Interested in learning more about security?

SANS Institute
InfoSec Reading Room

This paper is from the SANS Institute Reading Room site. Reposting is not permitted without express written permission.

MacOS X: User Friendlier Security for Unix

One of the problems of computer security in practice is providing an easy mechanism for the user of a system to take advantage of the security features present in an operating system. A system may have significant security features, but absent an interface that allows the user to easily make use of those features the effective security of the system may be low. FreeBSD is an Open Source Unix system primarily designed for use as an Internet server or development system by knowledgeable users. MacOS X ("Jaguar") is a con...

Copyright SANS Institute
Author Retains Full Rights

AD

MacOS X: User Friendlier Security for Unix
Raleigh F. Romine
GSEC Practical (v1.4b), Option 1

Summary

One of the problems of computer security in practice is providing an easy mechanism for the user of a system to take advantage of the security features present in an operating system. A system may have significant security features, but absent an interface that allows the user to easily make use of those features the effective security of the system may be low.

FreeBSD is an Open Source Unix system primarily designed for use as an Internet server or development system by knowledgeable users. MacOS X ("Jaguar") is a consumer-oriented system for the Apple Macintosh based in large measure on FreeBSD. In this paper, we explore the additions and modifications Apple has made to the FreeBSD core to enhance the security of the users of MacOS. We begin with a short history of the two systems, and continue with a discussion of installation and administration. In addition to the strictly program based aspects of security, we examine how the target marketing of MacOS has affected the effectiveness of its security.

The History of FreeBSD and MacOS

FreeBSD has its roots in the BSD version of Unix, specifically the 4.4BSD Lite version that provided the base for BSD as an open source Unix system. Building on the initial port of BSD to the Intel 386 platform done by William Jolitz, the FreeBSD team, led by Jordan Hubbard, produced a full distribution of 4.4BSD incorporating the core Berkeley operating system along with packages from other Open Source projects, notably the X Window system from XFree86 and many utilities from the GNU project. The distribution itself has always been free but has received significant support from Walnut Creek and Wind River Systems in the form of Internet and CD-ROM distributions and developer support. BSDi and later Wind River Systems marketed their own version of BSD, termed BSD/OS, while supporting development of FreeBSD. Currently, FreeBSD is maintained by its user community via a small group of "committers" enabled to modify the official source code. Two other major distributions of Open Source BSD exist: NetBSD, which is focused on portability across platforms, and OpenBSD, which emphasizes security.[1]

FreeBSD has attained a significant measure of acceptance in the commercial world with Yahoo! and, for a time, Hotmail using it as their operating system. It is

[1] See Howard 2001 for a discussion of the differences among the various Open Source BSD's.

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

tailored towards the server and developer communities, but with the addition of Linux compatibility has become usable as a desktop environment.[2]

Apple's operating systems for the Macintosh have long been noted for their user-friendliness, however by the 1990's it was becoming apparent that the core of MacOS had reached its limits. Rather than continue incremental improvements to the base system, Apple chose the radical approach of replacing the entire system and providing backward compatibility for its installed user base by means of an emulation program. The new operating system ("Darwin"), released in 2001, is based on the CMU Mach 3.0 kernel with FreeBSD providing system services and the NetInfo system providing centralized system administration. The GUI, "Aqua", is Apple proprietary, but the majority of the operating system is Open Source. This approach has enabled Apple to take advantage of the strengths of the FreeBSD system while adding their own expertise in user interface design. Apple has implemented several security enhancements such as Keychain, CDSA (the Common Data Security Architecture) and a basic AES encrypted file system in addition to relying on standard Open Source projects such as OpenSSH and OpenSSL when appropriate.[3] In addition, Apple has made much of the operating system available for peer review and modifications via the OpenDarwin and OpenOSX web sites.[4]

Initial Setup

Both systems follow the now standard practice of a GUI-based installation procedure, but the difference in intended users is apparent from the beginning. FreeBSD is designed to run on several types of CPU (although the vast majority of installations run on the Intel x86 platform) with a large variety of peripherals. MacOS only supports Apple Macintosh computers and standard peripherals such as USB and Firewire devices. For example, the initial stage of the FreeBSD installation procedure involves selecting various possible hardware combinations and resolving device conflicts, partitioning and formatting disks, and installing a boot manager before even loading the initial kernel; MacOS need only query the user for the installation drive and optional partitioning.

The FreeBSD installation procedure continues with a series of screens allowing the user to configure the network hardware, enable optional services such as NFS and FTP and install optional software. Following the default options leads to a system reasonably secure from outside intrusion. The only network services enabled are sendmail, SSH, and the X Window system. The inetd daemon is not started by default and the XFree86 server does not accept TCP connections.

---

[2] Lehey 2002
[3] Apple 2003a
[4] Apple 2003b

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

As a system oriented towards the naïve user, MacOS has few install-time options. The most significant information needed is for network setup and that is directed towards ISP connection data rather than the lower level IP address or DHCP information needed for FreeBSD. At the end of the installation, MacOS connects to Apple to register the installation and check for system updates released after the version of the system shipped.

Both systems set up user and administrator accounts as part of the installation process, but take different approaches. FreeBSD uses the traditional Unix root account for all administration. The FreeBSD installation procedure requires the entry of a root password and offers the option of setting up user accounts. MacOS has the root account disabled by default and only establishes user accounts with optional administrator privileges. Although neither system enforces the use of strong passwords, both follow the SANS recommendation of eliminating default accounts and accounts with no passwords.[5]

Both MacOS X and FreeBSD come supplied with OpenSSH as part of the standard distribution and configure keys as part of the installation process.

Administration

One of the major security enhancements in MacOS is the replacement of the root login account with the concept of administrator privileges. Traditional UNIX systems such as FreeBSD have essentially no granularity in granting access rights. The root account has no limitations and thus has total control of the system. Any system administration tasks that need to be performed as root present the security problem of preventing the user logged in as root from accidentally or deliberately taking actions outside the desired task. FreeBSD provides the standard access controls of optionally restricting root logins to specified (possibly no) terminals and restricting access to the su command to user accounts in a specific group. Setting up the sudo system to allow root access only to specific commands can provide some additional granularity, but since there remain tasks, such as installing new software versions, which require root access the potential of unauthorized use of the root account remains. Similarly, tasks that involve editing configuration files tend to require unmediated root privileges. Commands using the setuid facility (for example, passwd) can constrain operations, but are limited to a predefined set of actions

In MacOS, the root account is largely unnecessary and is in fact disabled by default. Only certain users of the system are designated as administrators able to perform administrative tasks. As an example, enabling FTP is done by clicking on a checkbox in the Sharing System Preferences application rather than by editing inetd.conf. This mechanism is certainly not bulletproof. MacOS retains the

[5] SANS 2001, Section G2

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

concept of total privilege associated with UID 0; as with any Unix system, the UID of a process is the controlling factor, not the fact that the username is “root”. The significant advantage comes in that most standard system operations are mediated by applications that limit the opportunity for mistakes or abuse.[6]

The implementation of the Administrator concept was not bug-free. In adapting the NetInfo subsystem from NextStep to run in the combined Unix and Macintosh environment, Apple did make some mistakes with at least one leading to a trivial root compromise. Raba describes how merely opening a Terminal window while in the NetInfo Manager application would produce a root shell.[7] Apple did promptly include a fix in the next Security Update.

Updates and Upgrades

FreeBSD follows the traditional Open Source model for operating system updates and upgrades. Mailing lists are available that announce both general and security oriented updates. Since FreeBSD is primarily a source-based distribution, these notifications are composed of a description of the problem and a pointer to the new source code tree. Responsibility for subscribing to the lists, assessing the applicability of a fix, obtaining, compiling and installing the latest code is in the hands of the user. There is no automatic mechanism for announcing and installing fixes supplied as part of the operating system distribution. It is possible to schedule automatic downloads of snapshots from the core CVS source tree, but this is an undertaking fraught with peril for a user not intimately familiar with recompilation. As there is no distinction in the CVS tree between security fixes and system enhancements, one could easily end up with a secure, but unbootable system.

Major system releases are available over the Internet as ISO CD images for binary installation. Several companies also provide these CD’s on a subscription basis. However, as with the CVS recompilation method, these upgrades encompass both enhancements and fixes rather than providing a mechanism for closing a specific security hole. FreeBSD maintains several distinct development and maintenance trees so the user can chose whether to upgrade to the most up-to-date version of the OS, install a fully patched version of the current running OS, or install the current release version.

MacOS addresses the problem of patch distribution both through mailing lists and an application named Software Update. On demand, or at regularly scheduled intervals, Software Update will check with a central patch repository at Apple and compare available updates with those currently installed on the

---

[6] As an obvious example, running “sudo sh” from the command line gives an administrative user free rein on the system.

[7] Raba 2001

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

system. If new updates are found, the user is presented with a dialog box announcing the update and giving a brief summary of the problems to be fixed. After prompting for authorization by an administrative account, the program will install and log the patch and reboot the system if necessary.

There are two paths for major system upgrades in MacOS. Some upgrades are treated as “jumbo patches” and are available through the Software Update application as outlined above. Others are available only through the purchase of a CD distribution from Apple. Security fixes have always been available via direct download.

With both systems, there exist some problems. One practical difficulty is in the size of the downloadable patches. A corporate or educational user may have no difficulty in receiving a multi-megabyte update via the Internet but a home user accessing the patch repository via a dialup may have second thoughts at the amount of time necessary to download a security patch. This tends to affect MacOS more than FreeBSD due to the composition of the user communities: MacOS is marketed much more to the home user; FreeBSD users tend to be more technologically advanced and to have correspondingly higher bandwidth connections.

There have been specific security related issues with both systems. The original Apple software update mechanism used an unauthenticated HTTP connection to the central server. This opened the possibility of a man-in-the-middle attack that could lead a user to downloading a trojaned patch. Apple addressed the issue after a demonstration exploit using ARP and DNS spoofing was posted. An update to the Software Update program was released that uses authenticated HTTP. Note that Apple had to address the nontrivial problem of updating an application via a known insecure channel. Including SHA1 checksums, a link to a secure URL, and explicit instructions for checksum verification in the update announcement enabled users concerned with security to verify the authenticity of the patch.[8]

FreeBSD faced a different problem when they were found to be distributing a trojaned version of OpenSSH. In this case the normal installation mechanism detected the Trojan and aborted the installation, but more sophisticated attacker could have replaced the MD5 checksum file at the distribution site and remained undetected. As the Trojan was detected and could only have been installed by specific user action in overriding an alert, FreeBSD did not change its update procedures. It should be pointed out that the actual compromise of the code was not done at the FreeBSD site but only propagated via the repository.[9]

---

[8] Dalrymple 2002
[9] MavEtJu 2002

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

In addition to the core operating system and applications, both systems ship with "standard" Unix software packages such as BIND, sendmail, SSH, and Samba. FreeBSD provides a /usr/ports hierarchy with a wide variety of system add-ons. These pose difficulties for both systems in that a security vulnerability may be found to exist in an application over which the distributor has no control and which is often not even installed. FreeBSD provides update notification for the ports packages, but cannot check whether a package is in use and requires patching. Apple issues patches for the core system and the packages supplied with the core such as Samba even if the package is not enabled at the time of the update. Patches to Apple's proprietary applications (iTunes, QuickTime, etc) are also installed automatically. There is no equivalent to the FreeBSD ports facility. Although a wide variety of shareware and commercial products exist for MacOS, the responsibility for updating them is left to the user. This can lead to a false sense of security in a user who regularly sees Apple supported packages updated automatically and does not realize unpatched vulnerabilities may exist in other programs

Network Services

As pointed out in the original SANS Top Ten/Twenty Vulnerabilities[10], "default installations nearly always include extraneous services and corresponding open ports. Attackers break into systems via these ports. In most cases the fewer ports you have open, the fewer avenues an attacker can use to compromise your network." FreeBSD and MacOS both take this advice seriously. The network services configured during a basic installation of each system are limited to those necessary for the operation as a workstation providing essentially no services to an external user. FreeBSD enables only sendmail, SSH, and syslog. No services are enabled in inetd. Syslog and the X server are configured to not accept connections from the network. MacOS enables nothing other than syslog for external access and NetInfo for local connections.

Enabling additional services in FreeBSD follows the traditional procedure of editing /etc/inetd.conf or adding a daemon to the system startup process. Some services such as NFS and anonymous FTP can be enabled using the /stand/sysinstall script, but the majority depend on the skills of the system administrator to configure correctly and securely. FreeBSD does require by default the use of the TCP Wrappers facility for inetd services, giving the system administrator both a mechanism and reminder to maintain security when adding services.

MacOS provides a system preferences pane for adding Internet services such as file sharing, SSH, and web services. One click in a checkbox enables the service and sets up a standard configuration. A problem that arises with this approach is that a user attempting to make use of facilities outside of the standard

[10] SANS 2001, Section G1

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

configuration may find functionality and security compromised. Brenninkmeyer describes how an Apple upgrade to the Apache web server could silently overwrite the httpd.conf file. This could have the effect of disabling user added access controls with no warning.

User Services Philosophy

One significant contribution to the effective security of MacOS comes not from modifications to the operating system but rather from the different user communities that are address by the systems and, perhaps surprisingly, from Apple's overall commercial strategy.

FreeBSD is aimed at a community seeking a cost-effective standalone system that can provide a variety of services with no external dependencies. Using FreeBSD as a Web or e-Mail server implies accepting the risks that vulnerabilities may occur in the applications implementing those services. It is expected that an experienced and conscientious systems administrator will be configuring and maintaining the system.

Apple's intended audience[11] is a user primarily focused on applications running on the local computer with the option of sharing information with external users. Rather than configuring a mail transport agent such as sendmail or postfix, the MacOS user will normally send and receive mail using POP or IMAP via an ISP. Local transport programs are provided as part of the system, but their use is not necessary. Apple's .Mac product provides drag-and-drop Web services and network backup facilities without the need to configure and run local servers. Eliminating the need to configure and operate local servers for the most common uses of a home computer, specifically e-mail and serving simple web pages greatly increases the security of the system; if there is no web server running, configuration errors or program vulnerabilities are of no consequence.

Suggestions for Improvement

A disappointing omission in both distributions is the absence of an integrated mechanism for processing secure e-mail using the de facto standard PGP program. Encrypted and authenticated e-mail is an important aspect of security and the lack of an easy to use facility is a major barrier to its wider utilization.[12]

---

[11] Apple markets a version of MacOS designed for server use as well as a hardware platform on which to run the system. This paper focuses on the consumer version of MacOS.

[12] Whitten and Tygar cover at length the problems of using an earlier version of PGP with Eudora on the Macintosh and conclude that it is a failure from the standpoint of usability.

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

In the case of FreeBSD, the omission can be justified to a certain extent by the design decision to allow the user to choose the mechanism for mail transport and processing, although one notes that sendmail and Mail/mailx are installed and configured as part of the core installation. The wide variety of mail user agents available in the ports collection (MH, pine, mutt, etc.) would preclude a standard interface to PGP. Ports of the open source distribution of PGP are available, but require the system administrator to install and configure it.

More regrettable is the omission of PGP integration with Apple's Mail application. The PGP product from PGP Corporation provides, as of version 8.0.2, plug-ins for both Apple's Mail application and Microsoft's Entourage. Were Apple to bundle PGP with MacOS X, perhaps enhancing it with Apple's expertise in interface design, and provide a user-friendly mechanism for key exchange via the .Mac product, the number of users actively sending secure e-mail could be significantly enhanced.

As another example, both systems come with firewall capabilities but FreeBSD provides a no mechanism for configuration other than command line and configuration files. This is to be expected with FreeBSD due to its positioning as a system for experienced administrators. With MacOS, firewall configuration is done via a set of checkboxes in the Sharing preferences pane enabling the user to straightforwardly enable basic services such as SSH or Web access without opening up other services. To be fair, one should note that configuring the firewall for rules as simple as "allow file sharing to the local network, but not the Internet" require the user to abandon the preferences pane and revert to manual editing of the ipfw configuration files.[13] Third-party packages exist to provide GUI changes of the firewall, but an extension to the Apple supplied preferences would be preferable from the perspective of a typical user. In addition, any changes made by hand or by a third-party firewall disable the Apple firewall. This can cause considerable confusion to a user attempting to go beyond the limits of the built-in rules.[14]

Conclusion

Unix has for a long time been known for its robustness and security. Unfortunately, it has also been known for its lack of user friendliness. With MacOS X, Apple has taken a version of Unix popular with experienced system administrators and made it usable for a much wider range of users. In the process, they have lessened security in some areas, but have enhanced it overall. By choosing an Open Source distribution as the base for their commercial product, Apple is able to continue to make use of core security enhancements while adding its own contributions to enhance the security available to the average user.

---

[13] Cochella 2002
[14] Ribe 2002

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

References

Apple 2003a. Apple Computer, “Mac OS X v10.2 Technologies: Security”, http://www.apple.com/macosx/technologies/security.html

Apple 2003b. Apple Computer, “The Open Desktop”, http://www.apple.com/macosx/technologies/darwin.html

Brenninkmeyer 2002. Brenninkmeyer, Lawrence. “MacOS 10.2.4 update & httpd.conf replacement”, Risks Digest, http://catless.ncl.ac.uk/Risks/22.56.html#subj6.1

Cochella 2002. Cochella, Chris, “Configuring Jaguar's Firewall “, http://www.macdevcenter.com/pub/a/mac/2002/12/27/macosx_firewall.html

Dalrymple 2002. Dalrymple, Jim, “Mac OS X Software Update security issue uncovered”, http://maccentral.macworld.com/news/2002/07/08/update/

Lehey 2002. Lehey, Greg. “Explaining BSD”, http://www.freebsd.org/doc/en_US.ISO8859-1/books/handbook/history.html

Howard 2001. Howard, James. “The BSD Family Tree”, http://www.daemonnews.org/200104/bsd_family.html

Hubbard 2000. Hubbard, Jordan. “A Brief History of FreeBSD”. http://www.freebsd.org/doc/en_US.ISO8859-1/articles/explaining-bsd/index.html

MavEtJu 2002. Mavetju, Edwin. “Slashdotted”, http://slashdot.org/comments.pl?sid=37188&cid=3991288

Raba 2001. Raba, Nicholas. “Security Concern with Mac OS X: Setuid root applications allow root shell access”, http://www.securemac.com/macosxsetuidroot.php

Ribe 2002. Ribe, Scott, et al., “Using IPFW disables Jaguar Firewall control”, http://www.intellexcorp.com/4DMessages/1800/1859.html

SANS 2001. The SANS Institute. “The Twenty Most Critical Security Vulnerabilities (Version 2.501)”, http://www.sans.org/top20/top20_oct01.php

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

Whitten 1999. Whitten, Alma & Tygar, J. D., “Why Johnny Can’t Encrypt: A Usability Evaluation of PGP 5.0”, http://www.usenix.org/publications/library/proceedings/sec99/whitten.html

© SANS Institute 2003, Author retains full rights

Last Updated: June 20th, 2013

Upcoming SANS Training

Click Here for a full list of all Upcoming SANS Events by Location

Digital Forensics & Incident Response Summit 2013	**Austin, TXUS**	**Jul 09, 2013 - Jul 16, 2013**	**Live Event**
SANS London Summer 2013	**London, GB**	**Jul 09, 2013 - Jul 16, 2013**	**Live Event**
SANS Rocky Mountain 2013	**Denver, COUS**	**Jul 14, 2013 - Jul 20, 2013**	**Live Event**
SEC528 SANS Training Program for the CompTIA® New Advanced Security Practitioner (CASP) Certification	**Washington, DCUS**	**Jul 22, 2013 - Jul 26, 2013**	**Live Event**
SANS Mumbai 2013	**Mumbai, IN**	**Jul 22, 2013 - Jul 27, 2013**	**Live Event**
SANS San Francisco 2013	**San Francisco, CAUS**	**Jul 29, 2013 - Aug 03, 2013**	**Live Event**
SANS Boston 2013	**Boston, MAUS**	**Aug 05, 2013 - Aug 10, 2013**	**Live Event**
SANS SEC 560: Network Penetration Testing @ Bangalore 2013	**Bangalore, IN**	**Aug 05, 2013 - Aug 10, 2013**	**Live Event**
Critical Security Controls Summit	**Washington, DCUS**	**Aug 12, 2013 - Aug 18, 2013**	**Live Event**
Industrial Control Systems Security Training - DC	**Washington, DCUS**	**Aug 12, 2013 - Aug 16, 2013**	**Live Event**
SANS Virginia Beach 2013	**Virginia Beach, VAUS**	**Aug 19, 2013 - Aug 30, 2013**	**Live Event**
SANS Thailand 2013	**Bangkok, TH**	**Aug 19, 2013 - Aug 31, 2013**	**Live Event**
SANS Melbourne 2013	**Melbourne, AU**	**Sep 02, 2013 - Sep 07, 2013**	**Live Event**
SANS Capital City 2013	**Washington, DCUS**	**Sep 03, 2013 - Sep 08, 2013**	**Live Event**
Network Security 2013	**Las Vegas, NVUS**	**Sep 14, 2013 - Sep 23, 2013**	**Live Event**
SANS Canberra 2013	**OnlineAU**	**Jul 01, 2013 - Jul 13, 2013**	**Live Event**
SANS OnDemand	**Books & MP3s OnlyUS**	**Anytime**	**Self Paced**