Global Information Assurance Certification Paper

Mainframe CA-TOP SECRET Security of UNIX System Services

GSEC CERTIFICATION

Practical Assignment Version 1.4b

By

William H. Wiegman

April 27 2004

Attended: Local Mentor Class

Introduction

The classic IBM mainframe has evolved and continues to evolve to support today's networking strategies. No longer are they isolated, monolithic giants locked behind doors with only privileged, highly-trained specialists having access. They now support Internet access, TCP/IP, LDAP and USS (UNIX System Services) just to name a few. A Computer Associates (CA) USS developer told me understanding and implementing USS security can be a lifelong endeavor. IBM's UNIX System Services Planning[10] Manual is over 600 pages long with 8 chapters dedicated to describing just the differences between different releases. The IBM UNIX System Services Command Reference[11] Manual is over 900 pages. Other manuals are in the Reference section[12, 13]. Obviously, this kind of detail is beyond the scope of this paper. This paper will describe basic CA-TOP SECRET security features that can and should be implemented to secure USS on the Mainframe. Two implementation strategies will be presented. The appropriate one for your installation depends on your situation and will be left for you to decide.

Mainframes Today

The demise of the Mainframe has been greatly exaggerated. I believe this twist on Mark Twain's quote[1] is appropriate for today's Mainframes. By Mainframe, I mean IBM's zSeries or S/390s computers running z/OS or OS/390[2] -- basically, the descendents of the classic old "dinosaurs" of the 1960's and 1970's[3]. There is no question that the dominance they had back then is gone, simply because you can now select hardware and software appropriate for your computing needs. Back then, if you needed significant computing power, your options were pretty much limited to a mainframe whether it was appropriate for your needs or not. This was especially true for business applications, which were a significant portion of computer applications in the early days of computing. Now you can select the appropriate computer type, size and operating system for your computing needs. However, when major business computer power is needed, often the IBM mainframe is still the appropriate choice[17].

CA-TOP SECRET Standard Basics

Before going into CA-TOP SECRET USS security, a brief description of CA-TOP SECRET basics is needed to lay down a foundation from which to build.

CA-TOP SECRET has three basic components: Program, Control Options File, and Security File.

Program: The actual program that must be running as a Started Task or Daemon to enforce the security rules. It intercepts requests for computer resources and determines if the requestor has the permission to be granted the access.

Installing and maintaining CA-TOP SECRET is usually done by System Programmers, hopefully, under the watchfully eye of Security Administrators. The Security Administrator must always test that the program is performing correctly whenever changes are made to it (i.e., new releases, versions or when PTFs/patches are applied).

Control Options file: The settings in the Control Option file are used to customize how the program will enforce security for your particular installation; i.e., what computer resources are protected, how and what action should be taken under given situations. (Similar to an ".ini" file).

Although the Control Options file is normally created by the System Programmer during installation, only the Security Administrator should set the values in the file. Many of these options have default settings, but I recommend explicitly defining them in this file. Defaults can change over time with different releases; and, more importantly, this will clearly document what the setting should be. These settings can be overridden four different ways. Explicitly setting them in this file will clearly document what they were set to.

Three ways of overriding these settings when the CA-TOP SECRET program is started are,

1. by changing options on the start command used to start the CA-TOP SECERT program when IPLing the Mainframe (initial program load – IBM's version of boot),

2. by changing the PARM field in the JOB card that starts the CA-TOP SECRET program, or

3. by the computer operator issuing a start or modify command[7].

I strongly recommend not using these three methods. The change is logged, but it may be very difficult to determine who made the change and when it was made. Also, and maybe more importantly, the Security Administrator didn't have to be involved with the change.

The fourth way is by issuing the CA-TOP SECRET "modify" command. Only knowledgeable Security Administrators should be permitted to use this command and only under documented controlled conditions. However, this method is auditable and does require a Security Administrator with specific authority.

These four override methods have their uses; and although other installations may disagree, for the production environment I strongly recommend only changing settings in the Control Option file under change control procedures. Using the CA-TOP SECRET modify command should only be allowed in Production for emergencies under very strict conditions. As stated before, these settings control how the security program functions and changing it on the fly should be used with extreme caution.

In the test environment, using the CA-TOP SECRET modify command is a good way of testing changes without having to IPL or re-cycle the CA-TOP SECRET program.

Security File: The Security file contains the accounts of actual people or entities and the permissions given to these users or entities for specific computer resources. This is the information the program needs to permit or deny requests to access computer resources. (Similar to Microsoft's Active Directory[9])

The data in the Security File should be totally controlled by the Security Administrators. The size and maybe the location of the file should be worked out with the Systems Programmer during installation. CA has a formula for determining the size of this file, but make it much larger. The size can be changed later, but risking the security system to save a little disk space isn't worth it. Make it large enough in the beginning to accommodate substantial growth. This is where the actual security rules are stored that CA-TOP SECRET enforces, and understanding how the program interacts with this file is critical in understanding how security is enforced and administered. First let's start with the structure of the file.

The Security file is an encrypted vendor propriety database whose logical structure is hierarchical, similar to a DOS inverted tree. All entities (records) in the file are called ACIDs (Accessor IDs).

The Security file has three types of ACIDS: Organizational, Control and Functional.

Organizational ACID

There are three Organizational ACIDs that are used to create the internal structure of the file. They are normally used to align the security file's structure with the company's reporting structure, but they can be used in any manner that facilitates your security administration. This should be thought out carefully before implementation. It can be changed but doing so can be disruptive and time consuming.

> **ZONE ACIDs** – are at the top level, basically just below what could be called the ROOT. In a single company zones may represent geographical

locations or company functions. In a multi-company organization, they normally would be used to represent the different companies.

Division ACIDs – are at the middle level and can be used to further separate resources and people. Divisions are normally assigned to a Zone, but can be stand alone for specific purposes. As the name suggests, they were originally designed to represent divisions within a single company.

Department ACIDs – are at the lowest level and are used to group people and resources at the smallest structural unit. Departments are normally assigned to a Division but can stand alone. They can't be assigned to a Zone. Normally, Department ACIDS are used to represent actual company departments.

Control ACIDs

A Control ACID is actually a User ACID with administrative permissions. There are six (6) types arranged in hierarchical order of scope of control and authority.

MSCA - Master Security Control ACID
Has unlimited scope of control (the entire security file) and implicit unlimited administrative authority (can issue any administrative command). Only one can exist. It cannot be deleted, but can and should be renamed.

SCA - Central Security Control ACID
Has unlimited scope of control (the entire security file) but explicit administrative authority (must be permitted). Many can exit, but only the MSCA can create them.

LSCA - Limited Central Security Control ACID
Has limited scope of control (over specific zones) and explicit administrative authority (must be permitted). Many can exit.

ZCA - Zone Control ACID
Has limited scope of control (over one zone) and explicit administrative authority (must be permitted). Many can exit.

VCA - Divisional Control ACID
Has limited scope of control (over one division) and explicit administrative authority (must be permitted). Many can exit.

DCA - Department Control ACID
Has limited scope of control (over one department) and explicit administrative authority (must be permitted). Many can exit.

Functional ACIDs

Functional ACIDS are actual Users (people) or Entities (computer programs) making requests for computer resources or groups of Users, Entities or permissions.

Profile ACIDs: contain resource permissions that should be logically combined to facilitate security administration. Profile ACIDS are added to user ACIDs. They are similar to UNIX Groups.

Group ACIDs: are a collection of User ACIDs that share permissions. It is similar to a profile ACID in that it logically combines permissions and users. The difference is that groups are recognized by IBM USS. Each User ACID must be associated with a group ACID if it is to be recognized by IBM USS. A Group ACID defines the GID() group number.

User ACIDs: define an actual person or entity and explicit permissions granted that person or entity. This is an important concept to understand with CA-TOP SECRET. Permissions are stored with the person or entity not the resource. This should be based on their job requirements and may be modified by location within the company. If a User ACID has CA-TOP SECRET administration permissions, it is considered a Control ACID.

Other files are needed for normal Data Processing housekeeping, (Audit/Tracking file, Backup file, and Recovery file), but their use is secondary to security and common to most security systems and, therefore, won't be addressed in this paper.

Putting all this together allows security requests to be sent to CA-TOP SECRET by mainframe sub-systems or applications to determine if a User or entity has the authority to access the computer resource. The CA-TOP SECRET program will interrogate the Security File, given the settings in the Control Options file, to determine if the request should be granted or denied.

CA-TOP SECRET USS Basics

In today's distributed environments, many hardware platforms and operations systems are networked together to optimize users' access to numerous applications regardless of where the application is running. This creates a major security issue. Obviously, the same level of control and accountability for data and resources accessed in each environment should be equivalent. This is a very complicated, if not impossible, goal with the current state of the art of security. To keep up with this and, hopefully, remain a major player in this changing environment, IBM keeps adding network functions with each new release of its mainframe Operating Systems z/OS or OS/390. One of these functions is USS (UNIX system services) which provides an online, interactive UNIX environment running within the IBM mainframe operating system. USS is similar to the older online environments, IMS or CICS in that it currently has its own internal security. Like them, the internal security is being migrated out into the MVS external security system, in this case, CA-TOP SECRET. Consequently, new versions of CA-TOP SECRET keep adding new features to perform UNIX security administration necessary to manage the USS environment. Currently, it is a mixed bag making security difficult. CA-TOP SECRET supports administering most resource permissions and enforces MVS security when external USS access is requested; but, depending on control options, much of the USS enforcement is still internal based on UNIX security.

CA-TOP SECRET USS resources[5] are:

UID() is a numeric value form 0 to 2,147,483,647 and must be unique except for 0, which indicates a superuser. Within USS a user is defined by this number.

GID() is a numeric value form 0 to 2,147,483,647 and must be unique for each group. 256 groups can be assigned to a user. Within USS every group is defined by this number.

HOME defines the initial directory path used when a user enters the OMVS command or enters the ISPF shell. HOME is optional and if not defined, defaults to the Users root directory. It can be 1024 upper and lower characters.

OMVSPGM is the user's shell program which is the first program started when the OMVS command is entered or when a batch job is started using the DPXBATCH program. OMVSPGM is optional and if not defined the default shell program ('/bin/sh') is used. It can be 1024 upper and lower characters.

NOOMVSDF is a user ACID attribute that prevents the ACID from inheriting the default UID() and GID(). The administrator must have this administration permission in MISC8.

CA-TOP SECRET UNIX System Services authorities[5] are:

BPX.SUPERUSER – allows non-superusers to gain superuser authority (can use UNIX command SU)

BPX.DAEMON – allows daemon programs to validate a user's password and change the identity of a spawned address space (control over setUID() and seteUID()).

BPX.SERVER – allows daemon program to customize the security of an environment thread.

BPX.SMF – to restrict access for 'c' applications to generate SMF records without APF authorization

BPX.DEBUG – to allow users to use dbx to debug programs that run APF authorized or with BPX.SERVER authority.

BPX.WLMSERVER – to allow users to use WLM server functions.

BPX.STOR.SWAP – to allow users to make address spaces non-swappable.

BPX.FILEATTR.APF – to allow users to turn on the APF-authorized attribute for an HFS file.

BPX.FILEATTR.PROGCTL – to allow users to turn on the program controlled attribute for an HFS file.

UNIXPRIV resources:[8 page 1-16]

UNIXPRIV resources were created by IBM to allow non-superusers to be granted superuser privileges with a high degree of granularity. At OS/390 version 2 release 8 and above, a user can be permitted only the privileges needed and not superuser authority UID(0). It allows you to minimize the number of required superusers thus reducing the security exposure.

SUPERUSER.FILESYS.FILE (READ or higher) - Allows a user to read any HFS file and read or search any HFS directory

SUPERUSER.FILESYS.FILE (UPDATE access or higher) - Allows a user to write to any existing HFS file.

SUPERUSER.FILESYS.FILE (CONTROL access) -Allows a user to write to any HFS directory.

SUPERUSER.FILESYS.CHOWN -Allows a user to change ownership of any file.

SUPERUSER.FILESYS.MOUMT -Allows a user to issue mount, unmount, quiesce, and unquiesce requests.

SUPERUSER.FILESYS.PFSCTL -- Allows a user to call pfsctl ().

SUPERUSER.FILESYS.VREGISTER -Allows a user to issue vregister() to register as a vfs file server.

SUPERUSER.IPC.RMID -Allows a user to do ipcrm calls to clean up leftover IPC mechanisms.

SUPERUSER.PROCESS.GETPSENT -Allows a user to view all processes.

SUPERUSER.PROCESS.KILL - Allows a user to send signals to any process.

SUPERUSER.PROCESS.PTRACE -- Allows a user to use dbx to trace any process.

SUPERUSER.PROCESS.SETPRIORITY -Allows a user to increase his priority.

CA-TOP SECRET USS control options[7]:

OMVSGRP – specifies the ACID used to provide the OMVSGRP segment for an extract for any group that doesn't have an OMVSGRP segment. I recommend permitting each group a specific GID() number and setting this option to none. It takes more effort but maintains control over groups which is vital.

OMVSTABS – requests that CA-Top Secret refresh the internal UID() and GID() tables used by USS processing. This updates all UID() and GID() administration changes done since the last IPL or refresh.

OMVSUSR – specifies the ACID used to provide the OMVS segment for an extract for any user who doesn't have an OMVS segment. I recommend permitting each ACID needing USS access a unique UID() and setting this option to none. It takes a little more effort but maintains accountability.

OPTION (32) – Enables USS logging feature. Can only be set in the Control file.

OPTION (69) – When a GROUP is specified in a signon (i.e. TSO, CICS, IMS), and it doesn't match a group to which the user is authorized, TSS will fail the signon. Without this option, TSS substitutes * for the group and allows the signon.

CHOWNURS() – Allows users to use the CHOWN function to change file ownership for their files.

(OFF) – restricts users from changing file ownership unless he/she is a superuser or permitted UNIXPRIV(SUPERUSER.FILESYS.CHOWN). This is the default setting.

(ON) – Allows a user to issue a CHROWN command to change the owner UID or GID for a file he/she owns.

HFSSEC() – turns HFS security ON or OFF. (a very important option)

(OFF) – Disables CA-TOP SECRET HFS security checking. Normal UNIX security checking is enabled, including file permission bits checking, superuser status and normal UNIX security services.

(ON) – Enables CA-TOP SECRET HFS security checking. Normal UNIX security checking is disabled, including file permission bits checking, superuser status and normal UNIX security services.

CA-TOP SECRET USS security

As stated earlier in this paper, new versions and releases of the IBM operating system add more features and functions to USS, and new versions and releases of CA-TOP SECRET not only keep pace with these new features and functions but expand the granularity and scope of the control.

The latest version of CA-TOP SECRET supports many but not all UNIX features, defining user ACID UID(), GID(), HOME, initial program, controlling some system services, HFS, Auditing and others. But, the major control provided by CA-TOP SECRET is access to the USS environment.

ACCESS:

The primary and most significant function CA-TOP SECRET provides is controlling access to USS. If you can't access USS, there isn't a security issue. Before a user or entity can access the USS environment, a CA-TOP SECRET ACID must exist and be permitted, a UID(), GID() and default Group. CA-TOP SECRET can be set up to provide a universal UID() and GID() using Control

Options OMVSUSR and OMVSGRP, but I don't recommend it. It does reduce administration work but only marginally, and meaningful auditing won't be possible. If you do use this option, you can still assign unique numbers to user ACIDs when appropriate. And if you don't want certain users to have access to USS you can add the attribute NOOMVSDF to their ACID, thus preventing them from inheriting the default UID() and GID(). I recommend assigning unique UID() GID() numbers to every ACID using USS, even if only for future accountability.

ACID UID() assignments

Developing a strategy for assigning unique UID() numbers for every user ACID is important but at many companies a standard may already exist. If they already have UNIX computers there may or should be a UID() standard which should be used for USS. This will keep user ACID assignments consistent across platforms. If there isn't a standard, many payroll systems use employee numbers which could be used to uniquely identify people. If there isn't an obvious choice and the number assigned isn't important, CA-TOP SECRET has an option to assign unique UID() numbers to user ACIDs given a specified range of numbers. If a standard doesn't exist, take the time to develop a good one for your company.

One UID() number that doesn't have to be unique is zero UID(0). This is a standard UNIX convention where a user defined with UID(0) is a superuser and has unlimited authority. The equivalent in CA-TOP SECRET is the MSCA (Master Security Control ACID). Both have unlimited authority and are needed to perform specific tasks no other user can do. However, CA-TOP SECRET allows only one MSCA. The name can be changed but it can't be deleted and it can't be permitted to USS[8 page 1-8]. In USS, many users can be superusers, assigned UID() of 0. Even a normal user, if permitted DBX.SUPERUSER authority, can change to a superuser with the UNIX SU command. However, even with this option, depending on the version/release of MVS and CA-TOP SECRET, certain Users and Started Tasks must be defined as superusers. Newer versions of MVS have reduced this dependency but it still remains for certain tasks; i.e., restoring HFS (UNIX hierarchical file system). If a user ACID only requires superuser authority for specific tasks, assign it a normal UID() and permit it DBX.SUPERUSER authority. The user must then sign on USS with the assigned UID() and only needs to change to superuser status when needed. This is auditable. Better yet, if the user only needs certain administration functions, assign only the ones needed via UNIXPRIV permissions.

Group GID() assignments:

CA-TSS group ACIDS define GID() group numbers for USS. Again, if you have UNIX computers, hopefully, group number assignments have already been standardized. USS itself requires group OMVSGRP UID(1) and TTY UID() (2).

Each Group UID() number must be unique. Hopefully, GID() 1 and 2 doesn't conflict with your standards. A user ACID can be assigned up to 256 group ACIDS, although, I would contend this is excessive. A user ACID can and should be assigned a default group by permitting it the DFLTGRP field. Obviously, the user ACID must be permitted to the group GID() for the default group DFLTGRP it is assigned. The DFLTGRP is used whenever the user ACID signs on and doesn't explicitly specify what GID() to use. The default group can always be overridden at signon time as long as the user ACID is permitted the group specified.

Since CA-TOP SECRET dataset (file) permissions are based on permissions in the user or profile ACIDS and USS file permissions are based on UID() and GID(), numbers keeping data security consistent in both environments is a difficult task. One strategy would be to associate a profile ACID with a group ACID. Users permitted to both would have the same dataset access in MVS from the profile and USS file access from the group ACID. This is much easier said than done.

Other USS assignments:

CA-TOP SECRET can assign two other USS parameters HOME and OMVSPGM. If they aren't defined when a user accesses USS, the directory is defaulted to root, and control is passed to the default shell program. This is documented at logon:

> FSUM2386 No shell program was specified in the user profile. The default shell ('/bin/sh') is used.
> FSUM2383 No initial directory pathname was specified in the user profile. The home directory is set to root.

The above messages were cut and pasted from my terminal.

Maintaining USS tables.

CA-Top Secret maintains the USS tables of UIDs, GIDs in memory. These tables are built in the computer's memory during the initial startup of the CA-TOP SECRET program. Whenever USS security is defined, deleted or modified the changes will be made to the Security File but won't take affect until CA-TOP SECRET is re-cycled (shut down and restarted). Waiting for CA-TOP SECRET to be re-cycled usually isn't an option since most installations run days if not weeks without re-cycling. Waiting this long isn't advisable, and re-cycling CA-TOP SECRET just to update these tables isn't recommended or needed. Anyone with the administration authority to make these USS changes should have the authority to issue the TOP SECRET command to refresh the USS

tables in memory - TSS MODIFY (OMVSTABS).[6 page 6-49] The refresh is fast and doesn't endanger the security system.

Hierarchical File System (HFS)

USS uses the Hierarchical File System which is a standard UNIX tree-structured file system consisting of directories and files.[7] It resembles the DOS file system, using "/" instead of "\". Internal USS file security is standard UNIX security. Each directory and file is assigned an owning UID() and GID() number which is saved with the file not in CA-Top Secret. As with standard UNIX security, three categories of access are defined for each directory and file in HFS:

1. The owner of the file or directory (normally, the user that created it)
2. The group that owns the file or directory
3. Everyone else.

Each category has three levels of access, READ, WRITE and EXECUTE[16]. This is represented in three groups of three bits (i.e. rwx, rwx, rwx.). Therefore, as the chart below displays (111,101,100), would permit Read, Write and Execute access to the owner, Read and Execute to the group, and only Read to everyone.

	Owner	Group	Everyone
Read	1	1	1
Write	1	0	0
Execute	1	1	0

Permissions for a given file or directory can be displayed using standard UNIX commands "ls –l" or "ls –E". Permissions can be changed using shell commands:

1. CHMOD – Change permissions bits for a file or directory
2. CHOWN – Change owner or group for a file or directory
3. CHGRP – Change the group of a file.

These are standard security tools UNIX administrators' use.

CA-TOP SECRET can override the USS internal file security by setting the control option HFSSEC to ON. However, this disables the internal UNIX security including bit permissions, superuser and security services. If you do turn HFSSEC ON, make sure you add resource HFSSEC of ROOT to CA-TOP SECRET, otherwise it won't have the authority to protect the files under ROOT. This is standard CA-TOP SECRET procedures. HFS file security will then be maintained in CA-TOP SECRET.

However, if you already have UNIX computers and a working Group security strategy, you may elect to set HFSSEC OFF and use your UNIX strategy for the internal USS security. One issue is CA-TOP SECRET dataset permissions, which range from ALL, UPDATE, READ, WRITE, CREATE, FETCH, SCRATCH, and CONTROL to NONE. Each User or Profile ACID can be permitted any dataset with any combination of these permissions. This granularity cannot be duplicated in USS or in any other UNIX system. But, if your installation already has an acceptable UNIX security strategy, it can certainly be used for USS.

The advantage of moving HFS security into CA-TOP SECRET is the permissions for HFS file and MVS dataset can be permitted in the same user or profile ACID for easier administration. Standard CA-TOP SECRET logging and reporting could be used for both MVS and USS. The disadvantage is normal UNIX security is disabled; and as stated, if you have other UNIX systems and administrators this may not a good option.

Each installation must decide to keep internal USS security or migrate it to CA-TOP SECRET based on their situation and whether they believe the future is MVS, UNIX or a mixed shop.

Reporting/Tracing USS

CA-TOP SECRET has very good tracing and reporting facilities.

Tracing should be used carefully since it can significantly impact performance and generate considerable output. Tracing is usually only used at the direction of the CA support center. In USS, Command CHAUDIT is used to audit files and SECTRACE is used to trace USS requests[8]. Both are beyond the scope of this paper.

The reporting utility TSSOERPT[15] can and should be used to log and monitor USS activity. It runs as a normal MVS job and accesses USS data from the standard MVS system SMF (System Management Facilities) datasets not from the CA-TOP SECRET logs. SMF datasets are standard MVS system logging datasets maintained by system programmers. Report selection criteria are:

TITLE – 35 characters at the top of the page

LINECNT – Lines per page

SDATE & EDATE – Start and end date of the report

STIME & ETIME – Start and end time for report

UID - The UserID on which to collect data for the report

GID – The GroupID on which to collect data for the report

USER – The user ACID on which to collect data for the report

GROUP – Group ACID on which to collect data for the report

SEVICE – The callable service on which to collect data for the report. There are 54 documented.[15 pg9-18]

SUMMARY – Only prints three lines for each event

DETAIL – Prints all information available for each event

Following is a small report sample. The titles are original but the data has been modified for confidentiality.

```
03/03/04  04.063 06.42.06  -  OMVS/USS UNIX REPORT FOR MARCH   -  PAGE  12

    SERVICE            USERID   GROUP        UID()      GID()   SAF  RC   RSN
     DATE              TIME     JOBNAME   SOURCE   SYSID   CPU

 CHECK_ACCESS     ACID01   OMVSGRP         0         1       0      0      0
 03/03/04  04.063        05:09:18  MYPROGRAM          IBMSYS  SYS0
  Requested Access: Write
  Function: unlink              User Type: Security Defined Local User
  Pathname: /usr/local/myfile/tcat/apps/mcs/data/mcs/mydata1
  Filename: mydatafile
  Volume  : XYZ123  Owner: rwx   Group: r-x   Other: r-x
  File Identifier:   00110B0011B7330230
  Owning UID:        11      Owning GID:       1100
  User Audit Options   : Read Failure    Write Failure  Exec/Search Failure
  Auditor Audit Options: Read None     Write None     Exec/Search None
```

Conclusion

If you're an IBM mainframe installation and believe it is here to stay, setting Control Option HFSSEC to ON will enable CA-TOP SECRET to enforce most internal USS security. Putting the USS file permissions and MVS dataset permissions in the same profile will assist in keeping data access consistent between the two environments. UNIX System Services could be controlled using BPX permissions and superuser privileges by using UNIXPRIV resources. Although not all USS functions can be controlled by CA-TOP SECRET (NFS in particular), new releases will continue to include more USS security until everything will be in CA-TOP SECRET.

However, installations that believe UNIX is the future or that have a UNIX/Mainframe strategy in place that they want to employ for USS would set Control Option HFSSEC to OFF. The USS internal UNIX security will be enforced. In this case I would suggest pairing a Group ACID with a Profile ACID to keep the security between the environments as consistent as possible.

Whichever strategy you prefer, CA-TOP SECRET performs two significant functions:

> First, it controls access to USS by requiring UID and GID permissions be made to any ACID before it can log on to USS. This access should only be permitted ACIDs needing USS to do their job. This will greatly reduce security issues.
>
> Second, it provides good reporting. If internal USS security is in force, the TSSOERPT report utility should be run whenever the standard CA-TOP SECRET logging utility TSSUTIL is run. Review both reports frequently -- daily if possible -- if for no other reason than to become familiar with what is happening in USS and your mainframe. Even if you can't review them frequently, run them daily and archive them for at least a year. If an incident does occur, you can use them to determine what happened.

As stated earlier, this topic can be a life-long endeavor. I hope this paper gives you a starting point and a glimpse of what is possible. Develop your USS security strategy as early as possible and keep refining it as new releases make more options available.

References

[1] Mark Twain quote
http://www.twainquotes.com/Death.html

[2] IBM mainframe site
http://www-1.ibm.com/servers/eserver/zseries/

[3] Mainframe definition.
http://www.techweb.com/encyclopedia/defineterm?term=mainframe

[4] More mainframe information
http://en.wikipedia.org/wiki/IBM_mainframe

[5] eTrust[tm] CA-Top Secret Security for z/OP and OS/390
User Guide 5.3 sp01

[6] eTrust[tm] CA-Top Secret Security for z/OP and OS/390
Command Functions Guide 5.3 sp01

[7] eTrust[tm] CA-Top Secret Security for z/OP and OS/390
Control Options Guide 5.3 sp01

[8] eTrust[tm] CA-Top Secret Security for z/OP and OS/390
Security Cookbook May2003

[9] "Active Directory Features". June 15, 1999
http://www.microsoft.com/windows2000/server/evaluation/features/adlist.asp#heading2

[10] IBM UNIX System Services Planning manual GA22-7800-03

[11] IBM UNIX System Services Command Reference manual SA22-7802-04

[12] IBM UNIX System Services Users Guide manual SA22-7801-03

[13] OS/390 Version 2 Release 6
IBM UNIX System Services Implementation and Customization SG24-5178-00

[14] Debugging UNIX System Services, Lotus Domino, Novell Network Services and other Applications on OS/390

[15] eTrust[tm] CA-Top Secret Security for z/OP and OS/390
Control Options Guide 5.3 sp01

[16] "UNIX System Services – A Powerful Blend"
http://www-1.ibm.com/servers/eserver/zseries/zos/unix/release/nmas.html

[17] Donoghue, Andrew. "Mainframe: Still Going Strong after all these years"
http://insight.zdnet.co.uk/specials/upgrades/0,39021190,39116898,00.htm

[18] Mainframe definitions. Used in [3] for mainframe, but, other definitions are available. http://www.techweb.com/encyclopedia/