Security in the Solaris Environment

Howard I. Solomon
SunSoft, a Sun Microsystems, Inc. Business
his@east.sun.com
March 1, 1993

Executive Summary

Security safeguards in the Solaris environment are stronger than generally acknowledged. The most significant of these are the basic responsibilities of each individual user. A system superuser must administer a superset of these responsibilities to ensure a secure operating environment. A substantial suite of security features are bundled directly into Solaris although most of these are not enabled in the default configuration. These include the features needed to meet government C2 security requirements as well as the Data Encryption Standard (DES) for network–based authentication. The Sun SHIELD product family includes C2 security, the DES, secure Remote Procedure Call (RPC), Account Resource Management (ARM), and the Automated Security Enhancement Tool (ASET) which together provide a set of tools useful for security needs in most environments. For users with strict security requirements, Solaris CMW provides a labeled environment with secure X windows among its major features. Security is an important component of the Solaris 2.0 operating environment as well. In summary, Solaris–based systems can be configured to be secure enough to meet the needs of a wide variety of users.

Introduction

As UNIX has matured, so have the demands placed upon it. UNIX is no longer just the plaything of engineers; customers range from universities to banking houses, from government agencies to engineering teams. For each of these classes of users, security is becoming an ever–increasing issue. Today, many options are available for the security–conscious Solaris–based system user.

Although widespread misconceptions still exist, UNIX is actually rather secure. Many important security features are built into the operating system. However, it is important to realize that UNIX was designed to be open, flexible, and inviting. As such, many of the security mechanisms inherent in UNIX are not activated in the default configuration delivered to customers; the user or system administrator must activate those security features that are desired or required in a given environment. Further, it is important to keep in mind that security is as much the responsibility of the user and system administrator as the system software vendor. The world's best security implementation is of no use if the basic responsibilities of the individual are ignored.

This paper will serve as an introduction to security in the Solaris environment. In it, I will discuss several features of the operating system, relevant unbundled products that

assist in securing an environment, future directions, and resources available to assist users. Equally important will be the sections on user and administrator responsibility since these serve as the cornerstone to achieving the most basic of security goals; many of the basic features of UNIX security depend on the individual, be they administrator or user. In no way is this document designed to be all–inclusive; there are many books and papers on this subject as well as much knowledge in the minds of system administrators everywhere. This paper should serve to show the reader some of the important options available and whet the appetite for more research if security is of significant concern.

User Responsibility

The most critical component of a system's security rests with the individual user. Control of the most basic system actions rests here. Each user can individually become the weak link in the security chain of a given system environment. Proper education of users and enforcement of rules regulating account usage cannot be underestimated.

The first area of interest is login and password entries. Login entries are fairly benign as most environments use a particular system which is easy to guess such as initials or first initial followed by last name. These entries can be designed to not be easily guessed through devising a more complicated scheme such as a mix of numbers and characters for logins. Knowledge of login names, though, is not generally a serious threat. The protection provided by passwords puts less significance on login entries. However, using password entries that are easily guessed constitutes one of the strongest potentials for security breaches in computers. Simple examples of this include:

* spouse's (or anyone's name)
* driver's license, social security, or other commonly known number
* name of automobile or similar possession

and, for that matter, any

* dictionary entry.

Each site must determine its own regulations. Some may go so far as to require a combination of characters with numbers while many require a minimum number of characters or numbers. Programs that attempt to match passwords to dictionary entries in order to disqualify them are common. Password lifetime is another key issue. Many sites expire passwords in some locally determined fashion as password modification is necessary. Further, free distribution of one's password is equally dangerous. Each user must be sure to keep his/her password to his/herself. Also important is to not write down one's password as someone else may eventually find it. This last precaution is often weighed against the complexity requirements of the password. Sites must individually determine whether a more obscure password is desired or whether passwords that would never need to be written down are preferable. The Account Resource Management (ARM) product addresses many of these requirements and will be discussed below.

Discretionary Access Control (DAC) is an important cornerstone of the security model. The basic file permission structure most users take for granted addresses this important

security criterion. UNIX file permissions determine who can read, write, or execute a given file, or have access to a given directory at three levels of granularity: user, group, and everyone. Maintaining files at their appropriate access levels will do much to ensure a secure operating environment. Further, proper protection of executables and configuration files is equally important.

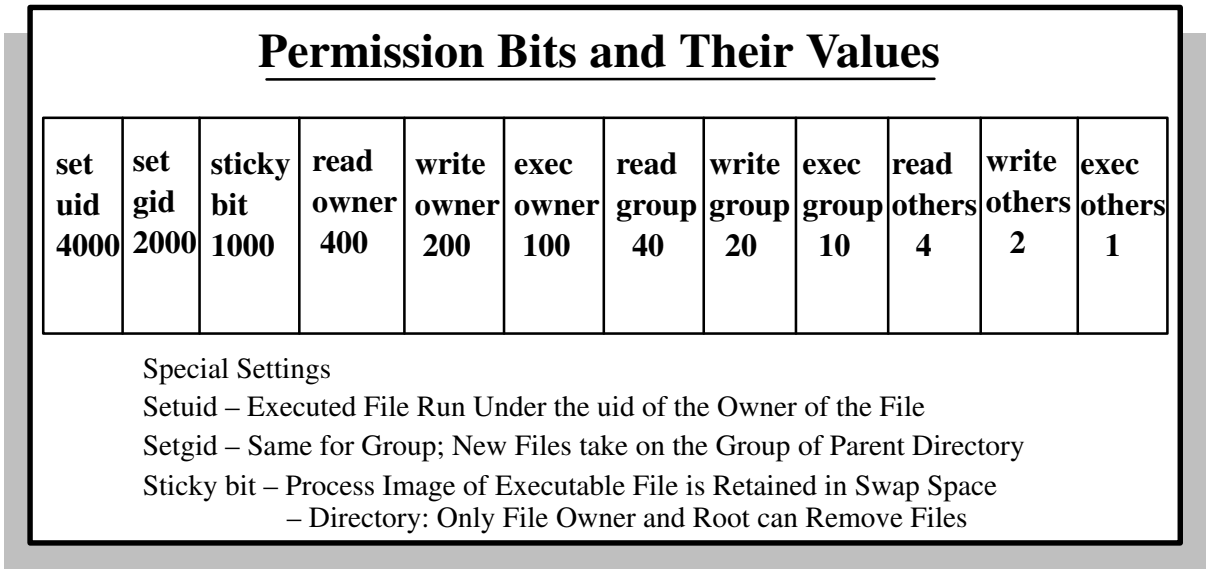

Figure 1 – Permission Bits and Their Values

In more secure environments, it becomes important to limit certain system operations. Programs which operate *setuid* or *setgid* can be dangerous as these are run under the *uid* or *gid* of the owner of the file. This may temporarily provide an individual user with increases access or authority. Users' configuration files (such as *.cshrc* and *.login*) and home directories should be owned and writable by that user only. Always logging out or setting a lockscreen is also important to insure that others cannot access one's data or system without permission. The environment check feature of the ASET product can search for the use of an insecure pathname variable or insecure file creation permissions as discussed below. The ARM product contains a feature to automatically logout a user or initiate a lockscreen when the system remains idle for a certain period of time as discussed below.

Although much of the information discussed here seems rudimentary, it is through misuse of basic user functionality that systems are most often at risk. It is important that all system users bear their individual responsibility toward ensuring a secure system. Use of advanced security capabilities is of little use if users allow more access than they should or are careless in their daily operations.

Superuser Responsibility

The system superuser, *root*, has more responsibilities than other users. Password protection and configuration takes on an added dimension of importance when access to

the entire system is at stake. Setting up user accounts, group environments, object modes, and filesystems must all be done with security in mind.

The superuser has a password as does any system user. Restricted access to this password, however, is much more significant as this account has free reign over the system. The root password must be well protected and access to it limited. The superuser also maintains the password and group files. It is his/her responsibility to ensure that no user accounts exist without passwords; this would constitute one of the most basic and dangerous security holes. In a more security–conscious environment, group entries in the */etc/group* file should also have passwords so that DAC prevents members from entering unauthorized groups, allowing access to users' files. The Automated Security Enhancement Tool (ASET) can check the system for missing passwords as discussed below.

The superuser has other responsibilities specific to system administration. Restricted access to files, directories, and devices is important. Access to the kernel, memory files, and device drivers should be limited to the superuser to prevent tampering. Programs that operate *setuid* or *setgid* should be used with caution as the program is executed as the owner, not as oneself. Programs that operate 'setuid root' can be especially dangerous as the person executing a given program "becomes" root for that period of time.

File systems can be mounted read–only to protect against accidental alteration when users do not need to be able to write there. Further, physical security of systems and peripherals is the responsibility of the administrator. CPUs, disks, tapes, and modems, etc., should be kept as secure as local needs dictate; these are often locked away in server rooms or the like.

Advanced superuser issues such as encryption and network trust must also be addressed. Domestic systems can use the optional Data Encryption Standard (DES) algorithm for basic data encryption with a public key. This product has been available in the United States and Canada as an unbundled option since SunOS 4.0 (pre–Solaris release). A different algorithm is used for encoding and decoding. Secure Remote Procedure Call (RPC) enhances network security through enhanced authentication via the DES. NFS and NIS, built upon the RPC mechanism, may use secure RPC to authenticate data access requests. Secret keys are passed between machines to validate users. Use of secure NFS mounts regulates access to network data. User and host names are verified through the DES discussed above.

Network trust is another key issue. The *.rhosts* file in a user's home directory allows network access to given users and machines without password verification. The */etc/hosts.equiv* file performs a similar function systemwide for remote machine trust. Use of these should be curtailed or avoided as they can subject a network to the insecurities of a given machine or user.

It is advisable to avoid or limit use of common directories that are readable by all, such as */tmp* and */usr/tmp*. Also, the current directory (.) should be the last entry in the pathname variable, $PATH, if present at all. The local directory is the most likely place for a worm

or other program that could compromise security. Proper administration of the *umask* through a setting of 27 or better is important. The *umask* determines the default DAC for files created by a given user. Labeling the console "insecure" in */etc/ttytab* further enhances security. This disables root login to the system, forcing root access via the *su* command. It ensures the ability to audit all work done as superuser through the individual user *uid*. This also requires a password for single user booting ensuring restricted access to a system that has crashed or has been brought down for some reason. Additionally, the system *eeprom* can be set as another method to require a password for single user booting. The ASET product can check for the current directory entry in the pathname variable, the *umask* setting, insecure ports, and the *eeprom* setting as discussed below.

C2 Level Security

Solaris, in response to the need for further security–related specifications, provides features required for government–approved computing. The Department of Defense (DoD) developed guidelines for varying levels of computer system security in the 1980s. These requirements are defined in the DoD Trusted Computer System Evaluation Criteria (TCSEC) manual (DoD 5200.28–STD) otherwise known as the "Orange Book" after the color of its cover. The National Computer Security Center (NCSC) of the National Security Agency (NSA) evaluates systems based on these criteria.

Orange Book Requirements

The Orange Book defines 7 different levels of trust:

D	Minimal protection
C1	Discretionary access control
C2	Authentication, auditing
B1	Mandatory access control, labels
B2	Configuration control, covert channel
B3	Access control lists
A1	Formal proofs

It is recognized as the standard by most of U. S. DoD and NATO.

Figure 2 – Orange Book Requirements

Meeting C2–level requirements is considered a baseline requirement for many government security–conscious computer users. This same feature set has application in the non–government sphere as well. The C2 requirements are varied. Important C2 features include DAC, object reuse, identification and authentication, auditing, assurances, and relevant documentation. The basic UNIX system includes features

which include DAC (discussed above) and user identification and authentication through the login/password paradigm. Object reuse ensures that no user or process can use resources which were previously used by another user or process without being cleared; this is also covered by basic UNIX operations. Auditing is a component of the C2 package which will be discussed below. Assurances basically guarantee that the system will be secure and operate in the manner it professes to operate. This includes system architectural issues, tools to verify system integrity, and testing of both the system functionality and possible penetration. The documentation package provides usage information as well as design, test, and functionality information.

Since SunOS 4.0, SunOS (now a Solaris component) has provided most C2–level security features. An update to the C2 feature set is provided as a software patch to Solaris 1.0 and Solaris 1.0.1. This patched version of the C2 package is also known as the Basic Security Module (BSM). The BSM auditing features form the base of SunOS CMW auditing, a product which will meet the Compartmented Mode Workstation (CMW) and B1 security specifications and which is discussed below. It is available from the SMCC Solution Center via the 1–800–USA4SUN phone line. It is also available for Solaris 2.1 as an option for a nominal media fee.

The majority of security–conscious customers, both government and otherwise, will find C2 security satisfactory. This feature set is general enough to be of use to commercial, university, and other customer environments; the features have widespread applicability and are used by a varied group of users from financial to technical.

The C2 feature set adds to the UNIX security base. First, passwords are more secure through the use of a file called */etc/passwd.adjunct*. While standard UNIX encrypts passwords, itself a basic security mechanism, the */etc/passwd* file is readable by anyone. Use of the */etc/passwd.adjunct* file splits the sensitive password information from the general background information of the */etc/passwd* file. The */etc/passwd.adjunct* file containing the encrypted passwords is only readable by root, further safeguarding the encrypted passwords. The C2 system also allows for requiring a password be entered for single user booting. This protects against someone halting a machine, rebooting it single user, and deleting the root password. This can be done through the */etc/ttytab* file or via a change in the system's *eeprom* as discussed above.

Auditing, a key component of C2 security, provides a record of system events for later examination. Activities can be traced to individual users of the system, providing accountability for system actions. Auditing can be performed on given commands, given users, or for all system activity. Commands exist which enable the system administrator to later analyze the audit data. It is important to ensure that the system has sufficient disk space allocated for audit data. If the file system(s) allocated to store audit data fill, the system will halt. This is intentional operating procedure as security concerns are paramount in this type of operating environment. It is therefore wise to audit selectively and clean out audit data regularly. The system does, however, warn the administrator at specific intervals that the audit space is near capacity.

One of the benefits of the BSM C2 package is its modularity. Individual site administrators can decide how secure to make their environment by implementing only desired features. For strict C2 operation, several things much be set in place. A site security policy should be defined. This includes several components:

* select limited number of trusted users
* provide desired physical security including machine and network cable access
* decide which users and which commands will be audited
* determine how and when audit data will be analyzed

Administration of a C2 system is critical to its success. All systems should be administered centrally with the Network Information Service (NIS). The NIS maintains network–wide databases of information such as the host file or the password file. This simplifies network administration, centralizing the maintenance of such database files. The BSM should be run on all machines on the network. Individual users should not have superuser passwords. A master */etc/fstab* file should be created providing a set of common mount points. All systems should utilize a subset of that file. The network can be physically separated, if desired. The password option of the boot PROM should be utilized insuring that crashing a system will not allow anyone to reboot and take over; "full" provides the highest level of security and is required for true C2 security. Root logins can be disallowed to insure that all audited actions can be traced to an individual user ID; allow root access only through the wheel group, group 0. The */etc/passwd.adjunct* file, which only allows root access to this repository for encrypted passwords, is required for auditing; the last two entries in this file are used for this purpose.

Device allocation is an important component of the BSM. This feature allows for secure sharing of devices and provides mutual exclusion. The ability to clean a device is a significant component of this feature. A prime example would be removing the last number dialed from a modem. Furthermore, device access can be restricted to certain users.

Inter–operation is also important. The BSM will interoperate with basic encryption and the ARM and ASET Sun SHIELD components discussed below. The Solaris BSM is an important component of a secure UNIX operating environment.

Sun SHIELD

The Sun SHIELD family of security products has been created to address widespread need for general security. Five products comprise the Sun SHIELD product family.

These are:

* C2 security

* the Data Encryption Standard (DES)
* Secure Remote Procedure Call (RPC)
* Account Resource Management (ARM)

and

* Automated Security Enhancement Tool (ASET).

The first three products are discussed elsewhere in this paper. The remaining two products are ARM and ASET. ARM addresses needs relating to system administration

security issues. The features of ARM include:

* password aging
* disabling logins after a certain number of invalid attempts
* account expiration
* system access hour limitation
* autolockscreen and autologout
* requiring password to logon to protected port

and

* password qualification of potential passwords.

These features cover a wide range of security requirements. The password aging feature determines how long the current password is valid and the minimum time allowed between changes. This protects against continued use of a given password by forcing it be changed and not allowing it to be immediately changed back. Upon password expiration, the system can be set to either notify the user of the expired password but allow login, run a program to prompt the user to change his/her password, or disable the account, disallowing login until after consultation with the system administrator. Account validity can be set for both an activation and expiration date. Any attempted use after the expiration date would result in immediate logout from the system or the system administrator could write a different program to use. Account disabling after invalid attempts can be set in a variety of ways on a per port, per user, or per domain basis, among others. When the specified number of attempts has been reached, account access is denied.

Autologout and autolockscreen safeguard against an idle system being accessed. After a defined period of inactivity, the system will either perform a logout or activate a lockscreen. Dialpass requires the entry of an additional password when certain ports are used. This can be used for dial–in modem ports or to access a login shell with permissions broader than that of the average user. Access hours can be set for particular accounts to limit the time of day when access to that account is permitted. Password qualification tests a selected password against the system password requirements. ARM provides an example password qualifier, but an alternate check can be substituted to enforce different verification standards.

ASET plays a complementary role. Its features include:

* automatically bringing system to low, medium, or high security setting
* tightening system file permissions
* checking system configuration files
* checking user and group information
* environment checking
* *eeprom* checking

and

* firewall function (which turns off IP forwarding and does not advertise routing information).

The ASET product places the system in one of three pre–determined security states: low, medium, and high. For a system in low security mode, ASET will ensure that attributes are set to standard release values. Several checks will be performed and any potential security weaknesses reported; system behavior is not altered in the low configuration. This should reduce the risk of security–related attacks.

The medium security setting provides adequate security for most environments. This operating environment will modify some system files' settings and parameters to restrict system access, reducing the security risks of the system. Security weaknesses and access modifications are reported. The medium security mode of operation does not affect system services.

The high security setting produces an extremely secure system. Many system files and parameters are set to minimum access levels. A few system applications and commands may be affected as security is the foremost concern. The firewall function is of particular effectiveness. The infamous Internet worm of 1988 had virtually no effect on the computer network at Sun Microsystems. This is because of the network environment in place at Sun. For security reasons, Sun has only one Internet connection; all communication must go through that system to enter or leave Sun. Further, IP forwarding is disabled on that machine and routing information is not advertised. The Internet worm attacked that system as it did all systems on the Internet. However, due to the security precautions in place, it could only affect the gateway machine; it could not propagate throughout Sun's internal network. The system administrator noticed performance degradation on this machine, diagnosed the problem, and removed the worm. This security precaution protected Sun from the intrusion many other organizations faced. The firewall component of ASET is a productization of much of this functionality.

Both ARM and ASET are configurable. This flexibility matches the product features to the desired environment for each customer. Capabilities can be assigned to different users, machines, etc., in the user configuration.

Compartmented Mode Workstation (CMW)

Solaris CMW is the flagship of offerings to make the Solaris client/server computing environment more secure. Compartmented Mode Workstation (CMW) is a specification put forth by the Defense Intelligence Agency (DIA) to specifically address the security needs of a workstation user. This system will meet the needs of government, intelligence, military, and commercial users who need security protection not available in standard commercial operating systems. Solaris CMW is based on Solaris 1.0 (SunOS 4.1.1) with substantial enhancements in the areas of security and administration.

Solaris CMW provides a broad set of security features. The product merges the security features and assurances of the DIA Security Requirements for System High and Compartmented Mode Workstations (DIA Document No. DDS–2600–5502–87) with the B1 level features of the NCSC Orange Book described above. In addition, certain Orange Book requirements from higher levels are included. Furthermore, Solaris CMW will be the only CMW product available which provides for a networked security model; Solaris CMW extends the notion of the Trusted Computing Base (TCB) to include a local area network of Sun workstations running Solaris CMW. This contrasts sharply with the standard CMW offering: a single workstation capable of being networked with no distributed computing security requirement. The Sun TCB is logically linked into one system model through the use of the Network Information Service (NIS) component of the Open Network Computing (ONC) de facto standard suite of upper layer networking products.

Major CMW features include:

* Mandatory Access Control (MAC) through Sensitivity Labels
  – Hierarchical 16 Classifications
  – Non–hierarchical 128 Compartments
* Information Labels
  – Hierarchical 16 Classifications
  – Non–hierarchical 128 Compartments
  – Non–hierarchical 128 Markings
* Trusted Facility Management
  – No longer an all–powerful privileged root or superuser
  – e.g.: Operator, Security Officer, System Administrator
* Least Privilege
  – a fine–grained privilege set
  – applications and processes run with minimum required privileges
* Distributed Trusted Computing Base over a network
* DNSIX (DODIIS Network Security Information Exchange)
* Multi–level Secure X11 window system
  – Includes OpenWindows look and feel
  – XView and OpenLook Intrinsics Toolkit (OLIT) Toolkits

* C2 and B1 Security Features
* Discretionary Access Control
* Accreditation Range
* Object Reuse

and

* Trusted Path.

Labeling determines the security levels of users, files, and other system objects. System objects include processes, files, windows, and the like. Solaris CMW includes two types of labels: sensitivity labels and information labels. Sensitivity labels are assigned to any system object and reflect the highest security level that object will attain.

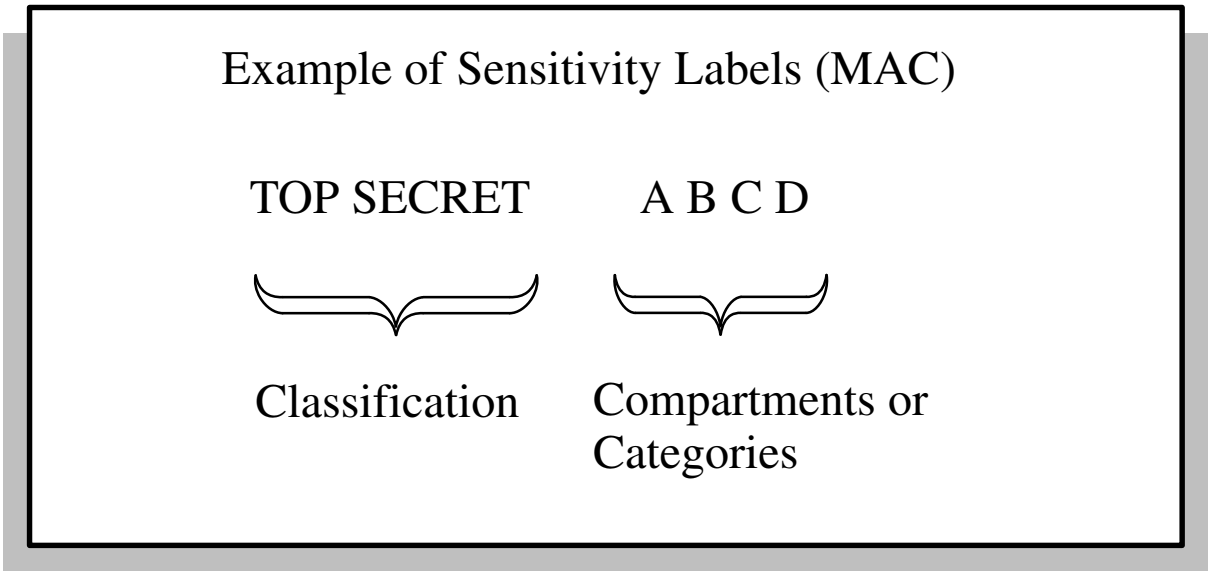

Figure 3 – Example of Sensitivity Labels (MAC)

Information labels are assigned to objects and reflect the actual classification of the object and any relevant handling caveats.

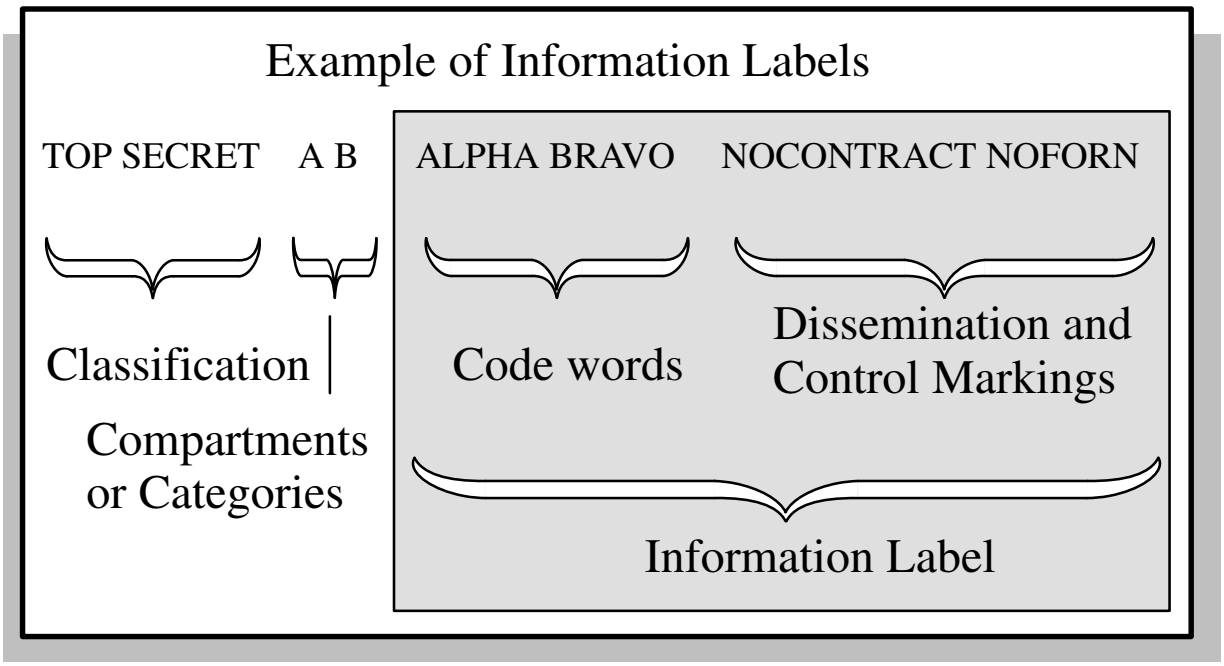

Figure 4 – Example of Information Labels

MAC is enforced through use of the sensitivity labels, determining who should have access to system information. DAC, discussed above, restricts access to system objects such as files via the UNIX permission mechanism.

System management is performed in a security–conscious way. The Trusted Facility Management (TFM) package consists of OPEN LOOK window tools, a trusted administrative command shell, and administrative utilities. These tools centrally manage the entire distributed system. By default, the TFM functions are divided into three roles: an Information System Security Officer (ISSO), a System Administrator, and an Operator. This separation of roles distributes the superuser's power among several individuals. The notion of UNIX superuser is replaced by the three administrative functions above.

The OpenWindows window system has been extended to provide a secure X environment. Each window is labeled and color coded to match its relevant sensitivity label. Solaris CMW allows standard OpenWindows features such as cut & paste and drag & drop. Transfer of data is mediated by a trusted selection agent. This agent prevents unauthorized data transfer and all attempts can be audited. Users may upgrade or downgrade information if they have been given such authorization by the security officer.

The windowing environment has a label stripe across the bottom called the screenstripe. This screenstripe provides information about the label status of the window system and of keyboard/cursor input. User access to the Solaris CMW TCB is only possible through a trusted path, preventing the user from being "spoofed". The LoginTool is the first

contact with the trusted path, but other security–related operations including entering passwords or modifying sensitivity and information labels also use this path. The window stripe across the bottom of the screen provides an indication of whether the user is communicating via the trusted path. Trusted path access is through this screenstripe; this is how authorized users assume administrative roles.

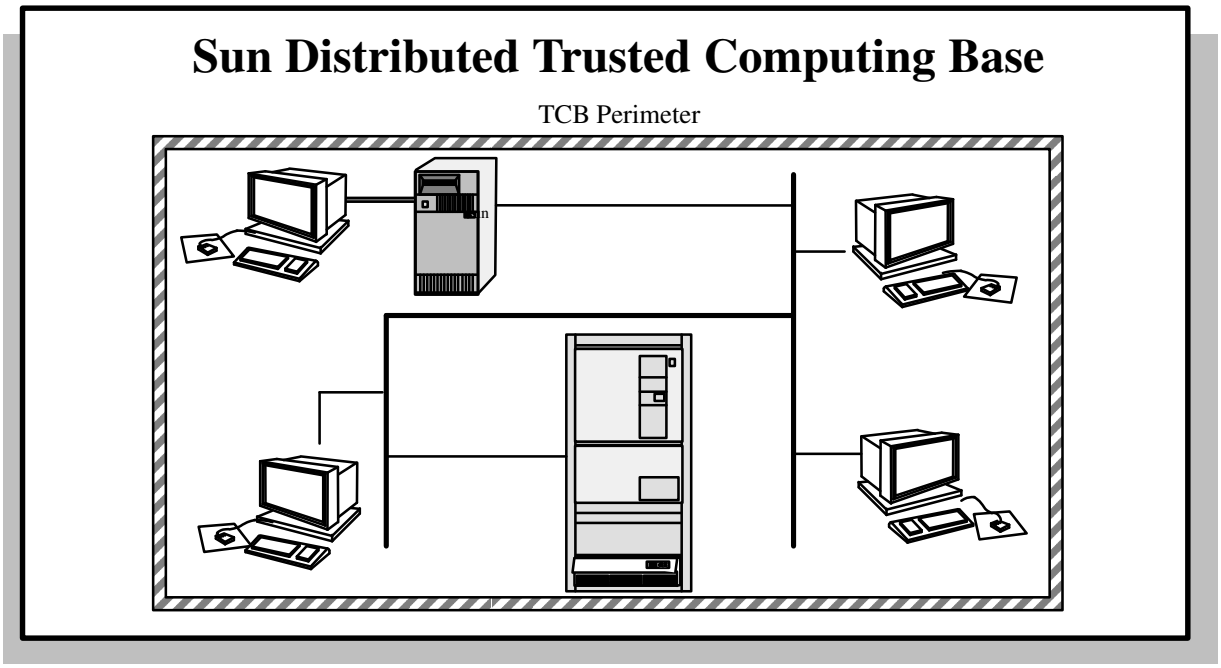

Figure 5 – Sun Distributed Trusted Computing Base (TCB)

An important aspect of Solaris CMW is networking. This is the only CMW product to extend the TCB across the network. This permits the use of established networking paradigms, such as NFS, NIS, RPC, etc., by security–conscious customers. The NFS filesystem model has been extended to allow labeling of filesystems, both locally and remotely. Diskless clients are fully supported using NFS. Support is also provided for both multi–level and single–level gateway functionality. Solaris CMW NeWSprint is available as an unbundled product which provides the first secure PostScript capability for laser printers. This product meets the CMW requirement for labeled headers and footers as well as front and back banner pages. Printers can be accessed from anywhere within the networked TCB.

Solaris CMW runs only on Sun SPARCsystems. Formal evaluation by the NCSC will be on seven models:

* SPARCstation 1, 1+, and 2
* SPARCstation IPX

* SPARCstation ELC

and

* SPARCsystem 470 and 490.

Other models can be accredited by a government Designated Approving Authority.

While designed to meet government requirements, this product holds relevance for the commercial user as well. The use of compartments equates nicely with the layout of an organization. I will use Sun Microsystems as an example. Within Sun are distinct business units such as SunExpress, SunSoft, Sun Microsystems Computer Corporation (SMCC), etc. Further, each business unit is divided into departments such as engineering, marketing, sales, and human resources. Much of the data held by any one organization should not be accessible by the others. The classifications are not particularly useful to this category of user. However, a single level system (ie: ignoring the classifications) with distinct compartments could serve a company like Sun well. Compartments covering each of the categories of employee mentioned above could provide adequate separation of classes of user within the corporation. This simple example demonstrates the usefulness of one of the CMW features in the commercial sphere. Other features are useful as well.

Security Information Resources

Sun provides a Customer Warning System (CWS) as a forum for action on Sun customer security needs. The CWS provides a mechanism for reporting security bugs or holes. Email is the preferred mechanism; emails can be sent to security–alert@sun.com. An alternative for those who cannot send email is to call (415) 688–9081. A Sun Security Coordinator exists to man the hotlines and evaluate security problems. A security “swat team” has been established to handle severe security concerns. Contacts at user sites can be established to receive CWS information. Further, security–related patches are available through dial–up access at Sun Microsystems. Additionally, Sun has written an “Introduction to Sun System Security” document available to users to provide a basic security overview.

The Computer Emergency Response Team/Coordination Center (CERT/CC) is another source of assistance. Carnegie Mellon has established this service to assist users who have a suspected security breach and to assist users with any security problems that might exist. They can be reached at (412) 268–7090.

Solaris 2.0

The new major revision of Solaris is based on System V Release 4 (SVR4). The first release of this new software base, Solaris 2.0, became available in June of 1992. This release is targeted primarily at Sun Original Equipment Manufacturers (OEMs), Independent Software Vendors (ISVs), and system administrators. The next Solaris

release is targeted to reach the general Solaris enduser community before the end of calendar year 1992.

Solaris 2.0 includes many important security features. There is a new file, /etc/shadow, that holds encrypted passwords and is only readable by the superuser. It includes the basic functionality currently in */etc/passwd.adjunct* as well as password aging controls. Aging controls include setting the number of days since the password last changed, the minimum number of days between changes, and the maximum number of days a password is valid. Another SVR4 feature is the expansion of restricted shells. This includes the C shell, the Bourne shell, and the Korn shell. A user can be placed in a directory which has limited functionality for a new or untrusted user. The user is not

allowed to perform any of the following:

* cd
* set $PATH
* specify path or command containing '/'

or

* redirect output (> and >>).

It also can leave the user in a non–writable and non–executable directory, only permitting the user to read files.

ASET functionality is bundled with Solaris 2.0. Plans are for ARM to be bundled into a future Solaris 2.x release. The C2 BSM will also be available for a future Solaris 2.x release. Solaris 2.0 also includes client support for Kerberos version 4.0, an alternate authentication mechanism that is currently popular. Kerberos, developed at MIT's Project Athena and based on a private key model, identifies network clients and requestors. Solaris 2.0 enables users to write Kerberos authenticated network applications or use Kerberos as an alternative to DES for RPC and NFS authentication. SunSoft, the Sun Microsystems, Inc. subsidiary which produces the operating system, will evaluate future directions for the product. One area already under investigation is SVR4 ES, the extended security product. Other technologies will be investigated as appropriate.

SunSoft, the Sun Microsystems, Inc. subsidiary which develops the operating system, has announced federated network services as a key element of the Solaris distributed computing roadmap. Federated security is an important aspect of the federated services framework, enabling flexible security services that extend to a global computing environment.

Network authentication and authorization are the cornerstones of network security in Solaris. Authentication involves verification of a client's identity for network operations and authorization refers to verification of the client's right to perform the desired operation. In addition to providing robust and reliable mechanisms for authentication,

Solaris gives the user the unique capability to choose from an array of authentication schemes. Solaris 2.0 includes UNIX, Diffie–Hellman, and Kerberos 4.0 mechanisms for RPC authentication. The Solaris network security product direction includes plans to provide a flexible model for customization by system and network administrators. Application developers will not have to choose an authentication scheme at the time of development. For example, an application can be 'kerberized' without changing the application source code or binary code. System administrators can choose an authentication flavor such as Diffie–Hellman, Kerberos, or RSA (a third party solution) at installation time or runtime of an application or network service.

SunSoft plans to make pluggable authentication services available to multi–vendor network services through an emerging standard that defines an Application Programming Interface (API) for such authentication interfaces. This is the Generic Security Services (GSS) API that is being defined by the Internet Engineering Task Force (IETF); it is being considered by POSIX and ISO for standardization.

Conclusion

Many varied options exist for users and administrators to make Solaris–based system environments more secure. While some features are part of the base operating system, all features are easily attained and implemented. All features operate within the distributed computing framework familiar to workstation users. It is important to remember that UNIX is open by design; activating the desired or required security mechanisms is a simple task. The user and administrator must be aware of their responsibility to address security in environments they create to help make the operating environment as secure as possible. Through the various security options available, from bundled functionality within Solaris 1.x through ARM and ASET to Solaris CMW and into the future with Solaris 2.x, workgroups as varied as government users, university networks, development groups, and commercial application developers can operate in the environment that meets their specific security requirements. Secure UNIX is not the contradiction its reputation might suggest.

Acknowledgements

Special thanks to Ginny Beyer, Joe Borgia, Dale Ferrario, Chuck Meyer, and Beverly Ulbrich, all of Sun Microsystems, Inc., for reviewing this paper and providing invaluable insight.

Many people have provided information that I have used in this paper including Larry Baron, Ginny Beyer, Joe Borgia, Alex Choy, Dale Ferrario, Laura Matthews, Chuck Meyer, Barry Shurtz, and Beverly Ulbrich, all of Sun Microsystems, Inc.

Several Sun documents, both internal and external, were helpful in the preparation of this document.

Trademarks

Sun Microsystems, Sun, Solaris, SunOS, SunSoft, NFS, NIS, RPC, and OpenWindows are trademarks of Sun Microsystems, Inc.

UNIX and OPEN LOOK are registered trademarks of UNIX Systems Laboratories, Inc.

PostScript is a registered trademark of Adobe Systems Incorporated.

The X Window System is a product of the Massachusetts Institute of Technology.

SPARC is a registered trademark of SPARC International, Inc.