MACINTOSH OS X
SECURITY TECHNICAL IMPLEMENTATION GUIDE
Version 1, Release 1

15 JUNE 2004

Developed by DISA for the DOD

"The STIG is intended to provide configuration guidance and is not to be construed as "endorsement/approval' for the use of Macintosh OS X. Per the DODD 8500.1, "All COTS IA or IA-enabled IT hardware, firmware, and software components or products incorporated into DOD information systems must comply with the evaluation and validation requirements of NSTISSP 11. Such products must be satisfactorily evaluated and validated either prior to purchase or as a condition of purchase; i.e., vendors will warrant, in their responses to a solicitation and as a condition of the contract, that the vendor's products will be satisfactorily validated within a period of time specified in the solicitation and the contract. Purchase contracts shall specify that product validation will be maintained for updated versions or modifications by subsequent evaluation or through participation in the National IA Partnership (NIAP) Assurance Maintenance Program." For exceptions to this policy, please see the updated NSTISSP 11 (July 2003) for specific guidance on Exemptions and Deferred Compliance.

TABLE OF CONTENTS

Page

1. INTRODUCTION 1
1.1 Background 1
1.2 Authority 1
1.3 Scope 2
1.4 Writing Conventions 2
1.5 STIG Distribution 2
1.6 Document Revisions 3

2. INTEGRITY 4
2.1 Hardware Integrity 4
2.1.1 System Equipment 4
2.2 Software Integrity 5
2.2.1 Free and Open Source Operating System Software 5
2.3 Data Integrity 6
2.3.1 File Integrity 6
2.3.2 Availability and File Location 7
2.4 Patch Control 7
2.4.1 DOD Patch Repository 8

3. DISCRETIONARY ACCESS CONTROL 10
3.1 User Account Controls 10
3.1.1 Interactive Users 11
3.1.2 Logon Warning Banner 11
3.1.3 Account Access 13
3.1.4 Inactivity Timeout 13
3.2 Password Controls 14
3.2.1 Password Guidelines 14
3.2.2 Keychains 15
3.3 Special Privilege Access 16
3.3.1 Root Account 16
3.3.2 Groups 18
3.4 Resource Controls 18
3.4.1 File and Directory Controls 18
3.4.1.1 Home Directories 21
3.4.1.2 Startup Files 22
3.4.2 Device Files 26
3.5 Special Purpose Access Modes 27
3.5.1 Set User ID (suid) 28
3.5.2 Set Group ID (sgid) 28
3.5.3 Sticky Bit 29
3.6 Umask 29
3.7 Development Systems 30
3.8 Default Accounts 31
3.9 Audit Requirements 31
3.10 Cron Access 33
3.10.1 Access Controls 33

3.10.2 Access Permissions and Owners 33
3.10.3 Cron on Mac OS X server 33
3.10.4 Locations 34
3.11 At Access 35
3.11.1 Access Controls 36
3.11.2 Access Permissions and Owners 36
3.11.3 At on Mac OS X Server 36

4. NETWORK SERVICES 38
4.1 Network Services Descriptions 39
4.1.1 Apache 39
4.1.2 Rlogin and rsh 39
4.1.3 Rexec Command 39
4.1.4 Finger 40
4.1.5 Remote Host Printing 40
4.1.6 Traceroute 40
4.1.7 Client Browser Requirements 41
4.2 Sendmail 42
4.3 Ftp 42
4.4 Trivial File Transfer Protocol (tftp) 43
4.5 Domain Name Service (DNS) 43
4.6 System Logging Daemon (syslogd) 44
4.7 Secure Shell (ssh) 44
4.8 Mac OS X Built-in Firewall 45

5. TRUST RELATIONSHIPS 46
5.1 Network Information Service (NIS) 46
5.2 Network File System (NFS) 46
5.3 Samba 48

APPENDICES

APPENDIX A. Related Publications 50
APPENDIX B. File and Directory Permissions Table 52
APPENDIX C. Procedures for Bringing a Mac OS X System Into STIG Compliance 53
APPENDIX D. Acronym 56

1. INTRODUCTION

The Macintosh (Mac) OS X Security Technical Implementation Guide (STIG) provides the technical security policies and requirements for deploying a secure Information System (IS) running Macintosh OS X in a Department of Defense (DOD) Network environment.

The intent of this Macintosh OS X STIG is to address security considerations for adding an IS running Mac OS X to a DOD network with an acceptable level of risk.

Most of the checks that are in this document are based on the UNIX side of the Macintosh OS. Some of these are carried over from the UNIX STIG and are designed to be a baseline for security. Included are several checks, which are specific to the Mac OS X side of the environment.

This STIG is designed for the Mac OS X 10.2 workstation and Mac OS X 10.2 server.
It should be noted that FSO Support for the STIGs, Checklists, and Tools is only available to DOD Customers.

1.1 Background

In its infancy, the Macintosh was looked at as a computer for the education and the home sectors. However, with the latest operating system being built upon the BSD UNIX kernel, the Macintosh is seeing a revival in the government and business sectors. With this new growth in mind, it is important to stay on top of the security situation and to assess any vulnerability that may be present. Mac OS X has both components of a workstation and a server therefore, it is important that it is locked down as both while keeping in mind that the standard user must be able to perform the day-to-day functions of their job.

The biggest advantage to using a Mac with OS X is that; it allows for the flexibility of the UNIX OS while offering the simplistic point-and-click options.

This advantage brings with it a big security issue as well. Mac OS X versatility makes it a powerful tool but that versatility can also make a system vulnerable. The UNIX OS has been developed to be open for modification and the Mac is no exception. The UNIX side as well as the Mac Interface side of the OS must be secure.

It should be noted that FSO Support for the STIGs, Checklists, and Tools is only available to DOD Customers.

1.2 Authority

DOD Directive 8500.1 requires that “all IA and IA-enabled IT products incorporated into DOD information systems shall be configured in accordance with DOD-approved security configuration guidelines” and tasks DISA to “develop and provide security configuration guidance for IA and IA-enabled IT products in coordination with Director, NSA.” This document is provided under the authority of DOD Directive 8500.1.

The use of the principles and guidelines in this STIG will provide an environment that meets or exceeds the security requirements of DOD systems operating at the MAC II Sensitive level, containing unclassified but sensitive information.

1.3 Scope

This document applies to all DOD administered or managed Macintosh systems. The requirements set forth in this document are designed to assist Information Systems Security Officers (IAOs) and System Administrators (SAs) in support of protecting DOD network infrastructures and resources.

It is important to note that even though the Mac OS X is based on BSD UNIX and all UNIX systems share common characteristics, they each implement features differently. They do not all implement the same features, and use different methods for implementing some of the same features. This document is limited to the Mac OS X 10.2 system, although additional system support will be included as necessary.

1.4 Writing Conventions

Throughout this document, statements are written using words such as "**will**" and "**should**." The following paragraphs are intended to clarify how these STIG statements are to be interpreted.

A reference that uses "**will**" implies mandatory compliance. All requirements of this kind will also be documented in the italicized policy statements in bullet format, which follow the topic paragraph. This will make all "**will**" statements easier to locate and interpret from the context of the topic. The IAO will adhere to the instruction as written. Only an extension issued by the Designated Approving Authority (DAA) will table this requirement. The extension will normally have an expiration date, and does not relieve the IAO from continuing their efforts to satisfy the requirement.

A reference to "**should**" is considered a recommendation that further enhances the security posture of the site. These recommended actions will be documented in the text paragraphs but not in the italicized policy bullets. Nevertheless, all reasonable attempts to meet this criterion will be made.

For each italicized policy bullet, the text will be preceded by parentheses containing the italicized Short Description Identifier (SDID), which corresponds to an item on the checklist and the severity code of the bulleted item. An example of this will be as follows "(*G111: CAT II*). "If the item presently has no Potential Discrepancy Item (PDI), or the PDI is being developed, it will contain a preliminary severity code and "N/A" for the SDID (i.e., "[*N/A: CAT III]*").

1.5 STIG Distribution

Parties within the DOD and Federal Government's computing environments can obtain the applicable STIG from the Information Assurance Support Environment (IASE) web site. This site contains the latest copies of any STIG, as well as checklists, scripts, and other related security information.

The NIPRNet URL for the IASE site is http://iase.disa.mil/. The Secret Internet Protocol Router Network (SIPRNet) URL is http://iase.disa.smil.mil/. Access to the STIGs on the IASE web server requires a network connection that originates from a **.mil** or **.gov** address. The STIGs are available to users that do not originate from a **.mil** or **.gov** address by contacting the FSO Support Desk at DSN 570-9264, commercial 717-267-9264, or e-mail to **fso_spt@ritchie.disa.mil.**

1.6 Document Revisions

Comments or proposed revisions to this document should be sent via e-mail to **fso_spt@ritchie.disa.mil**. DISA FSO will coordinate all change requests with the relevant DOD organizations before inclusion in this document.

2. INTEGRITY

Sites achieve Mac OS X system integrity by managing the overall processing environment. Proper security and system management protects system hardware, software, applications, and data from unauthorized access and improper modification and leads to the secure operation of Mac OS X systems. Total system integrity is most vulnerable to malicious intrusion before systems have been completely configured for secure operation. Newly built or configured systems could have their data integrity compromised as soon as they are connected to a production network if they are not STIG compliant before connection to the production network. All Mac OS X systems will conform to the security directives in this STIG before they are connected to a production network.

2.1 Hardware Integrity

Hardware resources include Central Processing Units (CPUs), Direct Access Storage Devices (DASDs), terminals, X terminals, workstations, and printers. A Security vulnerability may be created in the operating environment when any hardware component is incorrectly installed, operated, or maintained.

Controlling access to hardware resources is essential. Access control reduces the risk of theft, damage, and unauthorized access. Specific installation guidelines apply to classified equipment.

The operating environment must be capable of protecting the integrity of the hardware through physical means. The following sections define the hardware integrity requirements.

2.1.1 System Equipment

The Mac OS X operating system resides on, stores information on, and is accessed by, a number of different devices. The devices associated with the Mac OS X operating system must be protected. A person familiar with Mac OS X, who has physical access to a CPU, can boot the system in the single-user mode with the default root shell. In the single-user mode, the standard Mac OS X Identification and Authentication (I&A) process can be bypassed. (Configure all systems that support the requirement for single-user passwords to support that feature.)

To provide minimal physical protection for other systems and certain peripherals, locate them in a controlled access area that requires positive identification (i.e., a swipe card) for entry. The IAO will document and justify all deviations from this requirement. All systems will be furnished with a maintenance log. Enter all single-user and maintenance actions in the maintenance log to provide a history of actions that may be needed for possible recovery operations.

- *(OSX1026PYS0001: CAT II) The IAO will ensure that all Mac OS X system equipment (e.g., CPUs, storage devices, consoles) is physically located within a controlled access area.*

- *(OSX1026PYS0002: CAT II) The IAO will document the location, access method, authorized user(s), and reason for placement of any Mac OS X system equipment (e.g., workstations, terminals) that is not physically located within a controlled access area.*

- *(OSX1026PYS0003: CAT II) The IAO and SA will ensure that all Mac OS X systems are configured, where possible, to require a password for access to single-user and maintenance modes.*

- *(OSX1026PYS0003: CAT II) The following requirements will apply to all Mac OS X systems that cannot be configured for password access to single-user and maintenance modes:*

  - *The IAO will maintain a list of all such systems.*

  - *Each system will be protected in a manner that precludes physical access by anyone but authorized System Administrators.*

  - *A maintenance log noting the date, time of day, name of authorized System Administrator(s), and purpose for single-user or maintenance mode access will be maintained for each system.*

2.2 Software Integrity

2.2.1 Free and Open Source Operating System Software

Open Source Software

DOD has clarified policy on the use of open source software to take advantage of the capabilities available in the Open Source community as long as certain prerequisites are met. DOD no longer requires that operating system software be obtained through a valid vendor channel and have a formal support path if the source code for the operating system is publicly available for review.

Open source software takes several forms:

1. A utility that has publicly available source code is **acceptable**.

2. A commercial product that incorporates open source software is **acceptable** because the commercial vendor provides a warranty.

3. Vendor supported open source software is **acceptable**.

4. A utility that comes compiled and has no warranty is **not acceptable**.

Mac OS X is acceptable based on the availability of source code, in some instances, and the support and guarantee of the vendor (i.e., Redhat) and the support and guarantee of vendors who incorporate the software in their common release. However, any operating system in use in a production environment must be capable of STIG compliance as verified by an SRR. Operating systems are still subject to the requirements for NIAP certification described in DODI 8500.2.

Freeware and Shareware

Public Domain software distributed as freeware or shareware where the software is only publicly released in a compiled form and the product is not supported by any vendor is still restricted for use in DOD. Department of Defense Directive (DODD) 8500.1, Section 4.19 states:

> *"Public domain software products, and other software products with limited or no warranty, such as those commonly known as freeware or shareware, shall only be used in DOD information systems to meet compelling operational requirements. Such products shall be thoroughly assessed for risk and accepted for use by the responsible DAA."*

These requirements conform to the spirit of the draft policy memorandum from the Assistant Secretary of Defense, *Guidance and Policy for Department of Defense Information Assurance, 24 June 1999, ASD (C3I)*. *Paragraph 4.11* requires COTS and GOTS security-related software, hardware, and firmware, to be evaluated prior to purchase.

2.3 Data Integrity

This *Mac OS X Security Technical Implementation Guide (STIG)* is not intended to address data-level integrity in detail, but to provide techniques that can be used to ensure security of the data residing under the control of Mac OS X.

File system controls play a critical role in maintaining the integrity of Mac OS X systems. Several key areas of control requirements are discussed in the following sections.

2.3.1 File Integrity

File integrity is a key factor in the protection of Mac OS X systems. System files that must be protected are found in *APPENDIX B. FILE AND DIRECTORY PERMISSIONS TABLE*.

The SA will ensure permissions for the files listed in *APPENDIX B. FILE AND DIRECTORY PERMISSIONS TABLE are implemented*.

- *(OSX1026SVR0001: CAT II) The SA will for the Mac OS X server be responsible for checking and reporting to the IAO the integrity status of system/sensitive files on a weekly basis.*

- *(OSX1026SVR0002: CAT III) The SA will for the Mac OS X server ensure access to the software that performs this function is granted only to a limited number of privileged accounts.*

- *(OSX1026SVR0003: CAT II) The SA will for the Mac OS X server ensure methods used to check file integrity will alert the SA and the IAO via email if a security breach or a suspected security breach is discovered.*

- *(OSX1026GEN0106: CAT II) The IAO and the SA will for the Mac OS X server be responsible for initiating actions when file integrity breaches are detected.*

- *(OSX1026SVR0041: CAT III) The SA will for the Mac OS X server ensure a procedure is in place to set the system date and time-of-day daily*

NOTE: This may be accomplished through a local time server.

2.3.2 Availability and File Location

Data availability is the ability to deliver timely, reliable access to data and information services for authorized users. This can depend on many things, such as hardware availability, but data location, especially where and how backup data is maintained, is sometimes an overlooked factor in data availability. Ensure the effects of hardware failures on system availability are minimized. Avoid collocation of files such as primary and alternate databases or online and backup data files. The loss of a physical volume containing data should not also cause the loss of the backup data because of their collocation. The following only applies if you are running Mac OS X server.

- *(N/A: CAT II) The SA will ensure backup/baseline files are not located on the same physical device/medium as the primary/online files.*

- *(N/A: CAT II) The SA will ensure that the system backup of OS data is performed on a daily basis and the backups retained for at least one month.*

2.4 Patch Control

Maintaining the security of a Mac OS X system requires frequent reviews of security bulletins from http://www.apple.com and then going under the OSX tab and then on to the Updates page. Many security bulletins mandate the installation of a software patch (**Software Updates**) to overcome security vulnerabilities.

SAs and IAOs should regularly check OS vendor web sites for information on new security patches that are applicable to their site. All applicable security patches will be applied to the system. A security patch is deemed applicable if the product is installed, even if it is not used or is disabled.

FSO does not test or approve patches or service packs. It is the site's responsibility to test vendor patches within their test environment

- *The IAO will ensure that the Standard Operating Procedure (SOP) for each system includes the requirement to monitor Department of Defense Computer Emergency Response Team (DOD-CERT) bulletins at http://www.cert.mil. Select the link to the DOD-CERT bulletins.*

- *(N/A: CAT II) The IAO and SA will subscribe to the DOD-CERT/VMS bulletin mailing list.*

- *(OSX1026GEN0101: CAT I) The IAO will ensure that Mac OS X has the most current Security Patch installed.*

- *(OSX1026GEN0105: CAT I) The IAO will ensure that all security related software patches and updates are applied and documented.*

- *(OSX1026GEN0106: CAT II) The IAO will ensure that the latest OS service packs are applied and documented*.

NOTE: Generally a couple of months will be allowed for any problems to be reported to the vendor prior to new service packs being required.

2.4.1 DOD Patch Repository

DISA maintains a repository of software patches and hot fixes that is available for downloading fixes for IAVM related vulnerabilities. At current most Mac OS X fixes that are listed on this site pertain to Microsoft ONLY fixes that are under the other software category of the Microsoft Section.

This patch server can be accessed at the following locations:

https://patches.csd.disa.mil for NIPRNet
https://patches.csd.disa.smil.mil for SIPRNet

This page is intentionally left blank.

3. DISCRETIONARY ACCESS CONTROL

This section discusses discretionary access control (DAC) and the Identification and Authentication (I&A) criteria necessary to ensure that access to system resources is effectively managed and controlled for the Mac OS X system. In this sense, it is also discussing confidentiality, which consists of assurance that information is not disclosed to unauthorized persons, processes, or devices. This entails the concept of "least privilege" necessary to accomplish authorized tasks. *Least privilege* includes confidentiality, integrity, and availability, and states that users have only the authority to access those resources necessary to perform their functions. DAC places a large part of the responsibility for data confidentially, integrity and availability directly into the data owners hands by relegating to the owner the ability to determine who can access his data and how they may access it (read, write/delete). This STIG attempts to provide secure methods of accomplishing DAC, and other operations, while still protecting the data owner, the data user, and the platform's operating system.

3.1 User Account Controls

DOD directives require unique identification for each system user. Authorized users should be granted access only to the resources needed to accomplish the mission. A user is either an individual or an executing process/task that accesses a computer resource. The account name and corresponding user identification number (uid) identifies the user. Typically, *uids* are assigned according to the following scheme:

- Privileged *uids* generally range from **0** to **20**.
- Application *uids* generally range from **100** to **999**.
- Interactive/normal *uids* generally range above **1000**.
- Some systems reserve *uids* and *gids* (group identification numbers) from **0** to **30**.

Security requires individual user accountability. This precludes the use of shared accounts (accounts where multiple users are allowed to log on directly to the same account). Applications may require that a specific account be used for certain administration tasks. The user will still be required to log on with that user's account name and **su** to the application account. That action retains the individual accountability (through audit files). If there is an absolute requirement for logging directly into an account the IAO will obtain justification and documentation from the vendor that states the necessity.

- *(OSX1026GEN0006: CAT IV) The IAO will ensure that shared accounts within the Mac OS X server are not being used.*

***NOTE**:* If shared accounts are need for an application the IAO will document the shared account and the application need.

- *(OSX1026GEN0007: CAT II) The IAO will ensure a shared account within the Mac OS X server logon will be accomplished by invoking the su - (switch user) command from an individual user's Terminal.*

- *(OSX1026SVR0007: CAT II) The SA will ensure that an account will be locked after 35 days of account inactivity.*

- *(OSX1026SVR0008: CAT II) The SA will ensure that an account will be disabled after 35 days of inactivity. Account information, files, etc., will be retained for one year, if deemed necessary. The owner of the files will be changed to root, in the interim, and the IAO and SA will ensure none of the Mac OS X server files violates the requirements for files owned by root.*

3.1.1 Interactive Users

8500.2 security requires all users accomplish identification and authentication (I&A) to a computing system with a minimum of a legitimate, authenticated account name and password pair before access to computing resources is granted. The IAO controls access to Mac OS X resources by authorizing functionality to accounts as documented on proper documentation received from the prospective user's supervisor. For DOD personnel, that documentation will be a *DD Form 2875 or an equivalent form*. The IAO will direct the SA to assign unprivileged users a *uid* greater than 20 (unprivileged user uids generally begin with 1000), and a primary *GID* greater than 19 (unprivileged user GIDs usually start at 100). Users may be assigned to more than one group, as necessary. Systems usually reserve the first 30 uids and gids for system use.

- *(OSX1026GEN008: CAT III) The SA will ensure each user is assigned a unique account name.*

- *(OSX1026GEN0009: CAT II) The SA will ensure each user is assigned a unique uid.*

- *(OSX1026ADM0005: CAT II) The IAO will ensure all user access rights are documented on DD Form 2875 or an equivalent form.*

3.1.2 Logon Warning Banner

Recent criminal court cases involving unauthorized access to official Government computer systems has prompted the need for a logon warning banner to be presented to anyone accessing a Government computer system. Refer to *3.1.2.1 Logon Warning Banner Implementation*. The following points must be made in the banner: 1) The system is a DOD system. 2) The system is subject to monitoring. 3) Monitoring is authorized in accordance with applicable laws and regulations and conducted for purposes of systems management and protection, protection against improper or unauthorized use or access, and verification of applicable security features or procedures. 4) Use of the system constitutes consent to monitoring. 5) The system is for authorized US government use only.

- *(OSX1026GEN0010: CAT II) The SA will ensure a logon-warning banner is displayed on all devices that allow application or command-level access.*

- *(OSX1026GEN0010: CAT II) The SA will ensure a logon-warning banner is displayed before the actual logon attempt is made.*

- *(OSX1026GEN0011: CAT II) The IAO will ensure the Legal Notice Logon Warning Banner includes the following five points:*

  - *The system is a DOD system.*

  - *The system is subject to monitoring.*

  - *Monitoring is authorized in accordance with applicable laws and regulations and conducted for purposes of systems management and protection, protection against improper or unauthorized use or access, and verification of applicable security features or procedures.*

  - *Use of the system constitutes consent to monitoring.*

  - *This system is for authorized US government use only.*

3.1.2.1 Logon Warning Banner Implementation

System supplied methods to display logon banners vary. For Mac OS X this will be implemented in two ways. Using both of these ways will cover the warning banner for both the Aqua interface and the terminal login.

Add a line to the .plist of the login window to show the Warning Banner text on the same screen where the user enters their username and password.

- By Using the Property List Editor
  - Open the Property List Editor.
  - Open the /Library/Prefernces/com.apple.loginwindow.plist file.
  - Expand Root.
  - Highlight Root and Select the New Child Button.
  - Add LoginwindowText as its name.
  - Place the Warning Banner Text in the Value Field.
  - Save the plist file.

- Editing the motd file to show the Warning Banner after a terminal session is invoked. The system displays the contents of this file via the global /etc/.login and the /etc/profile files, depending on which shell is started.

- Edit, or create, /etc/motd.
  - Insert the banner text.
  - Write **/etc/motd**.
  - Chmod **444 /etc/motd**.
  - Chown root **/etc/motd**.
  - Chgrp sys (or bin) **/etc/motd**.

NOTE: It is important to note that if a service is removed from a Mac OS X machine, the machine will be more secure and a warning banner will not need to be added on that

service. If there are unneeded open services, remove them and comment them out in the /etc/**inetd.conf** file.

3.1.3 Account Access

Many computer compromises occur as the result of account name and password guessing. Someone with an automated script that uses repeated logon attempts until the correct account and password pair are guessed generally does this. Logon and logout logs (for users as well as root), session locking, and session disconnect are effective methods of controlling potential malicious account access. Some systems do not support account lockout. Some systems disconnect a session after three consecutive failed logon attempts. Some systems allow five attempts before initiating a session disconnect. If a system allows five consecutive failed logon attempts before disconnect, to provide a larger margin of safety, increase the delay between logon attempts to four seconds, where possible.

- *(OSX1026GEN0012: CAT II) The SA will ensure that all logon attempts (both successful and unsuccessful) will be logged to a system log file (e.g., /var/log/logins.log).*

- *(OSX1026NET0001: CAT II) The SA will ensure that the systems that support actions based on three failed logon attempts will be configured for a delay of at least two seconds between logon attempts.*

NOTE: This is supported on Mac OS X server and with any Mac using authentication from a Windows Domain.

- *(OSX1026NET0002: CAT II) The SA will ensure, after the supported number of consecutive failed logon attempts for an account, the account is locked until the IAO unlocks it or the system unlocks it after a minimum 30-minute delay.*

NOTE: This is supported on Mac OS X server and with any Mac using authentication from a Windows Domain.

- *(OSX1026NET0003: CAT II) The IAO will review the circumstances causing locked accounts to ensure there are no security concerns.*

3.1.4 Inactivity Timeout

Whenever a user is logged on to a Mac OS X system, the system is susceptible to alteration or damage. A user may become busy or distracted and inadvertently leave a logon session unattended or a user process may be left *orphaned* by some unforeseen circumstance. Such idle sessions leave the Mac OS X system vulnerable to unauthorized user exploitation. Screen lock programs can be configured to activate if terminals are idle for a specified period. If a screen lock device is available, it should be able to be invoked by the user when the user wishes to leave the terminal unattended. The IAO and individual users should work together to determine and implement the correct inactivity timeout for their needs.

- *(OSX1026SEC0003: CAT II) The SA will configure the screen-lock out feature to log out interactive processes (i.e., terminal session) after 15 minutes of inactivity unless a password protected screen lock mechanism is used and is set to lock the screen after 15 minutes of inactivity.*

- *(OSX1026SVR0009: CAT II) The SA will ensure applications executing on Mac OS X servers requiring continuous, real-time screen display (i.e., network management products) will be exempt from the inactivity requirement provided the following requirements are met:*

  - *The logon session is not a root session.*
  - *The inactivity exemption is justified and documented with the IAO.*
  - *The display station (i.e., keyboard, CRT) is located in a controlled access area.*

The Mac OS X operating system uses a tool called the Screen Lock from within the program Keychain Access (which is different then actually using the keychain function) to allow the user the ability to manually lock the screen. This will be added to the menu bar so that the user just needs to click on the menu item and hit Lock Screen. See *APPENDIX C: PROCEDURES FOR BRINGING A MAC OS X SYSTEM INTO STIG COMPLIANCE*.

- *(OSX1026GEN0080: CAT II) The SA will ensure that the Lock Screen feature is added to the menu bar.*

3.2 Password Controls

Mac OS X operating systems allow specification of a password. The following guidelines will be used for password creation.

3.2.1 Password Guidelines

Users must take precautions to protect passwords by choosing passwords wisely. Studies show that users are more likely to remember their passwords if they are allowed to choose them themselves. Passwords so complex or obscure that they require being written down introduce the hazard of becoming accessible to unauthorized persons. The following rule will be used in password creation: The IAO will ensure all passwords will be a minimum of eight alphanumeric characters in length and will include at least one capital letter, one lower case letter, one numeric character, and one special character.

- *(OSX1026GEN0019: CAT II) The IAO will ensure all passwords are a minimum of eight alphanumeric characters in length and will include at least one capital letter, one lower case letter, one numeric character, and one special character.*

- *(OSX1026GEN0019: CAT II) The IAO will ensure all passwords do not contain personal information such as names, telephone numbers, account names, dictionary words, etc.*

- *(OSX1026GEN0019: CAT II) The IAO will ensure all passwords do not contain consecutively repeating characters.*

- *(OSX1026GEN0004: CAT II) The SA will ensure user passwords are changed every 90 days.*

- *(OSX1026GEN0020: CAT II) The SA will ensure passwords are not be reused within 10 password changes.*

- *(OSX1026ADM0005: CAT III) The SA will ensure application passwords are changed at least once a year and anytime an application administrator is reassigned. This includes ftp account passwords for ftp accounts used by applications or users.*

- *(N/A: CAT II) The IAO will ensure if a system cannot be configured to automatically enforce the above password directives, that users are properly trained in password policy and proper password construction.*

***NOTE**:* The training requirements will be a part of the standard operating procedure (SOP) documentation.

- *(OSX1026GEN0019: CAT II) The SA will ensure the root password is changed on the same 90-day schedule as for users.*

- *(OSX1026ADM0006: CAT III) The IAO will ensure the root password is changed whenever someone who knows the root password is reassigned.*

- *(OSX1026ADM0009: CAT II) The IAO will be responsible for updating the documentation and storage of root passwords whenever the root password changes.*

- *(N/A: CAT II) The IAO will limit the number of people who know the root password to security and administrative personnel.*

- *(N/A: CAT II) The SA will assign the Open Firmware Application, system monitor (which can be used Mac on OS X server), and other privileged user passwords, and they will be treated the same as root passwords.*

- *(OSX1026ADM0007: CAT II) The SA will ensure that the root account is disabled after the password is changed to meet the strong encryption requirements.*

NOTE: This is done from the Netinfo Manager or from the command line.

The Mac OS cannot be configured to automatically enforce the above password directives for local passwords; the IAO will ensure that users are properly trained in password policy and proper password construction. However, this only pertains to the local accounts of the machines. If the machine connects to a Windows or Mac OS X server then the network password will conform to standards.

3.2.2 Keychains

Another security related item that is of interest is the Keychain Feature of the Mac and its applications. Keychains can be saved and transferred from machine to machine for easy access

to things such as websites and network shares.

- *(N/A: CAT II) The SA will ensure that Mac OS X Keychains are NOT allowed within the DOD.*

3.3 Special Privilege Access

Mac OS X systems provide special privileges that, when assigned to an account, allows the owner of the account to modify the security environment, perform auditing tasks, and perform functions that could circumvent security requirements. Therefore, no account will be granted privileged access unless authorized by the IAO. A privileged account is an account with a *uid* of 20 or less, or a group with a *gid* of 19 or less, depending on the system defaults. This can be modified and checked through the Netinfo Manager under the Utilities folder in that Applications directory.

- *(OSX1026SEC0101: CAT II) The IAO will authorize all privileged accounts (i.e., accounts with a uid less than or equal to 20), but only upon receipt of written documentation signed by the user's supervisory personnel. For DOD, the documentation will be a DD Form 2875 or an equivalent form.*

- *(OSX1026SEC0102: CAT II) The IAM, or site security office, will maintain separate documentation to identify all privileged accounts and list the privileges the accounts possess. For DOD, all account information will be documented on a DD Form 2875 or an equivalent form. This is both for Mac OS X server and for any special privilege accounts needed on a Mac OS X workstation.*

3.3.1 Root Account

The root account is used to accomplish system administrative functions. The system uses the account to run privileged programs. Because root enjoys access to all files and programs, root has no security constraints.

By default, the root home directory is "/" which is readable by all Mac OS X users. It is desirable to have the root home directory in a directory other than "/" to afford root's startup and work files the same protection as is afforded to all other users.

Sites usually designate one or more *primary* and *alternate* System Administrators who require root access. The sharing of the root account and password results in a breach of the DODI 8500.2 IAIA 1/2 security requirements for individual I&A and audit requirements. Enforcing a requirement where users log on with their individual account and use the su - command, can minimize the individual breach. Use of the su - command and the /var/adm/authlog file results in the ability to identify a user who uses a shared account (particularly the root account) and to audit their actions.
The only user with a *uid* of 0 will be root. If another *uid* of 0 is in the password file, it may be an indication of system compromise.

There may be several accounts that are root capable (i.e., they are alternate administrators who share the root password and are able to switch user to root). These accounts will be bound by the

same restrictions of the root account. They will log on to their named accounts. They will invoke the "su –" command to reach root, if necessary. Their PATHs will be the same as the root PATH once the command is completed. In any case, their personal PATH statement will be bound by the same restrictions as the root PATH statement. This restriction protects against the root capable account accidentally typing "**su**" instead of "**su –**", and dragging a default environment with an incorrect PATH variable along with it.

- *(OSX1026GEN0026: CAT II) The IAO will enforce users requiring root privileges to log on to their personal account and invoke the su - command to switch user to root.*

- *(OSX1026GEN0021: CAT II) The SA will ensure only root has a uid of 0.*

- *(OSX1026GEN0022: CAT IV) The SA will ensure root is assigned a home directory other than "/" (such as /roothome) and the directory will have permanent permissions of 0700.*

NOTE: Do not change the permissions of the "/" directory to anything other than 0755.

- *(OSX1026GEN0022: CAT IV) The SA will ensure that the root home directory has permanent permissions of 0700.*

- *(OSX1026GEN0024: CAT II) The SA will ensure the root search PATH (and the search path of root capable accounts) does not contain "**.**", "**::**", or start or end with a "**:**".*

NOTE: All are equivalent to "**.**".

- *(OSX1026GEN0025: CAT II) The SA will ensure root's PATH (and the search path of root capable accounts) does not contain directories or files that are world writable.*

- *(OSX1026GEN0026: CAT II) The SA will ensure root can only log on "as root" from the system console, and then, only when necessary to perform system maintenance. This applies to both Mac OS X server and workstation.*

- *(OSX1026SVR0010: CAT III) The IAO will ensure when administrators log on to Mac OS X server as root from the system console, they record all non-auditable actions with an entry in the system log book, recording the date, time action performed, why and whether they were successful or not.*

- *(OSX1026GEN0027: CAT II) The SA will ensure successful and unsuccessful root logon and logout attempts are recorded in a system log file such as /var/adm/syslog, /var/adm/messages, /var/sulog, etc.*

- *(OSX1026GEN0027: CAT II) The SA will enforce the requirement for all switch user (su -) attempts will be logged to the /var/adm/authlog log file.*

- *(OSX1026ADM0005: CAT II) The IAM, or Security Officer will authorize and document all root account access privileges. They will be documented with the IAO.*

- *(OSX1026SYS: CAT II) The SA will ensure the root account will have a default shell of /sbin/sh.*

3.3.2 Groups

Groups are collections of users with common resource requirements. Users are given resource access by the rights provided to a group. All users will belong to at least one group. Systems normally reserve *gids* lower than 20 for privileged system use. Therefore, the SA will not assign users a *gid* less than 20 unless the user is a privileged user. All *gids* that appear in the password file will be defined in the group file in order to maintain order and to maintain the integrity of the password file and group file. Only privileged users and groups should have access to kernel capabilities. All User and Groups can be maintained by the Netinfo Manager and there should be no need to actually go into the /etc/passwd and /etc/group files.

- *(OSX1026ADM0010: CAT II) The IAO will document group membership through DD Form 2875 or an equivalent form, for all users.*

- *(OSX1026ADM0006: CAT III) The SA will ensure that every account is assigned to at least one group.*

- *(OSX1026ADM0007: CAT II) The SA will assign unprivileged users to a group with a gid greater than 19.*

- *(OSX1026ADM0008: CAT IV) Every group referenced in the /etc/passwd file will be defined in the /etc/group file*, this can be done in the terminal or with Netinfo Manager.

3.4 Resource Controls

Resource controls are the base capabilities supplied by the Darwin system to control access to system-level resources. These include file controls, device controls, printer spool controls, and sensitive utility controls.

3.4.1 File and Directory Controls

Mac OS X is a multi-user system. This means that multiple users may be concurrently logged on to a machine, and those users can read and use files belonging to each other if they have been granted permission to do so. The owner of a file, or root, can grant permissions to a file by changing the permission bits, the file owner, or the group that is allowed to access it. In general, however, no user will possess a more permissive access to a file than the owner does. This is referred to as uneven file permissions. Before a system is connected to a production network and after required software has been loaded, a baseline of system and application files and directories will be recorded. The system will be checked weekly, in conjunction with the weekly system file baseline check, to ensure that there are no uneven file permissions. When a need to change the basic system file and directory baseline occurs, the SA document the required changes and be responsible for generating a new system file baseline after the required changes are approved. Every file and directory can be assigned three basic file permissions. These file permissions are as follows:

Read – Users with this type of permission can view the contents of a file.
Write – Users with this type of permission can change the contents of a file.
Execute – Users with this type of permission can execute a program or search a directory.

This group of three permissions is assigned to three classes of users:

Owner – Usually the person who created the file.
(Owning) Group – All users in the same group as the Owner, who have been grouped together by the System Administrator, perhaps by task assignment.
Other (or world) – Any other user on the system.

If files are other (or world) writable, they can be accessed and changed by any friendly or malicious user who gains access to the system. In other words, the files could be populated with erroneous, malicious, and harmful information, or even deleted from the system. For that reason, world writable directories will only be allowed if they are public directories, such as /tmp, /var/tmp, /var/spool/uucppublic, etc. World writable files will only be allowed within those public directories. Files can exist without a discernable owner or group owner by having the *uid* number and the *gid* number of a previous user (a user who has been deleted from the system). If a new user is added to the system and assigned the same *uid*/*gid* numbers as the previous user, the new user inherits *all* of the access permissions that previously belonged to the former user. That could mean unauthorized access to sensitive information. For that reason, un-owned files and/or files without a group owner will not be allowed.

Permissions are assigned by octal values. The *read* permission has a value of 4. The *write* permission has a value of 2. The *execute* permission has a value of 1.

The first octal value shows the owner's permissions. The second octal value shows the group permissions. The third octal value shows the other permissions.

For example, a file with a file access permission of 764 would grant the following permissions:

Owner – *Read*, *write*, and *execute* (4 + 2 + 1)
Group – *Read* and *write* (4 + 2)
Other – *Read* (4)

There is one change in interpretation for permissions of a directory. In a directory, *execute* means *search*. For example, if the above example were a directory, not a file, a directory access permission of **764** would grant the following permissions:

Owner – *Read* (the contents), *write* (into), and *search* (4 + 2 + 1)
Group – *Read* and *write* (4 + 2)
Other – *Read* (4)

Only the owner of a file or directory, or the root user, can assign or modify the file permissions. The ability to *write* also implies the ability to *delete* a file. The rights of a process to access a file are checked when the file is first accessed. The many rules that exist for system file ownership and access permissions must be observed in order to protect system security. Obviously, all

system files will be owned by a privileged user such as root, sys, bin, lp, and others. Access permissions for system files and directories are set up to allow access by privileged users and to deny, or strictly limit, access by group owners and the world.

The italicized bullets below state the requirements for files, directories, and types of files and directories. Daemons refer to the service daemons, network or otherwise, that run in the background (or on demand from within inetd.conf) and service user requests. The telnet daemon (telnetd or in.telnetd) is just one example. System log files refer to logs of system activities, such as the /var/log/syslog file, the /var/messages file, and others. Skeleton dot files refer to the default files that are copied into a newly added user's directory to be used as startup files (files that condition the user's operating environment such as .profile and .cshrc). In general, system executable files require permissions of 755, or more restrictive.

System library files (files used when compiling and running programs), manpage files (files that contain instructions for executing commands), and shells (programs such as **sh** and **csh** that determine the overall user operating environment) require access permissions that limit user access privileges in order to preserve system integrity. One other file, that requires special protection from malicious intruders in order to protect the account security of every user (including root, applications, and application data) is /etc/passwd. *APPENDIX B. FILE AND DIRECTORY PERMISSIONS TABLE,* of this document offers the recommended file ownership and permission settings for Mac OS X system and device files.

- *(OSX1026SVR0011: CAT II) The SA will check the permissions of all system directories and files of Mac OS X servers weekly to ensure there are no uneven file permissions. The exception will be in WWW server directory trees where some files will be allowed a permission of 460.*

- *(OSX1026SVR0012: CAT II) The SA will ensure that workstations do not host WWW servers.*

- *(OSX1026SVR0013: CAT III) The SA will ensure that any changes (additions, deletions, and modifications) to the Mac OS X server system directory and file permissions baseline are documented.*

- *(OSX1026SVR0014: CAT II) The SA will perform a Mac OS X server system files baseline backup before a Mac OS X system is connected to a network other than an isolated test network.*

- *(OSX1026SVR0015: CAT II) The SA will ensure a new system files baseline backup of the Mac OS X server is generated after changes to system directories and files are applied.*

- *(OSX1026SVR0016: CAT II) The SA will ensure files are checked on the Mac OS X server for a valid owner and group on a weekly basis, and files without a valid owner or group will be deleted or corrected.*

- *(OSX1026GEN0150: CAT II) The SA will ensure that world writable files are only allowed in public directories, such as /tmp, /var/tmp, etc.*

- *(OSX1026GEN0151: CAT II) The SA will ensure that world writable directories are only allowed if they are public directories, such as /tmp, /var/tmp, or other documented directories, and have the sticky bit set (Example: 1777).*

- *(OSX1026GEN0152: CAT II) The SA will ensure that all daemons have permissions of 755, or more restrictive.*

- *(OSX1026GEN0153: CAT II) The SA will ensure that all system log files have permissions of 644, or more restrictive.*

- *(OSX1026GEN0154: CAT II) The SA will ensure that all default/skeleton dot files have permissions of 744, or more restrictive.*

- *(OSX1026GEN0155: CAT II) The SA will ensure that all NIS/NIS+/yp files will be owned by root, have a privileged group owner, and have permissions of 755, or more restrictive.*

- *(OSX1026GEN0156: CAT II) The SA will ensure that all manpage files (i.e., files in the man and cat directories) have permissions of 644, or more restrictive.*

- *(OSX1026GEN0157: CAT II) The SA will ensure that all library files have permissions of 755, or more restrictive.*

- *(OSX1026GEN0158: CAT II) The SA will ensure that all shells have permissions of 755, or more restrictive.*

- *(OSX1026GEN0159: CAT II) The SA will ensure that all system commands have permissions of 755, or more restrictive.*

- *(OSX1026GEN0160: CAT II) The SA will ensure that all system files, programs, and directories are owned by a privileged account (i.e., an account with a uid less than 21).*

- *(OSX1026GEN0161: CAT II) The SA will ensure that all system files, programs, and directories belong to a privileged group (i.e., gid less than 20).*

- *(OSX1026GEN0162: CAT II) The SA will ensure that root owns the password file.*

- *(OSX1026GEN0163: CAT II) The SA will ensure that root is disabled from within Netinfo Manager.*

- *(OSX1026GEN0164: CAT II) The SA will ensure that the /etc/passwd file has permissions of 644, or more restrictive.*

3.4.1.1 Home Directories

A home directory contains a user's files and exists for that user's exclusive use. The user has access to all files in, and subordinate to, the directory (or by root in the case of startup or configuration files). Home directories should have an initial access permission of 700. DAC

allows a user to change the home directory access permissions, but they will never be more permissive than 750, which would allow group *read* access for selected files.

The user will own that user's home directory and the group owner will be the user's primary group.

- *(OSX1026GEN0165: CAT IV) The SA will ensure that all home directories are defined in the user entry of NetInfo Manager under the home property.*

- *(OSX1026SVR0016: CAT IV) The SA will ensure all home directories of the Mac OS X server, defined in NetInfo Manager exist or are justified and documented with the IAO.*

- *(OSX1026GEN0166: CAT II) The SA will ensure that user home directories have initial access permissions of 700, and never more permissive than 750 unless fully justified and documented with the IAO.*

- *(OSX1026GEN0054: CAT II) The SA will ensure the uid of a home directory is that of the account under which the directory is defined or is justified and documented with the IAO.*

- *(OSX1026GEN0055: CAT II) The SA will ensure the gid of an account home directory is the primary gid of the account (i.e., the one assigned in NetInfo Manager), except in the case of application directories for which the SA will furnish the IAO with documentation.*

3.4.1.2 Startup Files

3.4.1.2.1 User Startup Files

User startup files (i.e., files in a user's home directory with a name that begins with ".") are files that are normally read by the kernel (or utility programs) and used to customize the user's environment. These files include .login, .profile, .cshrc, and other files used by a system's shell or other utilities to set the initial working environment whenever users log on or execute an application or system utility User startup files will be owned by the user or root and will be no more permissive than 740. If a user startup file, such as .profile, sets the PATH variable, it will not contain a "." or "::" except in the last position. The PATH variable defines the search sequence the shell uses to find executable programs. A PATH variable may be observed by typing the env or set command, or by typing echo $PATH. The PATH is normally placed in the /etc/.profile or /etc/.login (for global settings), or in each user's .profile, .cshrc, or .login file (depending on the user's shell). The PATH is constructed in the following format (for sh or ksh):

PATH=/bin:/usr/bin:/oracle/bin:/usr/local/bin

This indicates that when a user types a command name the shell will search **/bin** for it first, and, if the command is not found there, the shell will search for the command in **/usr/bin**, and so on. A "**.**" (or "**::**") represents the current directory. If a PATH variable is written as follows:

PATH=/bin:.:/usr/bin:/oracle/bin:/usr/local/bin

Then the shell would search the current directory for the command immediately after it searched

/bin. Assume the user was in the /tmp directory (the current directory) when attempting to execute the ls command. Assume a malicious user created an executable program in /tmp named ls. Assume the ls program in /tmp executes a command to delete all of the user's files. If the user typed ls and the kernel did not find it in /bin, it would search the current directory, execute the malicious ls, and destroy all of the user's files. For this reason, it is preferable to never have a "." in the PATH variable. Since it would be more disastrous if the above scenario happened to root, root will never have a "." in the PATH variable. Use an editor such as vi to change the PATH variable to remove the ".". The PATH variable above would become the following after editing:

PATH=/bin:/usr/bin:/oracle/bin:/usr/local/bin

Ensure that system and user startup files are not executable by others and do not have the suid or sgid bits set that could allow a malicious user to gain expanded privileges. Help protect against implementing Trojan horses by ensuring that system and user startup files do not execute world writable programs or scripts. Root's startup files are startup files in root's home directory that serve the same purpose for root as other user startup files do for users. Finally, startup files will not execute the mesg -y command that would make their terminal devices world writable and open for possible exploitation.

- *(OSX1026GEN0056: CAT II) The SA will ensure that user startup files are owned by the user or root.*

- *(OSX1026GEN0056: CAT II) The SA will ensure that user startup files have permissions of 740, or more restrictive.*

- *(OSX1026GEN0056: CAT II) The SA will ensure that user startup files do not have a "**.**" or a "**::**" in the PATH variable definition except as the last entry.*

- *(OSX1026GEN0056: CAT II) The SA will ensure that user startup files do not have the suid bit set.*

- *(OSX1026GEN0056: CAT II) The SA will ensure that user startup files do not have the sgid bit set.*

- *(N/A: CAT II) The SA will ensure that user startup files do not execute world writable programs.*

- *(OSX1026GEN0057: CAT II) The SA will ensure that user startup files do not contain the command mesg -y.*

3.4.1.2.2 System Startup Files

System startup files are scripts executed by the system and/or kernel when the system is booted. They are also executed (with a different argument such as stop) when the system is shut down in an orderly manner. They may also be executed by root at any time. The numbers associated with the rc directory name relate to the run state at which the system executes the startup files. Files in rc2.d, for instance, would only be executed when the system is going into run state 2.

System startup files set parameters for the Kernel and start or stop applications and system utilities (such as daemons). Their names and locations are dependent on the system architecture. There are some common system startup files, such as /etc/profile and /etc/.login, in which global parameters, such as PATH variables, may be set each time a user, or root, logs on. There are also system default startup files that are placed in a new user's directory to get them started. They are normally located in /etc or /etc/skel and have names such as .profile.d, .login.d, and others. In Mac OS X their are login hooks that will allow the system to run programs and execute tasks upon startup. Since login and logout hooks require some functions of the root user, they can be configured/written to use another account instead of root, but this will limit some functionality. This section will cover the basics of the startup files and to that end it will not cover login hooks. However, if the use of login hooks should become widespread a section will be added to cover them in a future version of this document.

Startup files normally refer to the files in, and subordinate to, /etc that begin with the letters "rc" or reside in a directory such as rc0.d, rc1.d, and so on. The number relates to the run state at which they are invoked. The startup files are linked between the directories. One startup file may appear five times with different names. System startup files may also be located in /etc/init.d and /sbin/init.d, as well as /sbin/rc*.d.

System startup files will not execute programs that are world writable and will only execute programs owned by a privileged *uid* or an application owner. Additionally, since executing the command mesg -y opens up the user terminal to writing by all users, the mesg -y command will not be executed by a startup file.

- *(N/A: CAT II) The SA will ensure that system startup files are owned by root.*

- *(OSX1026GEN0102: CAT II) The SA will ensure that system startup files have a group owner of bin, sys, or the system default.*

- *(OSX1026GEN0058: CAT II) The SA will ensure that access permissions for system startup files are 755, or more restrictive.*

***NOTE**:* This requirement will not apply to symbolic links, which may be **777 (lrwxrwxrwx)**.

- *(OSX1026GEN0058: CAT II) The SA will ensure that system startup files do not contain "**.**", "**::**" (or a "**:**" as the last entry) in the PATH variable.*

- *(OSX1026GEN0058: CAT II) The SA will ensure that system startup files do not have the suid bit set.*

- *(OSX1026GEN0058: CAT II) The SA will ensure that system startup files do not have the sgid bit set.*

- *(OSX1026GEN0059: CAT II) The SA will ensure that world writable programs are not executed by system startup files.*

***NOTE**:* This includes executing programs via Login Hooks and via the system startup files in the System directory.

- *(OSX1026GEN0060: CAT II) The SA will ensure that system startup files only execute programs owned by a privileged uid or an application default.*

- *(OSX1026GEN0061: CAT II) The SA will ensure that system startup files contain the command mesg –n, where it is technically feasible.*

3.4.2.4 User Files

User files are files owned by a user (except for the possibility of user startup files that may be owned by root) and maintained by the user in the user's home directory tree. A user's files will have an initial access permission of 740 and will never be more permissive than 750 (for group access). All files in a user's directory will be owned by the user with the possible exception of startup files that may be owned by root.

- *(OSX1026GEN0063: CAT II) The user, application developers and the SA will ensure that regular files (not startup files) in user home directory trees will have initial file permissions of 700 and will not exceed 750.*

3.4.2.5 Shells

A shell is a program that serves as the basic interface between user and operating system. It is essentially a command interpreter that talks with the user, finds out what is needed, and calls the appropriate kernel functions to accomplish requests. The shell also establishes the environment that a user operates in, or controls the user's view of the system. It may be modified to suit almost any user, and it may run additional programs that serve as additional layered front-end interfaces. Every system comes supplied with several shells (sh, ksh, jsh, csh, and others) that may be defined as the default shell for users. The IAO may define the default shells that users are allowed to have in a file called /etc/shells. If a user does not have a default shell authorized through inclusion in this file, that user will not be able to log on. The IAO will ensure the SFUG instructs users not to change their default shell without authorization, and that it contains instructions prohibiting the use of unauthorized shells. The SA may use shells not listed in the /etc/shells file to disable accounts. These are /usr/bin/false, /bin/false, or /dev/null. They will not appear in the /etc/shells file because that could allow ftp to be logged on to and negate the reasons for assigning a false shell.

- *(OSX1026GEN0070: CAT II) The SA will list all authorized shells in the /etc/shells file.*

- *(OSX1026GEN0071: CAT II) The SA will ensure that the /usr/bin/false, /bin/false, and /dev/null will be considered valid shells, and that they are not listed in the /etc/shells file.*

- *(OSX1026GEN0092: CAT II) The SA and IAO will ensure accounts are set up so that inactive accounts (accounts with no activity for 35 consecutive days), and accounts that are never used for logging into the system (such as system accounts) have /bin/false, /usr/bin/false, or /dev/null as the default shell in the /etc/passwd file or be disabled in the shadow or adjunct file, or equivalent.*

- *(OSX1026GEN0192: CAT II) The SA will ensure each account in the /etc/passwd file will invoke an authorized shell listed in the /etc/shells or use /bin/false, /usr/bin/false, or /dev/null.*

- *(OSX1026GEN0064: CAT I) The SA will ensure that no shell has the suid or sgid bit set.*

- *(OSX1026GEN0065: CAT II) The SA will ensure that all shells are owned by root or bin.*

- *(OSX1026GEN0066: CAT II) The SA will ensure that shells have access permissions of 755, or more restrictive.*

3.4.2 Device Files

A device file is a special Mac OS X file that is configured with major and minor device numbers. Major and minor device numbers identify the device special file and its characteristics to the Mac OS X kernel. They provide a linkage from the user to the Mac OS X device drivers that control peripheral and memory operations. Device drivers reside in the kernel. Device files reside in special directories. The device directory and device file access permissions, as well as device driver major and minor number integrity, are critical to system security. The function of a Mac OS X device file can be changed by changing the major and/or minor numbers associated with it. If the device directory, device special file, or a device driver is compromised, then the entire system could be compromised.

The console device file can be compromised to intercept root's commands or password. Therefore, it will not be world readable or writable. Terminal devices for other users can also be compromised and will not be world writable when a user is logged on to it. Device files located outside the normal locations may indicate attempts to compromise the system. For this reason, the system will be scanned weekly for extraneous device files. If extraneous device files are located, the IAO will investigate to identify the source and take appropriate action. The IAO will justify and document the device file or delete it. Backup devices present a more subtle security hazard. If they are world writable, a backup could be destroyed accidentally or maliciously. Files not usually accessible to users may be accessible from a world readable and writable backup device. Therefore, backup devices (normally devices controlling tape drives and system floppy disks) will not be world readable or writable unless justified and documented with the IAO.

Audio and video devices that are globally accessible have proven to be another security hazard. There is software that can activate system microphones and video devices connected to user workstations. Once the microphone has been activated, it is possible to eavesdrop on otherwise private conversations without the victim being aware of it. This action effectively changes the user's microphone to a bugging device. Vendor procedures normally install /dev/audio (or the equivalent) with the device file permissions set to 666 (globally writable and therefore vulnerable). The SA and IAO will ensure that the access permissions for the audio device are 644, or more restrictive. The audio device will be owned by root with a group owner of root, bin, or sys.

- *(OSX1026GEN0167: CAT II) The SA will ensure that the console device (i.e., /dev/console) is not world readable or world writable.*

- *(OSX1026GEN0167: CAT II) The SA will ensure that ttyXX, ptyXX (where XX represents the device number, such as in tty01), and other pseudo-terminal devices are not world readable or world writable when a user is using the device.*

- *(OSX1026GEN0054: CAT II) The SA will ensure that all device files are located in the directory trees as installed and designated by the vendor.*

- *(N/A: CAT II) The SA will identify the source/owner/creator of any out-of-place device file and report it to the IAO.*

- *(OSX1026GEN0077: CAT II) The SA will ensure that the device file directories are not writable except by the owner or as configured by the vendor.*

- *(N/A: CAT II) The SA will ensure backup devices (tape and floppy disk device) of the Mac OS X server and files are readable and writable by root unless justified and documented with the IAO.*

- *(OSX1026GEN0080: CAT II) The SA will ensure the audio devices access permissions are 644, or more restrictive.*

- *(OSX1026GEN081: CAT II) The SA will ensure the audio devices are owned by root with a group owner of root or sys.*

3.5 Special Purpose Access Modes

Special operating characteristics may be assigned to a file or directory by the chmod command. These special characteristics are as follow:

set-user-id (suid)
set-group-id (sgid)
set sticky bit

- *(OSX1026GEN0082: CAT II) The IAO will ensure all locally developed programs (especially those with the suid or sgid bit set) are justified and documented and have been approved by the local CCB.*

- *(OSX1026GEN0083: CAT II) The IAO will document any changes made to the location or permissions on any file having the suid or sgid bit set.*

- *(OSX1026GEN0084: CAT II) The SA will ensure a suid files baseline backup is maintained for weekly comparison with the online suid files.*

- *(OSX1026GEN0085: CAT II) The SA will ensure a sgid files baseline backup is maintained for weekly comparison with the online sgid files*

- *(N/A: CAT II) The IAO will investigate any discrepancies when comparing suid and sgid files baseline backups with the appropriate online files.*

3.5.1 Set User ID (suid)

Authorized, vendor-supplied suid programs are crucial to the correct operation of the Mac OS X operating system, but unauthorized suid programs present a security hazard. When the suid attribute is set on the access permissions of a program, a user executing the program has the same privileges as the owner of the program. If the owner of the program is root, then the user, while executing that program, has all the powers of root, at least for the scope of the program being executed. It is extremely important; therefore, that any program that has the suid bit set is of known origin and scope.

Refer to the specific vendor's Mac OS X documentation for details concerning suid programs. Commercial and Government-supplied applications may also contain programs with the suid bit set.

If so, the vendor/proponent instructions must be followed. Where possible, require vendor/proponent integrity statements that guarantee there are no *back doors*, such as shell escapes, built into the applications.

The following command may be invoked to find all suid programs on a system and produce a listing of the owner and other pertinent information.

find / -type f -perm -4000 -exec ls -ld {} \;

If a mounted filesystem has any suid executable scripts or programs, a user who invokes the executable takes on the *uid* of the executable's owner. The owner of such suid executables is typically a privileged user, usually root. If a filesystem is exported, a remote user, who may be normal or privileged, may execute an suid file and alter files mounted, but not exported, on the exporting host system. This is a serious vulnerability, which must be managed with the **mount** command options.

- *(OSX1026GEN0086: CAT II) The SA will ensure user filesystems, removable media, or remote filesystems are mounted with the nosuid option invoked.*

3.5.2 Set Group ID (sgid)

Authorized, vendor-supplied sgid programs are crucial to the correct operation of the Mac OS X

operating system, but unauthorized sgid programs present a security hazard. The sgid bit only affects executable programs. When this attribute is set, the user executing the program has the same privileges as the group owner of the program. It is extremely important; therefore, that any program that has the sgid bit set is of known origin and scope. Programs with the sgid bit set must never allow escapes to the command line.

Refer to the specific vendor's Mac OS X documentation for details concerning sgid. Commercial and Government-supplied applications may also supply programs with the sgid bit set. If so, then vendor/proponent instructions must be followed. Where possible, require vendor/proponent integrity statements that guarantee there are no *back doors* (such as shell escapes) built into the applications.

The following command will identify all sgid programs on a system, producing a listing of the owner and other pertinent information:

find / -type f -perm -2000 -exec ls -ld {} \;

3.5.3 Sticky Bit

When the sticky bit is set on a directory, only the owner of a file within that directory, the owner of the directory, or root may delete or change files in that directory. The feature prevents users from accidentally or maliciously deleting or changing files that could adversely affect the operation of another user's applications or cause data corruption in another user's temporary files. The setting is normally reserved for directories used by the system and by users for temporary file storage (in /tmp, for instance) and for directories that require global *read*/*write* access. Since the public directory owner can change or delete any file within the public directory, all public directories will be owned by root and the sticky bit will be set. The group owner of all public directories will be root, bin, sys, or the COTS/GOTS default.

- *(OSX1026GEN0087: CAT III) The SA will ensure the sticky bit is set on all public directories.*

- *(OSX1026GEN0088: CAT III) The SA will ensure the owner of public directories is root.*

- *(OSX1026GEN0089: CAT III) The SA will ensure the group owner of all public directories is root*, *sys*, *bin, or the COTS/GOTS default.*

3.6 Umask

The umask is a kernel variable that controls the file access permissions assigned to newly created files and directories. Data and program integrity, confidentiality, and availability are directly affected by the system and user umask. If the umask is too permissive, newly created files and directories will be accessible to unauthorized and possibly malicious users. If the umask is too restrictive, applications may not function correctly. Therefore, the umask is a critical component of every user and system process.

The umask controls access permissions for the following three groups:

- File owner (or creator)
- Owner's default group
- Rest of the world (others)

To determine what permissions a given umask will assign to a newly created file, subtract the umask from 777. A umask of 022, for instance, would assign the file creator *read*, *write*, and *execute* permissions. The group and others would be assigned only *read* and *execute* permissions. The access permissions are read as 755. All Mac OS X systems are fielded with a default umask of 022. This allows the access permissions listed above. This allows access permissions of 755. It is desirable to only allow access to the owner of a file, by default, and only after explicit action by the owner (called discretionary access control [DAC]) if access is allowed to group users, as appropriate. To accomplish this, the system and user umask will be set to 077, and will not be reset unless justified and documented with the IAO. Exceptions to this will be during software installation when the installation process demands a more permissive value, during database access by users, and during administrative actions. All requirements will be justified and documented with the IAO.

- *(OSX1026GEN0089: CAT II) The SA will ensure the system and user umask is 077.*

- *(OSX1026GEN0090: CAT II) The SA will ensure application umasks are not less restrictive than 022.*

3.7 Development Systems

Application developers often ignore security requirements in favor of development expediency. One of the most important parts of applications today, however, is security. Therefore, development systems will be subject to the same security requirements as production systems. Development systems are often connected to live networks and, because security requirements have not been observed, jeopardize the entire network. If network connectivity is a requirement for development systems, they will be connected to a testing network that is completely isolated from all other production systems and networks. Applications will be designed to work correctly in a secure environment.

- *(OSX1026DEV0001: CAT II) The developers, the SA and the IAO will ensure systems used for development are completely isolated from all production systems and networks, such as through an isolated subnetwork.*

3.8 Default Accounts

Mac OS X systems come configured with default accounts and, when software is installed, applications have default accounts. These accounts usually have standard passwords. Default system accounts are normally listed in NetInfo Manager and they have names such as mysql(even though it is not installed), nobody, smmsp(even thought it is not installed), sshd, unknown, www, and daemon. The IAO will be responsible for inspecting NetInfo to ensure that default passwords are changed whenever new operating systems or applications are installed. The IAO will also ensure that system default accounts, other than root, are disabled. The IAO will ensure that new passwords are assigned for applications, both internally

Default accounts will be disabled by entering /dev/null as the default shell in NetInfo or by disabling the password in NetInfo as well. It is preferable to do both but either will do. It should be documented which was done on a given IS.

- *(OSX1026GEN0092: CAT II) The SA will ensure logon capability to accounts bin, lib, uucp, news, sys, guest, daemon, and any default account not normally logged onto is disabled by making the default shell /dev/null, or by disabling the password.*

- *(OSX1026GEN0091: CAT I) The SA will ensure application passwords, internal to the application and at the system level, is changed after application implementation.*

3.9 Audit Requirements

Auditing is not system logging and is not system accounting. System logging is done via the **syslog** facility. System accounting, when activated, collects data useful for charging timeshare customers and for system capacity planning.

Due to Mac OS X not having a built in auditing system, auditing on a Mac OS X system needs to be accomplished by a third party program.

- *(OSX1026AUD0001: CAT II) The SA will ensure that auditing is implemented.*

Security requires monitoring of user and process activity almost to the keystroke level. It records much more detail about what users are doing and records system actions. Most systems provide system software for that purpose. Each is configured differently and has unique utilities for reading audit data files. Audit utilities can extract information about specific users and processes from the audit files.

These flags will be implemented and all deviations will be justified and documented with the IAO. The IAO and SA will ensure that audit files are only accessible to authorized personnel. All users, including root, will be audited. In Mac OS X, not all of the auditing features other operating systems have are implemented in the OS at this time. According to Apple Computer, this issue as it relates to NIAP compliance: "Our work so far indicates that Mac OS X meets the requirements except for the Auditing feature which we have under development." Some features are implemented but others are not. Because of this any auditing that can be done, will be done on the workstations and servers for now until stronger measures are put in place by Apple.

- *(OSX1026GEN0093: CAT II) The SA will ensure that audit files have permissions of 640, or more restrictive.*

- *(OSX1026GEN0093: CAT II) The SA will ensure that all audit files and directories are readable only by personnel authorized by the IAO.*

- *(OSX1026SVR0017: CAT II) The SA will review Mac OS X server audit files daily for anomalies.*

- *(OSX1026SVR0018: CAT III) The IAO will ensure audit files are retained at least one year.*

- *(OSX1026SVR0019: CAT II) The SA will ensure that for all users, including root, the audit system are configured to audit at least the following events:*

  - *Logon (unsuccessful and successful) and logout (successful)*

  - *Unauthorized access attempts to files (unsuccessful)*

  - *Use of privileged commands (unsuccessful and successful)*

  - *Application and session initiation (unsuccessful and successful)*

  - *Discretionary access control permission modification (unsuccessful and successful use of chown/chmod)*

  - *System startup and shutdown (unsuccessful and successful)*
  - *All system administration actions*

  - *All security personnel actions*

- *(OSX1026SVR0020: CAT I) The IAO will ensure the auditing software is able to record the following for each audit event:*

  - *Date and time of the event*

  - *Userid that initiated the event*

  - *Type of event*

  - *Success or failure of the event*

  - *For I&A events, the origin of the request (e.g., terminal ID)*

  - *For events that introduce an object into a user's address space, and for object deletion events, the name of the object, and in MLS systems, the object's security level*

  - *Root and other administrative actions*

3.10 Cron Access

Cron is a job scheduling utility. It controls jobs configured to run in the background on a recurring schedule. Cron determines the schedule and the jobs from configuration files called crontabs. It keeps track of each specific crontab creator and executes the programs with all the privileges of the crontab creator. Because of that, crontab entries will not execute world or group writable programs nor will the programs be in a world writable directory or a directory tree that contains a directory that is world writable. Cron will be enabled only for root on all Mac OS X workstations. This is so the Mac can run a nightly job that cleans up the system and refreshes the locate databases. Cron for jobs should not be used on the workstation (server is covered later). To do this the following three things will need to be done:

- Create an allow in /var/adm/ and put NO-ONE in it except for root.
- Set permissions to 700 on allow.
- Give Cron permissions of 700.

- *(OSX1026SVR0021: CAT II) The SA will ensure no Cron jobs execute on Mac OS X workstations.*

3.10.1 Access Controls

Access to the use of Cron facilities will be authorized and documented with the IAO. In addition, Cron uses a file called allow, populated by the SA, to determine which users are authorized to create crontabs. It uses a file called deny, also populated by the SA, to deny access to specific users. The allow and deny files, if they exist, are usually located in/var/adm/. Specific locations can be determined by performing the man Cron command, which should mention their locations. If allow is used, there is no absolute need to also have a deny file, because users not in the allow file will not have access anyway. If there are no allow and deny files, the system assumes either everybody can access Cron or nobody can access Cron, depending on the system. Therefore, every system will have either a allow file listing authorized Cron users, or a deny file, listing users not authorized to use the Cron.

3.10.2 Access Permissions and Owners

The maximum access permissions for the allow and deny files will be 700. The owners of the allow and deny files, where they exist, will be root, bin, or sys. The owner for the cronlog files will be root. The group owner of the cronlog file will be root or another privileged user such as sys or bin.

Other files and directories associated with Cron will be owned by root or bin with a group owner of root, bin, or sys. Crontabs will be owned by root with a group owner the same as the group of the crontab creator. Crontabs will have a maximum access permission of 600. The access permissions for the Cron and crontab directories will be 755, or more restrictive.

- *(OSX1026SVR0022: CAT II): The group owner of the cronlog file will be root or another privileged user such as sys or bin.*

3.10.3 Cron on Mac OS X server

The crontab files will be created with the same name as the creator of the file. A crontab entry, or any program executed by the crontab entry, will not relax the system umask unless the requirement has been justified with, approved by, and documented with the IAO. A crontab entry will not execute locally developed suid or sgid programs unless they have been approved by the local Config Control Board (CCB) and documented with the IAO. Exceptions include programs supplied with the operating system. Default accounts (with the possible exception of root) will not be listed in the allow file. If there is only a deny file, the default accounts (with the possible exception of root) will be listed there (the size cannot be zero).

- *(OSX1026SVR0023: CAT II): The IAO will ensure a crontab entry is not executing locally developed suid or sgid programs unless they have been approved by the local Configuration Control Board (CCB) and documented with the IAO.*

Users will use the crontab -e command to create or edit all Cron jobs associated with their account name. This utility provides file locking to prevent multiple users from editing the same file at the same time and notifies the Cron daemon when crontabs have changed so the Cron daemon knows to reread the crontabs. It should also provide the correct access permissions to the crontab.

Cron has the capability to log its actions, and their success or failure, to a log file called cronlog. This is a configuration item for all systems. The SA and IAO will ensure the system is configured to log all Cron actions. The SA will also ensure the cronlog access permissions are set to 600, or more restrictive.

3.10.4 Locations

The cronlog will be created in **/var/cron/log**. The allow and deny files are located in /var/cron.

- *(OSX1026GEN0203: CAT II) The SA will ensure crontab entries do not execute group or world writable programs.*

- *(OSX1026GEN0200: CAT II) The SA will control access to the cron utilities via the allow or the deny file.*

- *(OSX1026GEN0204: CAT II) The SA will ensure crontab entries do not execute programs located in, or subordinate to, world writable directories.*

- *(N/A: CAT II) The IAO will authorize and document all users who are allowed to create crontabs.*

- *(OSX1026GEN0200: CAT II) The SA will ensure every system has either a allow file or a deny file.*

- *(OSX1026GEN0200: CAT II) The SA will ensure no allow or deny file has a size of zero.*

- *(OSX1026GEN0201: CAT II) The SA will ensure the allow file access permissions are 700, or more restrictive.*

- *(N/A: CAT II) The SA will ensure the deny access permissions are 700, or more restrictive.*

- *(OSX1026GEN0205: CAT II) The SA will ensure access permissions for crontab files are 600, or more restrictive.*

- *(OSX1026GEN0206: CAT II) The SA will ensure access permissions for the cron and crontab directories are 755, or more restrictive.*

- *(OSX1026GEN0207: CAT II) The SA will ensure the owner of the cron and crontab directories is root or bin.*

- *(OSX1026GEN0208: CAT II) The SA will ensure the group owner of the cron and crontab directories is root, bin, or sys.*

- *(N/A: CAT III) The SA will ensure cron jobs do not execute a program that sets the umask to a value more permissive than 077 unless it is documented and justified with the IAO.*

- *(N/A: CAT IV) Users and the SA will ensure the command crontab -e is used to create and edit crontabs.*

- *(OSX1026GEN0209: CAT II) The SA will ensure cron logging is implemented.*

- *(OSX1026GEN0210: CAT II) The SA will ensure cronlog access permissions are 600, or more restrictive.*

- *(OSX1026SVR0025: CAT II) The SA or the IAO will review the cronlog daily.*

- *(OSX1026GEN0210: CAT II) The SA will ensure the cronlog owner is root and the group owner is root, bin, or sys.*

- *(OSX1026SVR0026: CAT II) The SA will ensure the owner and group owner of the allow file are root, bin, or sys.*

3.11 At Access

The at utility reads commands from standard input and groups them together for deferred execution at the times specified by the user. Access to at will be controlled using the at.allow and at.deny files to list authorized or unauthorized users respectively. At uses cron for program execution and at actions are logged, by cron, in the cron log. Because at executes jobs with the privileges of the user, at will not execute world writable files. No one should be using at for jobs on the workstation. To do this three things need to be done.

- Create an at.allow in /var/adm/ and put NO-ONE in it except for root.
- Set permissions to 700 on at.allow.
- Give at permissions of 700.

- *(OSX1026SVR0024: CAT II) The SA will ensure that at jobs are not run on Mac OS X*

workstations.

3.11.1 Access Controls

Access to the use of at facilities will be authorized and documented with the IAO. At uses a file called at.allow, populated by the SA, to determine which users are allowed to create at jobs. It uses a file called at.deny, also populated by the SA, to determine which users are specifically denied use of the at facilities. Users specifically allowed to use at appear in the at.allow file. Users specifically denied access appear in the at.deny file. If neither at.allow nor at.deny exists, then root is the only user allowed access to use at. However, if only an empty at.deny file exists, then anyone may use at. The at.allow file may exist without the at.deny file. The at.deny file may exist without the at.allow file, but may not be empty. Users not listed in the at.allow file, if it exists, will not be allowed access to at. To control access to at, an empty at.deny file will not exist if there is no at.allow file that lists authorized users.

3.11.2 Access Permissions and Owners

The access permissions for the at.allow and at.deny files will be 700. The owner will be any privileged system user such as root, bin, or sys. The group owner will be root, bin, or sys. Access permissions for the at (or equivalent) directory will be 755 or more restrictive.

3.11.3 At on Mac OS X Server

The at.allow and at.deny files, if they exist, are usually located in /var/cron/. Executing the command “man crontab” will usually give information on the location of the allow and deny files.

- *(OSX1026SVR0027: CAT II) The SA will be responsible for ensuring jobs initiated by the ”at” utility do not execute world or group writable programs.*
- *(OSX1026SVR0028: CAT II) The SA will ensure access to “at” is controlled via the at.allow or at.deny file.*
- *(OSX1026SVR0029: CAT II) The SA will ensure “at” job entries do not execute programs in or subordinate to world writable directories.*
- *(OSX1026SVR0030: CAT III) The IAO will authorize and document all users allowed to use “at”.*
- *(OSX1026SVR0031: CAT II) The SA will ensure every system has either an at.allow or an at.deny file.*
- *(OSX1026SVR0032: CAT II) The SA will ensure neither the at.allow nor the at.deny files are empty.*
- *(OSX1026SVR0033: CAT II) The SA will ensure access permissions of the at.allow file are **700**, or more restrictive.*

- *(OSX1026SVR0034: CAT II) The SA will ensure access permissions of the at.deny file are 700, or more restrictive.*

- *(OSX1026SVR0035: CAT II) The SA will ensure access permissions for the "at" (or equivalent) directory are 755, or more restrictive.*

- *(OSX1026SVR0036: CAT II) The SA will ensure the owner and group owner of the "at"(or equivalent) directory is root, bin, or sys.*

- *(OSX1026SVR0037: CAT II) The SA will ensure "at" jobs do not use a umask less restrictive than 077.*

- *(OSX1026SVR0038: CAT III) The SA will ensure "at" jobs do not execute a program that sets the umask to a value more permissive than 077 unless it is documented and justified by the IAO.*

- *(OSX1026SVR0039: CAT II) The SA will ensure default accounts do not appear in the at.allow file.*

- *(OSX1026SVR0040: CAT II) The SA will ensure programs executed via "at" are not world or group writable.*

- *(OSX1026SVR0041: CAT II) The SA will ensure programs executed by "at" are writable only root, the user, or the application.*

- *(OSX1026SVR0042: CAT II) The SA will ensure programs executed by "at" are not in a directory tree where one or more directories, in the tree, are world writable.*

- *(OSX1026SVR0043: CAT II) The SA will ensure the owner and group owner of the at.allow file are root, bin, or sys.*

4. NETWORK SERVICES

Most system services that can be accessed via the network are defined in the inetd.conf file. The inetd.conf file contains the configuration for the inetd program. The inetd program is a daemon that listens for network connection requests and services them by spawning another process. If the requested service is not defined in its configuration file, inetd will refuse to provide the service. Sites can limit the types of network services provided by commenting out the lines that define the service in the inetd.conf file. A list of services that are normally commented out is shown below. In most cases, only telnet, ftp, and other system and application services are enabled. On all Mac OS X workstations the inetd.conf file will be renamed to noinetaccess.txt and placed in /var/adm/ directory. Then a blank inetd.conf file will be put in the place of the old one. This is in case the file is needed later for troubleshooting the Mac. There should be no reason to alter the blank file, but if one is needed the SA and IAO will document the reasons.

NOTE: When running Mac OS X servers the restrictions for the inetd.conf can be altered to allow for web services and other network services.

- *(OSX1026SEC0100: CAT I) The IAO will ensure the following non-exhaustive list of potential network services shows services that are not usually necessary for operations. These services are disabled in the inetd.conf file unless justified and documented with the IAO.*

admind	nsed	rpc_keyserv	sysstat
chargen	nsemntd	rpc_sched	talkd
echo	pfilt	rquotad	tfsd
etherstatd	portd	rsh	tftpd
fingerd	quaked	rstatd	timed
ICQ server	rexd	rusersd	ttdb
identd	rexecd	selectd	ugidd
named	rje_mapper	serverd	uucpd
netstat	rlogind	showfhd	walld
netstatd	rpc_3270	sprayd	
nit	rpc_alias	statmon	
nntp	rpc_database	sunlink_mapper	

- *(OSX1026SVR0044: CAT III) The SA will ensure all network services required for operations are justified and documented with the IAO*

- *(OSX1026GEN0107: CAT II) The SA will ensure the inetd.conf file is owned by root or bin.*

- *(OSX1026GEN0108: CAT II) The SA will ensure the inetd.conf file has permissions of **440**, or more restrictive.*

- *(OSX1026SVR0045: CAT III) The SA will ensure inetd logging/tracing is enabled.*

4.1 Network Services Descriptions

The following descriptions are not intended to endorse the use of the services described. They are merely to familiarize the reader with the purpose of the service.

4.1.1 Apache

The Apache web server comes as a factory install on Macs running OS X. This program is used to serve web content off a server/workstation. For all Mac workstations the Apache program will need to be deleted. Referencing: *APPENDIX C. PROCEDURES FOR BRINGING A MAC OS X SYSTEM INTO STIG COMPLIANCE* in the section: Removing Apache from OS can assist in deleting this program. If you are running Mac OS X server then you will want to refer to the Web Server STIG to ensure that you are creating a safe Apache running environment.

- *(OSX1026SVR0046: CAT II): The SA will ensure that the Apache Web Server is removed on all Mac OS X workstations and on Servers that do not need web hosting services running.*

4.1.2 Rlogin and rsh

The rlogin and rlogind programs provide remote terminal service similar to telnet and telnetd. The client program is rlogin, and the server program is rlogind. The important difference between rlogin and telnet is that if the rlogin connection is coming from a trusted host or a trusted user (meaning .rhosts and/or hosts.equiv is properly configured), no password is required. On a Mac OS X workstation rlogind and rlogin will both be given permissions of 000 so they can stay on the system but not usable by anyone but root which is disabled by virtue of the account portion of this STIG.

- *(OSX1026SVR0047: CAT II): The SA will ensure that rlogind and rlogin are given permissions of 000 on Mac OS X workstations and Servers that do not need remote services running.*

Secure shell provides a functional alternative to the typical requirements for rlogin and rsh.

4.1.3 Rexec Command

The remote command execution daemon, rexecd, allows users to use rsh or remsh to execute commands on other systems. A password may or may not be required depending on the use of .rhosts and/or hosts.equiv. Unlike login and telnet, rexecd returns different error messages for invalid accounts and passwords. If an invalid username is supplied the error message returned is login incorrect. If an invalid password is supplied, it returns password incorrect. This allows a potential unauthorized user to probe the system to find a valid user account name and then to work on the password. Therefore, if rexecd is required, it will be justified and documented with the IAO. This will have a permissions set of 000 on all Mac OS X workstations.

- *(OSX1026SVR0048: CAT II): The SA will ensure that rexecd has permissions of 000 on all Mac OS X workstations.*

4.1.4 Finger

The finger command makes personal information available to users on the network. Hackers use this feature to obtain and exploit information about users and to help obtain unauthorized access to accounts. The syntax is simple—finger user@host. The output contains information about the user. This will have a permissions set of 000 on all Mac OS X workstations.

- *(OSX1026SVR0049: CAT II): The SA will ensure that finger has permissions of 000 on all Mac OS X workstations.*

4.1.5 Remote Host Printing

The /etc/hosts.lpd enables remote host printing on most systems. It is possible for unauthorized remote systems to print to hosts (as a print server) if the printer configuration files are not configured properly. In addition, the SA and IAO should know and document all systems that are authorized to use a host as a print server.

- *(OSX1026SVR0050: CAT II) The SA will for all Mac OS X servers obtain the approval of the IAO for all hosts that are implemented as clients to a print server.*

- *(OSX1026SVR0051: CAT II) The SA will for all Mac OS X servers supply all print server – client configuration documentation to the IAO.*

- *(OSX1026SVR0052: CAT II) The IAO will for all Mac OS X servers maintain documentation clearly depicting all print server – client configurations.*

- *(OSX1026SVR0053: CAT II) The SA will for all Mac OS X servers ensure the local UNIX host printer configuration file, if one exists, will not contain the "-" (minus) or "+" character.*

- *(OSX1026SVR0054: CAT II) The SA will for all Mac OS X servers ensure the printer configuration files will be owned by root, bin, sys, or lp.*

- *(OSX1026SVR0055: CAT II) The SA will for all Mac OS X servers ensure printer configuration files will have permissions of 664, or more restrictive.*

4.1.6 Traceroute

Traceroute is a utility used to determine the path a packet takes between two endpoints. Sometimes when a packet filter firewall is configured incorrectly, an attacker can traceroute the firewall, gaining knowledge of the network topology inside the firewall. This information may allow an attacker to determine trusted routers and other network information. Traceroute is often used by network management software, and this is acceptable as long as it is documented and justified.

- *(OSX1026SVR0056: CAT I) The SA will ensure that the traceroute of the Mac OS X server command is owned by root.*

- *(OSX1026SVR0057: CAT I) The SA will ensure that the traceroute command of the Mac OS X server has a group owner of sys, bin, or root.*

- *(OSX1026SVR0058: CAT I) The SA will ensure that the permissions for the Mac OS X server traceroute command are 700, or more restrictive.*

- *(OSX1026SEC0110: CAT I) The SA will ensure that the to Mac OS X workstation traceroute has its user permissions set to 000, by doing a chmod 000 traceroute.*

Note: The SA will ensure that the Mac OS X workstation traceroute is used by the SA to troubleshoot problems if needed; in this case it can be reactivated by using a chmod u=sr,go=rs tracerout and this will be documented and justified with the IAO, the IAM, and the NSO (Network Security Officer), it will then be set to a 000 status.

4.1.7 Client Browser Requirements

Mac OS X ships with Internet Explorer 5.2 by Microsoft. The latest security patches will be applied to the software.

- *(OSX1026WEB0018: CAT II) The SA will ensure that any web browser that is being used on Mac OS X is PKI enabled.*

- *(OSX1026WEB0005: CAT II) The SA will ensure any web browser, is the latest approved version and at the latest patch level.*

- *(OSX1026WEB0007: CAT III) The SA will ensure the browser is capable of 128-bit encryption.*

- *(OSX1026WEB0008: CAT II) The SA will ensure the SmartUpdate, or software update feature, of a browser is not enabled.*

- *(OSX1026WEB0006: CAT II) The SA will configure browsers to accept cookies only from the connected site.*

- *(OSX1026WEB0009: CAT II) The SA will configure browsers to disallow secure content caching unless encrypted.*

- *(OSX1026WEB0004: CAT III) The SA will configure browsers to display a warning when submitting non-encrypted form data to an html page.*

- *(OSX1026WEB0003: CAT III) The SA will configure browsers to display a warning when viewing documents with both secure and non-secure content.*

- *(OSX1026WEB0010: CAT III) The SA will configure browsers to disallow automatic downloading of active content.*

- *(OSX1026WEB0011: CAT III) The SA will configure browsers to disallow active scripting.*

- *(OSX1026WEB002: CAT III) The SA will configure browsers to issue a warning when entering or leaving an encrypted or secure site.*

- *(OSX1026WEB0012: CAT II) The SA will configure browsers to issue a warning if form data is redirected.*

- *(OSX1026WEB0013: CAT III) The SA will disable JavaScript on browsers.*

- *(OSX1026WEB0014: CAT II) The SA will configure browsers to issue a warning when viewing data on a remote site containing a security certificate that does not match its Internet address.*

- *(OSX1026WEB0015: CAT II) The SA will configure browser home pages for the local site home page or a blank page.*

- *(OSX1026WEB0016: CAT I) The root account will not use a browser for any reason other than to control local applications.*

4.2 Sendmail

The Simple Mail Transfer Protocol (smtp) is the standard for transferring e-mail between hosts. The sendmail program or equivalent (e.g., mmdf, rmail, smail) implements both the client and server sides of the smtp protocol. Sendmail can deliver e-mail to local and remote users, mailing lists, and programs. E-mail addresses are located in an alias file in which users, working through their electronic mail administrator, may establish e-mail addresses and mailing lists.

The Sendmail program will be removed from all Mac OS X Workstations. To accomplish this refer to *APPENDIX C. PROCEDURES FOR BRINGING A MAC OS X SYSTEM INTO STIG COMPLIANCE*: Removing Sendmail from Mac OS X.

- *(OSX1026SVR0059: CAT II): The SA will ensure that Sendmail is removed from all Mac OS X workstations.*

4.3 Ftp

Ftp allows the transfer of files between systems. The client program is ftp, and the server program is ftpd. The system supplied ftp client will not be used on hosts inside the protected perimeter. The only data transfer client allowed will transmit and receive only encrypted passwords and data once an ftp protocol session has been established. The ftpd server supplied with the system will not be used on hosts inside the protected perimeter. The only data transfer server allowed will transmit and receive only encrypted passwords and data once an ftp protocol session has been established. Hosts located outside the protected perimeter may use ftp and ftpd, but the hosts should be considered “sacrificial lambs”, in that case. The following is a brief explanation of the ftp utility to allow users to understand the use.

- *(OSX1026SVR0060: CAT II): The SA will ensure that Ftpd is set to permissions 000 on all Mac OS X workstations.*

4.4 Trivial File Transfer Protocol (tftp)

Tftp is a file transfer program that requires no I&A. On all Mac OS X workstations the tftp will not run due to the blank inetd.conf file. In addition, the tftpd will have its permissions set to 000.

- *(OSX1026WEB0001: CAT I) The SA will ensure that the tftpd file permissions are 000 on all Mac OS X workstations.*

4.5 Domain Name Service (DNS)

BIND and named are equivalent. The name daemon, named, is the software that implements BIND. There are others, but the BIND DNS server is used on the vast majority of name serving machines on the Internet. The resolver library included in the BIND distribution provides the standard application programmer interfaces (APIs) for translation between domain names and Internet addresses. The resolver library is used for linking with applications requiring domain name service. Most implementations of BIND use a daemon called named. BIND has encountered some security problems. It is very important, therefore, to ensure that the latest version is being used. The minimum version that is allowable at this time is the newest version supported by the vendor. In general, BIND Version 8.2.2, Patch Level 7, is the latest and most trustworthy version at this time. To examine the version number of named for HP systems, use the command what /usr/sbin/named. The easiest way to examine the version number of named for Sun Solaris systems is to use the command strings /usr/sbin/in.named | grep –i version.

The BIND program will be removed from all Mac OS X Workstations. To accomplish this refer to *APPENDIX C. PROCEDURES FOR BRINGING A MAC OS X SYSTEM INTO STIG COMPLIANCE* : Removing BIND from Mac OS X.

- (OSX1026GEN000 CAT II) The SA will ensure that BIND has been removed from all Mac OS X workstations.

The configuration files associated with BIND are as follows:

/etc/resolv.conf	Contains the domain and the server to use for address lookups
/etc/named.boot or named.conf	Configuration boot file (contains locations of other files/tables)
The DNS translation tables defined in named.boot or named.conf	
/var/run/named.pid	Process ID of the named process
/var/tmp/named.run	Debug output file
/var/tmp/named_dump.db	Dump of name server database
/var/tmp/named.stats	Nameserver statistics data

Configuration files will be owned by root with a group owner of root, bin, or sys. Configuration file access permissions will be **600**, or more restrictive.

4.6 System Logging Daemon (syslogd)

The system-logging daemon (syslogd) reads and forwards system messages to the log files and/or users. Malicious users can flood the logging daemon with unauthorized messages unless syslogd is configured to accept messages only from designated hosts. System logging normally takes place over port 514. Services to this port should be restricted to local hosts at the firewall or premise router.

If syslogd is required to log system messages to the local machine, ensure that the system name in /etc/hosts contains the alias loghost. If the /etc/hosts file shows the loghost as some other system, then system log messages will be sent to that host instead of being logged on the local host. The IAO will maintain documentation of the machines using a non-local loghost. Local hosts will not be permitted to act as loghosts for systems outside the local network. Some messages need to be reviewed immediately by responsible parties such as root. Use the following example (or one similar) in the /etc/syslog.conf file to ensure alerts are written to the terminal screen of root or operator if they are logged on:

```
*.alert root,operator
```

Some systems are vulnerable to a syslog denial of service (flood) attack. If you are not using remote logging, use the "-r" option (or "-l" in BSDI) to turn remote logging off in your syslog daemon. You must then recompile the daemon. Contact your vendor or refer to your vendor's documentation for more information.

- *(OSX1026DNS0001: CAT II) The SA will ensure the /etc/syslog.conf file is owned by root with access permissions of 640, or more restrictive.*

- *(OSX1026DNS0002: CAT II) The SA will ensure the group owner of the /etc/syslog.conf file is a privileged uid.*

- *(OSX1026DNS0003: CAT II) The IAO will maintain documentation of the machines using a non-local loghost.*

- *(OSX1026DNS0004: CAT III) The IAO will maintain documentation of log servers and the machines that are permitted to log to them.*

4.7 Secure Shell (ssh)

Secure Shell (**ssh**) is communications software that uses encrypted communications to log on to and perform jobs on another computer through a network. It can also be used to execute remote commands and to move files between machines. Ssh communicates using encryption to protect data and passwords. It provides strong authentication and secure communications over insecure channels. Ssh also provides rlogin, rsh, rcp, and rdist services, but since the communications are encrypted, it is done in a much more secure manner than the traditional services.

Hackers, curious administrators, employers, and criminals, both industrial and government (friendly and antagonistic) can eavesdrop on network communications using sniffers to collect private and corporate information such as account names, passwords, and sensitive data. Communications packets also include information about destination and origination network addresses. A sniffer is a program that puts a network interface into promiscuous mode. The interface, when in promiscuous mode, listens to all communication packets passing through its network instead of just packets that contain its address.

It is also possible to hijack unencrypted network connections. This technique can be used to enter in the middle of existing connections to modify data in both directions and to insert new commands in sessions authenticated by one-time passwords. No security method based only on user I&A is safe.

Ssh connects to sshd on the server machine. It verifies that the server machine really is the machine it wanted. Ssh then exchanges encryption keys (protected from sniffers), and performs authentication, RSA (Rivest, Shamir, and Adleman) authentication, or conventional password based authentication. The server normally allocates a pseudo-terminal and starts an interactive shell or user program. Ssh will also work with X Windows.

It is recommended that Version 3.4p1 of OpenSSH that ships with Mac OS X 10.2.4 or higher is used. Ssh offers the ability to log on directly as root even when the system configuration files disable that feature for other access methods. Ensure that this feature is disabled. Ssh also allows the use of .rhosts. It is not recommended that the .rhosts file is NOT used unless the feature is operationally necessary.

Ssh will be disabled on all Mac OS X Workstations. It will be done by removing the program from all Mac OS X Workstations using the *APPENDIX C. PROCEDURES FOR BRINGING A MAC OS X SYSTEM INTO STIG COMPLIANCE*: Removing SSH from Mac OS X.

- *(OSX1026SYS0010CAT II) The SA will ensure that SSH is removed from all Mac OS X workstations.*

4.8 Mac OS X Built-in Firewall

Using a firewall has become a common practice for keeping unwanted connections off an IS. Mac OS X comes with its own built-in firewall, which can be used.

- *(OSX1026SEC0020: CAT I): The SA will ensure that all known DDoS ports and NetBIOS ports will be bi-directionally blocked by the built-in firewall. Refer to the Desktop STIG, for additional firewall guidance.*

- *(OSX1026SEC0021: CAT II): The SA will ensure that a "deny by default" posture is enforced on the built-in firewall. The SA will ensure that only ports or services required for operational use are open on the firewall and that all open ports are documented.*

***NOTE**:* By default, this is the configuration when the firewall is started.

5. TRUST RELATIONSHIPS

In the early days of computer use, all information necessary for an application was contained on storage media physically attached to the computer system on which the application executed. With the advent of networks and network technologies, many computer applications were designed to communicate with other computers to share information and to store data centrally. Initial communication protocols for sharing information did not consider checking the authority (I&A) for a request for data or command execution. Today, computer information must be guarded to assure privacy and accuracy. This guarding is handled by assorted encryption schemes and protocols that establish trust relationships between two or more computers. Communication protocols also ensure end-to-end data integrity.

5.1 Network Information Service (NIS)

Network Information Service (NIS) is a database system that provides a mechanism for sharing network objects and resources. It provides a uniform storage and retrieval method for network-wide information in a transport-protocol and media-independent fashion.

By running NIS, the System Administrator can distribute administrative databases called *maps* among a variety of servers (master, slaves, and clients), and update those databases from a centralized location in an automatic and reliable fashion to ensure that all clients share the same information in a consistent manner throughout the network. NIS stores information about machine names and addresses, users, the network, and network services. This collection of network information is referred to as the NIS namespace.

NIS addresses administration requirements of client/server computing networks common in the 1980s. Client/server networks were limited to no more than a few hundred clients and a small number of multipurpose servers. The clients and servers were spread across a few remote sites. Users were considered sophisticated and trusted so security was not a primary concern. The networks needed infrequent updates. NIS can only be updated by transferring an entire map to a slave or client. NIS uses no authentication between computers on a network. This poses a serious threat to security. NIS maps will be secured in such a way that a malicious user cannot easily obtain them. The best way to do this is to make the NIS domain name hard to guess. NIS can be easily misconfigured and contains several well-known vulnerabilities, making it difficult to secure systems using NIS. For that reason and others, NIS should not be used.

NIS will be removed from Mac OS X workstations.

- *(OSX1026SYS0011: CAT II) The SA will ensure that NIS is removed from all Mac OS X workstations.*

5.2 Network File System (NFS)

Network File System (NFS) allows clients to access filesystems located on remote servers as though the filesystems were resident on the clients. This allows a filesystem to be stored in one common location and securely *exported* to many clients at once instead of replicating it across many systems. NFS has the capability to enforce security policies for exported/shared filesystems. A security concern is presented with NFS because filesystems are physically

located on remote servers and users can access and change the data without logging on to the server. This would appear to defeat the I&A requirements. This is also true for remote databases. If access to files is properly restricted, however, file security can be greatly enhanced.

Several steps are required to secure NFS against most forms of unauthorized access. The file (either /etc/exports or /etc/dfs/sharetab) that indicates which filesystems the server exports and the level of access assigned to clients of those filesystems will be protected against unauthorized modification. Exported/shared system files will be owned by root and will not be world or group writable. Filesystems exported as other than *read only* will be documented with the IAO. These steps prevent sensitive system files from being modified or replaced.

Several options must be enabled in the NFS server file export configuration file (/etc/exports, for instance). The anon option should be set to disallow access from client requests that do not include a userid. The access option grants filesystem access only to those hosts or netgroups listed with the option. The secure option is used if secure RPC is enabled on the system (true if NIS+ is enabled on the system). This allows NFS to use DES (Data Encryption Standard) for encrypting the authentication session between the server and client. The root option overrides the default userid mapping of root access in NFS, and will not be used unless authorized and documented with the IAO. NFS clients will use the nosuid and nosgid options to mount filesystems from a server to prevent setuid and setguid executables of dubious origin from gaining root access on the client system. Port monitoring causes NFS requests that do not come from privileged ports to be rejected. Port monitoring will be enabled.

Because NFS presents such a target of opportunity for attackers, the NFS daemons will not be allowed to run unless NFS is actually being used.

- *(OSX1026GEN0178: CAT II) The SA will ensure that the /etc/exports (or the equivalent) file is owned by root and have permissions of 644, or more restrictive.*

- *(OSX1026GEN0181: CAT II) The SA will ensure that exported system files and directories are owned by root.*

- *(OSX1026GEN0180: CAT II) The SA will ensure that file systems are exported as read only unless an operational requirement warrants otherwise.*

- *(OSX1026GEN0180: CAT II) The SA will ensure that file systems containing system executables used by the local host are exported as read only.*

- *(OSX1026GEN0180: CAT II) The SA will ensure that any file systems that must be exported with permissions other than read only are documented.*

- *(OSX1026GEN0182: CAT II) The SA will ensure that the anon option in the /etc/exports file is set to anon=65535, 60001, or anon=-1.*

- *(OSX1026GEN0183: CAT II) The SA will ensure that the access and secure options are used for all entries in /etc/exports, /etc/dfs/dfstab, or the equivalent file, where available.*

- *(OSX1026GEN0185: CAT II) The SA and IAO will ensure that root access options are not*

used unless authorized and documented with the IAO.

- *(OSX1026GEN0186: CAT II) The SA will ensure that NFS clients will mount file systems with the nosuid and nosgid options set.*

- *(OSX1026GEN0177: CAT II) The SA will ensure that if NFS is running, NFS port monitoring is enabled.*

- *(OSX1026GEN0180: CAT II) The SA will ensure that NFS files will not be exported to a foreign domain (outside the local area network) without justification documented with the IAO, IAM, and NSO.*

5.3 Samba

Samba is a technology to allow file and printer sharing between Mac OS X and Microsoft Windows operating systems. Mac OS X systems use TCP/IP as their networking protocol, while Windows uses Session Message Block (SMB). Windows systems share files by using the Common Internet File System (CIFS), which uses SMB and the Network Basic Input Output System (NetBIOS) interface to share network resources. Samba was created to make a UNIX systems in this case Mac OS X appear to be a Windows system on a network, allowing it to become part of a Windows domain. This allows for easy sharing of files, directories, and printers.

Samba is actually a package of programs. The smdb daemon provides file and printer sharing, while the nmdb daemon provides NetBIOS name resolution and service browser support. Several utilities allow for FTP-like access, mounting and unmounting of shared directories, and checking status of the smb server. Samba also includes an administration tool called the Samba Web Administration Tool (SWAT) that provides a GUI to configure the /etc/smb.conf file through a web browser. When sharing network files and printers, access can be granted in two different ways. In *share mode* one password is set for each shared resource, and any user that knows the password can access it. In *user mode* each user has their own individual password, which is stored in the smbpasswd file (which is in the /etc directory by default, but may be placed elsewhere as determined by the smb.conf configuration).

While Samba provides a service that may be necessary, it does so with some risk. SWAT runs as a Linux service on port 901 by default, and requires a root logon to be accessed. If SWAT is to be used to administer Samba, it will be redirected through ssh to encrypt the root logon and the following configuration information. The /etc/smb.conf file will be owned by root, have a group of root, and have permissions of 644, or more restrictive. The smbpasswd file will be owned by root, have a group of root, and have permissions of 644, or more restrictive. The /etc/smb.conf file will be configured to allow access only to machines on the local network, require the user access mode, password encryption, and have shares defined with guest set to No.

- *(OSX1026SMB0001: CAT II) The SA will disable smb if file sharing with Windows is not implemented.*

- *(OSX1026SMB0003: CAT II) The SA will configure root as the owner of /etc/smb.conf.*

- *(OSX1026SMB0004: CAT II) The SA will configure root as the group owner of /etc/smb.conf.*

- *(OSX1026SMB0005: CAT II) The SA will configure the permissions of /etc/smb.conf to 644, or more restrictive.*

- *(OSX1026SMB0006: CAT II) The SA will set the immutable attribute on /etc/smb.conf.*

APPENDIX A. RELATED PUBLICATIONS

Government Publications

Department of Defense (DOD) Directive 8500.1, "Information Assurance", October 2002.

Department of Defense (DOD) Instruction 8500.2, "Information Assurance (IA) Implementation," February 2003.

Defense Information Systems Agency (DISA)/Chief Information Officer, Memorandum for Distribution, "DISA Standard Computer Configurations," Version 1999-A, November 1998.

Defense Information Systems Agency Instruction (DISAI) 630-230-19, "Security Requirements for Automated Information Systems (AIS)," July 1996.

Defense Information Systems Agency (DISA)/Defense Information Services Organization (DISO) Naming Convention Standards, March 1994.

National Security Agency (NSA), "Information Systems Security Products and Services Catalog" (Current Edition).

National Security Agency (NSA), "Guide to Securing Microsoft Windows 2000 Active Directory," Version 1.0, December 2000.

National Security Agency (NSA), "Guide to Securing Microsoft Windows 2000 File and Disk Resources," Version 1.0, 19 April 2001.

National Security Agency (NSA), "Guide to Securing Microsoft Windows 2000 Group Policy: Security Configuration Tool Set," Version 1.2, December 2002.

National Security Agency (NSA), "Guide to Securing Microsoft Windows NT Networks," Version 4.2, 18 September 2001.

Defense Logistics Agency Regulation (DLAR) 5200.17, "Security Requirements for Automated Information and Telecommunications Systems," 9 October 1991.

Field Security Operations Publications

DISA Computing Services Security Handbook

Windows 2000 Addendum

Desktop Application STIG

Network Infrastructure STIG

Web Server STIG

General Information Sites

http://www.apple.com	Macintosh creators and designers of the OS. This site contains all Security Related Documents for Mac OS X.
http://www.auscert.org.au	Australian Computer Emergency Response Team. They maintain security "how to" documents.
http://www.cert.mil	Defense Information Systems Agency (DISA) DOD-CERT (Department of Defense - Computer Emergency Response Team)
http://www.cert.org	A focal point for the computer security concerns of Internet users
http://www.ciac.llnl.gov/	The U.S. Department of Energy's Computer Incident Advisory Capability
http://www.cs.purdue.edu	COAST (Computer Operations, Audit, and Security Technology) focuses on real-world research needs.
http://www.csrc.nist.gov	National Institute of Standards and Technology's Computer Security Resource Clearinghouse
http://www.datahouse.disa.mil	Defense Information Systems Agency (DISA) Home Page
http://www.macosxhints.com	Macintosh and Unix guide for script writing
http://www.nsi.org	National Security Institute's Security Resource Net Home Page
https://vms.disa.mil	Vulnerability Compliance Tracking System (VCTS)
https://vms.disa.smil.mil	Vulnerability Compliance Tracking System (VCTS) (Secret and Confidential)

APPENDIX B. File and Directory Permissions Table

FILE and DIRECTORY	*PERMISSION*	*OWNER*	*GROUP*	*EXIST*	*FORBIDDEN*
/bin/csh	755	privileged	privileged	n	n
/bin/sh	755	privileged	privileged	n	n
/dev/kmem	640	root	sys	y	n
/dev/mem	640	root	sys	y	n
/dev/null	666	root	sys	y	n
/etc/ftpusers	640	root	root	y	n
/etc/hosts.equiv	600	root	root	n	n
/etc/host.lpd	664	root	root	n	n
/etc/inetd.conf	440	root	root	y	n
/etc/passwd	644	root	root	y	n
/tmp	1777	privileged	privileged	n	n
/usr/bin/rsh	755	privileged	privileged	n	n
/var/mail	1777	privileged	privileged	n	n
/var/tmp	1777	privileged	privileged	n	n

APPENDIX C. Procedures for Bringing a Mac OS X System Into STIG Compliance

ADDING THE LOCK SCREEN FEATURE TO THE MENU BAR

- Lock Screen Menu Item, by hand.
  - Open the Property List Editor.
  - Open the /Library/Prefernces/com.apple.systemuiserver.plist file.
  - Expand Root.
  - Expand menuExtras.
  - Highlight menuExtras and Select the New Child Button.
  - Let it auto number the entry.
  - Place /Applications/Utilities/Keychain Access.app/Contents/Resources/Keychain.menu in the Value Field.
  - Save the plist file.
  - Quit the application.
  - Restart the System.

Or

- Lock Screen Menu Item, Using Keychain Utility
  - Open the Keychain Access Utility from the Utilities Folder.
  - Click on the View Menu.
  - Choose Show Status in Menu Bar.
  - Quit out of the application.
  - Restart the System.

REMOVING APACHE FROM MAC OS X

Apache

chmod –rl 000 httpd from the /etc directory.
rm –rf /System/Library/StartupItems/Apache

REMOVING SENDMAIL FROM MAC OS X

Sendmail

rm /usr/sbin/sendmail
rm –rf /usr/share/sendmail
rm –rf /System/Library/StartupItems/Sendmail

REMOVING BIND FROM MAC OS X

DNS (BIND) or named

rm –rf /Library/StartupItems/DNS (If it exists.)
rm /usr/sbin/named
rm /usr/sbin/ndc
rm /usr/sbin/named-bootconf.
rm /etc/named.conf

REMOVING SSH FROM MAC OS X

SSH

rm /usr/sbin/sshd
rm /usr/bin/ssh
rm –rf /System/Library/StartupItems/SSH

This page is intentionally left blank.

APPENDIX D. ACRONYM

C&A	Certification and Accreditation
DECC	Defense Enterprise Computing Center
DECC-D	Defense Enterprise Computing Center-Detachment
DISA	Defense Information Systems Agency
DOD	Department of Defense
DSN	Defense Switched Network
DAC	Discretionary access control
FIPS	Federal Information Processing Standard
FSO	Field Security Operations
IAO	Information Assurance Officer
IAM	Information Assurance Manager
IASE	Information Assurance Support Environment
IAVA	Information Assurance Vulnerability Alert
IAVM	Information Assurance Vulnerability Management
IT	Information Technology
Mac	Macintosh
OS	Operating System
PDI	Potential Discrepancy Item
PIN	Personal Identification Number
SRR	Security Readiness Review
SRRDB	Security Readiness Review Database
STIG	Security Technical Implementation Guide
VCTS	Vulnerability Compliance Tracking System
VMS	Vulnerability Management System