

Perception, Security and Worms in the Apple

David Harley (ESET), Andrew Lee (K7 Computing) & Pierre-Marc Bureau (ESET)

About the Authors

David Harley is Research Fellow and Director of Malware Intelligence at ESET, a member of the Board of Directors of AMTSO (Anti-Malware Testing Standards Organization), Chief Operations Officer for AVIEN (Anti-Virus Information Exchange Network) and an independent security author, blogger and consultant. In his copious free time he maintains the Mac Virus and Small Blue-Green World web sites, including blogs on Mac security, hoaxes and other security and non-security issues. He also blogs for Securiteam, (ISC)², AMTSO and AVIEN. He has authored or co-authored over a dozen books on security, including "Viruses Revealed" and the "AVIEN Malware Defense Guide for the Enterprise" as well as many articles and conference papers.

Contact Details: c/o ESET, 610 West Ash Street, Suite 1900, San Diego, CA 92101, USA, phone +1-619-876-5458, e-mail dharley@eset.com

Andrew Lee is Chief Technology Officer at K7 Computing; the current CEO of AVIEN (Anti-Virus Information Exchange Network); a director of AVAR and a member of the Review Analysis Board of AMTSO. He has spent many years writing about security, including for many popular computer magazines and is a frequent speaker at security conferences. He was a co-author of the Syngress book "AVIEN Malware Defense Guide for the Enterprise" and frequently blogs on AVIEN.net. He is currently completing his Masters thesis on issues relating to Anti-malware product testing.

Contact Details: c/o K7 Computing Private Ltd, 6th Floor, Rayala Techno Park, 144/7 Old Mahabalipuram Road, Kottivakkam, Chennai, 600041, India. Email: alee@k7computing.com

Pierre-Marc Bureau is a Senior Researcher at ESET. Contact Details: c/o ESET, spol. s r.o., Aupark Tower, 16th Floor, Einsteinova 24, 851 01 Bratislava, Slovak Republic

Keywords

Apple, OS X, iPhone, vulnerability, malware, rootkit, DNSChanger, rogue anti-virus, adware, jailbreaking, white-listing, user education, vendor responsibility, user perception

Perception, Security and Worms in the Apple

Abstract

Apple's customer-base seems to be rejoining the rest of the user community on the firing line. In recent years, criminals have shown increasing interest in the potential of Mac users as a source of illicit income, using a wide range of malware types, while issues with jailbroken iPhones have highlighted weaknesses in Apple's reliance on a white-listing security model.

A recent survey carried out on behalf of the "Securing our eCity" community initiative, however, suggested that Mac (and, come to that, PC users) continue to see the Mac - or at any rate OS X - as a safe haven, while Apple seems wedded to the idea that it has no security problem.

However, analysis of hundreds of samples received by our virus labs tells a different story. While the general decline of old-school viral malware is reflected in the Macintosh statistics, we are seeing no shortage of other malicious code including rootkits such as WeaponX, fake codec Trojans, malicious code with Mac-specific DNS changing functionality, Trojan downloading and installation capability, server-side polymorphism, fake/rogue anti-malware, keyloggers, and adware (which is often regarded as a minor nuisance, but can sometimes have serious impact on affected systems).

Nor is this just a matter of Mach-O (Mach Object File) format binaries: scripts (bash, perl, AppleScript), disk image files, java bytecode and so on are also causes for concern. While neither the possibility nor the actual existence of a threat always equates to the probability of its having measurable impact, we take the position that the tiny proportion of compromised machines reflects, at least in part, the still limited market penetration of Apple products. The surprisingly swift escalation of exploits of a single iPhone vulnerability from PoC code to multi-platform hacker tool to functional botnet has perhaps been given more exposure than its impact in terms of affected machines might deserve, yet it demonstrates how closely criminal elements are watching for any weakness that might be turned to advantage.

A security model based on white-listing and restricted privilege, implemented on the presumption of the user's conformance with licence agreements, can fail dramatically where there is an incentive to circumvent security for convenience or entertainment. Some types of attack (phishing is an obvious example) are completely platform agnostic because the "infected object" is the user rather than something on the system. Security reliant on the inability of a user to gain privileged access may lead to disaster if it fails to anticipate the ingenuity of hobby hackers and criminals alike, or the possibility of a conjunction of social engineering and technical vulnerability.

This paper will compare the view from Apple and the community as a whole with the view from the anti-virus labs of the actual threat landscape, examining:

- *The ways in which the Apple-using community is receiving increasing attention as a potential source of illegitimate profit,*
- *Reviewing the directions likely to be taken by malware over the next year or two*
- *Assessing the likely impact of attacks against Apple users.*
- *The implications for business and for the security industry in an age of interconnectivity, interoperability, and the paradox of accelerated computing power on ever-shrinking devices.*

Introduction

Since the appearance in 2006 of OSX/Leap.A, often considered to be the first virus for OS X (disregarding definitional quibbles for the moment), criminals have shown increasing interest in the potential of Mac users as a source of illicit income, using a wide range of malware types, while issues with jailbroken iPhones have highlighted weaknesses in Apple's reliance on a white-listing security model. (Note that this is not entirely an Apple issue: "rooting" of other models of smart phone such as the Motorola Droid (Harley, 2009a) is also a concern.) But has it really affected public perception? Recent research suggests that a broad section of the Mac community still believes in Apple's claims that "Every Mac is secure right out of the box." (Harley, 2008)

Yet several anti-malware vendors have recently launched or are in the process of launching Mac-specific scanners. We're also seeing other forms of blackhat interest such as a rogue antispyware products that only detect imaginary malware, malware taking the form of various flavours of malicious/semi-malicious software ported across platforms (including Linux, FreeBSD, and OS X), and so on.

A Matter of Opinion

A recent survey carried out by CERC (CERC, 2009) on behalf of the "Securing our eCity" community initiative, suggested that Mac (and, come to that, PC users) in the US continue to see the Mac - or at any rate OS X - as a safe haven.

Computer(s) owned, if any	Percentage of survey population
PC	53.9
Mac	5.6
Some other type of computer	8.2
Do not own a computer	23.2
Own Mac(s) <i>and</i> PC(s)	6.9
Unsure	2.1

Table 1: Computer Ownership

Type	Not Vulnerable	Somewhat Vulnerable	Very Vulnerable	Extremely Vulnerable	Unsure
PC	2.1%	29.4%	33.4%	18.4%	16.8%
Macintosh	9.2%	41.8%	11.7%	7.7%	29.7%

Table 2: Perceived Vulnerability of PCs and Macs (Whole Survey Group)

Type	Not Vulnerable	Somewhat Vulnerable	Very Vulnerable	Extremely Vulnerable
PC	0%	15%	48%	37%
Macintosh	16%	68%	2%	13%

Table 3: Perceived Vulnerability of PCs and Macs (Mac Users Only)

Type	Not Vulnerable	Somewhat Vulnerable	Very Vulnerable	Extremely Vulnerable
PC	1%	37%	43%	18%
Macintosh	12%	60%	19%	9%

Table 4: Perceived Vulnerability of PCs and Macs (PC Users Only)

Type	Not Vulnerable	Somewhat Vulnerable	Very Vulnerable	Extremely Vulnerable
PC	3%	19%	41%	36%
Macintosh	28%	62%	5%	5%

Table 5: Perceived Vulnerability of PCs and Macs (Owners of Both Types)

We would guess that these figures would have shown a higher percentage for Macintosh in the “Not Vulnerable” column even a year or two ago, and we regard the relatively high proportion of Mac users acknowledging that God’s own operating system is even “somewhat vulnerable” as encouraging. Nevertheless, the estimation of the Mac’s defensive capabilities from all three groups seems very high when we look at the volume of malware that targets OS X (either exclusively or in addition to Windows-targeting versions).

Keeping an Eye on the Orchard

On the other hand, there are indications that information relating to Apple security is watched pretty closely. For example, Graham Cluley reports that of the ten most popular posts on his own blog at <http://www.sophos.com/blogs/gc/> included the following, all of which include some implication of Mac security (Cluley, 2009).

9 th	Apple ships a known vulnerable version of Flash with Snow Leopard
8 th	Mac malware adopts porn video disguise
5 th	Apple Mac malware: caught on camera
4 th	Leighton Meester sex video lure spreads Mac and Windows malware to Twitter users
2 nd	First iPhone worm discovered - Ikee changes wallpaper to Rick Astley photo
1 st	Erin Andrews peephole video spreads malware

Table 6: Clu-Blog 2009 Top Ten Entries Including Apple-Related Content

While we know that some fairly unsavoury people read security blogs in a general information-gathering sort of way, it's unlikely that the popularity of these particular posts is entirely due to the curiosity of criminals and PC users hoping to gloat. It's probable that more Mac users are starting to move away from a "Not listening! Not listening! La-la-la-la-la..." stance, and starting to take a healthier interest in their own security. In fact, this behaviour may indicate that many Mac users realise that there are indeed vulnerabilities that exist, but because of the paucity of cover by security products such as anti-virus, they attempt to ensure that they patch their systems more diligently.

Discussion

Small wonder, however, if Mac users are ambivalent, when Apple seems publicly wedded to the idea that it has no security problem (F-Secure, 2008), while less publicly taking baby steps towards some measure of acceptance of responsibility for the protection of its users.

In late-2008 the company hastily retracted its suggestion in a technical note – formerly available at <http://support.apple.com/kb/HT2550>, but later removed, apparently in response to a surge of media attention (CNET, 2008) – which not only indicated that Mac AV is a Good Thing (Sellar & Yeatman, 1930), but actually appeared to endorse products by Intego, Symantec and McAfee. Presumably its withdrawal was accelerated by the fact that it seemed to contradict Apple's own statements that "Every Mac is secure right out of the box" (Harley, 2008) and "Mac OS X doesn't get PC viruses. And its built-in defenses help keep you safe from other malware without the hassle of constant alerts and sweeps." (Apple, 2010). Apple's rather disingenuous claims that PC viruses are not a problem appear to be based on the rather obvious fact that the binaries are different for each platform, but fail to account for attacks that do have potential to work equally on Mac and PC – for instance, the rogue javascripts that set-up scams such as fake AV downloads.

However, in 2009 the company slipstreamed a rudimentary anti-Trojan capability into its 2009 "Snow Leopard" product release. Specifically, a file called XProtect.plist (Ziff-Davis, 2009) and containing signatures/detections for two Mac OS X Trojans (commonly known as OSX.RSPlug and OSX.Iservice). This defence takes the form of an extension of the quarantine facility previously used by Safari, Mail, and iChat. (Intego, 2009a; Apple, 2007) The file Exceptions.plist indicates that the facility can be made use of by a number of specified browsers and email clients, while in theory other application developers can extend the functionality of their own programs to use the quarantining facility by setting the LSFileQuarantineEnabled key in their own info.plist files, if they are aware of it (Apple, 2007; Apple, 2009).

However, some issues remain unresolved: the efficacy of the quarantining measure is compromised in that detection of blacklisted malware is restricted to some (not all) variants of two known malicious programs

This approach restricts detection to a few, very specific execution contexts (Harley, 2009b). In fact, Intego (Intego, 2009a) argues that “Apple’s anti-malware function will never detect any iServices Trojans” because the primary distribution channel of iService, BitTorrent clients, are not included in Exceptions.plist.

We’ll leave aside the disparities between the restricted functionality of this utility and that of a full-blown commercial scanner (though it’s worth noting that it has neither full on-access nor full on-demand scanning functionality by which to make use of its severely limited range of “signatures”). However, the product also fails at a level that you’d expect the simplest scanner to try to achieve. Despite the continuing and growing interest on the part of cybercriminals, there appears to be no interest at Apple in adding detections. By early January 2010, nearly six months after the appearance of Snow Leopard, Ryan Naraine and other researchers confirmed that no changes had been made to XProtect.plist to reflect subsequent variants and more recent malware (Naraine, 2010).. Even the very common DNSChanger malware (which exists in a number of Mac-specific variations) has not been included.

Sadly, Apple has contributed a codicil to a Mac security issue that predates OS X by many years. A long procession of non-commercial scanners that, with a few honourable exceptions, hinder as much as they help, by feeding false expectations of total security where, in fact, only a subset of malware issues was being addressed. Even John Norstad, whose freeware “Disinfectant” was, arguably, one of the most successful and well-maintained non-commercial scanners ever, was obliged to discontinue development of the freeware version (Norstad, 1998) and pass the core code over to a commercial vendor for further development, realizing that Mac users were expecting it to provide protection even in the case of the macro virus epidemic of the second half of the 1990s. (In fact, Disinfectant had never addressed the full range of Mac threats: however, its limitations were fully and clearly explained in the documentation.)

Other developers were and are, no doubt well-meaning but far less scrupulous about addressing such issues as accurate documentation, timely updates, False Positives (FPs) and other bugs (Harley, 2008).

Apple Purist Puree or “There are no OS X viruses”

Or, why Macs have no security problems, never had security problems, and never will.

Or will they?

In fact, there is a significant disparity between this perception of the Mac as a safe haven and the threat landscape as we see it in the industry. While that landscape is a long way removed from the avalanche-scarred slopes inhabited by the Windows-using community, we’re painfully aware that in terms of unique (mostly Trojan) binaries, there already more OS X-specific threats than there were individual malicious programs for earlier Mac OS version, though the implications of that fact are rather complex.

Nevertheless, in such a (comparatively) sparsely-populated threatscape, does it really matter? Do Mac users really need Mac antivirus? Why are so many vendors now starting to service the needs of a user community that doesn’t, in general, see the need of such provision?

A range of commentators from Apple to the Mac-focused media to such information security luminaries as Rich Mogull have offered arguments to demonstrate how low the risk to Mac users is from security threats. Inevitably, some of these are better-founded than others.

“OS X doesn’t get PC viruses” (Apple, 2010)

Well, that’s a matter of definition. While the most dramatic example to date of multi-platform malware, the Office macro virus, has gone into an equally dramatic decline, there is plenty of potential for other cross-platform attacks such as scripting attacks. Many people are running some flavour of Windows on OS X in some environment or other, and most of the same security issues apply on Mac-hosted Windows as on “real” PCs. To think of security threats and the Mac only in terms of Mac/OS X-specific malware ignores the need for corporate multi-platform multi-layering and platform-independent social-engineering attacks on “wetware” (human beings) such as phishing and other forms of spam.

As we shall discuss below, in these days where many services are now ‘in the cloud’ and accessed via the browser, the potential for exploitation is high. Safari opens up several other applications when run, such as the calendar, address book, mail and so on, so as to ensure smooth integration, but this means that, because MacOS doesn’t use sandboxing for all applications, including Safari (Naraine & Danchev, 2007), an attacker is able to exploit those other applications as well as the browser. Javascript is a now infamous tool for exploiting vulnerabilities in browsers, and there is no reason to suspect that Safari suffers any less vulnerability in this respect than any of the other popular browsers. The key point today is that malware is about exploitation of systems to gain access to data. For the malware author this is not about being able to make some fancy Proof-of-Concept virus in order to gain kudos, but rather about finding any possible weakness in popular operating systems and applications that may give them an opportunity to gain access to sensitive and profit-generating data.

“Only Viruses Matter”

Apples are not the only fruit (Winterson, 1985) and viruses aren’t the only malware. While pre-OS X malware was largely viral, the common assumption among Mac users and commentators that “only viruses matter” is pure fallacy. As is the case with current Windows malware, classification of known Mac malware (as discussed at length in “The Mac Threatscape”) indicates that replicative malware is a relatively small part of an increasing problem, embracing, among other bits and pieces:

- Replicative Malware
- Rootkits
- Trojans
- Adware
- Spyware
- Fake AV

The Mac Threatscape

OS X’s kernel can be subverted, like that of any other operating system (OS). Many books, papers (Miller & Dai Zovi, 2009; Baccas, 2008), and code examples available from sources like Packet Storm and Phrack have been published on the topic. Most of the rootkits publicly discussed to date

are at the proof of concept (PoC) stage, but we have seen compiled versions of the WeaponX rootkit (which contains a number of subverted programs and source code) submitted for analysis, suggesting that some attackers are making active use of the PoC code in an attempt to hide the presence of their malware on a system.

Other open source initiatives such as logkext (<http://code.google.com/p/logkext>) are actively developing kernel extensions to log keystrokes on OS X. This tool's functionalities are regularly updated (<http://code.google.com/p/logkext/updates/list>) and even offer log encryption for improved stealth on a system. This means that any malware author can easily integrate key logging capabilities into his creation. We have also seen binaries of this kernel extension in the wild, once again suggesting that this code is likely to have been used in real attacks.

The Mac/Leap.A (CME, 2006; Van Oers, 2006) malware has attracted a lot of media attention and is believed to be the first worm to attack Mac systems. It appeared at the beginning of 2006. This worm spreads through the iChat application as a file named *latestpics.tgz*. Like many other malware, this threat uses a fake icon to disguise a binary executable as an image.

In February 2006, Kevin Finisterre released the code for a Proof-of-Concept worm targeting OS X systems. This worm (most often called OSX/Inqtana) is written in Java and spreads through a vulnerability discovered the previous year (see <http://www.securityfocus.com/bid/13491/info>) in Apple's Bluetooth system. To ensure persistence, this malware modifies the setting of *launchd* to make sure its code is executed at boot time.

OS X users are not immune to scareware (fake security software and so on), either. Over the last couple of years, we have seen (Ferrer, 2009) rogue applications pretending to clean or optimize Apple computers that were in fact fraudulent and of no use to any computer. Notorious examples of such annoyances include OSX/Imunizator (Sophos, 2008), a DMG installer which drops and launches a Mach-o binary and OSX/MacSweeper (Wikipedia, 2008).

The Mac/Hovdy malware family is a set of scripts designed to gather information from a host and send it back to a potential attacker. In some variants, the information is sent back in an email with the subject Howdy, hence the name. Some variants were programmed as a bash script while other variants are programmed using AppleScript. We have seen a just under a dozen different variants of the Mac/Hovdy script malware.

Proof of concept malware was discovered in 2009 and has been called Mac/Tored.AA, a modification of the original name found in the binary file, which was OSX.Raedbot. This worm can spread through email using its own SMTP engine. It can also contact a command and control server on the Internet to receive additional commands. Functionally, it therefore closely resembles certain classic Windows massmailers as well as many bots. However, we have not seen any instance of Mac/Tored.AA in the wild.

The family of DNS changing malware includes binaries identified as OSX/Jahlav, OSX/DNSchanger, OSX/Puper, OSX/RSPug (and sundry variations according to individual vendor naming conventions). Some vendors regard it as consisting of more than one family originating with the same author (Ferrer, M., 2009), but such distinctions are not maintained consistently across the vendor community. This group is also closely related to the Zlob family, associated with similar malicious functionality on Windows platforms. This type of malware is the one for which we have found by far the most files in the wild. It is predominantly found as a DMG file containing an installation package named *install.pkg*. It has been distributed using various schemes such as fake codecs, an approach commonly used by malware on other platforms. The ultimate purpose of this malware is to change DNS settings of an infected host, potentially enabling the attacker to alter content accessed from an infected system. The malicious actions are taken by a script named

preinstall executed at the beginning of the installer process. This script launches a set of shell commands to write its script to disk and execute it. An interesting point relating to OSX/Jahlav is that this threat uses server side polymorphism to generate new copies of its binaries, probably in an effort to evade detection by intrusion detection systems and antivirus software. Script files are also obfuscated using various shell tools such as *uuencode*, *sed*, and *tail* to conceal, vary or reverse the order of the commands and hamper analysis.

File sharing networks have been used for a long time to spread malware. Infected versions of popular applications such as iWork have been distributed on peer-to-peer (P2P) networks with a Trojan horse (Intego,2009b) . This Trojan, named OSX/Iservice, is a binary executable which opens a backdoor on infected computers giving an attacker complete access to the infected system.

This is by no means a complete list of known OS X malware, but perhaps it's enough to prove that there is a problem, even if the current size of the problem is open to debate. While malicious files related to OS X are still rare, coverage by AV vendors can sometimes be inaccurate. In many cases, benign files are flagged as malicious simply because analysts don't have in-depth knowledge of the operating system and prefer to label everything contained in an archive as malicious instead of concentrating their efforts on better detection of truly malicious content. Nonetheless, our research indicates hundreds of unique binaries including rogue antivirus, adware, keyloggers, out-and-out Trojans, and worms.

This looks trivial compared to the tens of thousands of unique binaries processed by virus laboratories on a daily basis – actually a conservative estimate (Harley, 2010) – it's far from the picture of Port Macintosh as a safe haven that is so often painted by Apple and others. In terms of unique Mac-specific binaries, it's a marked increase over the numbers of pre-OS X malware (ignoring cross-platform malware, notably macro malware, and platform-independent social engineering attacks).

“Multi-layered protection”: the gospel according to Apple”

Although Apple makes great noise about its multi layered approach to protecting the machine, under the hood it's a different story (and has little in common with the sort of cross-platform multi-layered protection we associate with enterprise defence in depth. Central to MacOSX is a program called *launchd* that combines functionality from several standard UNIX programs into one single utility: basically, it replaces the following services that on more standard versions of UNIX remain discrete:

- SystemV Init and all its needed runlevel scripts – this is used to select and initiate the default runlevel
- *Cron*, which is used for scheduling tasks such as cleanup scripts, log rotations or any script that might need to be scheduled (for example, anti-virus definition updates)
- *xinetd*, which is used to start services on demand (for instance, an ftp server might be started once a connection to port 21 is initiated – this avoids having the service constantly in memory)
- *mach init* – the UNIX equivalent of the Mach microkernel) – which takes care of mapping ports to services and registration of new service ports

There have been several vulnerabilities reported for this program and since it runs as root, usually these are serious – for instance, CVE-2006-1471 (CVE, 2006:), a vulnerability caused by failure to validate input correctly. Since the service provides several traditionally separate services, this

increases its complexity and its attack surface, and since it is also dealing with setting up and managing networked services the likelihood is that much higher that vulnerabilities will be remotely exploitable. Mac OS itself is a non-standard combination of the Mach microkernel and BSD Unix, with a new driver model called IOKit thrown in. Mach is not used in the true microkernel sense. Drivers run in the usual kernel address space and programs written for MacOS can use a mix of Mach and BSD APIs. For this reason there is a huge potential for attacks since this whole model is unproven.

That said, the Apple security model includes many useful – though in some cases more limited than popularly realized – attributes and defensive techniques (Apple, 2010):

- **Sandboxing:** although Apple did include sandboxing facilities with the advent of Leopard, it turns out that only a very few selected applications are sandboxed, and, bizarrely, Safari is not one of them. This has led to several attempts, such as Sandboxed Safari (<http://www.tomsick.net/projects/sandboxed-safari>) to rectify this shortcoming.
- **Library Randomization:** this offers some measure of protection, but unfortunately does not go far enough. ASLR, as applied in Mac OS X, does not cover stack, heap or code randomisation, meaning that the implementation is incomplete (see <http://www.laconicsecurity.com/aslr-leopard-versus-vista.html>) and leaves many categories of attack available.
- **Execute Disable:** again, executable space protection was introduced with Leopard, but only applied to Intel processors (PPC systems remained unprotected), and in 32-bit systems, only the stack was protected, whereas in the 64-bit systems, the heap is also protected. As has been pointed out (<http://www.laconicsecurity.com/aslr-leopard-versus-vista.html>), since most applications are 32bit (and are likely to remain so for some time) this still leaves many systems vulnerable to heap spray/overflow attacks.
- **Update and Patching:** this is one area that Mac OS X handles well, and in a somewhat simpler manner than Windows. However, since security patches tend to be ‘rolled up’ into packages, patching takes something of an ‘all or nothing’ approach. That said, Apple has the distinct advantage of being available in far fewer hardware configurations, and therefore its Quality Assurance process tends to avoid the sort of problems that Microsoft’s patches can sometimes introduce (see TDSS MS010-15 blue screen for instance, as described by Brian Krebs at <http://www.krebsonsecurity.com/2010/02/new-patches-cause-bsod-for-some-windows-xp-users/#more-1003>)
- **Firewalling:** the MacOS approach to firewalling is very simple, but far less configurable than the Windows equivalent (at least, since XP-SP2). There is very little fine-grained control over the firewall (such as application-level firewalling), and indeed there seem to be few (if any) third-party firewalls that can provide such extended functionality. This means that in most cases, the user must either accept the default options and take the risk of opening up a particular service, or forgo desktop firewalling.

In some respects, such as patching and enforced adherence to the principle of least privilege (apart, perhaps for its penchant for running many of its own programs as SUID root), OS X has from time to time outshone its stepsister from Redmond: however, it is naive to assume that it has maintained its lead over recent generations of Microsoft operating systems. In some respects, especially those relating to malware, Microsoft’s appreciation of the threat landscape in which it operates is far more realistic than Apple’s. While Mac users – with the exception of those making significant use of Windows on Macs – operate in an environment prowled by infinitely fewer predators, Microsoft

and its more savvy customers are to some extent shielded by a more accurate assessment of the risks to which Windows users are exposed.

Apple's "hear no evil, see no evil" philosophy (and that of its more fanatical supporters) when it comes to malware and "wetware" attacks works to the ultimate detriment of those customers. Numerically, the victims of this philosophy are still fairly small, but as Apple's market share increases, so do the number of potential victims, criminal interest in exploiting those victims, and the likelihood of serious breaches analogous to the Autostart worm of the 1990s.

We must reiterate that it's not realistic to think purely about Mac-hosted malware, old or new. For example, Dancho Danchev has reported on "How the Koobface gang monetarizes Mac OS X" by compromising legitimate sites with a PHP backdoor shell in an attempt to direct OS X traffic to affiliate dating programmes. (Danchev, 2010) He has also posted information on a phishing campaign where the bad guys are impersonating Apple in order to steal sensitive device information from iPhone users (Naraine & Danchev, 2010). This isn't "the sky is falling" stuff, but these aren't isolated incidents, either.

Apple, Macs and the iPhone

What does the recent furore over iPhone (and other smartphone) jailbreak exploitation tell us about Apple security in general? More than you might think. When the Mac Virus site now maintained by one of the authors was first built in the 1990s by Susan Lesch, Apple's product range was more limited than it is now. These days, it doesn't make sense to restrict Apple coverage to desktops and Macbooks, so the revival of the Mac Virus site as an Apple security-focused blog pretty much began (see <http://macvirus.com.wordpress.com/2010/02/03/iphone-and-ipod-touch-news/>) with iPhone-related items such as a report on the vulnerability of the iPhone to a remote attack on SSL, flagged by vulnerability researcher Charlie Miller, who specializes in Mac issues, and Heise's summary of the the vulnerabilities addressed in iPhone/iPod OS 3.1, as well as a number of issues explicitly flagged by the Common Vulnerabilities and Exposures page at <http://cve.mitre.org>.

Vanja Svajcer's commentary (see <http://www.sophos.com/blogs/sophoslabs/?p=8580>) on presentations by Nicolas Seriot at Blackhat and Tyler Shields at SchmooCon makes an excellent point on the limited effectiveness of application whitelisting and certification by smartphone vendors. One of the interesting points about Seriot's presentation was that it talked about "unmodified" devices when demonstrating a rogue app that can access personal data "in spite of AppStore tight reviews".

Does the argument that jailbreaking a smartphone (the iPhone is not the only mobile device whose security is largely dependent on application whitelisting by the vendor) is unethical (debatable, but certainly not an unreasonable position), a breach of the agreement between Apple and its customers (difficult to argue with), and so on, relieve Apple of responsibility for security for jailbroken devices? Perhaps that depends on the risk that such devices pose to legitimate users, but that risk isn't really quantifiable (certainly in terms of future threats). So it doesn't seem entirely responsible to decline *any* responsibility for the very sizeable population of users who've chosen to go that route, irrespective of arguments about choice versus paternalism. After all, where you choose to stand on that continuum has direct security implications.

Despite the fact that all three of the authors are currently employed within the anti-malware industry, we don't claim that there is an unequivocal need for commercial antivirus on every Mac, still less every iPhone. We would, however, like to see more recognition by Apple that the company cannot offer unbreakable, out-of-the-box protection for all its users and over its entire product range.

Conclusion

In a recent Guardian blog, Jack Schofield (Schofield, 2010) answered the question “Does a Mac need anti-virus protection?” in the following words:

“I don’t know of any live malware attacking Mac OS X, so you probably don’t need either anti-virus or anti-malware software at the moment. However, this does not mean you shouldn’t run it. If you are a home user, you don’t have to care what happens to your data, but business users do. It may be wise to take precautions, even if they don’t appear to be necessary.”

Playing devil’s advocate for a moment, we don’t quite see why anyone should run anti-malware on a Mac even though they “don’t need” it. On the other hand, we don’t think that business data are necessarily more “important” than a home user’s data: there are certainly scenarios where loss of work data at work is a trivial annoyance, but loss of data at home is a disaster. (Mac Virus, 2010)

It *is* correct to distinguish between business and home users in that there are threats that transcend platform-specific vulnerability (phishing, adware redirection), and there are compelling reasons why any business that has a Mac-using population should extend its security software coverage beyond Apple’s “out of the box” security. As one of us wrote in response to Schofield’s blog:

“For home users, the situation may be a little less clear-cut. If you want to give anti-malware a miss at the moment because you’re too bright to fall for social engineering Trojans, you’re prepared to accept the relatively small risk in terms of volume, you aren’t worried about 0-day self-launching exploits, and so forth, be my guest...I would advise, though that you don’t act on the unfounded assumptions that there is no Mac malware, or that only viruses matter.” (Mac Virus, 2010)

Macs, Malware, and the Vendor Community

There is a clear resurgence of interest in Mac anti-virus (AV) products, the Mac Virus web site (<http://www.macvirus.com>) and so on, from the media and the vendor community, at any rate. It seems unlikely that there’ll be much interest at consumer level, though the inclusion of iPhone security material on the Mac Virus site does seem to have stimulated an unanticipated degree of interest.

It will take an malware drama like the data damage caused by the Autostart worm in the 1990s to persuade the average Mac-user that they need AV, and a *highly-publicized* disaster to persuade them that they need to pay for AV they have to pay for, so there is probably no un milked cash cow in the room (standing next to the elephant). At the enterprise level, some established vendors may feel a slight chill. Vendors who now have a Mac product will benefit from customers with multiple platforms who like the Windows and/or Linux products they already have, so will give their Mac product a try, to see if they can benefit from integrating products from the same source rather than mixing and matching. However, the big players in the corporate space are unlikely to lose much business in the short term, unless they have customers who are really dissatisfied with all of them.

There does seem to be an increase in raw hardware sales of course. As the Bad Guys have got more interested as a result of that swelling pool of potential victims using Apples rather than Windows, obviously the security community has taken a corresponding interest. A sound OS X sample collection now includes hundreds of unique binary samples, more than we ever needed for pre-OS X Mac-specific testing in the 1990s. That doesn’t mean that there is a single unique threat for each sample, but it does mean that there’s a lot more out there than the handful of variants Snow Leopard’s own utility is intended to recognize.

It's not essential right now for a vendor to have a Mac-specific product, though it's nice for those customers with a foot in both camps if they do. But vendors cannot afford to ignore threats on platforms they don't support with a native product. They should, at a minimum, detect Windows/Mac/Linux malware at the perimeter and on file servers.

Most of all, though, Apple needs to be more aware at many levels that Mac malware does exist, and is increasing in volume. It seems that even its own support staff are not aware that Snow Leopard itself contains countermeasures against a couple of Mac threats, and if they are, may be unaware of how seriously restricted those countermeasures are. And clearly, few Apple spokesmen are thinking about people running Windows under OS X, or in a multi-platform environment. (Mac Virus, 2010)

References

- Apple (2007). Launch Services Framework Release Notes for Mac OS X v10.5. Retrieved 20 March, 2010 from <http://developer.apple.com/mac/library/releasenotes/Carbon/RN-LaunchServices/index.html>
- Apple (2009). Launch Services Keys. Retrieved 20 March, 2010 from http://developer.apple.com/Mac/library/documentation/General/Reference/InfoPlistKeyReference/Articles/LaunchServicesKeys.html#//apple_ref/doc/uid/TP40009250-SW10
- Apple (2010) Mac OS X has you covered. Retrieved 10 March, 2010 from <http://www.apple.com/macosx/security/>
- Baccas, P. (2008).. P. Baccas (Ed.), OS X Exploits and Defense (pp122-131): Syngress,
- CERC (2009). Securing Our e-City National Cybercrime Survey: Competitive Edge Research and Communication, Inc.
- Cluley, G. (2009). The Top Ten Clu-Blogs of 2009. Retrieved 10 March, 2010, from <http://www.sophos.com/blogs/gc/g/2009/12/31/popular-clublog-posts-2009/>, <http://www.sophos.com/blogs/gc/g/2009/12/30/top-ten-clublogs-2009/>
- CME (2006). CME List. Retrieved 20 March, 2010, from <http://cme.mitre.org/data/list.html#4>
- CNET (2008). Apple suggests Mac users install antivirus software. Retrieved 10 March, 2010, from http://news.cnet.com/8301-1009_3-10110852-83.html
- CVE (2006) CVE-2006-1471. Retrieved 19th March 2010 from <http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2006-1471>
- Danchev, D. (2010). How the Koobface Gang Monetizes Mac OS X Traffic. Retrieved 10 March, 2010, from <http://ddanchev.blogspot.com/2010/02/how-koobface-gang-monetizes-mac-os-x.html>
- F-Secure (2008). Mac Case. Retrieved 10 March, 2010, from <http://www.f-secure.com/weblog/archives/00001388.html>
- Ferrer, M. (2009) A Closer Look at Mac OS X Threats. Virus Bulletin Conference Proceedings (pp153-164): Virus Bulletin..
- Harley, D. (2008). Malware Detection and the Mac. In P. Baccas (Ed.), OS X Exploits and Defense (pp122-131): Syngress,
- Harley, D. (2009a) Droid Avoids with an AppleJackHack. Retrieved 20 March, 2010, from <http://www.eset.com/blog/2009/12/11/droid-avoids-with-an-applejackhack>
- Harley, D.(2009b) Execution Context in Anti-Malware Testing. In E. Filiol (Ed.), 18th EICAR Annual Conference Proceedings (pp. 203-218): EICAR
- Intego (2009a). How the Anti-Malware Function in Apple's Snow Leopard Works. Retrieved 10 March, 2010 from <http://blog.intego.com/2009/09/02/how-the-anti-malware-function-in-apples-snow-leopard-works/>
- Intego (2009b). Mac Trojan Horse OSX.Trojan.iServices.A Found in Pirated Apple iWork 09. Retrieved 20 March, 2010, from <http://www.intego.com/news/ism0901.asp>
- Mac Virus (2010). Is there such a thing as Mac malware? Retrieved 20 March, 2010, from <http://macviruscom.wordpress.com/2010/02/04/is-there-such-a-thing-as-mac-malware/>

- Miller, C. & Dai Zovi, D. (2009). *The Mac Hacker's Handbook*: Wiley.
- Naraine, R. (2010). Apple Malware Blocker Left For Dead? Retrieved 20 March, 2010, from http://threatpost.com/en_us/blogs/apple-malware-blocker-left-dead-010410
- Naraine, R. & Danchev, D. (2010). Memory randomization (ASLR) coming to Mac OS X Leopard. Retrieved 20 March, 2010, from <http://blogs.zdnet.com/security/?p=595>http://news.cnet.com/8301-10784_3-9759132-7.html
- Naraine, R. & Danchev, D. (2010). Scammers phishing for sensitive iPhone data. Retrieved 20 March, 2010, from <http://blogs.zdnet.com/security/?p=5460&tag=col1;post-5460>
- Norstad, J. (1998). Disinfectant Retired. Retrieved 20 March, 2010, from <http://homepage.mac.com/j.norstad/disinfectant-retire.txt>
- Schofield, J. (2010). Does a Mac need anti-virus protection? (updated). Retrieved 10th March, 2010, from <http://www.guardian.co.uk/technology/askjack/2010/feb/03/apple-data-computer-security>
- Sellar, W.C. & Yeatman, R.I. (1930). *1066 And All That*, Methuen. Retrieved 10th March, 2010, <http://www.methuen.co.uk/titles.php/isbn/0413772705>
- Sophos (2008). Mac OS X Trojan horse aims to make money from Macintosh users. Retrieved 10 March, 2010 from <http://www.sophos.com/pressoffice/news/articles/2008/03/imunizator.html>
- Van Oers, M. (2006).. Macintosh OSX binary malware. Retrieved 20 March, 2010 from http://www.virusbtn.com/pdf/conference_slides/2006/MariusVanOersVB2006.pdf
- Wikipedia (2008). MacSweeper. Retrieved 20 March, 2010, from <http://en.wikipedia.org/wiki/MacSweeper>
- Winterson, J. (1985). *Oranges Are Not The Only Fruit*: Pandora Press.
- Ziff-Davis (2009). Snow Leopard's malware protection only scans for two Trojans. Retrieved 10th March, 2010 from <http://blogs.zdnet.com/security/?p=4139>