Mac OS X 10.0 Security Essentials

Roland Miller

Mac OS X 10.0 Security Essentials

Roland E. Miller, III

August 21, 2001

GSEC Version 1.2e

Introduction.

On March 24th, 2001, Apple shipped the first release of its next generation operating system called Mac OS X (Apple 7, p. 1). Version 10.0 was preceded by a Public Beta release on September 13, 2000 (Apple 4, p.1). A discussion of the lineage of the operating system and a preliminary analysis of the security implications of the Public Beta has already been conducted (Crow, p.1). This report constitutes an evaluation of the out-of-the-box security of the shipping version of Mac OS X. The security features and vulnerabilities evaluated in this report form the foundation for most security aspects of the operating system and therefore should be considered essentials for anyone involved in maintaining a Mac OS X system. As of this writing, the current version of Mac OS X is 10.0.4 Build 4Q12 for most systems. This report will evaluate this particular version and build with the Web Sharing Update 1.0 installed (Apple 10, p.1).

It should be noted that there is also a server version of Mac OS X shipping called Mac OS X Server 10.0, which is based on the Mac OS X core. The server version has essentially kept sync with the general version with regard to security patches, updates, and version numbers. Virtually all of what is discussed in this report will apply to the server version. However, the issues detailed here are not all inclusive for the server product as there is much more functionality with security connotations contained in the server version than in the general release. In addition, Mac OS X 10.1 has been announced for release in September of this year (Apple 3, p.1). However, no build of this version has been officially released to the public and therefore it will not be evaluated.

The security aspects of Mac OS X 10.0.4 will be divided into apparent strengths and weaknesses and discussed in some detail. Parallels to the Windows platform and the UNIX platform will be drawn when appropriate. When possible, corresponding solutions and pitfalls will also be presented. The report will conclude with an evaluation of the overall security aspects of the operating system.

2.0 Apparent Weaknesses: Vulnerabilities and Potential Solutions.

This report starts with the apparent weaknesses of the operating system; not because it is an inherently weak system, but because the effects of some of the security weaknesses presented in this section can be lessened by careful planning before installation occurs. It is also essential to understand what is installed by default, how it is installed and how this may introduce vulnerabilities into the system. When possible, the apparent weaknesses described will have a potential solution or solutions prescribed.

2.1 Installation: Dual Booting & HFS+ vs. UFS.

When it was first released, Mac OS X was available only as a separate shrink-wrapped product that required subsequent user installation on an existing system. On May 21, 2001, Apple began shipping all new systems with Mac OS X preinstalled (Apple 5, p.1). The installation on new systems is a dual-boot configuration between Mac OS 9.1 and Mac OS X, with Mac OS 9.1 being the default startup environment.

Shipping systems are preformatted with one HFS+ partition on which both operating systems reside. This is similar to installing Windows NT/2000 on the same partition as an existing install of Windows 9x. Just as in the Windows case, the overall security of the system itself is dictated by the less-secure operating system. Local security on a Mac OS 9.1 installation is essentially non-existent. By booting into Mac OS 9.1, a user will have full and complete access to all files on the drive including files that have root-only privileges under Mac OS X.

Clearly, one solution to this issue is to remove Mac OS 9.1 completely from the system. Unlike the Windows scenario where the two operating systems are truly separate, Mac OS X relies on Mac OS 9.1 to support its Classic environment. The Classic environment is analogous to the NT virtual DOS machine under Windows NT. Mac OS X boots up a copy of Mac OS 9.1 as a separate process under which Classic (i.e., legacy) applications run. Mac OS X cannot run older applications without a fully functional, and consequently fully bootable, Mac OS 9.1 installation. This is an issue at this point in Mac OS X's lifecycle as there are not many native applications available. If there is a need to run legacy applications and a concern over dual-boot security, a potential solution involving supported file systems exists.

Mac OS X supports two different file systems for installation, Mac Extended File system, abbreviated as HFS+, and UNIX File System, abbreviated as UFS. Mac OS 9.1 only supports Mac Standard File system (HFS) as well as Mac Extended File system (HFS+) for installation. This presents a situation nearly identical to the one between Windows 9x and Windows NT/2000 with regard to supported file systems. On the Windows platform, it is considered good practice to format the boot partition in secure environments with NTFS under Windows NT/2000. If Windows 9x is required, then it can be installed on a separate partition that is formatted as FAT32. Since Windows 9x cannot read NTFS, the less secure operating system is essentially walled off from affecting the more secure operating system through direct file manipulation. The same principal can be used to secure an install of Mac OS X by installing on a partition formatted as UFS and keeping Mac OS 9.1 on a separate partition formatted as HFS+. It should be noted that there are some caveats associated with using Mac OS X on a UFS formatted partition. There is actually a loss of functionality associated with this type of installation (Apple 6, p.1).

Apple suggests that Mac OS X should be installed on an HFS+ partition unless there is a clear reason to install on a UFS partition. However, there are security issues involved with installing Mac OS X on an HFS+ partition. These issues are due exclusively to the fact that HFS+ is a case-preserving file system (Apple 6, p.1). Mac OS X ships with Apache as its built-in web server and it has been shown that Apache security is compromised by simply changing the case of a web query (Arentz, p.1). Apple addressed this particular issue with Apache security with the release of Web Sharing Update 1.0. However, there is still the potential that the usage of HFS+ may have future security implications as more UNIX based utilities that assume they are operating on a case-sensitive file system are ported and compiled to Mac OS X.

2.2 Physical Security: Open Firmware and Single User Mode.

So many security implementations break down completely when there is a breach in physical security; Mac OS X is certainly no different. When someone can plug-in a FireWire hard drive to the system and boot from it, all bets are off regardless of partitioning and the file system used. Since most Mac OS X compatible hardware has built-in USB and FireWire ports, the easy addition of an alternate boot device or transfer disk is a real security consideration. In addition, it is possible to boot from a Mac OS 9.1 CD in the internal optical drive and have full access to the partitions on the system if they are formatted HFS+.

There is an additional risk to Mac OS X security from booting with a CD. Apple has a password reset utility on the Mac OS X installation CD. The procedure to reset the root password or the password of any other account is to boot off the CD, run the utility, select the user account and type in the new password. This is considered a feature of Mac OS X by Apple, which makes sense considering that this is an operating system currently shipping to the home consumer and education markets. While this application will only work if the system is booted from the Mac OS X installation CD, its existence on every copy of the operating system makes it a security vulnerability that should not be ignored by anyone trying to maintain a secure system.

All of these physical security issues have a potential solution in Open Firmware. Open Firmware on Macintosh systems is directly analogous to BIOS and BIOS Setup on PC-compatible hardware (Apple 1, p.1). Like BIOS Setup, boot-time variables such as boot device and boot password security can be managed through Open Firmware. Version 4.1.8 of Open Firmware contains security features that can be activated and password protected (CodeSamurai 2, p.1). It is possible to prevent booting from the CD-ROM drive (which is enabled by holding down the "c" key at boot) and booting from another device (which is enabled by holding down the "option" key at boot) by setting a security option. Booting into Open Firmware itself (which is enabled by holding down "command-option-o-f" at boot) and changing or resetting these parameters is protected by a password. There is now a third-party GUI called Startup Security that will implement these same options (Digital Specter, p.1-2). This utility runs under both Mac OS X and Mac OS 9.1. Implementing Open Firmware security options in environments where physical security cannot be guaranteed such as computer labs should be considered.

Just as resetting a BIOS password is typically a known procedure, the resetting of the Open Firmware security settings and password has been shown to be straight-forward (Marukka 3, p.1). Open Firmware security functionality can be reset by simply removing a DIMM from the system and resetting the PRAM (by holding down "command-option-p-r" at boot) multiple times. This was presented as a vulnerability, however, since it requires direct access to the inside of the case, it is most likely an intentional reset feature. The best way to counter this is to lock the case, which most Apple hardware allows quite easily; once again illustrating the significance of basic physical security.

In a very recent development, an application called FWSucker 1.0 has been released that displays the Open Firmware password once the system is booted (mSec, p.1). It reads the NVRAM and locates and decrypts the Open Firmware password. It is a Classic application; therefore it requires Mac OS 9.1 and the Classic environment to run. Limited testing of the application in Mac OS X under the Classic environment has shown that it does not work. However, it appears

to work inconsistently when run in Mac OS 9.1. This version does not appear to pose a threat to systems that cannot be booted directly into Mac OS 9.1.

There is one additional caveat concerning the security implementation of Open Firmware. Apple does not support its use at this time (Apple 9, p.1). This is somewhat unusual considering it is a listed benefit of the Open Firmware 4.1.8 upgrade. Nevertheless, use of the Open Firmware security feature currently will void the warranty if there is damage to the system resulting from its use. Presumably, Apple is working on a utility and instructions to use this security feature that will be available in the near future.

Once past the initial hardware startup phase, there exists a serious security vulnerability that results in root access on the system. Mac OS X, like many UNIX variants, can be booted into what is known as single user mode. This mode on Mac OS X comes up by default with full root privileges without any authentication. By simply holding down the "command" and "s" keys at startup, root access to the system from the command line interface (CLI) is provided. Exploitation of this vulnerability is well documented (CodeSamurai 1, p.1).

There are currently two very different solutions to this vulnerability. The first involves a patch to the mach_init so that the system cannot be booted into single user mode (Marukka 1, p.1). The un-patched version of mach_init is not saved; therefore removal of the patch requires a copy of the original mach_init. The second solution involves a perl script called SecureIt (Boor, p.1). The script forces authentication when booted into single user mode and it cannot be bypassed to get to the CLI. The password can be set to anything; it does not have to be the actual root password. This password is stored encrypted and readable only by root. In addition, there is an uninstall option for complete removal of this utility. Ultimately, the only real solution to this serious vulnerability will have to be provided by Apple in a future version of the operating system.

2.3 Non-essential Installs: BSD Subsystem and Developer Tools.

The default installation of Mac OS X has an optional installation package checked by default called BSD Subsystem. According to the Installer, the BSD Subsystem "contains the optional components of the 4.4 BSD operating system." An examination of the install package reveals that there are over 2900 files installed. The majority of these files are CLI utilities (installed into the /usr directory) and documentation. If there is a concern about what a legitimate user or a network intruder can do from the CLI on Mac OS X, then foregoing the BSD Subsystem install will greatly reduce the number of commands available. It should be noted that the BSD Subsystem includes most of the potentially troublesome NetInfo utilities that are discussed in Section 2.4.

The shrink-wrapped version of Mac OS X also includes another CD besides the installation CD called Developer Tools. It requires that the BSD Subsystem package already be installed and includes various CLI compilers, debuggers, Apple's IDE, documentation and example files. Having the Developer Tools installed could potentially help any hacker who likes to, or needs to "roll his/her own" by providing the necessary files for compiling locally. There really is no need to help an intruder out in this way, especially if the tools and programs contained on the CD are

not going to be used for legitimate purposes such as software development or UNIX utility compilation.

2.4 Account Security: NetInfo.

Probably the most discussed aspect of Mac OS X security has been its reliance on the directory service called NetInfo for major portions of the operating system's functionality. NetInfo has been around since the days of NeXT Computer and is responsible for, among other things, account maintenance (Apple 8, p.1). One of the most significant security issues for Mac OS X is that all the password hashes are stored with readable by all others permission set within NetInfo and consequently can be accessed by several utilities using any local account (SecureMac.com, p.1). Basically, anyone on the system can easily get the password hashes for root or any other account and begin cracking them.

Virtually all of the security advisories regarding this vulnerability only mention the NetInfo CLI utility nidump as the source of the exploit, however, the password hashes can also be easily accessed using nigrep, nireport and niutil, the NetInfo Manager graphical user interface (GUI) utility, or by accessing the NetInfo backup file located at /var/backups/local.nidump. Besides the built-in methods of accessing the password hashes, a program called Malevolence has been released that will do the job as well (Marukka 2, p.1). This program is a CLI utility that reads the NetInfo database and formats the password hash information into HTML saving it as a file called index.html. To complete the password recovery process there exists several password-cracking programs for the Classic environment as well as the UNIX program crack, which has been ported natively to Mac OS X.

One approach to minimize the effects of this vulnerability is to remove the permissions for all others from the NetInfo CLI utilities (all except niutil are part of the BSD Subsystem), the NetInfo Manager and the local.nidump file (SecureMac, 1). Making these changes does not appear to negatively affect the functionality of the operating system after weeks of normal use. This is a stopgap measure and does not prevent anyone on the system from bringing their own copy of nidump, NetInfo Manager or Malevolence to extract the password hashes. This particular vulnerability will only be put to rest once Apple alters the way NetInfo works with an eye towards closing this vulnerability.

2.5 MalWare.

To date, there has not yet been a report of a Mac OS X native virus, worm, trojan or other malware. At this point in Mac OS X's lifecycle, the operating system's risk comes primarily from the Classic environment in the form of malware written and targeted for earlier versions of the Mac OS. Additionally, Mac OS X can also be a carrier for malware, which will then become active when sent to an appropriate platform.

While it appears that most malware created is compiled in x86 code, which will not run on Mac OS X compatible hardware, or assumes a Windows naming and file structure, these facts should not be used as a predictor for future activity nor should it be a justification for complacency. Most of the underlying functionality that is exploited on other platforms also exists in Mac OS

X. For instance, Mac OS X includes scripting functionality similar to VBScript on Windows systems called AppleScript. This functionality was used to create the Mac/Simpsons@MM worm that is similar to the ILOVEYOU worm in terms of its methods (McAfee 2, p.1-2). It is interesting to note that the usage of AppleScript for malware in this way was predicted (Bishop, p. 7-47).

Like most operating systems, the job of virus detection and eradication for Mac OS X is left to third-party vendors. There are two native versions of antivirus software that are currently in public beta form only, Virex (McAfee 1, p.1) and Sophos Anti-Virus (Sophos, p.1). Fortunately, existing Classic application versions of antivirus software will also run in Mac OS X. With numerous Classic versions of virii, worms and trojans and native malware undoubtedly on the way, it is prudent that best practices with regard to malware be performed on Mac OS X systems.

3.0 Apparent Strengths: Built-in Security Features and Potential Pitfalls.

Mac OS X has many built-in security features, some of which are leveraged from the UNIX core of the operating system. Of course, there is no perfect security feature and the features in Mac OS X are no exception. Where appropriate, potential pitfalls regarding the security features discussed will be presented as well.

3.1 Features Disabled or Restricted by Default: Services and Accounts.

As was the case with the Public Beta (Crow, p.1), Mac OS X ships with a wide-variety of services of which virtually all are disabled by default. OpenSSH, FTP, AppleTalk, ntp, file sharing and HTTP are all disabled, but can be enabled easily through the GUI. In addition to these, there are many other services installed by default or are included with the BSD subsystem. These services are also disabled, but cannot be enabled via the GUI. These include numerous traditional UNIX daemons such as named, telnetd and the remote utility daemons. Activating these services requires using the CLI, a foundation in UNIX and an understanding of how the UNIX core has been altered to work with the overlying layers of Mac OS X.

When Mac OS X is booted for the first time, the Setup Assistant creates one initial account. This account is the administrator account, which is different from the root account in Mac OS X. This account can have a blank password, although Setup Assistant warns, “This is a security risk. Do you want to continue?” The administrative account has a subset of the privileges that are normally given to the root account on UNIX systems. Mac OS X is setup by default to login automatically with this account mimicking the startup and security state of Mac OS 9.1. This setting can be changed in the GUI.

One of the privileges of the administrator account is the ability to create new accounts on the system through the GUI. The password length must be at least 4-characters, although it is possible to go into NetInfo and make a null password for the account. This is an inconsistency with the Setup Assistant which has no password length restrictions for the administrative account. On Mac OS X, all system functionality tied to the UNIX core actually truncates the password at 8 characters. This is a common characteristic of BSD based operating systems.

Other system functionality such as the Keychain, which is not derived from BSD, is not bound by this 8-character limit.

The root account is setup at installation as well, but is completely disabled. The procedure to enable the root account from the GUI is somewhat convoluted. It requires being logged on as the administrator account (or an account with administrative privileges) and launching the NetInfo Manager utility. Under a submenu, the administrator account is required to authenticate once again against NetInfo. This will activate a submenu option that activates the root account. The root account has a blank password by default and a warning is presented that the root account cannot have a blank password. The password for the root account can now be set and the account can be left enabled or disabled again. Once enabled, the root account is prevented from logging into the system through any remote service.

The root password can later be changed through the same procedure. The GUI requires that the existing root password be entered before it can be changed. However, it is possible to substitute a known password hash for the existing root password hash in NetInfo, thereby making this entire procedure and the GUI authentication requirements moot. In addition, the root password can be set to null using this procedure. All of the functionality of the GUI application NetInfo Manager is duplicated in the NetInfo CLI utilities.

The concept in Mac OS X of an administrator account and a separate but disabled root account is clearly an attempt to preserve the operating system's integrity from less knowledgeable system owners as well as errant applications or malware. A direct result of this implementation is that Mac OS X is a more secure system by default than many UNIX distributions.

3.2 Software Update and Security Patches.

Dissemination and installation of security patches is a fundamental security procedure. Mac OS X's built-in software update functionality is an application appropriately called Software Update. Since its initial release in March of this year, Mac OS X has had four incremental version updates and a separate security patch. Three of the four version upgrades contain security patches and the Web Sharing Update 1.0 addressed several security issues as well (Apple 2, p.1). All of these updates were available first through Software Update.

Like all things, Software Update has some pitfalls associated with it. Besides the obvious fact that it is voluntary, the way the software works and is scheduled by default potentially leaves the system open to known vulnerabilities that have patches. While Microsoft's Windows Update feature is disabled by default, when it is enabled, it runs every five minutes and the timing cannot be changed. Mac OS X Software Update is enabled by default, but is setup initially to run once a week. This introduces a seven-day gap where the system may be vulnerable to a widely disclosed and actively probed exploit. In addition, some of Apple's updates are dependent upon the installation of previous updates. When the software runs and downloads an update, it won't come up automatically for another seven days to install additional updates. The recent Code Red worm illustrates how much can happen in just seven days and also how critical security patches can be.

One question that remains is how fast will Apple react to security issues. The software update feature of the operating system is clearly a very powerful tool in helping to maintain a secure system. However, it is virtually useless unless Apple provides the security patches in a timely manner.

3.3 UNIX Core: OpenSSH, ipfw and tcpwrappers.

Apple touts the power of the underlying UNIX core in its marketing for Mac OS X. The power of the UNIX core in the Public Beta as related to security has already been examined (Crow, p.1-2). What has changed since the Public Beta as well as additional resources will be presented in this section.

For the remote access functionality, Apple chose to replace telnet in Mac OS X Public Beta with OpenSSH in the shipping version. OpenSSH is not enabled by default; however, simply checking a box in the GUI activates it. With OpenSSH enabled, the system can be securely connected to remotely. Apple has updated the version of OpenSSH in Mac OS X twice already, keeping pace with the current open source release. As of this writing, OpenSSH is at version 2.9p2 and that is what is installed with the Web Sharing Update 1.0. In addition, there now exist numerous freeware GUI utilities that will help setup and manage OpenSSH.

Mac OS X still comes with a built-in IP firewall (ipfw) that is derived from FreeBSD. The operating system has no rule sets by default, which means all IP traffic is allowed. Setting up a rule set for ipfw using the CLI is detailed in the man pages. In addition, there are now two shareware GUI utilities that make implementing rule sets for ipfw much more manageable. They are BrickHouse (Hill, p.1) and FireWalk X (Boor, p.1). Both of these GUI utilities allow advanced rule sets to be generated and provide convenient access to the logs generated by ipfw.

Tcpwrappers is also still built-in, but the necessary configuration file hosts.allow and hosts.deny are not setup, therefore, all TCP traffic is allowed by default. A tutorial on how tcpwrappers works and how to configure it for an older version of Mac OS X Server exists (Swan, p.1-4). It appears that most of the information contained in the article is still valid for the current version of Mac OS X. There are no GUI utilities to setup tcpwrappers at this time. It should be noted that xinetd has been ported to Mac OS X and can be used in place of tcpwrappers (Curator, p.9-11).

Since the previous two services are essentially wide-open by default and there is no GUI for them included with Mac OS X, it can be debated whether these are inherent strengths of the operating system. They are listed here simply because they are powerful security tools that are available and installed by default and can be activated easily by a knowledgeable user using the CLI or any number of widely available third-party utilities.

It should be noted that along with all of this power comes many of the pitfalls associated with running a UNIX operating system. While Mac OS X brings file level security for the first time to the Mac OS, it also brings with it the necessity in a secure environment to audit for suid permissions for example. The potential for root-kits and other deep system compromises that may not be immediately obvious at the GUI level should be kept in mind. Fortunately, many of the tools and procedures used to combat these threats on a traditional UNIX system are present

and will work in Mac OS X. Tools and services such as tcpdump, netstat, nmap, lsof, syslog, wtmp are present and working in the operating system. In addition, many UNIX security utilities have been ported and compiled to run on Mac OS X such as snort and crack. While these tools do not ship with the operating system, the UNIX core of Mac OS X allows them to work exactly as they would in a pure UNIX environment, making some potentially powerful security tools available. Apple has already included many of the security tools of choice in the operating system and consequently may include others in the future.

3.4 Internet Resources: For More Information.

The old adage "information is power" is certainly true when it comes to working on either side of the information security battle lines. The most powerful security feature of any operating system is up-to-date and accurate information in the possession of the systems administrator. Besides the information and references presented in this paper there are a variety of information resources concerning the security aspects of Mac OS X on the Internet. A list of Internet security resources for the Macintosh has already been compiled (Harris, p. 4). For reference, information resources from that list that need to be updated along with links to additional information resources specific to Mac OS X are presented below.

Macintosh Security Information Sites

Apple Product Security – http://www.apple.com/support/security/
Apple Computer Security Mailing List – http://lists.apple.com/mailman/listinfo/security-announce
MacInTouch Security Resources – http://www.macintouch.com/security.html
Mac Security.org – http://www.macsecurity.org
Mac Security.org Mailing Lists – http://www.macsecurity.org/mailman/listinfo
SANS Institute: Mac Section – http://www.sans.org/infosecFAQ/mac/mac_list.htm

Antivirus Software

McAfee Virex – http://www.mcafeeb2b.com/products/virex/default.asp
Sophos Anti-Virus – http://www.sophos.com/products/antivirus/savmac.html

Firewall Software

BrickHouse – http://personalpages.tds.net/~brian_hill/brickhouse.html
FireWalk X – http://www.users.qwest.net/~mvannorsdel/firewalkx/index.html
Norton Personal Firewall for Macintosh – http://www.symantec.com/sabu/nis/npf_mac/

Mac Exploit Software

Freak's Macintosh Archive – http://freaky.staticusers.net/index2.shtml
mSec – http://www.msec.net
ParallaX Research Group – http://www.parallaxresearch.com
Team2600 – http://team2600.33holding.com

4.0 Conclusions.

Most of the apparent weaknesses of Mac OS X presented here represent local exploits. Many of these vulnerabilities have workable solutions or they can be minimized. The apparent strengths of the operating system include a default installation that is resistant to network exploits. In addition, there are many tools that can be used to further tighten the operating system and a very easy method to apply system updates is in place.

The positive and negative consequences of the UNIX subsystem on the security of Mac OS X have only been briefly touched upon and can certainly use further research. With Mac OS X 10.1 coming out soon, the effects of the new version on what was presented here will need to be investigated. In addition, a detailed look at the security needs and implications of the server version of Mac OS X would certainly compliment this report.

Clearly, Apple has given attention to the basic security of Mac OS X while still maintaining the end-user usability and experience that makes the platform popular. The product's role as a high powered, full-featured operating system and the need to cater to the lowest skill level users is a very difficult balancing act that can and will continue to have repercussions on the security of the operating system. While there are some problems that have cropped up, it does appear that Apple has managed to keep Pandora's Box closed with the shipping version of Mac OS X (Crow, p.2).

References

Apple Computer, Inc. “Apple Computer Open Firmware Home Page.” August 15, 2001. URL: http://bananajr6000.apple.com/.

Apple Computer, Inc. “Apple Computer Product Security Incident Response.” July 30, 2001. URL: http://www.apple.com/support/security/security_updates.html.

Apple Computer, Inc. “Apple Previews Next Version of Mac OS X.” July 18, 2001. URL: http://www.apple.com/pr/library/2001/jul/18macosx.html.

Apple Computer, Inc. “Apple Releases Mac OS X Public Beta.” September 12, 2000. URL: http://www.apple.com/pr/library/2000/sep/13macosx.html.

Apple Computer, Inc. “Apple to Pre-Install Mac OS X Ahead of Schedule.” May 21, 2001. URL: http://www.apple.com/pr/library/2001/may/21macosx.html.

Apple Computer, Inc. “Mac OS X 10.0: Choosing UFS or Mac OS Extended (HFS Plus) Formatting.” AppleCare Knowledge Base, Article ID: 25316. August 31, 2001.

Apple Computer, Inc. “Mac OS X Hits Stores This Weekend.” March 21, 2001. URL: http://www.apple.com/pr/library/2001/mar/21osxstore.html.

Apple Computer, Inc. “Mac OS X Server: What Is NetInfo?” AppleCare Knowledge Base, Article ID: 60038. February 2, 1999.

Apple Computer, Inc. “Macintosh: Open Firmware Password Protection.” AppleCare Knowledge Base, Article ID: 106292. April 25, 2001.

Apple Computer, Inc. “Web Sharing Update 1.0 : Document and Software.” AppleCare Knowledge Base, Article ID: 120040. July 17, 2001.

Arentz, Stefan. “Mac OS X – Apache & Case Insensitive Filesystems.” June 10, 2001. URL: http://www.shmoo.com/mail/bugtraq/jun01/msg00098.shtml.

Bishop, Matt. “Detailed Analysis of the ILOVEYOU Worm.” 1.1: Security Essentials I. Denver: SANS Institute, 2001. 7-32 – 7-56.

Boor, Jason. “Scripts for OS X & Darwin.” August 13, 2001. URL: http://users.ez-net.com/~jasonb/secureit.html.

CodeSamurai. “Mac OS X Single User Mode Root Access.” April 9, 2001. URL: http://www.securemac.com/macosxsingleuser.php.

CodeSamurai. “Open Firmware Password Protection.” May 15, 2001. URL: http://www.securemac.com/openfirmwarepasswordprotection.php.

Crow, Thomas. “MacOS X: Is Apple Opening Pandora’s Box?” December 4, 2000. URL: http://www.sans.org/infosecFAQ/mac/macOS_X.htm.

Curator. “An Unofficial Xinetd Tutorial.” July 4, 2001. URL: http://www.macsecurity.org/resources/xinetd/tutorial.shtml.

Digital Specter. “Startup Security.” August 17, 2001. URL: http://www.digitalspecter.com/startupsecurity.html.

Harris, Patrick. “Macintosh Internet Security Basics.” September 15, 2000. URL: http://www.sans.org/infosecFAQ/mac/mac_sec.htm.

Hill, Brian. “BrickHouse.” June 4, 2001. URL: http://personalpages.tds.net/~brian_hill/brickhouse.html.

Marukka. “Disable Single User Boot Mode under Mac OS X.” April 26, 2001. URL: http://www.securemac.com/disablemacosxsingleboot.php.

Marukka. “Malevolence 1.0.” April 23, 2001. URL: http://www.msec.net/software/index.html#malevolence.

Marukka. “OpenFirmware Password Bypass.” May 25, 2001. URL: http://www.msec.net/archives/of_pwd_bypass.html.

McAfee. “McAfee Anti-Virus Beta Program: McAfee Virex v7.0 OSX Beta 3.” July 18, 2001. URL: http://www.mcafeeb2b.com/beta/products/VIREX-intro.asp.

McAfee. “Virus Information Library: Virus Profile – Mac/Simpsons@MM.” June 8, 2001. URL: http://vil.mcafee.com/dispVirus.asp?virus_k=99102&.

mSec. “FWSucker 1.0.” August 15, 2001. URL: http://www.msec.net/software/index.html#fwsucker.

SecureMac. “Mac OS X nidump Security.” July 6, 2001. URL: http://www.securemac.com/macosxnidump.php.

Sophos. “Sophos Anti-Virus for Macintosh beta downloads.” August 16, 2001. URL: http://www.sophos.com/downloads/beta/mac.html.

Swan, Jay. “Configuring and using tcpwrappers.” April 13, 2001. URL: http://www.stepwise.com/Articles/Workbench/2000-04-08.01.html.

Vannorsdel, Mike. “FireWalk X.” August 15, 2001. URL: http://www.users.qwest.net/~mvannorsdel/firewalkx/index.html.

Last Updated: May 26th, 2019

Upcoming SANS Training

Click here to view a list of all SANS Courses

SANS San Antonio 2019	San Antonio, TXUS	May 28, 2019 - Jun 02, 2019	Live Event
SANS Atlanta 2019	Atlanta, GAUS	May 28, 2019 - Jun 02, 2019	Live Event
Security Writing NYC: SEC402 Beta 2	New York, NYUS	Jun 01, 2019 - Jun 02, 2019	Live Event
Enterprise Defense Summit & Training 2019	Redondo Beach, CAUS	Jun 03, 2019 - Jun 10, 2019	Live Event
SANS Zurich June 2019	Zurich, CH	Jun 03, 2019 - Jun 08, 2019	Live Event
SANS London June 2019	London, GB	Jun 03, 2019 - Jun 08, 2019	Live Event
SANS Kansas City 2019	Kansas City, MOUS	Jun 10, 2019 - Jun 15, 2019	Live Event
SANS SEC440 Oslo June 2019	Oslo, NO	Jun 11, 2019 - Jun 12, 2019	Live Event
SANSFIRE 2019	Washington, DCUS	Jun 15, 2019 - Jun 22, 2019	Live Event
SANS Cyber Defence Canberra 2019	Canberra, AU	Jun 24, 2019 - Jul 13, 2019	Live Event
Security Operations Summit & Training 2019	New Orleans, LAUS	Jun 24, 2019 - Jul 01, 2019	Live Event
SANS ICS Europe 2019	Munich, DE	Jun 24, 2019 - Jun 29, 2019	Live Event
SANS Cyber Defence Japan 2019	Tokyo, JP	Jul 01, 2019 - Jul 13, 2019	Live Event
SANS Paris July 2019	Paris, FR	Jul 01, 2019 - Jul 06, 2019	Live Event
SANS Munich July 2019	Munich, DE	Jul 01, 2019 - Jul 06, 2019	Live Event
SANS London July 2019	London, GB	Jul 08, 2019 - Jul 13, 2019	Live Event
SEC450 Security Ops-Analysis Beta 1	Crystal City, VAUS	Jul 08, 2019 - Jul 13, 2019	Live Event
SANS Cyber Defence Singapore 2019	Singapore, SG	Jul 08, 2019 - Jul 20, 2019	Live Event
SANS Charlotte 2019	Charlotte, NCUS	Jul 08, 2019 - Jul 13, 2019	Live Event
SANS Pittsburgh 2019	Pittsburgh, PAUS	Jul 08, 2019 - Jul 13, 2019	Live Event
SANS Rocky Mountain 2019	Denver, COUS	Jul 15, 2019 - Jul 20, 2019	Live Event
SANS Columbia 2019	Columbia, MDUS	Jul 15, 2019 - Jul 20, 2019	Live Event
SANS Pen Test Hackfest Europe 2019	Berlin, DE	Jul 22, 2019 - Jul 28, 2019	Live Event
SANS San Francisco Summer 2019	San Francisco, CAUS	Jul 22, 2019 - Jul 27, 2019	Live Event
DFIR Summit & Training 2019	Austin, TXUS	Jul 25, 2019 - Aug 01, 2019	Live Event
SANS Riyadh July 2019	Riyadh, SA	Jul 28, 2019 - Aug 01, 2019	Live Event
SANS July Malaysia 2019	Kuala Lumpur, MY	Jul 29, 2019 - Aug 03, 2019	Live Event
SANS Boston Summer 2019	Boston, MAUS	Jul 29, 2019 - Aug 03, 2019	Live Event
Security Awareness Summit & Training 2019	San Diego, CAUS	Aug 05, 2019 - Aug 14, 2019	Live Event
SANS Melbourne 2019	Melbourne, AU	Aug 05, 2019 - Aug 10, 2019	Live Event
SANS London August 2019	London, GB	Aug 05, 2019 - Aug 10, 2019	Live Event
SANS Crystal City 2019	Arlington, VAUS	Aug 05, 2019 - Aug 10, 2019	Live Event
SANS Krakow May 2019	OnlinePL	May 27, 2019 - Jun 01, 2019	Live Event
SANS OnDemand	Books & MP3s OnlyUS	Anytime	Self Paced