that are associated with people as a security risk
- The corrective actions that the author perform

Security administration and auditing are complementary functions. They must be closely coordinated to be effective. Mutual cooperation between these two departments is important because no one department has total jurisdiction over corporate data security.

Implementation Queries

Raise the following questions before and during an implementation:

- Are the data security exposures of the z/OS operating system critical applications and support personnel both evaluated and understood?
- Is the corporate data security policy or the upper management position regarding data security complete and adequate? Does the policy address all of the logical security exposures like:
  - Operating system?
  - Applications development?
  - Personnel responsibilities?
  - Password management and control?
  - Resource ownership?
  - Resource administration?

  - Delegation of responsibility?
  - Accountability?
- Is a security committee required? If it is, does it consist of members from:
  - Security administration?
  - Auditing?
  - System programming?
  - Application development?
  - General user community at large?
  - Operations?
- Is administration to be centralized or decentralized? If decentralized to what extent: divisional, departmental, or user?
- Has a general plan of attack for implementation been established and approved?
  - WARN mode first then escalation to FAIL mode?
  - WARN mode with special categories of users or groups in FAIL mode?
  - IMPL mode then FAIL mode?
  - When is CICS IMS implemented?
  - Objectives and dates established?
- How are users to be identified for batch jobs? Are JCL changes required?
- Have the procedures for access authorization changes been established including written authorization forms?
- Is there an established policy regarding data access for:
  - Application programmers and production data?
  - Operations and production data?
  - Systems and the overall maintenance of the system?
- Has a policy been established for emergency access to data?
- Has a list of critical resources been established? Do the accesses to these critical resources include stringent controls?
- Are CA Top Secret files adequately protected? Who is allowed to access them?
- Has a chart of group departmental and divisional access to resources been established allowing the auditor to easily verify scope and boundaries?

- Are the password controls adequate? Some additional auditor considerations are:
  - The minimum password length
  - Variable expiration periods by function sensitivity
  - Expiration warning message interval
  - Password violation threshold
  - Members of a restricted password list
  - Password masking
  - Who can list passwords
  - Penalties for disclosure
- Have the procedures for incident reporting been established?
  - Are the reports properly produced distributed and reviewed?
  - Are logging options for violations and auditors correct?
  - Is there an audit trail of changes to critical data components?
- Is online tracking to be used and by whom?
- What are the violation investigation procedures? Are violations acted upon immediately?
- Will a user be held accountable for their actions?
- Are any CA Top Secret exits used and if so for what purpose?
- Is the security officer reviewing batch reports daily or is online tracking being used? Are the procedures effective?
- Has a policy been established for external users through JES remote job entry stations or online dial-up facilities?
- Have default ACIDs been established?
- Do adequate procedures exist for backing up the Security File?
- How many auditors are required at central division and department levels?
- Have all policies been communicated to and acknowledged by all appropriate personnel (like legal action considerations and end user training)?
- Are recovery procedures tested and documented?
- Is the auditory group involved in development of a proper application benchmark?

Chapter 2: Misuse of CA Top Secret

This section contains the following topics:

Control Options

Control options let selected operators and administrators specify how CA Top Secret controls security. Control options:

- Determine the security MODE of operation
- Determine how CA Top Secret processes typically, and how it processes under specific security MODES and circumstances
- Indicate what features or products are on the operating system
- Indicate how CA Top Secret handles individual facilities
- Specify password selection rules and violation thresholds
- Issue the commands that force CA Top Secret to reset after shutdown or reinitialize after installation of new CA Top Secret maintenance

Depending on your environment, the following control options with the specified operands can cause security breaches:

AUTOERASE(NO)

Does not erase all residual information on the DASD volume.

AUTH

Changes the authorization algorithm and might impact access (which can be granted or denied).

BACKUP(OFF)

Discontinues automatic backup of the Security File. Backup is also unavailable if the BACKUP DD statement is missing from the CA Top Secret started task procedure.

BYPASS

Allows selected or all jobs/users to bypass security; only use in an emergency.

DOWN

Affects security processing if CA Top Secret becomes inactive.

DRC(nnn,NOVIOL)

Indicates a violation but does not treat the event as a fatal violation. It flags the event but does not FAIL the user.

DUMP

Displays CA Top Secret data areas.

EXIT(OFF)

Deactivates the installation exit.

FACILITY

Controls separate facilities and also displays status. The FACILITY suboptions are:

MODE(mode)

Sets the mode.

LOG(NONE)

Deactivates logging. Violations are always logged in FAIL mode.

INACT

Prohibits initiation/signon.

SIGN(M)

Allows multiple logons with the same ACID for the specified facility.

LOCKTIME = 0

Deactivates terminal locking.

DEFACID(acid)

Controls default ACID assignment.

NOWARNPW

Password violations are not fatal in WARN mode (except for administrators).

NOAUDIT

Deactivates facility-wide auditing.

HPBPW

Allows expired or changed passwords to be used for limited time in batch.

INACTIVE

Sets a threshold for how long an ACID connected to an expired password can be used before it is suspended.

INSTDATA(0)

Resets global site installation data area to zero in CSA.

JES(NOVERIFY)

Indicates that the JES Early Password Verification feature is not in effect (USER and PASSWORD are required on the jobcard).

LOG(NONE)

Deactivates extra SMF and Audit/Tracking File logging (violations and audited events are always written to the Audit/Tracking File).

LOG(SEC9)

Routes violation messages to the security console using route code 9.

LOG(MSG)

Displays violation messages for batch jobs, started tasks, or online.

MLACTIVE(NO)

Deactivates Multilevel Security (MLS).

MLFSOBJ(NO)

Deactivates the requirement for security labels for UNIX directories and files.

MLMODE

Changes MLS security mode and can lessen or destroy security.

MLNAME(NO)

Allows user to view data set names that were hidden from them.

MLSLBLRQ(NO)

Specifies that security labels are not required for all users, data sets, and resources in an MLS environment.

MLSPCOBJ(NO)

Deactivates the requirement for security labels for IPC objects.

MLWRITE(YES)

Allows the write-down of data in an MLS environment.

MODE

Changes mode globally and can lessen or destroy security.

MSUSPEND(NO)

Allows a user to make an unlimited number of guesses to determine the MSCA password.

NEWPW(NO)

Deactivates most new password rules, except the MIN= and MINDAYS= suboptions.

OPTIONS(NO)

Allows user to indicate which APARS apply from previous releases of CA Top Secret.

PDSPROT(OFF)

Disables the PDS member level protection for all data sets.

PTHRESH(0)

Allows unlimited access attempts at guessing user passwords.

RECOVER(OFF)

Deactivates recording of changes to the Recovery File. If the RECFILE DD statement is missing recovery is not in effect.

RPW(RESET)

Removes all password prefixes currently in the restricted password list if NEWPW(RS) is in effect.

TAPE(OFF)

Deactivates built-in tape security. Only specify when using external tape management packages such as CA-1®.

TEMPDS(NO)

Indicates that temporary data sets are not protected and cannot be audited.

TIMER

Controls frequency at which logging buffers are examined and data written to the Audit/Tracking File. If the frequency is too high, data might be lost.

VTHRESH(0)

Deactivates violation threshold controls.

Security Definitions

The attributes that you can apply to a single ACID are:

CONSOLE

Allows the ability to change control options.

DUFXTR, DUFUPD

Allows reading and writing of installation data.

NOADSP

New data sets are unsecured in a non-Alwayscall environment.

NODSNCHK

Allows access or use of any data set.

NOVOLCHK

Allows access or use of any volume.

NOLCFCHK

Allows use of any command, program, or transaction.

NORESCHK

Allows use of any terminal, program, CICS, IMS, CA IDMS, or user resource.

NOSUBCHK

Allows the jobs to be submitted with any ACID.

Any resource access that is allowed as a result of one of the NO*xxx*CHK Bypass attributes is logged as a bypass event. TSSUTIL and TSSTRACK show these events as OK+B.

Examples: resource access

This example allows use of any data set at the designated access level:

```
DSNAME(******) ACCESS(nnnnn)
```

This example allows use of any data set on VOLUME(x) in any manner:

```
VOLUME(x) ACCESS(ALL)
```

Permissions

The following list details permissions:

ACTION(NODSN)

Indicates that all data set restrictions are bypassed. The permitted volume access level controls the minimum and maximum access levels.

ACTION(EXIT)

Specifies to pass control to the installation exit (TSSINSTX) for all accesses to the resource granted by this permission. This option is valid for data sets and volumes.

Undefined Data Sets

In FAIL mode, CA Top Secret protects all undefined data sets, but only if z/OS requests security validation. In Alwayscall environments (OS/390/SP with DFP 1.1, or OS/390/XA with DFP 1.2, and all ICF catalogs), z/os calls CA Top Secret to validate access requests to data sets. In all other z/OS environments, CA Top Secret is called only if the RACF bit is set in the VTOC, or catalog entry describing the data set. The state of this bit for any or all data sets can be interrogated with the CA Top Secret utility TSSPROT with the SIM option. All data sets created under CA Top Secret have the RACF bit turned on to ensure that security is always called.

In IMPL mode, access to undefined data sets is controlled with the DEFPROT attribute in the DATASET RDT entry. With DEFPROT set, undefined data sets are treated as in FAIL mode. If NODEFPROT is set, undefined data sets are not protected. In either case, CA Top Secret is only called to validate the request as previously described.

PERMIT(ALL)

Improper or accidental use of TSS PERMIT(ALL) could give the wrong access to all system users. Any user with RESOURCE(XAUTH) administrative authority can permit all users to access resources within their scope.

Started Task (STC) Definitions

Started tasks that have the bypass privilege are a potential threat. A TSS LIST(STC) command shows these as BYPASS.

CA Top Secret STC Procedure

As CA Top Secret is activated with an O/S START command:

- Start CA Top Secret automatically at IPL, no matter which IPL parameters or IPL volumes are used
- Ensure the CA Top Secret procedure contains the Audit/Tracking, Recovery, and Parameter Files. The Backup File is also recommended.

Note: The BACKUP control option is available only if the BACKUP DD statement is entered in the CA Top Secret started task procedure.

CA Top Secret Installation Exit

CA recommends a periodic scrutiny of the code in the CA Top Secret exit (TSSINSTX).

Tampering

CA Top Secret periodically (every 30 seconds by default) checks the integrity of critical tables and actual machine instructions in its own programs. If detected during the period in which these critical areas remain altered, non-deletable messages appear on the O/S master console.

CA Top Secret tables and Security Records are found primarily in Key 3 storage; minimizing accidental storage overlays.

Tampering includes changing the code in the system program and CA Top Secret exits and dynamically adding user SVCs. To succeed at tampering, a programmer (or user) must be in privileged state or have made physical alterations using the system console (alter/display frame). Operations must be allowed to exercise some control over threatening situations.

Operations have an ACID with CONSOLE authority to enable them to issue CA Top Secret control options using the O/S MODIFY command for the following situations:

- For suspected penetration attempts, enter:

  ```
  F TSS,FACILITY(fac=INACTIVE)
  ```

  The facility (such as TSO) is deactivated preventing access by all ACIDs except the MSCA.

- For suspected subversive activity, enter:

  ```
  F TSS,FACILITY(fac=AUDIT)
  ```

  All activity by users of a given facility is audited.

Subversion

CA Top Secret controls prevent many forms of subversion, including preventing penetration by *password and ACID-guessing programs* on personal computers.

Users can be fooled into disclosing their passwords. One ploy uses a program that simulates a VTAM/TCAM solicitation screen that accepts ACID and passwords, stores them in a data set, then informs the user that the system is down. This is accomplished without the program having to become privileged and by using standard TSO.

This exposure is more evident in environments where terminals are shared among several users. The most effective means of minimizing this type of subversion is to restrict users to specific terminals using SOURCE restrictions.

Securing Remote CICS Region Signon

Failure to monitor and regulate the access that is permitted to and by a remote terminal can leave your system open to a serious security breach. When CA Top Secret is used to secure CICS Multiregion Operation (MRO) and Intersystem Communication (ISC) environments, three security levels can be defined:

Bind Time

Used to prevent unauthorized remote regions from accessing your CICS region. With Bind Time security, a check is made when a request to establish a session is received or sent to a remote region.

Link

Used to limit the access of a specified remote region to your resources. Link security is active once the session between regions is bound. When the session is broken, Link security is deactivated.

Attach-Time

Used to allow incoming requests to attach to requested transactions. The session must be established. Additional degrees of Attach-Time security are:

Local

Set if CA Top Secret is not securing the remote region; the default.

Identify

Set if CA Top Secret is securing the remote region.

Verify

Set if CA Top Secret is securing the destination region (in an ISC environment). Verify does not apply to MRO.

For CICS release 3.2.1 and above:

Persistent

Set if CA Top Secret is securing the destination region (LU6.2APPC only).

Mixidpe

Set if CA Top Secret is securing the destination region (LU6.2APPC only).

When using Bind Time, Link, and Attach-Time, certain parameters must be set in the Resource Definition Online (RDO) or the Resource Definition Macro (RDM).

If you are using the RDO, for:

- Bind Time—Set the SECURITYNAME parameter the same as the CA Top Secret region control ACID definition for the remote region.
- Link—Code the SECURITYNAME parameters the same as the CA Top Secret ACID for the remote region. Do not set the OPERSECURITY and OPERSSL parameters; use the default values instead.
- Attach-Time—Set the ATTACHSEC parameters on.

If you are using RDM, for:

- Bind Time—Set the XSNAME parameter the same as the CA Top Secret region control ACID for the remote region.
- Link—Code the XSNAME parameter the same as the CA Top Secret ACID for the remote region. Do not set the OPERSEC or OPERRSL parameters in the RDM DFHTCT Type= SYSTEM macro; use the defaults instead.
- Attach-Time—Set the USERSEC parameter on.

We recommend that you specify the NODSNCHK, NORESCHK, and NOLCFCHK attributes. If these attributes are not specified for the region control ACID, every resource (OTRAN) or LCF-protected transaction ID would have to be permitted to the region control ACID used to sign on the receive terminal.

Chapter 3: Auditing, Reporting, and Surveillance

This section contains the following topics:

Characteristic Auditing Authorities

Auditors must be given specific administrative authorities that relate to the scope of their job.

Job functions and type of administrative authority define the auditor role. Multiple auditors with different areas of responsibility can be defined; for example, central auditor and divisional auditor.

The ACID type that was assigned when the ACID was created defines the auditor's scope. For example, a central auditor would be defined as an SCA, while a divisional-level auditor would be defined as a VCA. The CA Top Secret actions that an auditor is authorized to initiate are a function of the administrative authorities assigned. The administrative authorities that are characteristically given to auditors are:

- ACID (AUDIT, INFO, REPORT)
- RESOURCE (AUDIT, INFO, REPORT)
- DATA (ALL, PROFILE, PASSWORD)
- MISC1 (TSSSIM)
- MISC5 (MLSADMIN)
- MISC8 (LISTRDT, LISTSTC, LISTSDT)
- MISC9 (GENERIC)

Example: Establishing a central auditor

This example gives the attributes and authorities that are required to define someone to CA Top Secret to perform the functions of central auditor. The authority that is established allows:

- CTLADT to audit resources and users
- The auditor to use TSSCFILE, TSSCHART, TSSUTIL, TSSTRACK, TSSAUDIT, and TSSSIM to perform inquiries about all resources and ACIDs
- CTLADT to list any information from any Security Record except the password.

Since CTLADT is an SCA, the scope is the entire installation.

```
TSS CREATE(CTLADT) NAME('CENTRAL AUDITOR')
                   TYPE(SCA)
                   PASSWORD(password)
                   FACILITY(TSO,BATCH)

TSS ADMIN(CTLADT) RESOURCE(AUDIT,REPORT,INFO)
                  MISC9(GENERIC)
                  ACID(AUDIT,REPORT,INFO)
                  MISC1(TSSSIM)
                  DATA(ALL,PROFILE)
                  MISC8(LISTRDT,LISTSTC,LISTSDT)
```

Users Audit

Adding the AUDIT attribute to a user ACID causes all security-related activity by that user to be logged. When AUDIT is added to a common profile, all users that are connected to that profile are audited. For example:

```
TSS ADDTO(PROF10) AUDIT
```

All Users in a Facility Audit

To audit an entire facility, use the FACILITY control option. Place it in the Parameter File or use the O/S MODIFY command. For example:

```
FACILITY(TSO=AUDIT)
```

Resources Audit

Any resource can be audited. For example:

- Data sets
- Volumes
- Programs
- Terminals
- Abstracts
- Applications
- UR1, UR2
- CICS resources (FCT, DCT, JCT, PPT, TST)
- IMS resources (PSB, AGN, DBD)
- CA-IDMS resources (SUBSCHEMA, AREA)

All access attempts defined to the AUDIT record for the resource are recorded in the Audit/Tracking File or the SMF.

To audit any accesses, enter:

```
TSS ADDTO(AUDIT) resource(resource-name)
```

To specify the access or accesses to be audited for a resource, enter:

```
TSS ADDTO(AUDIT) resource(resource-name)
                 ACCESS(level1, level2, …)
```

Any resource, whether defined or undefined, can be audited. The specific resources, or all resources matching a generic prefix, can also be audited.

Notes:

- If access is not specified, ACCESS(ALL) is assumed
- ACCESS(NONE) is ignored
- A resource or prefix that is defined to the AUDIT record cannot exceed 64 characters

Example: Resource audit

This example audit use of production payroll data sets beginning with the high level qualifier PAYPROD:

```
TSS ADDTO(AUDIT) DSNAME(PAYPROD)
```

Logging Options

Correctly specifying logging options is an important prerequisite to reporting and tracking. If logging options are incorrect, it could be difficult to obtain representative audit trails.

Each facility can be separately monitored. In addition, MSG or SEC9 must be in effect to produce messages when violations occur. Password violation messages are always produced.

Record Permissions

The ADMINBY control option allows sites to record and trace accountability for any permissions added to a user. The ADMINBY control option records any authorized resource permissions and facilities added to a user within the Security File. When enabled, all new ADDS to a facility and authorizations of a resource to user ACIDS record:

- Administrator name
- SMFID of the system on which they issued the command
- Date/time the command was issued.

Audit Utilities

CA Top Secret provides several utilities to assist auditors in performing their functional responsibilities:

TSSTRACK

Use this utility to monitor security-related events from an online terminal in a real-time manner. TSSTRACK can go back to a specified date and time to focus on a selected facility or CPU, or to focus on violations only. Use this utility from both display and nondisplay terminals. Its standard version can be run under CICS and TSO. All CPUs can be monitored from a single terminal if you are using a shared Audit/Tracking File. All displayed information is obtained from the CA Top Secret Audit/Tracking.

TSSUTIL

A flexible report generator/extract utility is used to provide batch reports of any security-related events that have been logged to the Audit/Tracking File or SMF. Multiple and varied reports can be produced which monitor all types of security events with selection criteria that include:

- ACIDs
- Jobs

- Specific resources
- Resource types
- Facilities
- Departments
- Dates
- Types of access
- CPUs
- Violations
- Audited incidents

TSSAUDIT

This batch utility allows auditors to monitor changes that are made to the security file and sensitive z/OS facilities and data areas. It can be used to list:

- ACIDs which possess administrative or special privileges (such as AUDIT, CONSOLE, or any of the security bypass attributes).
- Changes that are made to the Security File. It generates this information for a given date or time span by examining the Recovery File. All changes by a particular ACID can also be requested. The ACID must fall within the scope of the administrator running TSSAUDIT.
- Information about modules in APF-authorized libraries.
- Information about site-written (non-IBM) SVCs, the Program Properties Table (PPT), and the Terminal Monitor Program's (TMP) authorized program lists.
- The last two items are useful in pinpointing z/OS security exposure.

TSSCHART

This utility lets you generate the ACIDs and owned resource relationships within the CA Top Secret security database in the form of an organization chart. The auditor can generate these charts at the zone, division, department, profile, or user level. This utility allows auditors to review the Security database to ascertain that it has been designed effective.

TSSSIM

This utility enables the auditor to simulate access attempts to resources to test and verify resource permissions. As a result, it can aid an auditor in deciding whether the ACID has access to particular resources. The auditor can simulate any mode or conditions for ACIDs within their scope of authority. TSSSIM establishes and verifies access characteristics. In addition, TSSSIM indicates which specific CA Top Secret permission (the TSS PERMIT command function) gave or denied access authorization.

TSSCFILE

This batch utility produces a fixed-format output file whose records closely parallel the output of a TSS LIST command function. A four- to six-character record identifier is associated with each record type. The auditor can then generate custom reports using TSSCFILE.

TSSPROT

This utility is useful for auditors in a z/OS non-Always call environment to determine what data sets are not protected. To obtain this information, the auditor must use the following options:

```
PROTECT SIM
```

Note: To execute TSSPROT, the auditor must be at least an SCA.

TSSOERPT

This batch utility enables the auditor to monitor user activity in the OpenEdition environment. The Various reports can be produced that monitor access to OpenEdition related, SAF callable services. Report data is extracted from SMF TYPE 231 records.

For information about these utilities, see the *Report and Tracking Guid*e.

CPF Journal Files

The Command Propagation Facility (CPF) uses Journal Files to provide a historical record of the commands sent to and from CA Top Secret. An individual Journal File is usually a JES spool data set. If the proper software is available, a SYSOUT data set which can be printed off or viewed online. CPF allocates one Journal File for each remote node that is defined to it through the NODES control option. CPF also allocates one Journal File for all incoming traffic. NODES specifies the places that CPF can send to, but does not affect from where it can receive.

By examining the appropriate Journal File, an auditor can see exactly what came in, what went out, and the results of the action taken.

The following sample reports demonstrate the Journal Files information.

Example: journal on sending machine

```
TSS9811I ***** CPF SUBTASK INITIALIZED FOR NODE *****
TO: NYC00   ID: 000000001
TSS LIST(USRJOE) DATA(ALL) TARGET(NYC00)
FR: NYC00   ID: 000000001
ACCESSORID = USRJOE   NAME       = JOE PAZ  TYPE       = CENTRAL  FACILITY   = BATCH
FACILITY   = STC  FACILITY   = TSO  FACILIT
Y  = TSR  FACILITY  = CICSPROD  FACILITY  = IMSPROD  FACILITY  = VM  FACILITY  = RPGFAC
FACILITY   = RDFFAC  FACILITY   = ROST
EST  CREATED   = 01/20/88  LAST MOD  = 10/10/90  16:   PROFILES   = TDGPROF   TCSPROF
ATTRIBUTES = CONSOLE,DUFXTR  BYPASSIN
NODSNCHK,NOVOLCHK,NOSUBCHK? LAST USED = 10/10/90 16:09 CPU(XE05) FAC(TSO    )
COUNT(01078)  PHYSKEY  = 2356668  VMMDISK  = PAZ
J001
TO: NYC00   ID:?000000002
TSS ADD(USRJOE) TSOPROC($USRJOE) TARGET(N*)
FR: NYC00   ID: 000000002
TSS0351E  SPECIFY "UNDERCUT" TO TRANSFER OWNERSHIP  TSS0301I  ADD    FUNCTION FAILED,
RETURN CODE =  8
TO: NYC00   ID: 000000003
TSS WHOO TSOPROC($USRJOE) TARGET(*)
FR: NYC00   ID: 000000003
TSODEPT1 OWNS TSOPROC  $USRJOE                                   TSS0300I  WHOOWNS
FUNCTION SUCCESSFUL
TO: NYC00   ID: 000000004
TSS PER(USRJOE) DSN(JUNK) TARGET(NY*)
FR: NYC00   ID: 000000004
TSS0317E  DATASET/PREFIX NOT FOUND IN SECURITY FILE  TSS0301I  PERMIT  FUNCTION FAILED,
RETURN CODE =  8
```

Example: journal on receiving machine

```
FR: CHI01   ID: 000000001
TSS LIS(USRJOE) DATA(ALL) TARGET(A*) WAIT(Y)
TO: CHI01   ID: 000000001
ACCESSORID = USRJOE   NAME    = JOE PAZ  TYPE    = CENTRAL  FACILITY   = BATCH  FACILITY
= STC   FACILITY   = TSO   FACILIT
Y  = TSR  FACILITY  = CICSPROD  FACILITY  = IMSPROD  FACILITY  = VM  FACILITY  = RPGFAC
FACILITY   = RDFFAC   FACILITY   = ROST
EST  CREATED   = 01/20/88  LAST MOD   = 09/25/90  12:13  PROFILES   = TDGPROF  TCSPROF
ATTRIBUTES = CONSOLE,DUFXTR  BYPASSING  =
NODSNCHK,NOVOLCHK,NOSUBCHK  LAST USED  = 08/29/90 11:47 CPU(XE05) FAC(TSO     )
COUNT(01076)  PHYSKEY  = 2356668  VMMDISK  = PAZ
JO01
FR: CHI01   ID: 000000002
TSS ADD(USRJOE) LTIME(1) TARGET(NCY00)
TO: CHI01   ID: 000000002
TSS0300I  ADD      FUNCTION SUCCESSFUL
FR: CHI01   ID: 000000003
TSS PER(USRJOE) DSN(JUNK) ACC(ALL) TARGET(*)
TO: CHI01   ID: 000000003
TSS0317E  DATASET/PREFIX NOT FOUND IN SECURITY FILE  TSS0301I  PERMIT  FUNCTION FAILED,
RETURN CODE =  8
FR: CHI01   ID: 000000001
TSS LIST(USRJOE) DATA(ALL) TARGET(NCY00)
TO: CHI01   ID: 000000001
ACCESSORID = USRJOE   NAME       = JOE PAZ  TYPE       = CENTRAL  FACILITY   = BATCH
FACILITY   = STC  FACILITY   = TSO  FACILIT
Y  = TSR  FACILITY  = CICSPROD  FACILITY  = IMSPROD  FACILITY  = VM  FACILITY  = RPGFAC
FACILITY  = RDFFAC  FACILITY   = ROST
EST  CREATED  = 01/20/88  LAST MOD  = 10/10/90  16:11  PROFILES  = TDGPROF  TCSPROF
ATTRIBUTES = CONSOLE,DUFXTR  BYPASSING  =
NODSNCHK,NOVOLCHK,NOSUBCHK  LAST USED  = 10/10/90 16:09 CPU(XE05) FAC(TSO     )
COUNT(01078)  PHYSKEY  = 2356668  VMMDISK  = PAZ
JO01
```

Chapter 4: Threats and Exposures

This section contains the following topics:

z/OS Integrity

z/OS integrity is defined as the state in which the operating system is functioning correctly and according to specifications. System and user programs and functions are operating in privileged state only when they have been authorized to do so. Otherwise, they operate in problem state. Problem state usually prohibits tampering and unauthorized activity. Privileged state "opens the door" to any form of unauthorized activity, while the System/370 and System/390 architectures support two instruction sets: program and supervisor.

z/OS integrity prevents unauthorized programs from:

- Bypassing the store and fetch protection of internal storage
- Overwriting or reading data areas of DASD files outside allocated boundaries
- Bypassing the password checking of password protected data sets on both DASD and tape
- Bypassing CA Top Secret security checks through the standard z/OS Security Interface

z/OS software provides this protection by validating program requests. Any audit of z/OS must verify that the integrity of the system has not been compromised. The audit must also validate mechanisms have not been deactivated or circumvented.

To gain authorization, a program must be link-edited with AC(1) and must execute from an APF-authorized library. It might then request execution in privileged supervisor state. This is accomplished by executing a privileged SVC, such as MODESET, which sets the program's PSW to supervisor state. From then on, the whole z/OS environment is open to the program, including the ability to defeat security and perform unauthorized (pre-programmed, pre-planned) functions.

An authorized program can perform many functions that are denied to problem programs. In addition, the properly designed, authorized program can defeat security mechanisms that are inherent in z/OS, including CA Top Secret.

Ensure that users and programs do not operate in privileged state except when it is absolutely necessary. If a privileged status is required, that they operate within certain bounds to minimize or prevent unauthorized activity.

Another way of authorizing a program is by using alter/display functions of the hardware itself; for example, modifying the PSW. The only way to control this activity is through effective physical site security.

Authorized Programs

Any program that resides in an APF-authorized library is a *potential* threat. A program must be designed to defeat security, merely being authorized does not negate security.

Take the following precautions:

- Audit use of APF-authorized libraries
- Audit programs from APF-authorized libraries
- Examine the source and object code of programs within the APF libraries
- Perform an APF audit with TSSAUDIT
- Control APF authorizations with ABSTRACT(AC1)
- Deny APF-authorized libraries update access.

z/OS Utilities

z/OS utilities use standard z/OS mechanisms to access data, and therefore go through normal security validation. However, some utilities or utility functions do not go through security processing. For example:

SUPERZAP (IMASPZAP, AMASPZAP)

This utility can determine if APF-authorized can change any data anywhere including security indicators.

IEHPROGM

This utility can allow certain functions to bypass security.

IEHINITT

This utility can initialize any tape volume (overwrite header labels) with only operator authorization.

Utilities can automatically invoke the Standard Security Interface directly themselves or indirectly using OPEN. Therefore, CA Top Secret is automatically invoked.

Utilities that do not interface with security should:

- Have their use restricted by program protection or by restricting use of the libraries containing these programs
- Not reside in the LINKLIST where it is available to all users.

Program Properties Table

The Program Properties Table indicates those programs that operate in privileged state. Programs in this table must have one or more of the following privileges:

- Privileged state.
- Privileged key.
- Security bypass. This privilege should be carefully controlled and justified.

Address Space Security Bypass

z/OS allows for the total bypass of data set security and is activated within the Program Properties Table. (Certain vendor programs also use this feature.) A system programmer can activate this feature for their job or TSO session, only if the tool or program resides in an accessible APF-authorized library. Limit update access to APF libraries or to the volumes upon which they reside.

VTAM and TCAM Terminal Definitions

All online terminals in a VTAM or TCAM network are associated with one- to eight-character names for identification, definition, and control purposes. CA Top Secret uses these names to protect sensitive terminals and to provide SOURCE control for selected ACIDs. If these names are changed, security for them is defeated. Ensure the integrity of these names both internally in the machine, and externally in the terminal definition files.

JES Considerations

JES provides the system programmer with several opportunities to subvert the system, including:

- Access to the JES checkpoint data set (can contain passwords)
- Use of various exits
- Access to PROCLIBs
- JES JCL changes
- Remote name definitions.

JES Reader Definitions

JES uses one- to eight-character names to identify and control its local and remote readers. CA Top Secret can use the names to allow for restricted use of sensitive or remote readers and to provide SOURCE control for selected ACIDs. If these names are changed, security for them is defeated. Ensure the integrity of these names both internally in the machine, and externally in the terminal definition files.

JES Checkpoint

JES does not encrypt or eliminate job passwords in the JCT section of the SYS1.HASPCKPT data set unless the early verify feature is used. Although TSO commands (such as QUEUE and SDSF) allow viewing of spooled input and output, they do not allow viewing of JCT entries. The system programmer can modify these programs to access and display JCT information. Restrict the use or modification of these commands, or security modifications included, to limit exposures.

User SVCs

Your installation can have user-written SVCs for special functions. Examine the SVC table for authorized SVCs. User-written SVCs can be an opportunity to defeat security. An SVC can potentially change the privileges of its caller and thus open up a possible security bypass. An audit must verify these user-written SVCs by looking at the nucleus memory for dynamic changes to the operating system after IPL.

System Modifications

Obtain an inventory of the modifications to z/OS, JES, and vendor products, and understand what they do. The more modifications that are made to your system, the less integrity you can have. Also, include a verification of the system memory for evidence of dynamic modifications.

Note: CA Top Secret makes no external modifications that the system programmer can forget when CA Top Secret is re-installed.

Tape Security

The auditor should monitor when ensuring tape security:

Write Rings

When using the CA Top Secret built-in tape security, CA Top Secret forces the removal of rings for tapes opened only for input processing. A program cannot alter a tape with the ring removed (unless hardware modifications have also been made).

Tape Bypass Label Processing (BLP)

BLP under z/OS allows for unrestricted access to all files on any tape. This might be controlled through CA Top Secret using the BLP access level. If users do not have BLP access authority for tapes, then they cannot use BLP to access tapes.

TMS (CA 1) Tape Bypass

CA-1 allows for security bypass using the LABEL=EXPDT=98000 JCL option. The TMS interface, provided by CA, prohibits use of this feature unless the user has been permitted use of ABSTRACT(XDT98000). For information, see the *Implementation: Other Interfaces Guide*.

SYSGENs

The system programmer can use a SYSGEN to introduce various traps and holes into the system. A careful audit of the SYSGEN is required.

SMF

Use SMF as an audit trail. Subversive activity can often be traced through SMF long before a perpetrator can accomplish any deceptive maneuver. Look for an increase of activity to certain sensitive files and audit the SMF options in PARMLIB member SMFPRMxx for missing record types.

An APF-authorized program can obliterate all evidence of activity in its address space by altering memory locations; making it appear to z/OS that SMF is not active for the user. Using the CA Top Secret Audit/Tracking File is not dependent upon whether SMF is active, and its use cannot be subverted.

SYSLOG

Most activity in the system is recorded in the SYSTEM LOG. When auditing SYSLOG, look for holes in times of the log.

LOGREC

Abnormal activity gets recorded in LOGREC.

PARMLIB Members

The various members within SYS1.PARMLIB affects security directly or indirectly. Check for alternate PARMLIB members (for example, the suffix '00' in SMFPRM00 can be changed for alternates).

SMF Member - SMFPRMxx

This member controls the use of SMF. If LOG(SMF) has been specified in the CA Top Secret control options, ensure that SMF record type 80 is being recorded. The name of the CPU is stored here within the SMFID parameter. If your site is using CPU protection, the integrity of this four-character name must be ensured. If you change it, you lose CPU protection. Also, notice record types not being recorded.

APF Member - IEAAPFxx/PROGxx

This member indicates what libraries are to be APF authorized. The authorized programs within these libraries, if so designed, can bypass security.

LINKLIST Member - LNKLSTxx

This member indicates which program libraries are automatically searched for programs. These libraries are also APF authorized.

Dump Member - DMPOPTxx

This member provides processing options for dumping. DMPOPTxx allows for the dumping of protected storage (LSQA subpool 230, key 3) that includes CA Top Secret control blocks for users. Minimize this dumping.

Command Member - COMMNDxx or IEACMDxx

This member indicates what commands are automatically issued at system startup (IPL). It must include an S TSS command to start CA Top Secret automatically.

LPA Member - IEALPAxx

This member indicates which modules are loaded into the system link pack area. Modules in the LPA can be accessed without accessing the libraries from which they have been loaded.

Appendage Member - IEAAPPxx

This member provides the names of I/O appendages that are used upon certain I/O conditions. These modules can execute in privileged state, so examine them for discrepancies in design.

IPL Member - IEASYSxx

This member contains default parameters for system initialization (IPL). IEASYSxx includes the suffixes for all PARMLIB members. Check these values for possible conflicts. The option OPI=NO prohibits specification of alternate options by the operator during IPL.

PROCLIB

Monitor changes to STC procedures. If you do not use the CA Top Secret default of FAIL for undefined STCs, watch for new procedures being implanted into the PROCLIBs.

Storage Modifications

The use of corezap utilities must be restricted. Look for evidence of use of these utilities.

System Data Sets

A programmer can hide subversive programs in many available data sets. Be vigilant for misleading and inconspicuous names such as IEFBR14 that might be SUPERZAP in disguise.

Alternate Volumes

In performing an audit of z/OS, be attentive to alternate system packs and backups of older volumes. Data sets such as PARMLIB or LINKLIB on these volumes must be part of the audit.

Exits

Exits within z/OS and vendor products allow for legal modifications to the system without changing IBM or vendor code. This flexibility, however, provides the deceptive programmer with several opportunities to exploit the system.

The auditor should obtain a complete inventory of exits. Before these exits can be judged as exploitative, the auditor must be familiar with coding at the assembler level.

Exits exist in almost all portions of z/OS. Potentially vulnerable exits include, but are not limited to:

- CICS
- IMS
- RMF (ERBMFIUC/MFDUC/TRACE/MFRUR)
- VTAM
- UTILITIES
- WTO (change route codes, remove messages)
- JES (upwards of 20 exits)
- DSF (ICKUSER1 - data security)
- DFP
- HSM
- DADSM (IGGPRE00, IGGPOST0)
- RMF
- SMF (IEFUJV/UJI/USI/U29/UJP/U83/UJI/ACTRT)
- DCB
- VSAM/IDCAMS (exception exit)

Note: Most of these exits allow execution in privileged state.

Chapter 5: Other Types of Threat and Exploitation

This section contains the following topics:

Application Programmers

Exploitation by a deceptive application programmer can occur during application development and maintenance. Programs must be compared against object code or programs in active storage itself.

Inspect code for:

Trap doors

Allow special functions to be performed using a special code or userid.

Trojan horses

Allow for special processing after a certain date.

Bombs

Cause data set damage after a certain date, such as employee termination.

The auditor must be involved in all stages of a program's life, especially during the design stage. Ensure that personnel who develop a program are not the individuals who test it. Fraud is more likely if only one person develops, tests, and implements a program. Fraud can also occur if a conspiracy exists among the developers, testers, and implementers. Separating function minimizes the risk.

In auditing applications already in use, determine if:

- Transactions and inputs are complete and no data is missing
- Input data is accurate and, if inaccurate, procedures exist to correct mistakes

- Every transaction is processed completely and data in files is changed accordingly
- All changes to files are correct
- Changes were properly authorized and that authorization and security are guaranteed
- File integrity is maintained after changes are processed
- Unauthorized changes are detectable
- Proper procedures are documented and available.

System Programmers

Security over system programmers is a trade-off between function and exposure. Surveillance and auditing create some assurance that system programmers have not subverted the system. The greatest single exposure is update access of APF-authorized libraries. Restrict updates to the APF-authorized repositories. Performing continuous audits is mandatory and all updates must be justified. Include the programs in this library in your periodic audits.

Unless required every day, access to critical system data sets should be allowed only through a super ACID. This ACID's activity is audited to leave an audit trail.

The key to controlling system programming is the control of APF library alterations, the use of AC(1), PROCLIB control (STC control), and use of super ACIDs.

The TSSAUDIT utility determines the contents of APF-authorized libraries. For information, see the *Report and Tracking Guide*.

Passwords and User Accountability

Password research and evidence indicate:

- The longer the password:
  - The harder it is to crack.
  - The harder it is to remember.
  - The more likely it is written down.
- The shorter the password:
  - The easier it is to crack.
  - The easier it is to remember.

Ensure that passwords conform to the following rules:

- Personal information such as name of spouses, children, names of places, months, license numbers, and telephone numbers must not be used.
- A minimum length of at least four or five characters.
- Passwords that are generated by using vowels alternating with consonants.
- Passwords that can be pronounced.
- Possible consideration of passwords that are automatically generated for online users to eliminate password distribution problems and user-selected passwords.
- Passwords that are forced to be changed at least once a month.
- A password history that is maintained to prevent re-use of similar passwords (automatic with CA Top Secret).
- Passwords cannot be changed more than once per day.
- Common words are restricted.
- Password cannot match User id.

Provide password distribution for remote users through self-sealed mailers. Establish a procedure that the user acknowledges receiving the password. Never communicate passwords over the telephone!

Implement a system that makes users responsible for their actions. By setting up penalties for security breaches, data integrity is less likely to be compromised.

Computer Operators

Ensure that operators have only those privileges that are required to perform their job functions.

The operations personnel have physical access to data at your site and can access data through terminals and consoles.

Potential areas of threat include:

- Use of started tasks at the console.
- Use of programs that allow data to be displayed or even altered.
- Use of alternate IPL parameters.
- Misuse of CA Top Secret to change security control options.

Available CA Top Secret controls include:

- Use of started task passwords for critical/sensitive STCs.
- Use of started task accountability to provide an audit trail of sensitive started tasks.
- Ensure strict authorizations for use of backup/restore programs and functions, especially full-volume operations.
- Controls over job submission, including assigning default ACIDs for local readers.
- CA Top Secret exit for further measures providing operator ACIDs that can only perform specific tasks.

Ensure that all entries to the system, including physical readers, are controlled through CA Top Secret.

Production Control Personnel

Production control personnel have the authority to change data when errors occur. They must also review the output for correctness and make appropriate changes in case of error. Without adequate controls, access to production data can be compromised.

Use CA Top Secret to:

- Prevent unauthorized use of production control ACIDs whether from TSO, BATCH, or a production control package.
- Limit the access to production data.
- Limit the use of production programs.

Once you authorize a production control person to submit jobs with a specific ACID, you also allow them to submit jobs that are not part of their job function. An example would be listing and changing data. This type of authorization creates the greatest exposure! Controls must be entrenched in the system to:

- Prevent job submission from outside the production control system.
- Examine JCL as it is being submitted.
- Prevent/limit updating of production control libraries.

To restrict the use of undefined job submission started tasks, the STC default of FAIL should be in place. Audit trails must be established by auditing the production control libraries, and compliance jobstreams must be periodically run independent of production control to assure completeness of production jobs. Job submission should be restricted to certain terminals or readers.

Allow updates only through ACIDs that are audited. Review and justify use of these ACIDs.

External Personnel

Set audit trails and controls to prevent or detect userid guessing and password guessing. CA Top Secret control options, such as VTHRESH, PTHRESH, and NEWPW, can be used to prevent and detect this type of activity. Auditing and tracking utilities can also be used to prevent and detect this type of activity.

Dial-Up Terminals

The use of dial-up terminals presents a great threat to the security of a site. Terminal names for dial-ups are defined to VTAM and TCAM. Define these names to CA Top Secret and force controls that include time of day and day of week restrictions.

Use special ACIDs for outside access. Do not allow internal ACIDs to be accessed from any facilities outside the building. In addition, audit these ACIDs.

Vendor Packages

Many vendor packages provide varying levels of threat, usually permitting execution in privileged state, or dynamic subsystem or APF library manipulation.

Some packages that permit execution in privileged supervisor state, or allow security bypass include: CA-IDMS, CA-ASM2, CA-LOOK™, RESOLVE, OMEGAMON, COMPLETE.

Controls include restricting program access and requesting superzaps from vendors to minimize or eliminate security breaches.

Chapter 6: Verification and Compliance

This section contains the following topics:

Verification of Systems and Data Integrity

All jobs, data files, programs, and hardware devices are under the direct control of z/OS, therefore a complete audit is crucial. For example, it is possible to set up users that ignore passwords, subvert CA Top Secret, and modify production data. User can also destroy audit trails and access restricted files—through z/OS granted permissions. On the systems level, it is your responsibility to control:

- The z/OS system generation (SYSGEN) specifications and parameter library (PARMLIB) contents.
- Operating system maintenance, tuning, and change management.
- z/OS operator consoles, commands, routing codes, and the security of the z/OS PASSWORD file and its contents.
- Administration and control of the Time Sharing Option (TSO).
- Authorized Program Facility (APF) library administration.
- The Program Properties Table (PPT) specifications.
- The entire access control software implementation.
- Other vendor and site defined Supervisor Calls (SVCs).
- Program product and site defined Input/Output (I/O) Appendages.
- The System Management Facility (SMF) options, files, and contents.
- z/OS System Exit selection, coding, usage, control, and documentation.
- The Job Entry Subsystem (JES) options, parameters, exits, and PROCLIBs.
- Installation and control of site-defined z/OS subsystems.
- Usage and control of the System Modification Program (SMP).
- Installation and control of products like IMS and CICS.
- Administration and control of any job accounting facility.

- Contents of the system link pack areas (PLPA, FLPA, MLPA).
- Installation and control of all third-party vendor products.

Information From CA Auditor

Some of the information that is provided by CA Auditor includes:

System Level Function

Provides z/OS version, level number, SUs, IPL and SYSGEN information, CPU type and serial number, and so on. CA Auditor allows online retrieval of SMF data in English, explains SMF options parameters, files, and exits. System level functions can scan PROCLIBs, and provides JES and SMF IDs, JES PROCLIB identification, and JES2 options.

Hardware Information

Includes device display by address, name, type, or allocation status. Online summary of disk and tape error rates, display of operator consoles including routing codes, allowable command groups, and alternate console structure is also available.

System Library Analysis

Includes PARMLIB and z/OS release independent display of APF, LINKLIST, LPA, and key system libraries. Determination of APF problems, such as duplicate modules, unauthorized copies of superzap, duplicated module detection, APF TSO programs and commands, and so on. It also provides automatic detection of PARMLIB changes and mapping of system catalogs.

Technical Information

Provides unique analysis functions to display z/OS subsystem information; search and scan for site defined I/O appendages; locate and identify major exits; display and analyze PLPA, MLPA, FLPA, and the Program Properties Table; and detect intercepts and abnormal conditions for IBM, site-defined, and ESR SVC modules.

Job and Program Functions

Provides JCL scans, intelligent online program compare, program freezer, program origin determination, program statistical information, and SMF-based job and program scans.

File Functions

Provides Library and VTOC integrity analysis, password system validation, catalog, and volume based complex search capability, library source and load module correlation, file freezer, and file compare.

A z/OS system audit should include an evaluation of all threats. Occasional checks of program code, data files, and audit trails are required. Audit trails of changed data elements are facilitated through the Applications Interface.

Verification of Authorizations

Access authorizations must be periodically evaluated and any discrepancies justified. Some of the sources that are used to verify the authorizations are:

- System status displays
- Listing of all users and their authorizations and attributes
- Cross-reference report of user bypass attributes
- Listing of changes to security definitions
- Listing of access authorizations to critical data sets and other resources
- Spot checking of ACID (using TSSSIM) for certain resource availability

Compliance Testing

Ensure that CA Top Secret is installed and implemented in compliance with corporate security policy. Ensure adequate administrative and technical controls exist so that the required levels of data security are provided and maintained.

The auditor roles include:

- Review of the installation process for completeness
- Review during implementation
- Consultant for control decisions
- Testing for compliance policy
- Consultant for CA Top Secret decisions
- Representative of top management

Controls to be Evaluated Once CA Top Secret is in Place

After CA Top Secret is in place:

- Determine the options in use at the installation. From the master console enter:

  ```
  F TSS,SYSOUT
  ```

  ```
  F TSS,STATUS
  ```

  ```
  F TSS,FACILITY(ALL)
  ```

  ```
  F TSS,SYSOUT
  ```

  Obtain the SYSOUT listing of the TSS STC from the printer and examine the output. Look for deviations from the expected control options. Take particular note of MODE, PTHRESH, VTHRESH, logging options, NEWPW, and FACILITY options.

- For each facility that is secured, log on using your auditor ACID and perform various tests to ensure proper facility operation according to policy.

- Obtain a listing of all users and profiles within your scope. Enter the following command using the BATCH TMP:

  ```
  TSS LIST(ACIDS) DATA(ALL)
  ```

- Attempt to log on using sample ACIDs to determine effectiveness of various security controls.

- Obtain a listing of ACIDs that are not in FAIL mode using the following commands:

  ```
  TSS WHOHAS MODE(D)
  TSS WHOHAS MODE(W)
  TSS WHOHAS MODE(I)
  ```

- Determine if sensitive utilities are protected. For example:

  ```
  TSS WHOHAS PROGRAM(IE)
  ```

- Determine who has access to critical system and production data sets using the following command:

  ```
  TSS WHOHAS DSNAME(SYS1.)
  ```

- Examine the CA Top Secret started task procedure to ensure that proper Backup, Recovery, and Audit/Tracking Files are in place.

- Determine who has special bypass privileges (use the PRIVILEGES control statement of TSSAUDIT).

- Determine what privileges/accesses have been given to all users by issuing:

  ```
  TSS LIST(ALL)
  ```

- Determine what controls are active for started tasks. List the STC definitions by issuing:

  ```
  TSS LIST(STC)
  ```

  The default option DEF is first; it is not BYPASS or UNDEF. Ensure that the default ACID for STCs is FAIL or a specific ACID. If it is a specific ACID, it should have no BYPASS attributes.

- Determine who has special administrative privileges and whether they conform to corporate policy. Obtain a listing of all ACIDs using the following command:

  ```
  TSS LIST(ACIDS) DATA(ADMIN)
  ```

- Determine whether the CA Top Secret Security File changes are being properly recorded in the Recovery File. As a test, change your password and then run TSSAUDIT using the CHANGES control statement to ensure that the change was recorded.
- Locate all ACIDs that do not require passwords. Determine whether adequate source controls have been placed upon these ACIDs. Use of the AUDIT attribute is recommended.
- Ensure that critical CA Top Secret control options are in use.
- Determine what resources must be audited and that their prefixes or names are in the AUDIT record by using the following command:

  ```
  TSS LIST(AUDIT)
  ```

- Determine who has access to APF-authorized data sets. Check PARMLIB members LNKLSTxx and IEAAPFxx.
- Perform an APF audit using the APF control statement of TSSAUDIT or CA Auditor.
- Check that dial-up lines have protected terminal names.
- Determine if applications use the Application Interface to log changes to critical data elements.
- Ensure what functional units within the organization have adequate security guides or other relevant documentation.
- Determine who has access to protected CA Top Secret utilities. Any program starting with TSS should be owned and therefore protected. Use the following command to check:

  ```
  TSS WHOHAS PROGRAM(TSS)
  ```

- Ensure that users of critical facilities, such as production IMS or CICS, comply with corporate policy.
- Check to see who has access to all resources by type. For example:

  ```
  TSS WHOHAS DSNAME(**)
  ```

- Check periodically that terminated employees no longer have active ACIDs.

- Use the CHANGES control statement of TSSAUDIT to check that the listed changes have proper written authorizations.
- Determine what default ACIDs are in use on a facility basis by using the TSS MODIFY command. For example:

  ```
  TSS MODIFY(FACILITY(TSO))
  ```

- Determine to what extent vendor packages can be used that can bypass security, or for which there is no inherent security.
- Ensure that all data sets are protected. In a z/OS non-Always call environment, obtain the current listing from the TSSPROT utility with the following options:

  ```
  PROTECT SIM
  ```

- Determine whether authorizations are not too general.
- Ensure that anti-subversion measures are in place.
- Check that critical started tasks use operator accountability. List the STC record and look for STCACT attribute.
- Protect unauthorized use of the linkage editor SETCODE AC(1) option.

  ```
  TSS ADDTO(MSCA) ABSTRACT(AC1)
  ```

- Monitor use of SETCODE AC(1) if it is authorized to all users by using the following command:

  ```
  TSS PERMIT(ALL) ABSTRACT(AC1)
                  ACTION(AUDIT,NOTIFY)
  ```

  or monitor use of SETCODE AC(1) if it is restricted by using the following command:

  ```
  TSS ADDTO(AUDIT) ABSTRACT(AC1)
  ```

- Notify security personnel when certain events occur using:

  ```
  TSS PERMIT(ALL) DSNAME(x)
                 ACTION(NOTIFY)
  ```

- Audit update access to certain data resources using:

  ```
  TSS PERMIT(ALL) DSNAME(x)
                  ACCESS(R)
  ```

  ```
  TSS PERMIT(ALL) DSNAME(x)
                  ACCESS(U)
                  ACTION(AUDIT)
  ```