

Macs in the Age of APT

Tom Daniels, Aaron Grattafiori, BJ Orvis, Alex Stamos, Paul Youn

iSEC Partners

Black Hat USA 2011



Agenda

1 Motivation

- Preface and Background

2 Anatomy of an APT

- Social Engineering
- Initial Exploitation
- Local Privilege Escalation
- Network Privilege Escalation
- Persistence
- Exploration
- Exfiltration

3 Conclusion

- Summary

Outline

- 1 Motivation
 - Preface and Background
- 2 Anatomy of an APT
 - Social Engineering
 - Initial Exploitation
 - Local Privilege Escalation
 - Network Privilege Escalation
 - Persistence
 - Exploration
 - Exfiltration
- 3 Conclusion
 - Summary

What is APT?

Apple Purchases Tacos?

- **Advanced:** not your average Joe, may be government funded, may have zero-day vulnerabilities.
- **Persistent:** initial access leads to the creation of many access methods and long-term exploration
- **Threat:** defines the group of attackers with these capabilities, not an actual attack scenario

Case Study: Aurora

What the what?

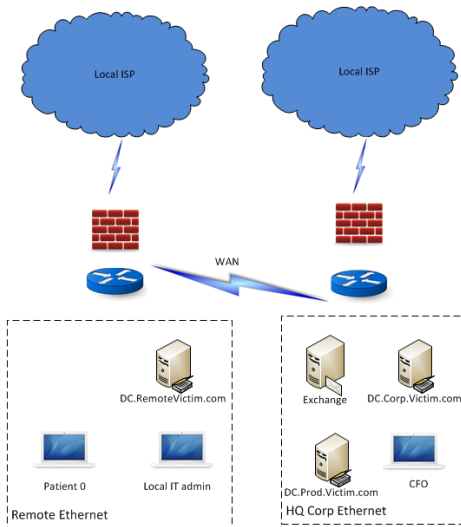
- Originally disclosed by Google on January 12th 2010
- Google discovered evidence of >30 other victims
- Attack was focused on Windows exploitation and escalation in AD
- Estimates range from dozens to hundreds of companies attacked¹
 - Google
 - DuPont
 - Adobe
 - Juniper Networks
 - Northrop Grumman
 - Sony
 - And many more

¹http://threatpost.com.mx/en_us/blogs/

[hbgary-e-mails-dupont-other-firms-hit-aurora-attack-031011](http://threatpost.com.mx/en_us/blogs/hbgary-e-mails-dupont-other-firms-hit-aurora-attack-031011)

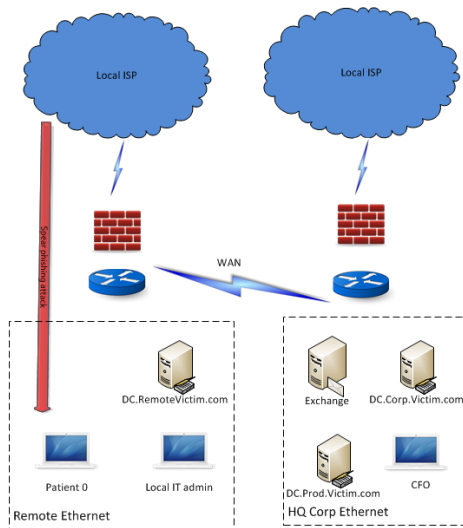
Case Study: Aurora

Socially engineer a victim to click on a malicious link



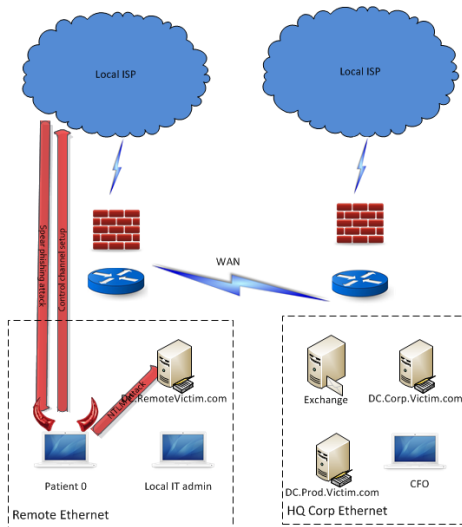
Case Study: Aurora

Socially engineer a victim to click on a malicious link



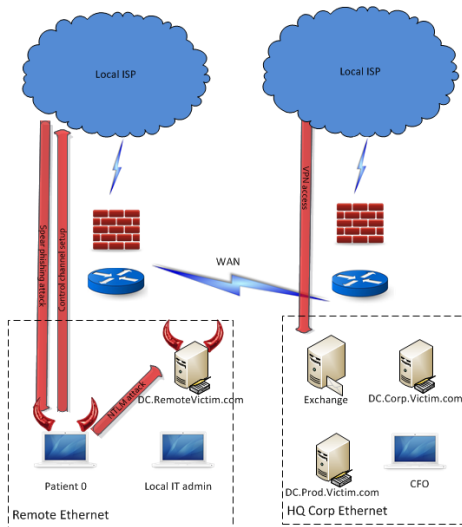
Case Study: Aurora

Escalate network privileges



Case Study: Aurora

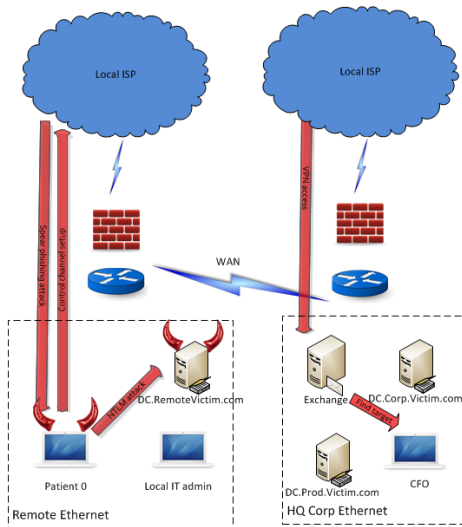
Make your attack more persistent



iSEC
PARTNERS

Case Study: Aurora

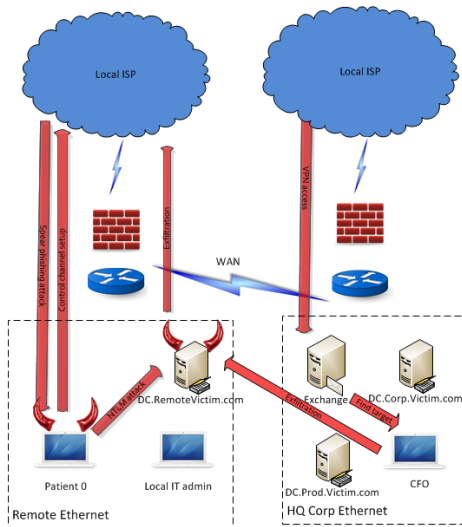
Explore



iSEC
PARTNERS

Case Study: Aurora

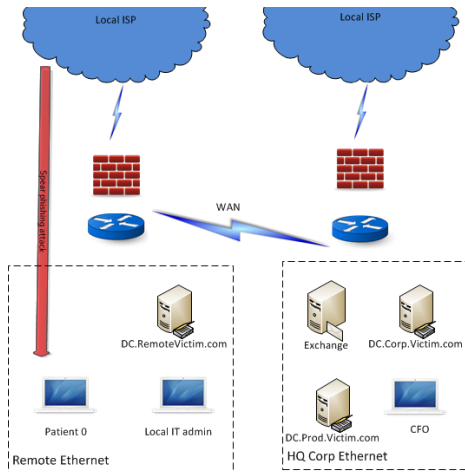
Exfiltrate the data



iSEC
PARTNERS

Outline

- 1 Motivation
 - Preface and Background
- 2 Anatomy of an APT
 - **Social Engineering**
 - Initial Exploitation
 - Local Privilege Escalation
 - Network Privilege Escalation
 - Persistence
 - Exploration
 - Exfiltration
- 3 Conclusion
 - Summary



Your Mac is Safer

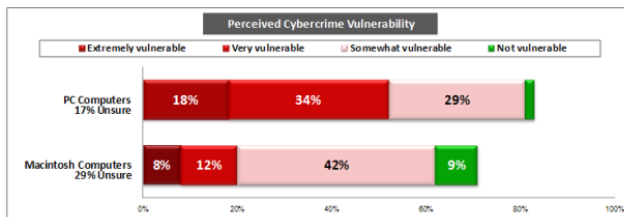
- Apple has a small computer market share (6-8%)²
- Building a bot-net? Go for Windows users
- There are fewer viruses and malware applications for Mac
 - No exploits included in common crimeware toolkits targeting Macs³
 - Attacks focus on social engineering (such as Mac Defender)

²<http://www.networkworld.com/news/2011/060611-mac-os-security.html>

³See iSEC consultant Dan Guido's research

Training Mac Users to Feel Safe

- A history of non-exploitation
- Go ahead, run this unsigned binary
- Who needs anti-virus?⁴



More than half of Americans believe that PCs are "very" or "extremely" vulnerable to cybercrime attacks, while only 20 percent say the same about Macs, according to this ESET survey.

(Credit: ESET)

⁴http://news.cnet.com/8301-27080_3-10444561-245.html

Apple Marketing is Misleading

Sort of like all marketing (unrelated: hire iSEC because we are the best at everything)

- “OS X doesn’t get *PC* viruses”^a
- Other things OS X can’t catch:
 - A Nintendo Wii virus
 - Mad cow disease, malaria, or chickenpox
 - Footballs (we tried)
- OS X is still vulnerable to malware (like almost any computer system)




Secure by design.

OS X doesn’t get PC viruses. And with virtually no effort on your part, the operating system protects itself from other malicious applications. Because every Mac ships with a secure configuration, you don’t have to worry about changing complex settings in order to stay safe. Even better, OS X won’t slow you down with constant security alerts and sweeps. Apple responds quickly to online threats and automatically delivers security updates. And with FileVault 2 in OS X Lion, all the data on your Mac is protected by powerful encryption.

^a<http://www.apple.com/macosex/security/>

Mac Users are Susceptible to Social Engineering


- Mac users aren't as paranoid as Windows users⁵


kairiebarie
Calculating status...

May 20, 2011 3:13 PM

I have a virus on my mac book

Like (0)


Michael Superczynski
Level 4 (3,445 points)

Re: I have a virus on my macbook
May 20, 2011 3:16 PM (in response to kairiebarie)

If you do, you will make history.

There are NO virii that can affect OS X. None. Nada. Zero. Zilch. Low-values. Binary zero. All bits off.

Like (0)

- Mac Defender
- Mac users may be easy to socially engineer

⁵<https://discussions.apple.com/message/15242642#15242642>

OS X isn't Safer

- 14.3% of publicly disclosed OS vulnerabilities affected OS X in 2008⁶

Operating System	Percentage
Apple Mac OS X Server	14.3%
Apple Mac OS X	14.3%
Linux Kernel	10.9%
Sun Solaris	7.3%
Microsoft Windows XP	5.5%

- Latest OS X security patch addressed 39 CVEs
- 1,151 CVEs reported in the last 3 years affect Apple (including third-party software)
- Similar number of Windows CVEs (1,325)
- Safety in numbers

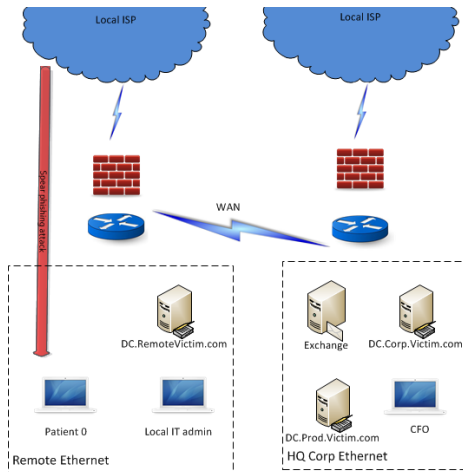
⁶Subsequent annual reports focused on mobile operating systems. Source: <http://www-935.ibm.com/services/us/iss/xforce/trendreports/xforce-2008-annual-report.pdf>

Back to APT

- Targeted attackers don't care what OS a corporation is running
- Mac users may be more vulnerable Social Engineering
- Plenty of vulnerabilities lead to "Initial Exploitation"

Outline

- 1 Motivation
 - Preface and Background
- 2 Anatomy of an APT
 - Social Engineering
 - **Initial Exploitation**
 - Local Privilege Escalation
 - Network Privilege Escalation
 - Persistence
 - Exploration
 - Exfiltration
- 3 Conclusion
 - Summary

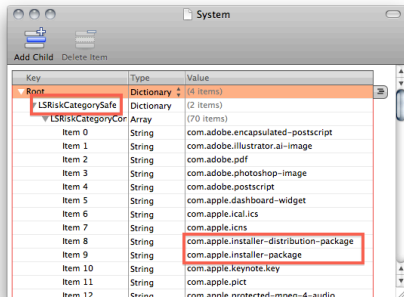


Exploitation in APT

- Get user to click a link
- And then exploit...
 - Railroad user into an installer with Safari's safe files
 - Browser or plugin exploit

Safari's open "safe" files includes installers

- .pkg and .mpkg files
- A .zip containing a .pkg runs Installer.app
- User must click through
- MACDefender⁷ and variants triggered a "4-5x higher than normal" call volume with AppleCare when it hit⁸



Key	Type	Value
Root	Dictionary (4 items)	
LSRiskCategorySafe	Dictionary (2 items)	
LSRiskCategoryCor	Array (70 items)	
Item 0	String	com.adobe.encapsulated-postscript
Item 1	String	com.adobe.illustrator.ai-image
Item 2	String	com.adobe.pdf
Item 3	String	com.adobe.photoshop-image
Item 4	String	com.adobe.postscript
Item 5	String	com.apple.dashboard-widget
Item 6	String	com.apple.ical.ics
Item 7	String	com.apple.icns
Item 8	String	com.apple.installer-distribution-package
Item 9	String	com.apple.installer-package
Item 10	String	com.apple.keynote.key
Item 11	String	com.apple.pict
Item 12	String	com.apple.protected-mpeg-4-audio

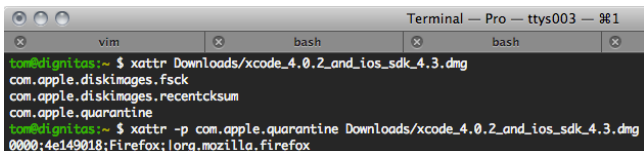
⁷<http://blog.intego.com/2011/05/02/>

macdefender-rogue-anti-malware-program-attacks-macs-via-seo-poisoning/

⁸<http://www.zdnet.com/blog/bott/>

an-applecare-support-rep-talks-mac-malware-is-getting-worse/3342?pg=1

File Quarantine and XProtect



```
Terminal — Pro — ttys003 — 061
vim bash bash
tom@dignitas:~ $ xattr Downloads/xcode_4.0.2_and_ios_sdk_4.3.dmg
com.apple.diskimages.fsck
com.apple.diskimages.recentcksum
com.apple.quarantine
tom@dignitas:~ $ xattr -p com.apple.quarantine Downloads/xcode_4.0.2_and_ios_sdk_4.3.dmg
0000;4e149018;Firefox;org.mozilla.firefox
```

- File Quarantine

- Part of the LaunchServices API
- Quarantine properties dictionary
- const CFStringRef kLSItemQuarantineProperties

- XProtect

- Signature-based scanner
- Piggy-backs on File Quarantine
 - Downloaded files marked with extended attribute
 - LaunchServices triggers scan
- In its infancy on Mac OS X (introduced in 10.6)
- Security Update 2011-003: Malware database now updates daily⁹

⁹<http://support.apple.com/kb/HT4657>

Anti-exploit Mitigations

Mitigation availability:

Mitigation	Windows	Mac OS X
Stack Protections	2003 (Visual Studio's /GS)	2007 (10.5/XCode 3.1)
Heap Protections	2003 (XP SP2) ¹⁰	2009 (10.6)
DEP	2004 (XP SP 2)	2006 (10.4.4 Intel)
ASLR	2007 (Vista)	2007 (10.5)

¹⁰<http://blogs.technet.com/b/srd/archive/2009/08/04/>

Smash the Stack

- GCC ProPolice can be used at compile-time ($GCC \geq 4.1$)
- 10.5/XCode 3.1: GCC 4.2 first included, but not the default (GCC 4.0)
- 10.6/XCode 3.2: GCC 4.2 the default, -fstack-protector enabled by default
- Binaries built using older toolchain may not have it enabled

Break the Heap

- Mac OS X
 - 10.5: checksum — not a security protection
 - 10.6: Include a security cookie — better¹¹
- Windows
 - XP SP2 and Server 2003¹²: Safe unlinking and heap entry header cookie
 - Vista and later: Numerous additional heap protections

¹¹<http://securityevaluators.com/files/papers/SnowLeopard.pdf>

¹²<http://blogs.technet.com/b/srd/archive/2009/08/04/>

NX/DEP/ED

- Supported on Intel architectures
- Sets the default `mprotect()` exec flag for heap and stack
- 10.6: heap always executable for 32-bit binaries
 - not even `mprotect()` can disable
- 10.7: 32-bit binaries compiled on 10.6 still have always-executable heaps

	10.4	10.5		10.6		10.7	
	i386	i386	x86_64	i386	x86_64	i386	x86_64
Stack	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Heap	No	No	No	No	Yes	Yes	Yes

ASLR

- 10.5: First introduced
- 10.6: No major changes
 - Not all libs use it
 - Not application code
 - Not the stack or heap
 - ROP exploits possible using dyld¹³
- 10.7: Supposedly improved¹⁴



Security

Enhanced runtime protection

Address space layout randomization (ASLR) has been improved for all applications. It is now available for 32-bit apps (as are heap memory protections), making 64-bit and 32-bit applications more resistant to attack.

¹³<http://securityevaluators.com/files/papers/SnowLeopard.pdf>

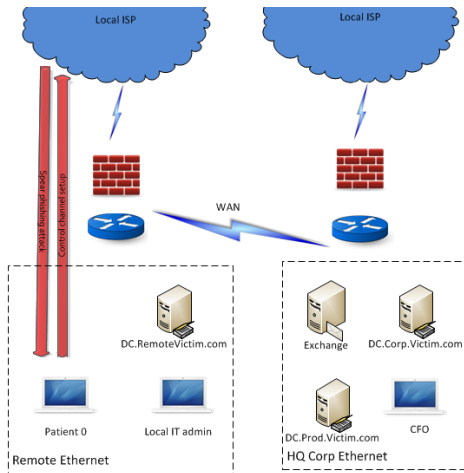
¹⁴<http://www.apple.com/macosx/whats-new/features.html#security>

Back to APT

- Been behind Microsoft, but finally catching up
- DEP and ASLR are not configurable
- Backwards compatibility threats

Outline

- 1 Motivation
 - Preface and Background
- 2 Anatomy of an APT
 - Social Engineering
 - Initial Exploitation
 - **Local Privilege Escalation**
 - Network Privilege Escalation
 - Persistence
 - Exploration
 - Exfiltration
- 3 Conclusion
 - Summary



Accessing Patient Zero's Data

Information stored on disc

- Locally stored E-mail
- Safari History, Bookmarks
- iChat logs
- Spotlight DBs

Escalating Privilege

Attacking the login keychain

- Code execution doesn't mean full account access
- The “Login Keychain” can be used to brute-force the user's password

Escalating Privilege

Sudo make me a sandwich¹⁵

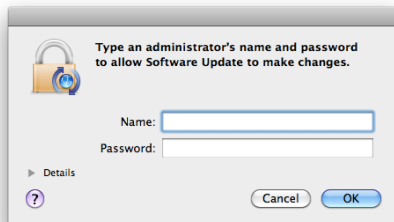
- If a user is a sudoer, password can directly escalate privilege
- User password can be used to decrypt the “Login Keychain”
- Privileged credentials in the keychain can be used to spread and explore

¹⁵<http://xkcd.com/149/>

Escalating Privilege

Phishing for admin

- OS X requires authorization for privileged action:

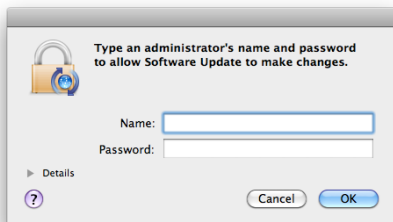


- Windows UAC screen slightly harder to spoof

Escalating Privilege

Phishing for admin

- This application sends admin credentials offsite in an HTTP “GET”



```
"GET /paul/Usernameis/isecadmin/Password/p@ssw0rd HTTP/1.1"
```

- UAC can be spoofed on Windows as well

Lion Improvements

AppSandbox: a safer place to play

- Subscription-based via plist

```
<key>com.apple.security.app-sandbox</key>  
<true/>
```

- Per application container

```
export $HOME=~/.Library/Containers/app.bundle.id/Data
```

- Per session entitlements
- Powerbox (pboxd)
 - sandbox-free broker process
 - transparent to developers (NSOpenPanel/NSSavePanel)

Lion Improvements

AppSandbox: cool kids use least privileges

- Entitlements

- `com.apple.security.documents.user-selected`
- `com.apple.security.assets`
- `com.apple.security.network`
- `com.apple.security.personal-information`
- `com.apple.security.device`

- Temporary Exceptions

- `$HOME/absolute` file access
- Send Apple Events
- Look up mach services
- Inherit

Lion Improvements

XPC: Intra-application privilege separation

- libSystem IPC API
- XPC binaries stored in `Bundle.app/Contents/XPC`
 - Address space isolation
 - Fully restricted sandbox by default
 - Elevating XPC service to root is unsupported
- On-demand launching
 - integration with GCD and launchd
- Quicktime Player uses a low-privileged process called `VTDecoderXPCService`¹⁶

¹⁶<http://arstechnica.com/apple/reviews/2011/07/mac-os-x-10-7.ars/9>

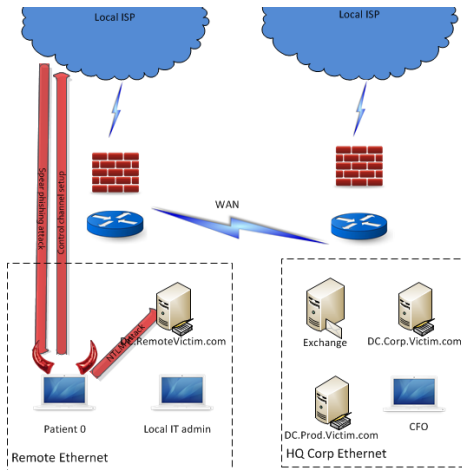
Back to APT

What can the local user do

- Access valuable local data
- Brute-force a valuable credential store
- Phish for admin credentials
- Help is on the way?

Outline

- 1 Motivation
 - Preface and Background
- 2 Anatomy of an APT
 - Social Engineering
 - Initial Exploitation
 - Local Privilege Escalation
 - **Network Privilege Escalation**
 - Persistence
 - Exploration
 - Exfiltration
- 3 Conclusion
 - Summary



Network Security Weaknesses

Application Level Firewall

- By default, signed binaries can open listening ports and holes in the firewall
- But some signed binaries are “dangerous”
- A case study...
 - Netcat is signed
 - Netcat is in a special blacklist
 - The blacklist is based on a path, the signature is within the file
 - Copy the file -> win the game
- Other signed binaries that can open ports (that are not blacklisted) likely exist
- And there are other weaknesses in Apple’s enterprise protocols

Lots of Services Makes Us Enterprise, Right?

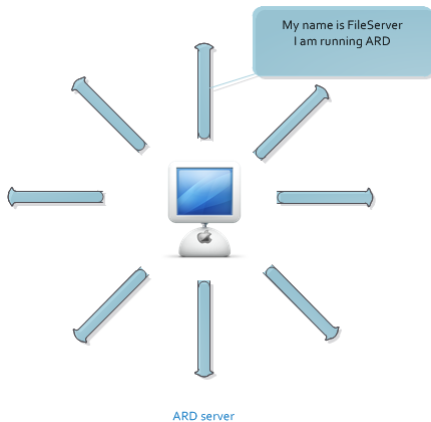
Right?

- Presented at SOURCE Seattle
- Looked at Snow Leopard Server (10.6)
 - 28 network ports open after default install!!!
- A quick (incomplete) look:

Service	Best Auth Method	Integrity?	Confidentiality?
AFP	Kerberos	No	No
ARD	Custom (DH)	No	Yes (AES)
Bonjour	None	No	No
ServerAdmin	Self Signed Cert	Yes (SSL)	Yes (SSL)

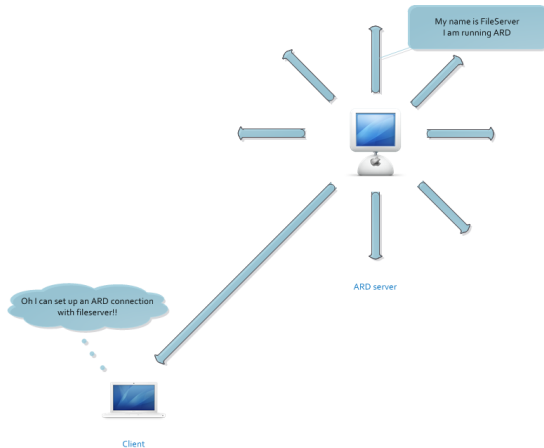
Bonjoof Beta

File server offering ARD services



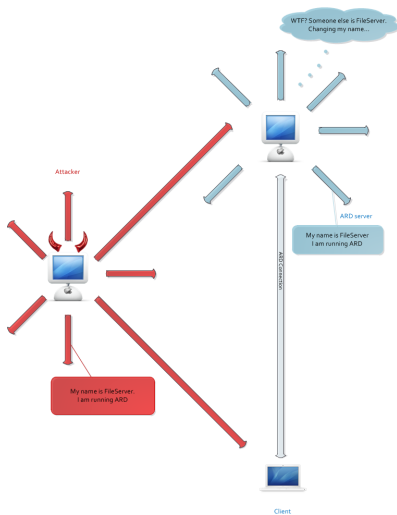
Bonjoof Beta

Administrator enjoys his coffee



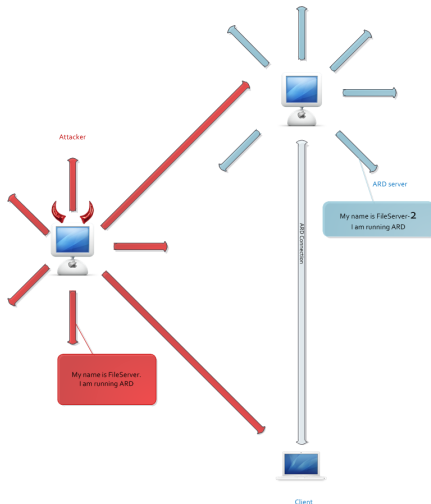
Bonjoof Beta

Spoofing mDNS



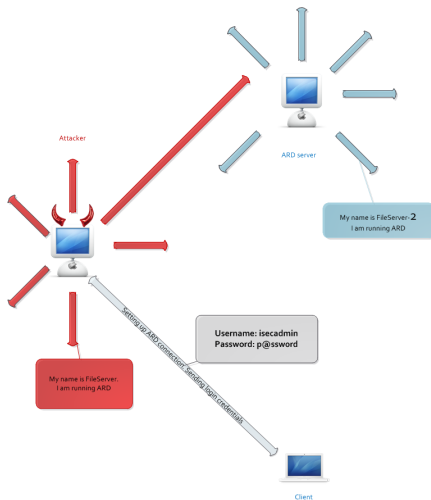
Bonjoof Beta

Claiming the hostname



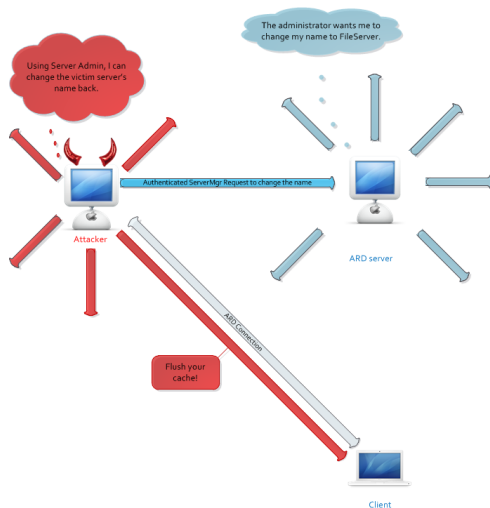
Bonjoof Beta

ARD client silently updates its stats



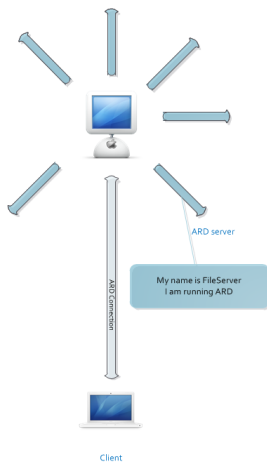
Bonjoof Beta

Reset the file server's hostname



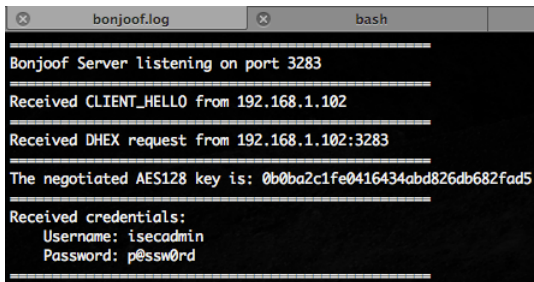
Bonjoof Beta

Where'd who go?



Bonjoof Beta

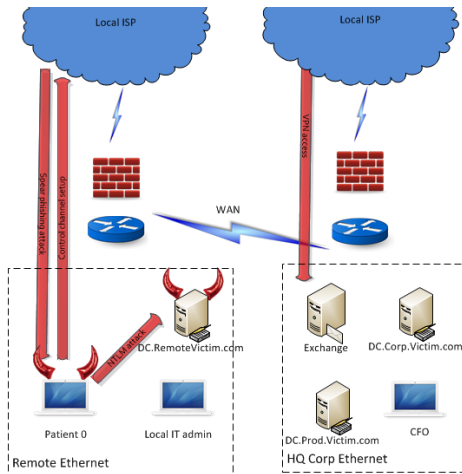
Some sample tool output

A screenshot of a terminal window with two tabs: 'bonjoof.log' and 'bash'. The terminal output shows the following text:

```
=====
Bonjoof Server listening on port 3283
=====
Received CLIENT_HELLO from 192.168.1.102
=====
Received DHEX request from 192.168.1.102:3283
=====
The negotiated AES128 key is: 0b0ba2c1fe0416434abd826db682fad5
=====
Received credentials:
  Username: isecadmin
  Password: p@ssw0rd
=====
```

Outline

- 1 Motivation
 - Preface and Background
- 2 Anatomy of an APT
 - Social Engineering
 - Initial Exploitation
 - Local Privilege Escalation
 - Network Privilege Escalation
 - **Persistence**
 - Exploration
 - Exfiltration
- 3 Conclusion
 - Summary



Maintaining Access

how to survive the reboot

- Create a hidden startup item
- Com.apple.SystemLoginItems.plist Exploit¹⁷
- Append to existing user startup scripts
- Hidden cronjob or automator script
- Modify existing binaries and services, which breaks signing but is generally not noticed
- Modify kernel extensions or cached extensions
- Persist in firmware

¹⁷http://www.macshadows.com/kb/index.php?title=Com.apple.SystemLoginItems.plist_Exploit

Maintaining Access

Attacking and hiding

- Execute arbitrary shell commands
- Run JavaScript in Safari to manipulate/create webpages in Safari
- Attach folder actions to hide data
- Send file transfer messages to your iChat contacts (may be Adium only)

Maintaining Access

At the network layer

- Issue VPN credentials to maintain foothold
- Issue soft tokens from access server
- Issue certificates
- Create new AD users

The Persistent Attack

Userland rootkits: a history...

- Nemo recreates PTRACE functionality and does great Mach ports research ¹⁸
- Dino publicly releases remotely controllable PoC Mach proxy rootkit¹⁹
- Jonathan Rentzsch creates tools and uses them for “hooking” and “swizzling”: methods of modifying existing binaries in memory or on disc
- Dino and Miller write “Mac Hacker’s Handbook” with excellent illustrative examples of persistent attacks using these techniques²⁰
- More followed

¹⁸nemo, Abusing Mach on Mac OS X. May 2006.

<http://www.uninformed.org/?v=4&a=3&t=pdf>

¹⁹<http://trailofbits.files.wordpress.com/2009/08/advancedmacosxrootkits.pdf>

²⁰C. Miller, D. A. Dai Zovi. Mac Hacker’s Handbook. 2009. pp300–318.

Fighting Persistence

Mac IR

- How do we handle IR on Macs?
- Commercial Products
 - EnCase, BlackLight, FTK
 - All handle standard HFS+ forensics
 - Some claim file hash checking (and fail)
- What's missing?
 - Easy checking of OS integrity
 - Binary and driver signing
 - Memory forensics²¹
- Is all of the system state captured on the HDD?

²¹Volatility <https://www.volatilitysystems.com/default/volatility-is-working-on-it>

Dealing with APT

Mac Hardware Forensics

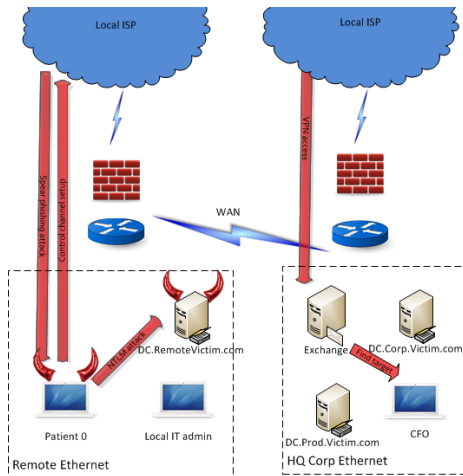


Mac Pro SMC
Firmware Update



Outline

- 1 Motivation
 - Preface and Background
- 2 Anatomy of an APT
 - Social Engineering
 - Initial Exploitation
 - Local Privilege Escalation
 - Network Privilege Escalation
 - Persistence
 - **Exploration**
 - Exfiltration
- 3 Conclusion
 - Summary



Who do you Love?

Are you for sure?

- Pick accounts to attack by examining the Open Directory users, groups, and privileges using unauthenticated ldapsearch
 - Engineers: source code
 - Product Management: release information
 - CFO's office, Controller: Financial data
 - In house counsel: Lawful intercept access
- Account home directories network mounted by default

Accessing Interesting Accounts

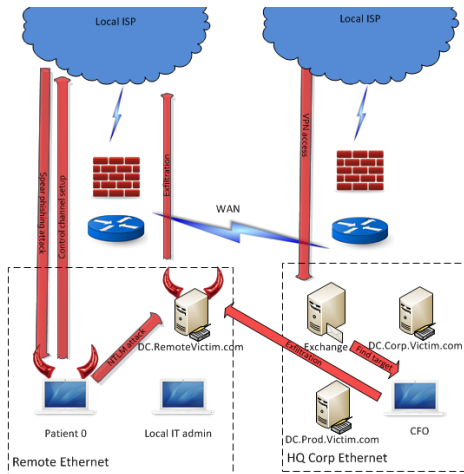
- Least intrusive/high privilege
 - using root privileges export the password directory with mkpassdb
 - mount an off-line brute-force attack on the passwords
 - login as users and access data
- Medium intrusive/high privilege
 - using root privileges copy the password directory
 - reset passwords and access accounts
 - restore previous directory
- Most intrusive/standard admin privilege:
 - change passwords and access accounts
 - run before anyone notices
- Maintain control by cracking more user/VPN credentials or creating new users with VPN access

Making Exploration Harder

- Don't allow server admin accounts to have root access
- Use strong password hash formats
- Regularly review audit logs and set up alerts to track password changes and VPN enrollment

Outline

- 1 Motivation
 - Preface and Background
- 2 **Anatomy of an APT**
 - Social Engineering
 - Initial Exploitation
 - Local Privilege Escalation
 - Network Privilege Escalation
 - Persistence
 - Exploration
 - **Exfiltration**
- 3 Conclusion
 - Summary



The Getaway

- Shawshank-style
 - Identify overseas internal drop server
 - Move data over corporate WAN to internal drop
 - Test for allowed outbound protocols
 - Bulk exfiltration though local office NAT to external drop server
- Covert Channels
 - ICMP
 - HTTPS
- Hide in plain sight²²
- PKI via embedded public keys

²²<http://invisiblethings.org/papers/passive-covert-channels-linux.pdf>

How can we mitigate the exfiltration threat?

Short term

- Coordinated egress restrictions in *all* offices
- DLP & proxy log monitoring
- 24x7 SOC ninjas

How can we mitigate the exfiltration threat?

Long term

- Time to rethink global architecture
 - Leased lines
 - Unified Forest
 - L3 routing directly between offices
- Alternatives
 - ADFS Federated domains
 - WAN accelerators
 - Limited, audited file sync

Outline

1 Motivation

- Preface and Background

2 Anatomy of an APT

- Social Engineering
- Initial Exploitation
- Local Privilege Escalation
- Network Privilege Escalation
- Persistence
- Exploration
- Exfiltration

3 Conclusion

- Summary



Dealing with APT

Comparison with Windows

- In each phase of an APT, how does OS X stack up?
- Assumptions:
 - Windows 7 and 2008R2
 - OS 10.7 Client and Server
 - No mixed environments

Windows vs Mac Comparison

Initial Exploitation:

Windows 7	OS 10.7 Lion	Advantage
Stack Canary	Stack Canary	Tie
Heap Hardening	Heap Hardening	?
Heap and Stack DEP	Heap and Stack NX	Tie
ASLR (32 and 64 bit)	ASLR (32 and 64 bit)	Tie
NT Priv Dropping	Broker service an XPC	OS X
Default all privs	New default sandbox	OS X
Configurable with EMET	Not configurable	Windows

Conclusion: OS X has now equalized anti-exploit technologies with Windows.

Windows vs Mac Comparison

Local Privilege Escalation:

Windows 7	OS 10.7 Lion	Advantage
NT Priv Dropping	Broker service and XPC	OS X
Default all privs	New default sandbox	OS X
UIPI and Secure Desk	Pop-up cred box	Windows
No default cred store	Login Keychain	Windows

Conclusion: Local privilege escalation on both platforms is still quite possible. Everybody loses.

Windows vs Mac Comparison

Network Privilege Escalation:

Windows 2008R2	OS 10.7 Server	Advantage
NTLMv2	Unsigned DH	Windows
Kerberos Only Option	Lots of fallback to DH	Windows
RPC Privacy and Integrity	No central protocol crypto	Windows
RDP with session security	Apple Remote Desktop	Windows
AD DNS with Secure Updates	mDNS	Windows

Conclusion: OS X networks are significantly more vulnerable to network privilege escalation. Almost every OS X Server service offers weak authentication methods allowing downgrade attacks.

Windows vs Mac Comparison

Persistence:

Windows 7	OS 10.7 Lion	Advantage
User-Mode Services	User-Mode Services	Tie
Kernel Rootkits	Kernel Rootkits	Tie
Many disk forensics options	Fewer disk forensics	Windows
Several RAM forensics tools	Almost no RAM forensics	Windows

Conclusion: Persisting malicious code on both platforms is not a problem for APT. Defenders have more options to detect modification of Windows and analyze code, but this need should be slowly met by open-source and commercial tools.

Windows vs Mac Comparison

Exploration and Exfiltration:

Windows 2008R2	OS 10.7 Server	Advantage
AD LDAP locked to unauthed users	Anonymous LDAP browsing	Windows
Configurable outbound FW	No outbound rules	Windows
Central logging requires product	Supports syslog UDP	OS X

Conclusion: These steps are mostly not dependent on the platform, although OpenDirectory can provide a better stepping stone than AD to an unauthenticated user.

Conclusion

Should you use Macs in your Enterprise

- Pros
 - Anti-exploit and sandbox technologies are looking good in 10.7
 - Getting “hacked by accident” is still harder
 - Slightly less body of knowledge in attacker circles
- Cons
 - Network privilege escalation is trivial
 - Local UI isolation allows for easy phishing of admin creds
 - No equivalent of GPO, hard to harden centrally
 - Fewer products to investigate incidents
- Bottom Line: Run your Macs as little islands on a hostile network.

QUESTIONS?

[HTTPS://WWW.ISECPARTNERS.COM](https://www.isecpartners.com)

THANKS TO ASTHA SINGHAL AND ROGER MEYER