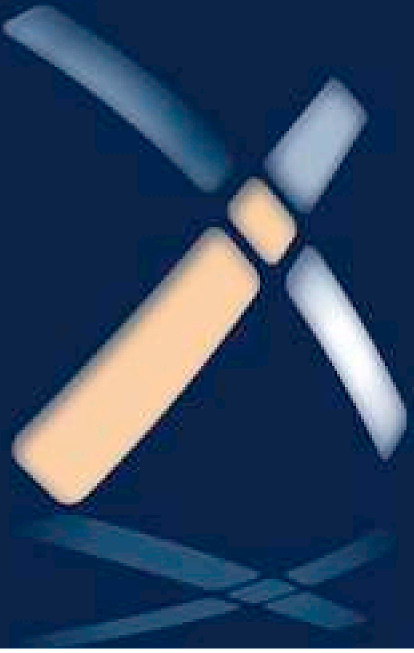
SYNGRESS

The Hacker's Guide to OS X

Exploiting OS X from the Root up



Rob Bathurst
Russ Rogers
Alijohn Ghassemlouei

The Hacker's Guide to OS X

This page is intentionally left blank

The Hacker's Guide to OS X

Exploiting OS X from the Root Up

Rob Bathurst

Russ Rogers

Alijohn Ghassemlouei

Pat Engebretson,Technical Editor

ELSEVIER

AMSTERDAM • BOSTON • HEIDELBERG • LONDON
NEW YORK • OXFORD • PARIS • SAN DIEGO
SAN FRANCISCO • SINGAPORE • SYDNEY • TOKYO

Syngress is an Imprint of Elsevier

SYNGRESS.

Acquiring Editor: *Chris Katsaropoulos*
Editorial Project Manager: *Meagan White*
Project Manager: *Priya Kumaraguruparan*
Designer: *Russell Purdy*

Syngress is an imprint of Elsevier
225 Wyman Street, Waltham, MA 02451, USA

Notices

Knowledge and best practice in this field are constantly changing. As new research and experience broaden our understanding, changes in research methods or professional practices, may become necessary. Practitioners and researchers must always rely on their own experience and knowledge in evaluating and using any information or methods described herein. In using such information or methods they should be mindful of their own safety and the safety of others, including parties for whom they have a professional responsibility.

To the fullest extent of the law, neither the Publisher nor the authors, contributors, or editors, assume any liability for any injury and/or damage to persons or property as a matter of products liability, negligence or otherwise, or from any use or operation of any methods, products, instructions, or ideas contained in the material herein.

Library of Congress Cataloging-in-Publication Data
Application submitted

British Library Cataloguing-in-Publication Data
A catalogue record for this book is available from the British Library.

ISBN: 978-1-59749-950-7

Printed in the United States of America
13 14 15   10 9 8 7 6 5 4 3 2 1

For information on all Syngress publications visit our website at *www.syngress.com*

Contents

This page is intentionally left blank

Foreword

As I write this, I'm contemplating the evolution of Apple/Mac, and the progress made around protecting Apple products. Mac systems have gained in market share over the last few years, and you can't argue with the amount of flexibility and pure performance power you can get out of the Mac. The late Steve Jobs and his team have developed outstanding products that I too have grown to depend on for my business and personal use. For the longest time I was anti-Mac. I couldn't justify in my mind the additional cost, the "attitude" of the Mac crowd, or that there was any chance it was that much better of a product. I was wrong. There is a reason that Apple products have been so popular. They are built to perform, and built to be easy to use.

But along with that popularity comes risk. Nothing can be 100% secure, and as Apple's market share has grown, so has the threat to their products. Unfortunately complacency has grown to a point where most users expect the Mac to be perfectly secured. And, therefore, the growth of the security of Mac OS has been lagging.

It is my belief that this situation must rapidly change, and does appear to be changing. The largest known Mac attack to date (2012) is the Flashback malware, which is estimated to have infected over 600,000 machines. In response to Flashback, Apple took an entirely new approach, and for the first time they were open about how they were addressing this malware issue. While their response method is a topic of debate in security circles, it was still an evolution from how they addressed any previous security issues that have been identified.

We also cannot forget about the technology breakthroughs that we lovingly call the iPhone, iPad, and any other products that run the Apple iOS. These devices have become the mobile computing platforms that we take with us everywhere. Apple's security, related to iOS, is considered fairly solid, but that doesn't mean it will remain that way. Any devices using the iOS are targets for exploitation, and as security holes are discovered, there will be further attempts to take advantage of those vulnerabilities.

This book gives you a strong foundation for securing your MAC OSX and iOS, and it gives you techniques for bettering your platforms for future protections. Take advantage of this information and protect yourself.

The bigger the target, the bigger the threat, the more likelihood of occurrence AND the necessity of demonstrating responsibility to the consumers of their products.

Greg Miles, Ph.D., CISA, CISSP, CISM

Peak Security, Inc.

CHAPTER 1

Introduction

INFORMATION IN THIS CHAPTER:

- Why You are Reading this Book
- What is Contained Within this Guide
- The Path Ahead

CONTENTS

WHY YOU ARE READING THIS BOOK?

The question in the large heading print may strike you ask, “Yes, why am I reading The Hackers Guide to Mac OSX?” Perhaps it was the word hackers, perhaps you picked it up at random, or perhaps you own an Apple product and suddenly grew concerned that there might be bad people doing bad things to your precious iSomething. Well rest assured, reader, that this book does not contain new vulnerabilities, exploits, or chapters of shellcode. The Hackers Guide to Mac OSX is here as a learning tool for students, professionals, and the curious reader to better understand the realm in which they are venturing forth. But, why should anyone care about testing the security of a company who commands such a small share of the over all pc market with less than 15%[1] [1]?

I’m glad you asked. Apple products, in terms of pc market share are indeed small, but their mobile platforms such as the iPad and iPhone account for commanding portions of the mobile device market and their pc share continues to grow annually. What this means for us as security professionals is that we will continue to see iOS and OSX use continue to grow in both consumer

[1] http://macdailynews.com/2011/10/12/gartner-apple-mac-grabbed-12-9-share-of-u-s-pc-market-in-q311/.

The Hacker’s Guide to OS X. http://dx.doi.org/10.1016/B978-1-59-749950-7.00001-0

and business sectors, and we must be able to properly assess the potential vulnerabilities of those systems.

What is Contained Within the Guide?

The Guide contains tools, tips, and techniques from our experiences as professional penetration testers and Apple enthusiasts to help you, the reader, gain a better understanding of the mindset needed to analyze Apple products from a security perspective. While there are many books available on how to think like a penetration tester, hacker, attacker, or generally aggressive person, there are a lack of books bridging the gap between the high level (This is how to turn on an Apple product) and the low level (Look at my awesome 1s and 0s). The following chapter listings have a brief synapses for each chapter, enjoy.

Chapter 1: The Introduction

You are reading it.

Chapter 2: OS History

The OS History chapter, much like its title suggests, is focused on the history and progression of the various Apple operating systems from OS8 through OSX 10.7. We will walk you through the significance of the legacy operating systems and how they relate to the design choices of the underlying systems such as Coca and EFI under the current OSX. In addition, the reader should understand that Apple is releasing its Mountain Lion version of OSX in August of 2012, which includes other changes that could impact the way you use your computer, including changes to the way the OS integrates X11 functionality.

Chapter 3: The Filesystem

In the Filesystem chapter we will cover HFS/HFS+ and how the filesystem provides an abstraction layer to the user. We will also cover the organization of the filesystem, including inodes and file caching.

Chapter 4: Footprinting OSX

Footprinting is a crucial piece of the recon process during a penetration test and offers valuable information such as open ports and OS versions. This information will allow you to narrow your possible payload choices and know when it is possible to use a remote exploit. We will show you how fingerprint an OSX machine, and what it looks like to industry tools such as NMap, Nessus, Metasploit. We will also provide advice on what useful and valuable information to look for in the output.

Chapter 5: Application and System Vulnerabilities

Now we get to the part everyone loves, how and what to break. In this chapter we will take you through targeting applications, how the applications interact with the operating system, exploiting vulnerable applications, code compiling, and much more.

Chapter 6: Defensive Applications

Every good offense needs a good defense, or something like that. While we focus on the weaknesses of the operating system and applications we must also know what we are facing as the system and users attempt to defend themselves. We will cover Firevault implementation, the built-in OSX firewall, anti-virus suites, kernel security, and pesky things users do to keep themselves safe.

Chapter 7: Offensive Tactics

We showed you how to break what was not meant to be broken, now we sharpen that mentality by showing you how to maintain presence and navigate through the system. This chapter covers modifying the kernel and drivers, command line tools (they help system admins, they help us too), pivoting from Metasploit through an OSX system, and attacker centric scenarios.

Chapter 8: Reverse Engineering

Reverse Engineering is a complex skill and we will not claim that this chapter will teach you how to be the world's greatest vulnerability finder, but what we will do is teach you to think about what happens to an application when it has a glitch. We will show you assemblers, compilers, reflectors, and basic fuzzing. If this sounds like we just made up a bunch of words, fear not, they all mean something.

Chapter 9: Mobile Platforms

Everyone loves the iSomething; hordes of people across the globe carry an iOS-based device with them every day and contained on these devices are untold amounts of personal information. We will look at the architecture of iOS, security implications, iOS signing, footprinting, and jailbreaking.

Chapter 10: Tips, Tricks, and the Future

This is the gift for all those who waited, or those of you who just skipped to the end of the book. Contained within these pages is all the information we could not fit into the other chapters, fun projects such as the Hackintosh, and handy reference lists for ports and processes.

THE PATH AHEAD

Now that you are as excited to read this book as we were to write it; we will offer some tips to help you as you move though the text. As you read through this book you will notice helpful tips in the sidebar and notes or references contained in the footer. Taking the time to review and read over these bits of extra information will help you to further understand the concepts we are discussing. We will often reference a website, whitepaper, or book that contains more information on the current topic than we can fit into the pages of the book and recommend browsing those resources should you wish to expand your knowledge.

REFERENCE

<http://macdailynews.com/2011/10/12/gartner-apple-mac-grabbed-12-9-share-of-u-s-pc-market-in-q311/>.

CHAPTER 2

History and Introduction to OSX

CONTENTS

HISTORY AND INTRODUCTION TO OSX

As a technical reader, I've always managed to devour technical books; often collecting them like some people collect bottle caps. In most of those books there is always a chapter on history, often full of dry, boring material that has limited relevance to the remainder of the book. Because of this, I've gotten into a habit of skipping these chapters on a routine basis.

However, with this publication, the history of how Apple came to the point of creating the OS X operating system has tremendous value to the remaining chapters. In the interest of fairness, and to alleviate the painful yawning, I've slimmed the content in this chapter down to just those concepts that will be the most useful to you as the reader. While it may be a shorter chapter, it will most certainly carry its value with rich, juicy tidbits of information, instead of the usual bland and boring history lessons we've all studied in the ancient textbooks.

OSX Origins

A Byte of History

Since this book is focused on OSX, the following sections will be rather targeted; not repeating the same Apple story we've all heard a dozen times. The goal here is not to create Apple zealots or fan boys, it's to provide relevant information so that you, as the reader, can form well-rounded opinions and decisions regarding the technical work that will be done.

Apple Computers was originally founded by Steve Jobs and Steve Wozniak on April 1st 1976, when they released the Apple I computer. By 1985, Steve Jobs had been ousted from Apple after a conflict with then CEO, John Sculley. When he left, Jobs founded a new company named NeXT, Inc., which was later split into two and renamed NeXT Computer, Inc. and NeXT Software, Inc. The new companies built computers, and an operating system, called NeXTStep, which was later used to invent the World Wide Web (WWW), by Tim Berners-Lee.

The Hacker's Guide to OS X. http://dx.doi.org/10.1016/B978-1-59-749950-7.00002-2

NeXTStep was built on top of a relatively unknown micro kernel architecture from Carnegie Mellon University, along with source code from the Berkeley Software Distribution (BSD). The end result was not an actual microkernel, but ended up much closer to the more familiar monolithic kernel most modern operating systems use. So looking back in hindsight, it's not really a huge surprise to find out that when Apple acquired NeXT in 1997 and brought Steve Jobs back as CEO of the company, that Apple began using the NeXTStep operating system as the foundation for what would eventually become the Mac OS X operating system we use today.

There are actually multiple components to the NeXTSTEP kernel itself. The kernel was comprised of version 2.5 of the Mach kernel and components of 4.3BSD, on top of which there was an object oriented API for writing drivers called Driver Kit. When Apple purchased NeXT the OS was revamped, the Mach component was upgraded to version 3.0 and code was used from the FreeBSD project to update the BSD sub-system. Driver kit was also replaced with what is now known as I/O Kit which is a C++ API for writing drivers. This kernel as it currently stands today is known as XNU. XNU is an acronym which stands for *X is not Unix*.

While Mach is a microkernel and technically allows running the various kernel responsibilities in separate programs in user space this generally leads to tremendous slowdowns and can be detrimental to having a fast speedy OS. The entire BSD subsystem was bolted on top of Mach to make what many would consider a hybrid kernel, although a lot of people dismiss that as merely marketing speak.

Mach provides many of the basic building blocks for an operating system; message passing, threading, virtual memory, kernel debugging support, and a console. The BSD subsystem provides the rest, a Unix process model (on top of Mach tasks), security policies, user id's, group id's, permissions, virtual file system support (allowing multiple file systems to be supported easily), a cryptographic framework (used extensively for Keychain, encrypted disks, and others), MAC (mandatory access control), and a whole range of other functionality, most importantly an POSIX compatible API.

The code for the BSD subsystem comes from the FreeBSD project, and Apple has, in the past, attempted to synchronize the API's that it exports to those that are available in FreeBSD, the last of such synchronizations was made with FreeBSD 5.[1] There is quite a bit of code that Apple shares with FreeBSD and vice-versa and various efforts to bring Apple code back to FreeBSD have sprung

[1] http://developer.apple.com/library/mac/#documentation/MacOSX/Conceptual/OSX_Technology_Overview/SystemTechnology/SystemTechnology.html

up in an attempt to take the best of what XNU has to offer back to the OS it came from.

The XNU kernel, along with various other tools and utilities, is what constitutes an operating system named Darwin.[2] Darwin can be considered a stand-alone OS, and there are various efforts to create a fully functioning OS, however it is better to consider it simply the underlying basis for Mac OS X. Some of the technology will sound familiar to those familiar with other open source projects, Darwin has used source from various projects such as GCC, GDB, Apache, Python and many others and, when needed, modified them to fit within their operating system (including many modifications to GCC and GDB to support Mach-O, the binary format much like ELF and PE on Linux/BSD and Windows respectively).

Darwin is open-source and available from Apple, however it is missing many critical components that would make it Mac OS X, notably Quartz, Apple's windowing system, Coca, Carbon, and many of the libraries such as CoreAudio, CoreImage, CoreAnimation, and many other important libraries that have yet to be mentioned. Now that we've got a grasp on what's going down when it comes to a general history with some kernel tidbits, lets jump deeper into the architecture of the processor itself.

PowerPC Architecture and Transition to Intel

Prior to Apple's adoption of Intel, PowerPC (PPC) reigned supreme. PowerPC stands for Performance Optimization with Enhanced RISC and was developed by an alliance between Apple, IBM and Motorola back in 1991. Apple began the integration of PowerPC processors into their Macintosh line in 1994 all the way through to 2006. But Apple felt that IBM's platform was not meeting the requirements Apple wanted to see in their future devices. In short, PPC was moving too slow. IBM wasn't able to deliver promises for faster chipsets, more efficient power consumption, and Apple pulled the trigger.

The transition to the Intel chipset on a software level took considerably less time than one would imagine. Specifically, with the release of OSX 10.5 (Leopard), Apple began support for both chipsets, shortly followed by 10.6 (Snow Leopard), which introduced the 64 bit architecture and began dropping support for PPC. Snow leopard allowed the end user to install an application called Rosetta to run certain outdated PPC applications, however, with did not allow installation onto PPC systems. And finally, with the release of 10.7 (Lion), Apple axed the PPC applications altogether, as they expected developers to have already upgraded their applications, given the previous four years of migration time.

[2] http://OSXbook.com/book/bonus/ancient/whatismacOSX/arch.html

So what are the benefits other than speed, cost, power consumption, and widespread adoption of the Intel chipset? Plenty of things come to mind given that question; emulation, virtualization, vast operating system support, directx support, easier code transitioning with OpenGL to name a few. With this chipset OSX was able to run many different operating systems far easier than before, without having to install PPC specific versions. The adoption of Apple hardware was definitely improved due to this change, Windows users were more comfortable knowing they could always fall back, newer Linux/Unix users had an easier time installing their favorite distros. At this point we can begin to pull back to review some information on EFI and Open Firmware and how it all relates back to OSX.

Firmware—EFI

Prior to what Apple currently ships, (Extensible Firmware Interface (EFI)) Open Firmware was the standard. Open Firmware allows the system to load platform-independent drivers directly from the PCI card, improving compatibility and whatnot. Below I've gone ahead and listed out a few of the original security concerns regarding Open Firmware:

- On a PowerPC-based Macintosh, the Open Firmware interface can be accessed by pressing the keys Cmd+Option+O+F at startup.
- Vulnerability allowed passwords to be disclosed to users via tool called FWsucker.[3]
- Passwords can be removed completely by removing DIMMs and reseting PRAM 3 times.
- Single user mode can be entered via holding down the s key.

So back to the relevant subject at hand. EFI has been around for quite some time (early 1990s). However, Apple announced its EFI adoption in mid 2005 and shipped devices with EFI support out in 2006. So what exactly is EFI? It's nothing more than the common BIOS interface you may already be familiar with, but with shell capabilities. And while it may not have such an easy interface as the old school IBM type BIOS', it does have its advantages. CPU independent architecture and drivers, flexible preboot OS with network support, modular, and 3TB HDD booting support are just a few cool things under the hood.

After all this is said and done, who really cares? Well, as the end user, this is a critical element in regards to futzing (read hacking) with the machine. There is a small partition to store files, one can set boot priority, change permissions in

[3] http://www.securemac.com/openfirmwarepasswordprotection.php

single user mode, load kernel extensions and have tons of fun just in this one area. At any rate, the next item is the file system itself!

File System—HFS+

OSX utilizes HFS+ for its file system, HFS Plus is also referred to as "Mac OS Extended" within the OS itself when partitioning the drives. It is the successor to Apple's older HFS file system, Hierarchal File System. The primary differences between the two are 16 bit vs 32 bit block addresses and Mac OS Roman vs Unicode support. The newer of the two file systems resolved one of the larger problems with the older file system; mainly that the allocation mapping was 32 bit thus allowing for more efficient use of space within the hard drive itself. OSX has full support for HFS while Linux and FreeBSD carry partial support with certain packages for read-only access. Other distributions require third party applications.

Common Misconceptions

Mac vs PC; which platform do you prefer? In an attempt to stand out from the market Apple has branded its systems as "Mac" as compared to the remainder of the market which we call PC's. Let's go out on a limb here and say that Macs are truly no different than PC's because they are PCs! The definition of a PC is a personal computer, right? Apple has done a wonderful job marketing its brand, and thus adequately confused and segregated many end users.

OSX is nothing more than another platform. Sure it has support for a limited set of hardware (Apple Hardware) so it will perform smoothly, it has a different user interface and much like any operating system, some unique features to set it apart from the rest of the market. Since Apple's gone ahead with this brilliant marketing strategy, how exactly does the larger public view OSX or Apple hardware in general?

Better for Designers/Creative People

While Apple has developed software targeting this audience (Aperture, Final Cut Pro) the industry also has plenty of other choices when it comes to video editing (Avid) and since Adobe has made their suite cross platform there is no legitimate reason for saying that OSX is better for designers and creative

TIP

Bootcamp Windows drivers allow HFS+ partitions to be read whether one is attacking or defending it is important to understand the underlying file system otherwise time will be spent attempting to resolve an issue that could've been avoided altogether.

people alike. But, you could say that since Apple enjoys making things minimalistic and easy to use it would be more attractive to those who are not engineers.

Secure

During the PowerPC era portions of the public/industry began touting Apple's operating systems as secure. Early on Apple's operating system didn't play nicely with the processors that Microsoft Windows supported. When any operating system has a large portion of the market, most malware writers will focus their efforts on that OS. With a different architecture, malware authors now need to go the extra step and either modify or rewrite their code to execute on Apple's operating systems. Up until recently that wasn't worth their time. The adoption of the Intel architecture has made it easier to port code over, so recently we have seen backdoor type trojans ported from Windows and other platforms over to OSX.

With the luxury of the smaller market share for quite some time now OSX has skirted by for the most part unscathed, it has had its share of vulnerabilities much like any piece of software. Apple's regular PR response approach doesn't really fit well with current security practices; this approach being a very slow, well thought out response. With the release of 10.6 (Snow Leopard) we saw the introduction of a bare-metal anti-malware system loosely integrated within the OS. This feature was slipped in and tended to be irregularly and silently updated while only identifying known malware via known signatures.

Apple did not directly acknowledge any security concerns, but even within the documentation provided on their website Apple promoted the use of third party antivirus software; even though their earlier advertising never addressed the issue. Much like any subject, there are individuals who aren't as informed as they should be on subjects they speak to. For example, "OSX is secure because it's Unix." Other than the blatant disregard for logic there, that statement has some validity but not much. Sure it's got the advantage of the underlying architecture being Unix-like but it is far from true Unix.

Unix, or any other operating system, has vulnerabilities, and OSX is no exception to that rule. Currently, Offensive Security's exploit database has approximately 120 usable exploits between 2003 and 2011. Again, this is nothing compared to the number Windows exploits which weigh in at a whopping 3,480 exploits within the same window of time. But bear in mind the market share and attack surface Windows occupies, compared to Linux or OSX. Now, Linux isn't too far off here with a total of 640 pieces of shellcode available to the public. With this slightly more informed perspective, let's take a look at how the larger population views Apple and its products.

Perceptions

With the release of iCloud, Apple has leveled the playing field when it comes to devices. They have claimed that OSX is just another device, it is just as important as your mobile devices (iPhone, iPod, iPad). Why does this matter? Even with MobileMe, iCloud's predecessor, major device syncing was done through iTunes on your computer. That data was then stored there and remained there until you deleted it. The focus has shifted to dare I say… THE CLOUD and with that, so has the data. Let's be honest, the data is what we care about, users need it and attackers want it.

Apple has shifted its focus to the consumer and prosumer markets; the enterprise is an afterthought as it currently stands. Dropping the XServe line back in 2008 and standardizing their notebook line, blurring the distinction between consumer and professional grade MacBook laptops. One could say that Apple hardware is nothing more than designer technology. Apple's mobile platform has had more of the enterprise treatment than OSX has had. This is made clear via Microsoft Active Sync support, configuration profiles, separate app store for in-house corporate applications, and a flurry of other fun things. BlackBerry has held the business market for quite some time, and iOS and Android are now giving RIM a dangerous run for its money. Readjusting the focus back onto OSX we'll cover some of the capabilities that the OS itself has.

Capabilities

OSX is just another operating system, Apple merely has a tighter grip on its hardware, making for a smoother end user experience. There's no real need for every driver under the sun, as it already has more efficient battery control, standardized trackpads for gestures, and plenty more.

On a software level, the following items set OSX apart right off the bat, as they are not available on any other platform.

- XSAN for distributed storage over fiber channel.
- Aperture for professional photography management similar to Adobe Lightroom.
- Final Cut and Logic Studio focusing on the audio and video industries.
- iLife, iWork, both directly relating to their target audience. iLife is a bundle of applications (iPhoto, iWeb, Garage Band, iMovie) for the average

TIP

The Apple tax is a term that is associated with the markup of products where they are overpriced when compared to their spec'd out counterparts.

user to be able to handle the creative side of the house. You also have iWork (Pages, Keynote, Numbers) for the office related stuff.

The items listed are Apple developed products. You still have Microsoft office and a variety of other choices for applications. Much like any other operating system there will be hundreds of thousands of applications or little things that one operating system has over the other. For instance, OSX Server in its latest form has been dwindled down to an additional package that can be downloaded to convert the standalone 10.7 install into a server. It's certainly not a novel concept but again we can see Apple targeting the prosumers more with the no brainer configuration and setup. Address book, file sharing, calendars, chat, mail, podcasts, time machine backups, VPN, web server, and wiki functionality are all baked into the server instance itself.

The one thing that really stands out for security folks is the BSD-Unix-like backend, where we can compile, install, and run all those applications we all know and love from the Linux and Unix worlds. Macports and Homebrew are applications that allow for easier package management, instead of having to manually install everything yourself. Not only can you run all those awesome commands, the services are usually there as well. The config files will be in slightly different locations, and certain daemons won't be running by default, but with a little digging online it's easy to pick up and tweak to your hearts desire.

In the following section you'll note how OSX is being utilized in different areas.

Environments Leveraging OSX

Home

Within the home environment the operating system has certainly come a long way since the early days of OS8 and OS9. I like to think this can be primarily attributed to the adoption of the Intel chipset around the time 10.5 was released. Combine that architecture with Apple's simplification of its user interface and minimalistic attitude and the end result is something everyone can use. We see the elderly migrating to OSX for the user support that Apple's one-to-one service provides, giving them training on how to use their devices and the applications within them. Technology is not something everyone is

TIP

DISA even releases security configuration guidelines to secure the platform that can be found at http://iase.disa.mil/stigs/os/mac/mac.html.

comfortable with and having physical stores with informed individuals there to help makes it an attractive environment.

There is also the more recent iOS environment helping bring new users to the Apple ecosystem. Specifically, the syncing service formerly known as MobileMe, now known as iCloud, which ties their desktop/laptop world to their mobile world via contacts, reminders, calendars, pictures, documents and much more. The mobile market has got to be one of the largest reasons for the recent wide spread adoption of OSX. The Apple ecosystem, much like the Microsoft ecosystem, is very smooth, depending on how deeply you buy into it.

One of the major benefits that has wrangled in many new users is the ease in which they can boot back into familiar territory, usually Windows, with the assistance of an application called Bootcamp. Bootcamp simplifies partitioning the hard drive and prepping the EFI for installation of another operating system other than OSX. For example, hold down the option key during the bootup process and choose which operating system you'd like to boot into! Dead simple. This single application has semi-resolved the gaming issue that Apple has been struggling with as developers are not willing to develop for OSX as readily as they would be for Windows. Although Steam, an online game platform with a large collection of games for purchase, has assisted in this effort by providing a client for OSX users which opens up a small portion of their gaming library.

Business

The larger business community has been slow to respond to OSX, mainly due to its lack of enterprise solutions for management. However, Apple has steadily been adding in key features here and there even though this is currently not Apple's primary focus. There has been an increase with the newer generations picking up OSX as opposed to the older folks sticking with Windows and more widely used systems. Small startups are also beginning to roll out the operating system, due to its ease of use and configuration.

We've seen pockets within businesses adopt OSX, but those pockets are usually concerned with identity (marketing, executives) and other creative areas (design and art). The larger IT community doesn't necessarily know how to handle OSX, as it's not as widely used as windows. So it remains an unknown and is often misconfigured.

But although there has been a slow transition for OSX into the corporate environment, on the mobile front we see rapid deployment of iOS devices due to its current popularity. iOS again charges ahead, blazing a path for OSX to follow.