# More Apple Sauce

macOS Security for the Windows Blue Team

# Overview

1. Whoami and Origin Story
2. Attack Surface and Prevention
3. Detection and IR
4. Big picture and automation
5. Notes on release, acks

# Whoami

- Reverse engineer by trade
  - Windows malware analyst
  - ICS and embedded devices (IoT)
  - InfoSec jack-of-all-trades
- Day-job
  - Support Blue team, help develop hunting techniques
    - Threat emulation
    - Malware analysis
- Mac security n00b
  - Have some ~~pain~~ “experience” to share
  - Hope this talk sparks interest, discussion about macOS security “gap”

@r3doubt

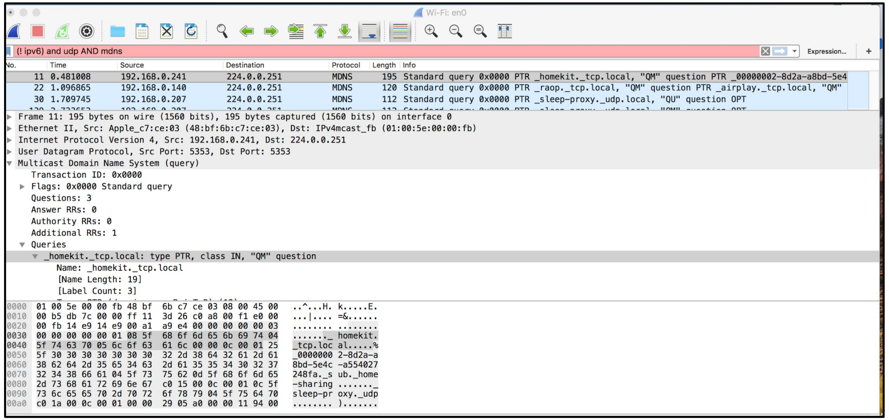
https://github.com/r3doubt/apple-sauce-in-a-bucket

https://blog.r3doubt.io
(tutorials coming soon, really!)

# Why This Talk is For You

- If you are responsible for security in a Windows enterprise environment
    - Or you want to be, someday
    - You have macOS endpoints
    - You aren’t a long-time Unix sysadmin or Apple super-user
    - You want a quickstart guide for hardening, monitoring, and IR
    - You don’t have a huge budget and think free is good
- Jamf 2016 Survey: Managing Apple Devices in the Enterprise
    - 44% offer Mac devices to employees
    - 91% have Mac devices in the enterprise
- Fortune 500s
    - IBM over 100,000, JPMC over 12,000
- Not just the numbers
    - Who has them? Developers, c-suite?
    - What can those accounts access? “Crown jewels”, intellectual property?

# Origin Story



```
(! ipv6) and udp AND mdns

No.  Time      Source         Destination  Protocol Length Info
  11 0.481008  192.168.0.241  224.0.0.251  MDNS     195 Standard query 0x0000 PTR _homekit._tcp.local, "QM" question PTR _00000002-8d2a-a8bd-5e4
  22 1.096865  192.168.0.140  224.0.0.251  MDNS     120 Standard query 0x0000 PTR _raop._tcp.local, "QM" question PTR _airplay._tcp.local, "QM"
  30 1.709745  192.168.0.207  224.0.0.251  MDNS     112 Standard query 0x0000 PTR _sleep-proxy._udp.local, "QU" question OPT

▶ Frame 11: 195 bytes on wire (1560 bits), 195 bytes captured (1560 bits) on interface 0
▶ Ethernet II, Src: Apple_c7:ce:03 (48:bf:6b:c7:ce:03), Dst: IPv4mcast_fb (01:00:5e:00:00:fb)
▶ Internet Protocol Version 4, Src: 192.168.0.241, Dst: 224.0.0.251
▶ User Datagram Protocol, Src Port: 5353, Dst Port: 5353
▼ Multicast Domain Name System (query)
    Transaction ID: 0x0000
  ▶ Flags: 0x0000 Standard query
    Questions: 3
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 1
  ▼ Queries
    ▼ _homekit._tcp.local: type PTR, class IN, "QM" question
        Name: _homekit._tcp.local
        [Name Length: 19]
        [Label Count: 3]

0000  01 00 5e 00 00 fb 48 bf  6b c7 ce 03 08 00 45 00   ..^...H. k.....E.
0010  00 b5 db 7c 00 00 ff 11  3d 26 c0 a8 00 f1 e0 00   ...|.... =&......
0020  00 fb 14 e9 14 e9 00 a1  a9 e4 00 00 00 00 00 03   ........ ........
0030  00 00 00 00 00 01 08 5f  68 6f 6d 65 6b 69 74 04   ......._ homekit.
0040  5f 74 63 70 05 6c 6f 63  61 6c 00 00 0c 00 01 25   _tcp.loc al.....%
0050  5f 30 30 30 30 30 30 30  32 2d 38 64 32 61 2d 61   _0000000 2-8d2a-a
0060  38 62 64 2d 35 65 34 63  2d 61 35 35 34 30 32 37   8bd-5e4c -a554027
0070  32 34 38 66 61 04 5f 73  75 62 0d 5f 68 6f 6d 65   248fa._s ub._home
0080  2d 73 68 61 72 69 6e 67  c0 15 00 0c 00 01 0c 5f   -sharing ......._
0090  73 6c 65 65 70 2d 70 72  6f 78 79 04 5f 75 64 70   sleep-pr oxy._udp
00a0  c0 1a 00 0c 00 01 00 00  29 05 a0 00 00 11 94 00   ........ ).......
```

- Investigating Suricata “RPC portmapper” alert
  - rpcinfo query, ps aux,  packet captures
  - Lots of noise
    - WTF is Bonjour?
- Helpdesk was issuing with default configs
    - Is there a STIG?
- Found hardening guides geared towards super-user
  - Manually managing single system
  - Often not updated
  - Focus on privacy stuff
- First effort bash scripted top recommendations from NSA tri-fold on Snow Leopard
- Idea was to push hardening scripts and security tools via configuration management tools
  - Then I learned about SIP

# macOS Internals and Security “Features”

# Fruitfly Malware

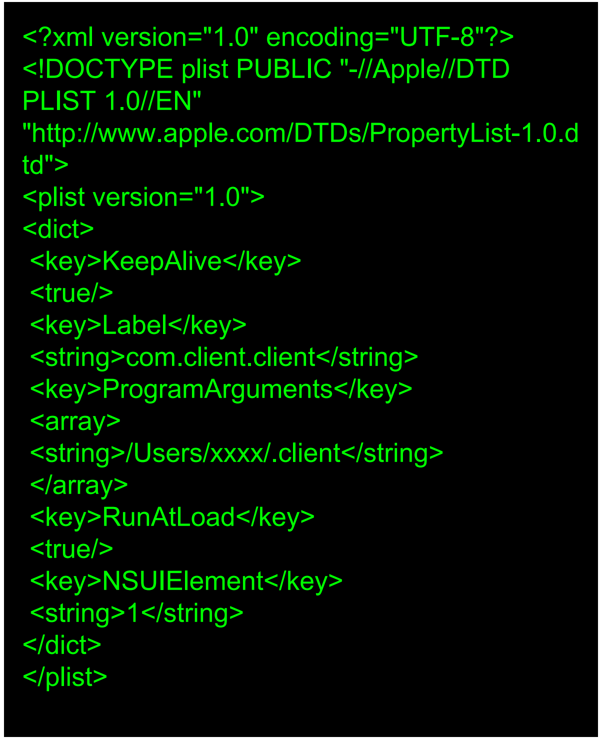
- Infosec blogs originally attributed to unknown APT
  - Two variants, one possible copycat (.B)
- Written by Phillip Durachinsky of North Royalton, OH
  - 16 counts of Computer Fraud and Abuse Act violations, Wiretap Act violations, production of child pornography and aggravated identity theft
- Targeted home users, private enterprises, universities, police department, US Department of Energy for 13+ years
- Obfuscated Perl script
  - Remote administration tool (RAT), keylogging, screen, audio, and webcam capture
  - Wrote embedded Mach-O binary for certain features
- Additional component performed network reconnaissance via mDNS
- Persistence via launchd

# Persistence with launchd and Property Lists

- macOS startup
  - Extensible Firmware Interface (EFI) bootloader boot.efi handles disk encryption and loads kernel environment
  - Passes to kernel_task (PID 0)
  - Which runs /sbin/launchd (PID 1)
- Launchd examines following directories and parses any .plist found
  - /Library/LaunchAgents/
    /Library/LaunchDaemons/
    /System/Library/LaunchAgents/
    /System/Library/LaunchDaemons/
    /User/username/Library/LaunchAgents/
- Property Lists or .plist
  - “Representation of a hierarchy of objects that can be stored in the file system and reconstituted later.” -developer.apple.com
  - Typically xml file in the application’s “bundle”, used for configuration
    - Bundle contains compiled executable and similar data to resources section of a PE file
- .plist in a /launchdaemon directory
  - Used by launchd for configuring a service that runs on startup

# Fruitfly .plist

- RunAtLoad key
  - Similar idea to Run / Run Once on Windows registry
    - HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run
- KeepAlive key ensures the service is running
- NSUIElement
  - Hides Dock icon
- /Users/xxxx/.client
  - Path to obfuscated malicious Perl script
  - Contains main RAT functionality
- Audit launchdaemon .plist files
  - Good way to find low-hanging “fruit”
  - “sudo launchctl list”

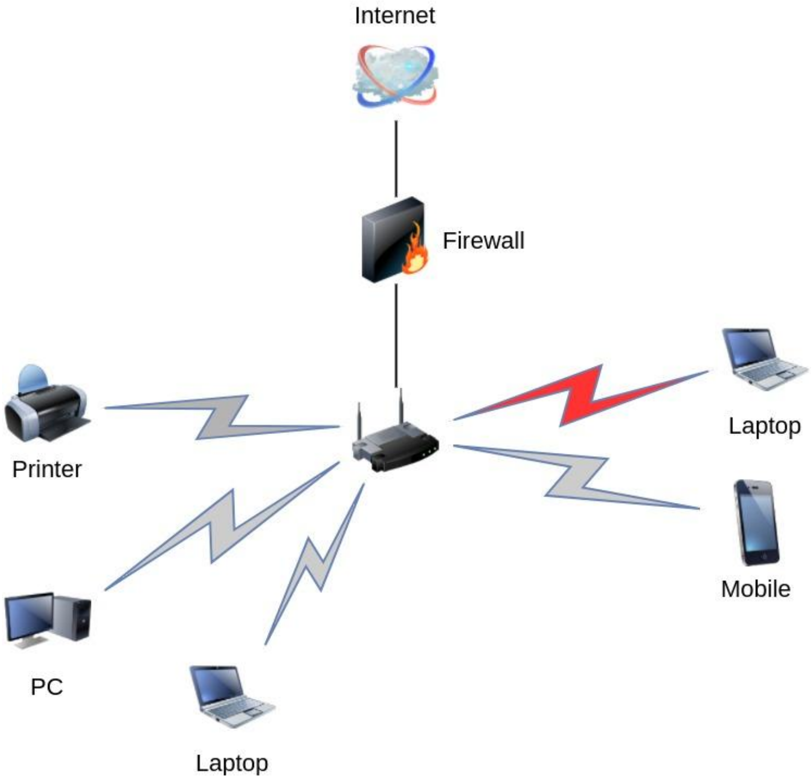


```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
 <key>KeepAlive</key>
 <true/>
 <key>Label</key>
 <string>com.client.client</string>
 <key>ProgramArguments</key>
 <array>
 <string>/Users/xxxx/.client</string>
 </array>
 <key>RunAtLoad</key>
 <true/>
 <key>NSUIElement</key>
 <string>1</string>
</dict>
</plist>
```

# Bonjour means “hello” to malware

- Fruitfly used network mapping script leveraging mDNS
  - Assumed mDNS would be available and allowed through firewall
  - Bonjour (mDNSResponder) enabled by default
- Bonjour on macOS provides automatic discovery and resolution of network resources on the local link (.local TLD)
- DNS-SD and mDNS used for “zero-configuration networking” a.k.a. ZeroConf
  - Multicast DNS (RFC 6762) (mDNS) allows DNS queries to be resolved over an IP multicast without a conventional DNS resolver (e.g. BIND 9)
  - DNS Service Discovery (RFC 6763) (DNS-SD) allows clients to discover a list of named instances of a desired service, using standard DNS queries

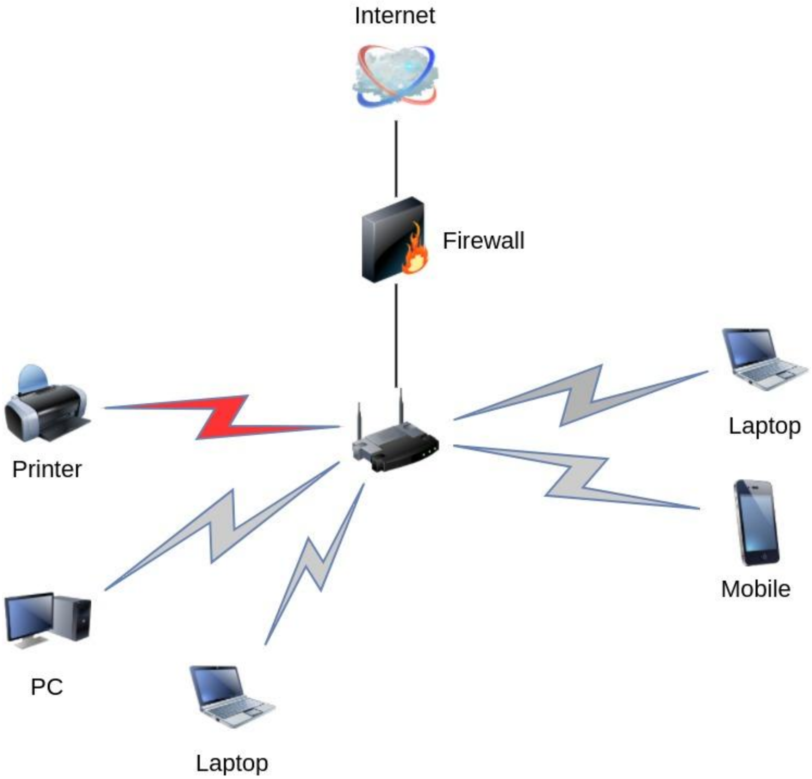
# Bonjour Print Spooler Service Example



Host sends multicast to IP 224.0.0.251 UDP port 5353 with query

“Who has _printer services?”

# Bonjour Print Spooler Service Example



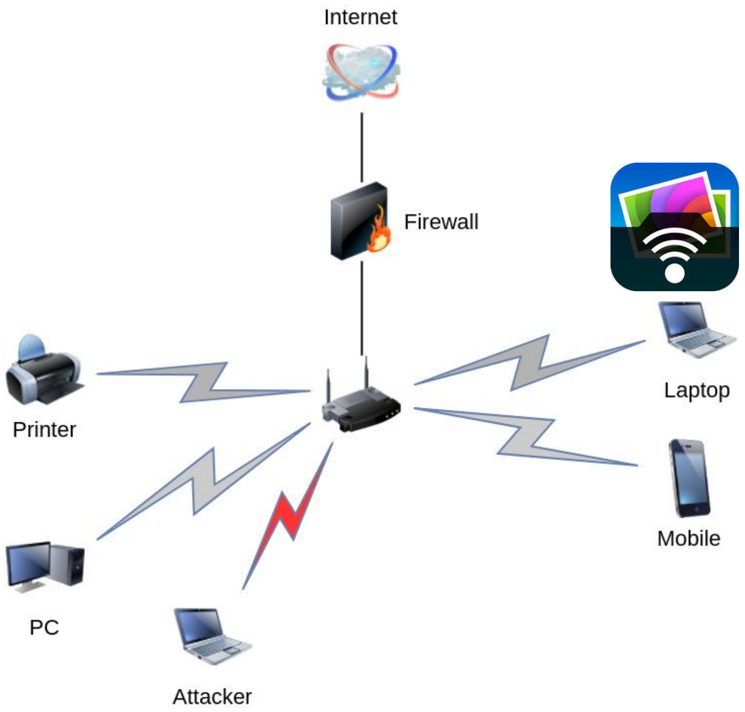
- Printers respond with PTR records of instances, print spooler named SirPrintsAlot
    - “_printer._tcp.local. 28800 PTR PrintsAlot._printer._tcp.local.”
- Query for SirPrintsAlot returns SRV record results with hostname
    - “PrintHost.local”
- Query for hostname returns A record for host.local with IP address and port
    - “192.168.0.4:9100”
- Can also send optional TEXT type

# Bonjour Man-In-The-Middle

- For some service types instance names are temporarily stored (example, printers)
- For some applications (example PhotoSync) discovery and resolution repeated each time
- Malicious devices can broadcast to claim ip hostname and service instance name
  - No authentication mechanism exists, this is a feature not a bug
- So you can man-in-the-middle to steal my file I tried to print, but you can’t get remote code execution, right?
  - Some services automatically do things like transfer and sync  files

# PhotoSync and CoreGraphics

- Bonjour MitM for PhotoSync (Bai/Xing, Black Hat USA 2016)
- PhotoSync application
  - Popular for transferring photos between iPhone and Macs
- CoreGraphics
  - “Handle path-based drawing, anti-aliased rendering, gradients, images, color management, PDF documents, and more.” -developer.apple.com
- CVE 2016-4673 (CoreGraphics)
  - Crafted JPEG files remote attacker executes arbitrary code
  - multiple similar vulnerabilities that have been discovered and patched
- So, Bonjour + PhotoSync could give us RCE



# Block Multicast From mDNSResponder Service

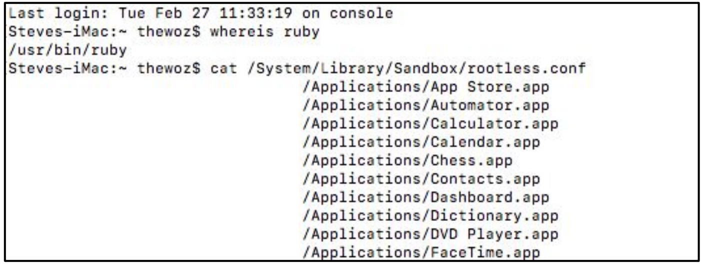
- Can try blocking via PF firewall (UDP port 5353)
  - IPv4 224.0.0.251
  - IPv6 FF02::FB
- Normally we can disable default apps by unloading and then moving the .plist
- Bonjour can be made “safer” by turning off multicast option for mDNSResponder
- launchctl utility provides editing for launchd .plist files

```
#!/bin/bash
#Remove multicast from Bonjour/Zeroconf mDNSResponder service
launchctl unload /System/Library/LaunchDaemons/com.apple.mDNSResponder.plist
launchctl unload /System/Library/LaunchDaemons/com.apple.mDNSResponderHelper.plist
defaults write /System/Library/LaunchDaemons/com.apple.mDNSResponder.plist \
ProgramArguments -array-add "-NoMulticastAdvertisements"
launchctl load /System/Library/LaunchDaemons/com.apple.mDNSResponder.plist
launchctl load /System/Library/LaunchDaemons/com.apple.mDNSResponderHelper.plist

#Hey, I used sudo, what happened?  Oh yeah, SIP
```

# System Integrity Protection (SIP)

- “Rootless mode” to limit damage if (when) attacker gets root
  - Limits actions even root user can perform
  - sudo won’t make me a sandwich
  - Check if on with csrutil status
  - Attacker would never get root, right? In a minute…
- File system protection for certain files and directories
  - Example /System, /sbin, /bin, pre-installed apps
  - Exceptions like /System/Library/Caches
  - Details in  /System/Library/Sandbox/rootless.conf
  - /user/local, /Applications, [~]/Library intended for developers
- Run Time Protections for protected processes
  - dtrace prevented from attaching to system processes
  - Dynamic link editor DYLD_ environment variables purged
    - Example DYLD_LIBRARY_PATH
- Kernel extensions (.kext) must be signed with special developer ID
  - Install to /Library/Extensions directory
  - This is another common persistence mechanism

```
#Posix  file system permissions
#Check individual file protection
ls -la
#or
xattr -l <filename>
drwxr-xr-x@ 4 root wheel restricted 136 2 Jan 13:03 System
# the @ symbol indicates extended attributes
# ‘restricted’ indicates a SIP protected binary
```



```
Last login: Tue Feb 27 11:33:19 on console
Steves-iMac:~ thewoz$ whereis ruby
/usr/bin/ruby
Steves-iMac:~ thewoz$ cat /System/Library/Sandbox/rootless.conf
                                /Applications/App Store.app
                                /Applications/Automator.app
                                /Applications/Calculator.app
                                /Applications/Calendar.app
                                /Applications/Chess.app
                                /Applications/Contacts.app
                                /Applications/Dashboard.app
                                /Applications/Dictionary.app
                                /Applications/DVD Player.app
                                /Applications/FaceTime.app
```

# Disable SIP with CSRUTIL

- Protections can be turned off with csrutil
  - Can be disabled entirely, or only certain protections
  - Can only disable from Recovery Mode
- Presents a challenge for remote administration
  - No current open-source tools to disable (user space)
  - Old hacks like RootFool .kext no longer work
  - Subverting OS security features bad policy, CYA
  - I have suggestions, but can’t give 100% answer
- Apple updates and other components bypass
  - Profile Manager on macOS Server
  - Third-Party Mobile Device Management (MDM) like Jamf with Apple issued cert
  - More on this later

```
#Restart and hold
COMMAND + R
#then utilities-->terminal
csrutil disable
# or <enable>  with options
--without kext
--without fs
--without debug
--without dtrace

#Clear current options
csrutil clear
reboot
```

# ‹blank› Gets You Root

- Why not just turn-off SIP?
- Vulnerability in High Sierra allowed anyone to login as root
  - Could enter blank password
  - Could make up a password
- Flaw in macOS API (com.apple.loginwindow)
  - Else if statement for failed return of shadow hash
  - Login failed, but hash for entered password attempt saved
  - Next login with same user and password combo works
  - Root disabled by default, so no shadow hash to return
- Fixed by patch
  - Temporary fix was enabling root user with a password
  - When update for Sierra to 10.13.1 broke again, (sad trombone)

# GateKeeper, Xprotect, and Sandbox

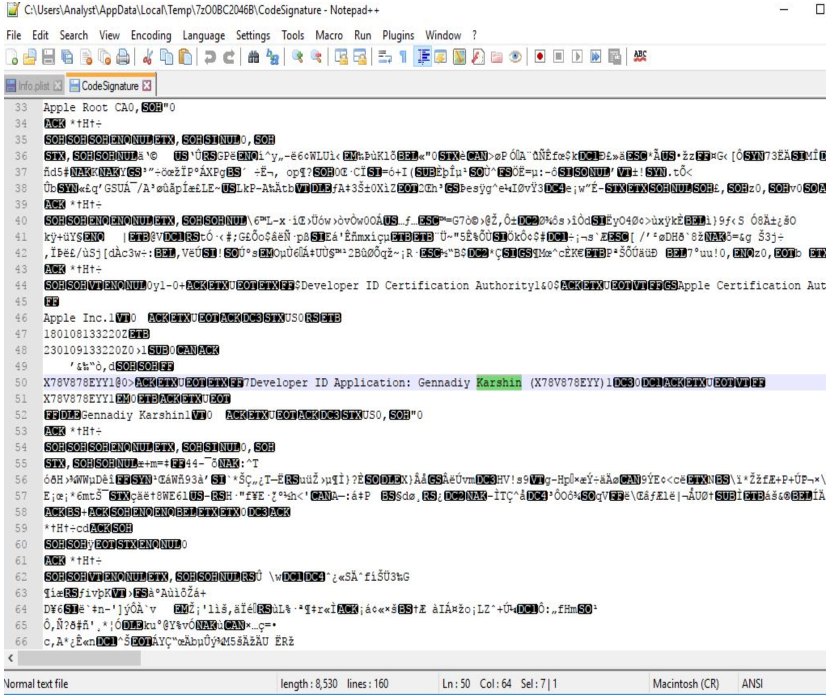
- Gatekeeper
    - Prevents install of untrusted apps
    - Settings defined by a policy
        - Apple signed
        - Signed third-party
        - Other apps
- XProtect
    - Apple's built-in "AV" protection (for downloads)
- Per application Sandbox

# Now the Bad News...

- GateKeeper
  - Only applies to “quarantine” files
  - User bypass, right-click
  - spctl disables
  - Quarantine can be removed
- XProtect
  - Only protects on downloads
  - Very simple hash and string defs
- Sandbox
  - Only applies to mac app store apps
- Apple signing can be abused
  - See Xorrior and Objective-See blogs
  - Fake or compromised dev accounts too

```
#To turn-off GateKeeper controls
sudo spctl --master-disable

#Let’s just remove that quarantine tag
sudo xattr -r -d com.apple.quarantine /path
```

# Shlayer, about that code signing...



C:\Users\Analyst\AppData\Local\Temp\7zO0BC2046B\CodeSignature - Notepad++

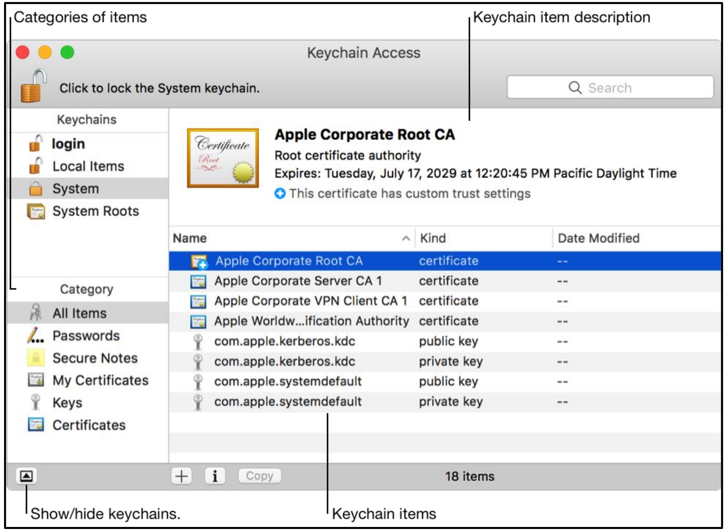
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
Info.plist
CodeSignature

```
Apple Root CA0,SOH"0
ACK *†H†÷
SOHSOHSOHENQNULETX,SOHSINUL0,SOH
STX,SOHSOHNULä'©  US'ÛRSGPëENQi^y„–ë6¢WLUi‹EM‰ÞùKl õBEL«"0STXèCAN>øP ÓÌA¨ûÑÊfœ$kDC1Ð£»äESC'ÂUS•žzFF¤G‹ [ÔSYN73ËÄSIMÍD
ñd5‡NAKKNAKYGS³"÷öœžÏP°ÁXPgBS´ ÷Ë¬, op¶?SOH0Œ·CÏSI=ó+I(SUBÈþÎµ¹SOÙ^FSÖË=µ:–õSISONUL'VT±!SYN.tÕ<
ÛbSYN«£q'GSUÄ¯/A³øûãpÍæ£LE~USLkP-A‰Ätb VTDLEfA‡3Š±0XìZEOT2Œh³GSÞesÿg^e¼IØvŸ3DC4e¡w"É-STXETXSOHNULSOH£,SOHz0,SOHv0SOA
ACK *†H†÷
SOHSOHENQENQNULETX,SOHSOHNUL\6™L-x·iŒ›Üów›òvÒw0OÂUS…f…ESC™=G7ò©›@Ž,Ô±DC2Ø¾ôs›ìÒdSIËyO4Ø¢>ùxÿkËBELì}9f‹S Ó8Ä±¿š0
kÿ+üY§ENQ   |ETB@VDC1RStÓ·‹#;G£Õo$âëÑ·pßSIEá'Êñmxíçµ ETBETB¨Ü~"5Ê%ÕÙSIÖkÔ¢$#DC1÷¡¬s`ÆESC[ /'²øDHð`8žNAKõ=&g Š3j÷
,ÏÞë£/ùSj[dÀc3w÷:BEL,VëÚSI!SOÚ°sEMOµÙ6Á‡UÙ§™¹2BûØÕqž~¡R·ESC‰"B$DC2*ÇSIGS¶Mœ^cÈK€ETBPª ŠÕÚäüÐ  BEL7°uu!0,ENQz0,EOTb ETX
ACK *†H†÷
SOHSOHVTENQNUL0yl-0+ACKETXUEOTETXFF$Developer ID Certification Authority1&0$ACKETXUEOTVTFFGSApple Certification Aut
FF
Apple Inc.1VT0  ACKETXUEOTACKDC3STXUS0RSETB
180108133220ZETB
230109133220Z0›1SUB0CANACK
    '&‰"ò,dSOHSOHFF
X78V878EYY1@0>ACKETXUEOTETXFF7Developer ID Application: Gennadiy Karshin (X78V878EYY)1DC30DC1ACKETXUEOTVTFF
X78V878EYY1EM0ETBACKETXUEOT
FFDLEGennadiy Karshin1VT0  ACKETXUEOTACKDC3STXUS0,SOH"0
ACK *†H†÷
SOHSOHSOHENQNULETX,SOHSINUL0,SOH
STX,SOHSOHNULæ+m=‡FF44-¯õNAK:^T
óðH›¾WWµDêìFFSYN¹ŒáWñ93à'SI`*ŠÇ„¿T–ËRSuüŽ›µ¶Ì}?ÈSODLEX}ÂåGSÂëÚvmDC3HV!s9VTg-HpÌ×æÝ÷äÄöCAN9ÝE¢<cëETXNBS\ì*ŽžfÆ+P+ÚP¬×\
E¡œ¡*6mtŠ¯STXçäë†8WE61US-RSH·"f¥E·¿°¼h<'CANA–:á‡P  BS$dø¸RS¿DC2NAK-ÌTÇ^åDC4³Ôõõ¾SOqVFFè\ŒáfÆlë|¬ÅUØ†SUBÌETBáš&®BELÍÀ
ACKBS+ACKSOHENQENQBELETXETX0DC3ACK
*†H†÷cdACKSOH
SOHSOHÿEOTSTXENQNUL0
ACK *†H†÷
SOHSOHVTENQNULETX,SOHSOHNULRSÛ \wDC1DC4^¿«SÄ^fíŠÜ3‰G
¶iæRSfivþKVT›FSà°AûìõŽá+
D¥6SIë`‡n-']ýÔÀ`v  EMŽ¡'lìš,äÏéÏRSùL%·ª¶‡r«ÌACK¡á¢«×šBS†Æ àIÁ¤žo¡LZ^+Ù¼DC1Ô:„fHmSO¹
Ô,Ñ?ð‡ñ',*¦ÔDLEku°@Y%vÔNAKùCAN×…ç=•
c,A*¿Ê«nDC1^ŠEOTÁYÇ"œÅbµÛý¾M5šÄžÄU ËRž
```

Normal text file
length : 8,530   lines : 160
Ln : 50   Col : 64   Sel : 7 | 1
Macintosh (CR)
ANSI

- Combination Mach-O and shell scripts
- Observed dropping adware
  - Bundlore
  - MacOffers
- Three variants observed
  - All three used “valid” developer code signing
- Developer Program accounts
  - "Harper Natalie"
  - "Murphy Rachel"
  - "Gennadiy Karshin

# macOS Keychain Services



**Apple Corporate Root CA**
Root certificate authority
Expires: Tuesday, July 17, 2029 at 12:20:45 PM Pacific Daylight Time
This certificate has custom trust settings

| Name | Kind | Date Modified |
|---|---|---|
| Apple Corporate Root CA | certificate | -- |
| Apple Corporate Server CA 1 | certificate | -- |
| Apple Corporate VPN Client CA 1 | certificate | -- |
| Apple Worldw...ification Authority | certificate | -- |
| com.apple.kerberos.kdc | public key | -- |
| com.apple.kerberos.kdc | private key | -- |
| com.apple.systemdefault | public key | -- |
| com.apple.systemdefault | private key | -- |

18 items

- Centrally store passwords, certificates, pins and other information
  - Idea is the same as other password management tools (LastPass, etc)
  - Can be local or via iCloud instance across devices
- By default all applications
  - Use single instance of keychain
  - Primary password set to user account password

# “Mr. Steal Yo Keychain” Vulnerability

- Discovered by Patrick Wardle in 2017
    - Objective-See blog
- Arbitrary applications could access data for other applications
- Could be leveraged to dump macOS Keychain
- Affected El Capitan (10.11) through High Sierra (10.13)
    - Not discovered until 10.13
- Worked for both signed (Apple) and unsigned applications
- Leveraged the macOS API available to developers
    - Apparently, not a “bug”, buffer overflow, null pointer, etc
- Not the only vulnerability discovered for Keychain
    - 'CVE-2015-5943' for example
- Keychain targeted by various Mac malware including Proton, Dok
    - Just used social engineering
- Requires Keychain to be unlocked
    - Keychain is unlocked by default when user logs in

# Securing Keychain Services

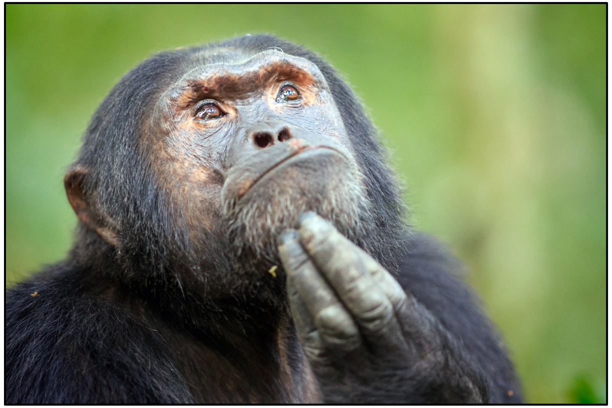
- NIST SP-800-179 Recommendations
  - Change default primary password
  - Create a separate keychain for sensitive information
    - This should stay locked unless needed
  - Set Keychain to “lock when sleeping”
- Other Recommendations
  - Delete local keychain logout
    - Keychain can create conflict with AD
- Turn off iCloud sign-in and iCloud Keychain
  - Edit com.apple.security.keychainsyncingoveridsproxy.osx.plist with launchctl
  - Profile Manager settings (more later on this)
  - Still testing this, will include options in scripts
  - iCloud is whole other talk

```
#remove local keychains
#!/bin/sh
rm -Rf /Users/$DUDE/Library/Keychains/*

#add logout hook
sudo defaults write com.apple.loginwindow \
LogoutHook /Library/foo.sh
```

# Part Two Detection and IR

# Endpoint Monitoring and IR Considerations

- Assume attacker gains a foothold
- Logistically feasible solution
  - Manageable volume of data
  - Acceptable resource consumption
  - Tunable for your environment
- Fits our policy constraints
  - Depends on the latitude you have to install stuff on endpoint
  - Something might already be imposed by existing policy
    - Example, standard AV product or proxy client for Mac
- Open-source, free
- Tools for alerting, hunting, and IR (after the fact)
- Within our level of effort
  - Some open-source options might be too DIY

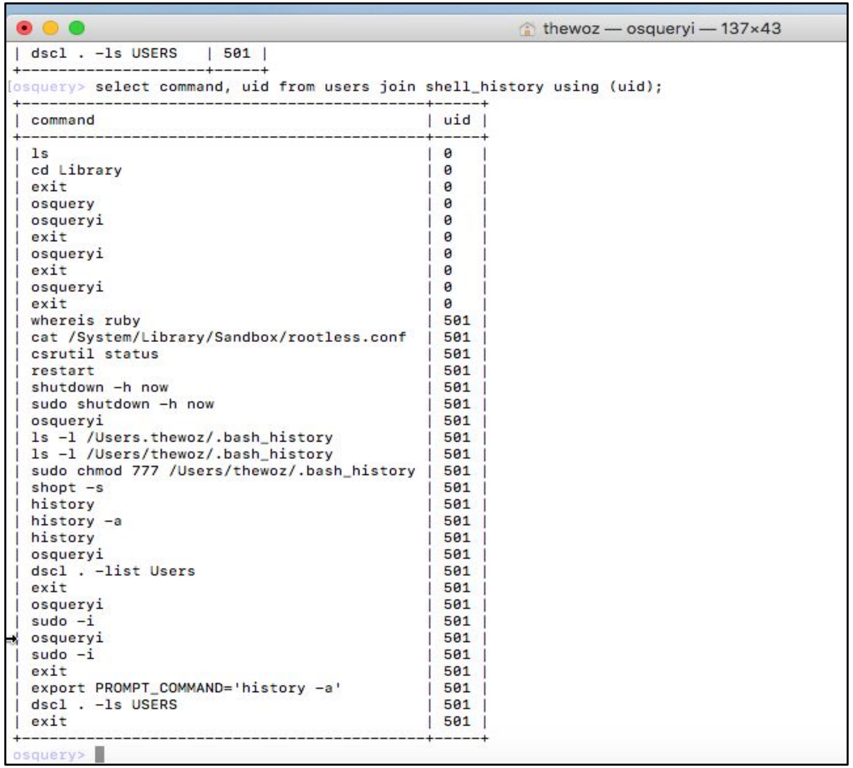


# OSQuery Overview

- By Facebook
- Leverages macOS api used by native logs
- Uses SQL style queries
- Queries imported as query packs
  - Contributed queries specifically for macOS threats
  - Create your own
  - Leverage other formats, example Yara rules
- Monitor for detection or use for IR
  - Rocksdb updated per change, regular interval, osqueryd
  - Can be used on the fly, interactively, osqueryi
- Push events to almost any solution you use for SIEM (HELK is nice)

# OSQuery Basics

- File Integrity Monitoring
  - Access, modify, create, delete
  - Leverages inotify and FSEvents
  - Include or exclude files
- Process monitoring
  - Leverages audit
  - Must be configured
- Mac Syslog
  - On by default
  - But uses asl.conf configs
- Yara rules
  - Applied on interval or once
- Export to SIEM
  - Logstash
  - Kafka
- Remote administer
  - Kolide Fleet, others



```
| dscl . -ls USERS   | 501 |
+--------------------+-----+
osquery> select command, uid from users join shell_history using (uid);
+--------------------------------------------+-----+
| command                                    | uid |
+--------------------------------------------+-----+
| ls                                         | 0   |
| cd Library                                 | 0   |
| exit                                       | 0   |
| osquery                                    | 0   |
| osqueryi                                   | 0   |
| exit                                       | 0   |
| osqueryi                                   | 0   |
| exit                                       | 0   |
| osqueryi                                   | 0   |
| exit                                       | 0   |
| whereis ruby                               | 501 |
| cat /System/Library/Sandbox/rootless.conf  | 501 |
| csrutil status                             | 501 |
| restart                                    | 501 |
| shutdown -h now                            | 501 |
| sudo shutdown -h now                       | 501 |
| osqueryi                                   | 501 |
| ls -l /Users.thewoz/.bash_history          | 501 |
| ls -l /Users/thewoz/.bash_history          | 501 |
| sudo chmod 777 /Users/thewoz/.bash_history | 501 |
| shopt -s                                   | 501 |
| history                                    | 501 |
| history -a                                 | 501 |
| history                                    | 501 |
| osqueryi                                   | 501 |
| dscl . -list Users                         | 501 |
| exit                                       | 501 |
| osqueryi                                   | 501 |
| sudo -i                                    | 501 |
| osqueryi                                   | 501 |
| sudo -i                                    | 501 |
| exit                                       | 501 |
| export PROMPT_COMMAND='history -a'         | 501 |
| dscl . -ls USERS                           | 501 |
| exit                                       | 501 |
+--------------------------------------------+-----+
osquery>
```

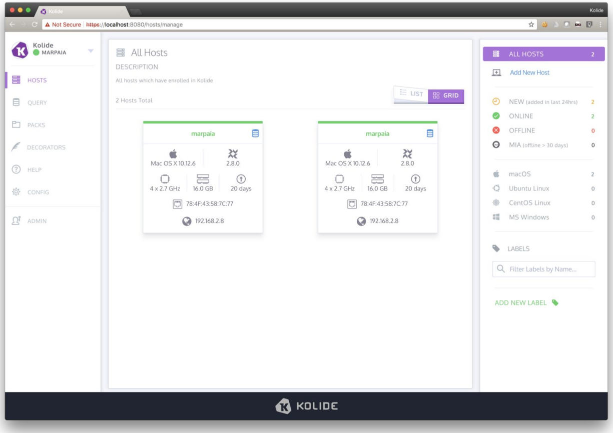
# Fruitfly Detection Redux

- Manually search for “suspicious” plist with launchctl
- Use osx-attacks query pack
  - Detects on static information
  - Log changes to launchd and look for anomalies
- Incident response query pack
  - Good for IR
  - Also good for anomaly detection
  - Focus on weirdness and change
- For IR team maybe use Kolide Fleet

```
#Good

 "OSX_FruitFly": {
    "query" : "select * from launchd where name = 'com.client.client.plist';",
    "interval" : "3600",
    "version" : "1.4.5",
    "description" : "FruitFly OSX Malware (https://blog.malwarebytes.com/threat-analysis/2017/01/new-mac-backdoor-using-antiquated-code/)",
    "value" : "Artifacts created by this malware"

#Better

SELECT path, label, program_arguments, inetd_compatibility, root_directory
  FROM launchd;
```

# Syslog, Audit, Santa, Kolide Fleet

- Santa
  - From Google
  - Say “binary whitelisting and blacklisting”, but for IR
  - Reminds me of Proc Explorer, sort-of
    - Example, get process tree for a process
  - Michael George talk from DerbyCon 2017
    - “macOS Monitoring the Open Source Way”
- Syslog and Audit
  - Need audit reduce to convert to xml etc
  - Requires more knowledge effort to setup, but…
  - More data
- Kolide Fleet
  - IR follow up and investigation
  - w/ ELK or similar for hunting



Not Secure https://localhost:8080/hosts/manage

All Hosts

DESCRIPTION

All hosts which have enrolled in Kolide

2 Hosts Total

| marpaia | marpaia |
|---|---|
| Mac OS X 10.12.6 | Mac OS X 10.12.6 |
| 2.8.0 | 2.8.0 |
| 4 x 2.7 GHz | 4 x 2.7 GHz |
| 16.0 GB | 16.0 GB |
| 20 days | 20 days |
| 78:4F:43:58:7C:77 | 78:4F:43:58:7C:77 |
| 192.168.2.8 | 192.168.2.8 |

| | |
|---|---|
| ALL HOSTS | 2 |
| NEW (added in last 24hrs) | 2 |
| ONLINE | 2 |
| OFFLINE | 0 |
| MIA (offline > 30 days) | 0 |
| macOS | 2 |
| Ubuntu Linux | 0 |
| CentOS Linux | 0 |
| MS Windows | 0 |

# Hidden Lotus Malware, because “APT”

- FireEye calls “APT 32” (Ocean Lotus Group)
  - @ItsReallyNick (Nick Carr) blog and reports
  - Campaign against companies with business interests in Vietnam, overseas dissidents
- Qihoo 360 attributes campaign against Chinese maritime related targets
- Several variants (and generations) of macOS specific malware
  - Not ported version of Windows RAT etc
  - Have evolved over several versions since 2015
  - Also used COTS red-team tools including Cobalt Strike
- Ocean Lotus
  - Spear-phishing delivery of malicious documents with macros (OceanLotus)

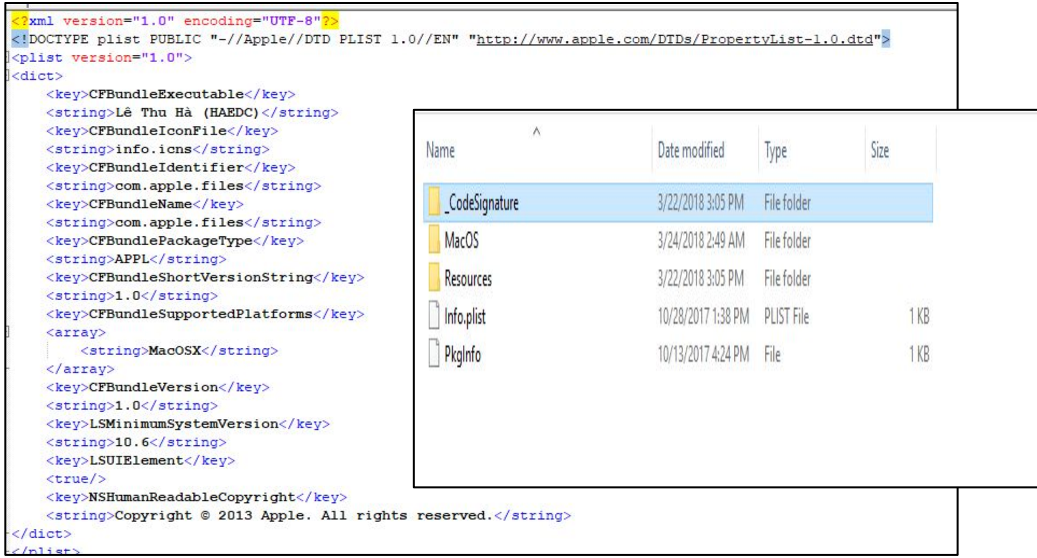
# Hidden Lotus OSQuery Detection

```
"OSX_HiddenLotus": {
    "query" : "select * from launchd where
name = 'com.apple.hidd.shared.plist';",
    "interval" : "3600",
    "version" : "1.4.5",
    "description" : "Apple added XProtect rules
for this sample:
(https://www.virustotal.com/en/file/f2618159
05e77eebdb5c4ec06a7acdda7b68644b1f5155
049f133be866d8b179/analysis/1509567775/)"
,
    "value" : "Artifacts created by this
malware"
```

- Most recent variant attributed to “APT32”
  - Qihoo 360 first reported
  - Employs malicious “PDF”
- macOS Attacks Query Pack now has rules
  - Based on static plist indicator
- Homework assignment
  - Write better rule!

# Hidden Lotus Bundle

- Just an archive
- Contains
  - Code signature
  - Mach-O binary
  - Resources
  - Bundle Plist
  - PkgInfo
- Was named with .pdf
  - “d” was a roman numeral
  - Launcher ignored extension



```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
    <key>CFBundleExecutable</key>
    <string>Lê Thu Hà (HAEDC)</string>
    <key>CFBundleIconFile</key>
    <string>info.icns</string>
    <key>CFBundleIdentifier</key>
    <string>com.apple.files</string>
    <key>CFBundleName</key>
    <string>com.apple.files</string>
    <key>CFBundlePackageType</key>
    <string>APPL</string>
    <key>CFBundleShortVersionString</key>
    <string>1.0</string>
    <key>CFBundleSupportedPlatforms</key>
    <array>
        <string>MacOSX</string>
    </array>
    <key>CFBundleVersion</key>
    <string>1.0</string>
    <key>LSMinimumSystemVersion</key>
    <string>10.6</string>
    <key>LSUIElement</key>
    <true/>
    <key>NSHumanReadableCopyright</key>
    <string>Copyright © 2013 Apple. All rights reserved.</string>
</dict>
</plist>
```

| Name | Date modified | Type | Size |
|---|---|---|---|
| _CodeSignature | 3/22/2018 3:05 PM | File folder | |
| MacOS | 3/24/2018 2:49 AM | File folder | |
| Resources | 3/22/2018 3:05 PM | File folder | |
| Info.plist | 10/28/2017 1:38 PM | PLIST File | 1 KB |
| PkgInfo | 10/13/2017 4:24 PM | File | 1 KB |

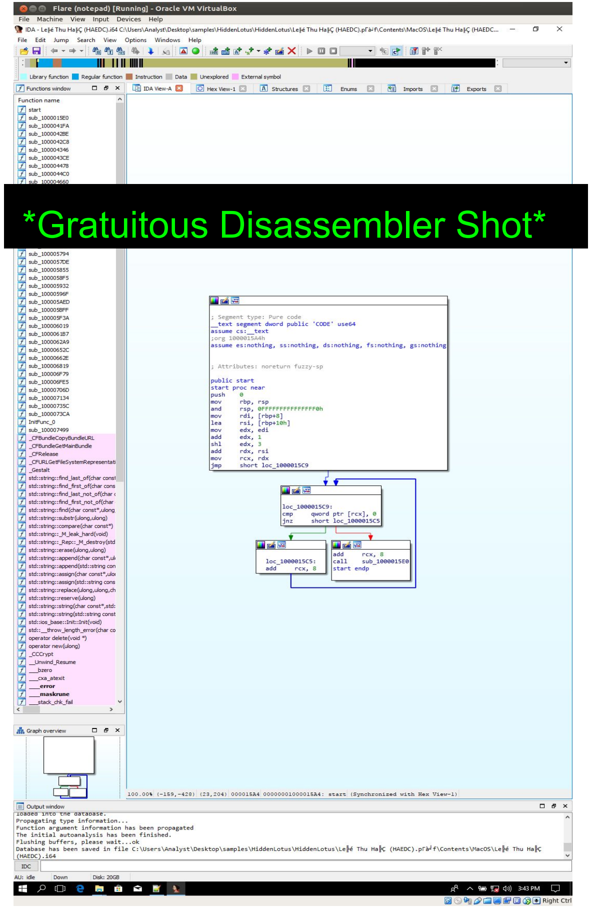
# Mach-O

- Binary format for macOS executables typically .o, .dylib (think .exe, .dll)
- Compiled objective C
  - Can be for different chipsets
- dot Oooh yeah!

| | |
|---|---|
| Header | |
| Load Commands | Table of Contents for segments |
| __PAGEZERO | Reserved (null pointer dereference protect) |
| __TEXT | Executable code |
| __DATA | Readable Writeable data Globals |
| __LINKEDIT | Reserved for DYLD use |

# Mach-O Analysis

- Object Tool cli tool
  - otool -options /file
  - Mach-O binaries explore bundle, loaded .dylibs
- lldb debugger for macOS
- Disassemblers
  - I have used IDA Free 7
  - Cutter (Radare 2 GUI)
- dtrace
- Sure there are more



*Gratuitous Disassembler Shot*

Flare (notepad) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
IDA - Le╠é Thu Ha╠Ç (HAEDC).i64 C:\Users\Analyst\Desktop\samples\HiddenLotus\HiddenLotus\Le╠é Thu Ha╠Ç (HAEDC).pΓà┘f\Contents\MacOS\Le╠é Thu Ha╠Ç (HAEDC...
File Edit Jump Search View Options Windows Help
Library function
Regular function
Instruction
Data
Unexplored
External symbol
Functions window
IDA View-A
Hex View-1
Structures
Enums
Imports
Exports
Function name
Graph overview
Output window
IDC
AU: idle
Down
Disk: 20GB
3:43 PM
Right Ctrl

```
start
sub_1000015E0
sub_1000041FA
sub_1000042BE
sub_1000042C8
sub_100004346
sub_1000043CE
sub_100004478
sub_1000044C0
sub_100004660
sub_100005794
sub_1000057DE
sub_100005855
sub_1000058F5
sub_100005932
sub_10000596F
sub_100005AED
sub_100005BFF
sub_100005F3A
sub_100006019
sub_1000061B7
sub_1000062A9
sub_10000652C
sub_10000662E
sub_100006819
sub_100006F79
sub_100006FE5
sub_10000706D
sub_100007134
sub_10000735C
sub_1000073CA
InitFunc_0
sub_100007499
_CFBundleCopyBundleURL
_CFBundleGetMainBundle
_CFRelease
_CFURLGetFileSystemRepresentati
_Gestalt
std::string::find_last_of(char const
std::string::find_first_of(char cons
std::string::find_last_not_of(char c
std::string::find_first_not_of(char
std::string::find(char const*,ulong,
std::string::substr(ulong,ulong)
std::string::compare(char const*)
std::string::_M_leak_hard(void)
std::string::_Rep::_M_destroy(std
std::string::erase(ulong,ulong)
std::string::append(char const*,ul
std::string::append(std::string con
std::string::assign(char const*,ulo
std::string::assign(std::string cons
std::string::replace(ulong,ulong,ch
std::string::reserve(ulong)
std::string::string(char const*,std:
std::string::string(std::string const
std::ios_base::Init::Init(void)
std::__throw_length_error(char co
operator delete(void *)
operator new(ulong)
_CCCrypt
__Unwind_Resume
___bzero
___cxa_atexit
____error
____maskrune
___stack_chk_fail
```

```
; Segment type: Pure code
__text segment dword public 'CODE' use64
assume cs:__text
;org 1000015A4h
assume es:nothing, ss:nothing, ds:nothing, fs:nothing, gs:nothing


; Attributes: noreturn fuzzy-sp

public start
start proc near
push    0
mov     rbp, rsp
and     rsp, 0FFFFFFFFFFFFFFF0h
mov     rdi, [rbp+8]
lea     rsi, [rbp+10h]
mov     edx, edi
add     edx, 1
shl     edx, 3
add     rdx, rsi
mov     rcx, rdx
jmp     short loc_1000015C9

loc_1000015C9:
cmp     qword ptr [rcx], 0
jnz     short loc_1000015C5

loc_1000015C5:
add     rcx, 8

add     rcx, 8
call    sub_1000015E0
start endp
```

```
100.00% (-159,-428) (23,204) 000015A4 00000001000015A4: start (Synchronized with Hex View-1)
```

```
loaded into the database.
Propagating type information...
Function argument information has been propagated
The initial autoanalysis has been finished.
Flushing buffers, please wait...ok
Database has been saved in file C:\Users\Analyst\Desktop\samples\HiddenLotus\HiddenLotus\Le╠é Thu Ha╠Ç (HAEDC).pΓà┘f\Contents\MacOS\Le╠é Thu Ha╠Ç
(HAEDC).i64
```

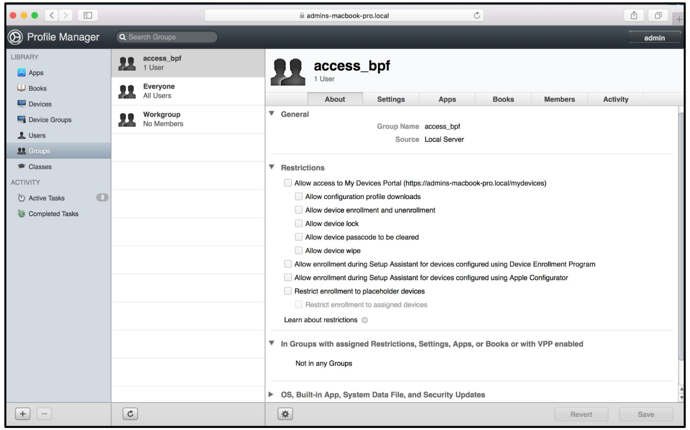
# Part Three Automation and MDM

# The Big Picture

1. Image
2. Disable SIP
3. Provision with dynamic configuration via configuration management
   a. We will have Ansible Playbooks, but easily converted to Chef, Puppet, etc.
4. Convert a Mac to macOS Server, $20 from Apple
5. Setup Profile Manager server
   a. Use to manage settings remotely, overrides SIP
6. Bind to our AD domain
   a. Gives us access to existing user objects
   b. See earlier comments about keychain and passwords
7. Connect the plumbing
   a. Connect up logs from Syslog, Audit, OSQuery, to our SIEM

# macOS Server and Profile Manager

- macOS Server upgrade for $20
- Profile Manager gives you
  - Mobile accounts for local and directory auth
  - Manage things like Gatekeeper policy
  - May be a hack via login scripts
    - com.apple.loginwindow
- Signed by apple so overrides SIP
- Uses XML like .plist format with key value pairs
- Alternative to pay products like Jamf MDM



**access_bpf**
1 User

▼ General

Group Name access_bpf
Source Local Server

▼ Restrictions

☐ Allow access to My Devices Portal (https://admins-macbook-pro.local/mydevices)
  ☐ Allow configuration profile downloads
  ☐ Allow device enrollment and unenrollment
  ☐ Allow device lock
  ☐ Allow device passcode to be cleared
  ☐ Allow device wipe
☐ Allow enrollment during Setup Assistant for devices configured using Device Enrollment Program
☐ Allow enrollment during Setup Assistant for devices configured using Apple Configurator
☐ Restrict enrollment to placeholder devices
  ☐ Restrict enrollment to assigned devices

Learn about restrictions

▼ In Groups with assigned Restrictions, Settings, Apps, or Books or with VPP enabled

Not in any Groups

▶ OS, Built-in App, System Data File, and Security Updates

# Notes About Code Release

- TEST, TEST, TEST, I am not responsible!!!
- https://github.com/r3doubt/apple-sauce-in-a-bucket
    - “beta” -ish v0.1 something something
    - Will release Ansible Playbooks  soon
    - Intended to be “light” and “medium paranoia”
        - Deployment via automation
        - Focus is managed network and mobile devices
- https://blog.r3doubt.io
    - Tutorials will come out with more details
- Respond if you contact me
    - Twitter @r3doubt
    - Github
- Recommend reading NIST SP-800-179 and GCHQ EUD for 10.12, 10.13

# Acknowledgements

- All of you for listening
- BloomCon, Dr. Polstra for giving me the soapbox (and swag)
- Nate Guagenti @neu5ron
- Michael George @pickmansec
- Roberto Rodriguez @Cyb3rWard0g for feedback and ideas
- Everyone else I missed
- I follow on Twitter (in case you wondered) for macOS stuff
  - Patrick Wardle @patrickwardle
  - Chris Ross @xorrior

# Questions??

# References

https://www.jamf.com/blog/fortune-500-companies-follow-in-ibms-footsteps-with-mac-choice-programs/
https://9to5mac.com/2017/03/07/apple-enterprise-data-mac-ios-jamf/
https://blog.malwarebytes.com/threat-analysis/2017/01/new-mac-backdoor-using-antiquated-code/
https://www.blackhat.com/docs/us-15/materials/us-15-Wardle-Writing-Bad-A-Malware-For-OS-X.pdf
https://www.blackhat.com/docs/us-17/wednesday/us-17-Wardle-Offensive-Malware-Analysis-Dissecting-OSXFruitFly-Via-A-Custom-C&C-Server.pdf
https://www.infosecurity-magazine.com/news/fruitfly-malware-creator-spied/
https://msdn.microsoft.com/en-us/library/windows/desktop/aa376977(v=vs.85).aspx
https://busylog.net/mac-osx-boot-sequence-launchd/
https://developer.apple.com/library/content/documentation/CoreFoundation/Conceptual/CFBundles/BundleTypes/BundleTypes.html#//apple_ref/doc/uid/10000123i-CH101-SW1
https://developer.apple.com/library/content/documentation/MacOSX/Conceptual/BPSystemStartup/Chapters/CreatingLaunchdJobs.html#//apple_ref/doc/uid/10000172i-SW7-BCIEDDBJ
http://www.multicastdns.org/
https://www.openbsd.org/faq/pf/
https://www.wavether.com/2016/11/pf-firewall-macos-jetbrains
https://tools.ietf.org/html/rfc6762
https://support.apple.com/en-us/HT204899
https://eclecticlight.co/2017/04/28/sierras-system-integrity-protection-sip-beyond-root/
https://eclecticlight.co/2018/01/02/the-app-you-cant-trash-how-sip-is-broken-in-high-sierra/
https://objective-see.com/blog/blog_0x24.html
https://developer.apple.com/library/content/documentation/Security/Conceptual/System_Integrity_Protection_Guide/Introduction/Introduction.html
https://arstechnica.com/information-technology/2017/11/macos-bug-lets-you-log-in-as-admin-with-no-password-required/
https://developer.apple.com/legacy/library/documentation/Darwin/Reference/ManPages/man1/dns-sd.1.html
https://tools.ietf.org/html/rfc6763
https://www.exploit-db.com/exploits/41873/
https://hacker.house/lab/gns-3-ubridge-local-privilege-escalation-attack-0day/
https://www.pentestpartners.com/security-blog/exploiting-suid-executables/
https://developer.apple.com/documentation/coregraphics
https://developer.apple.com/bonjour/
https://support.apple.com/en-us/HT208315
https://objective-see.com/blog/blog_0x24.html
https://developer.apple.com/library/content/documentation/Security/Conceptual/Security_Overview/Introduction/Introduction.html#//apple_ref/doc/uid/TP30000976
https://support.apple.com/en-us/HT204012
https://developer.apple.com/library/content/documentation/Darwin/Conceptual/KernelProgramming/About/About.html#//apple_ref/doc/uid/TP30000905
https://developer.apple.com/library/content/documentation/Security/Conceptual/keychainServConcepts/01introduction/introduction.html#//apple_ref/doc/uid/TP30000897-CH203-TP1
https://opensource.apple.com/tarballs/mDNSResponder/

# More References

https://medium.com/@JoshuaAJung/managing-your-mobile-devices-in-the-cloud-using-apples-own-mdm-solution-8a58
https://www.macworld.com/article/3016066/macs/a-primer-in-profile-manager-set-up-open-directory.html8d9724b6
https://www.securityweek.com/rpc-portmapper-abused-ddos-attack-reflection-amplification
https://xkcd.com/149/
http://tech.masterofsql.com/os-x/unload-disable-unwanted-agents-daemons-os-x.html
https://support.apple.com/kb/PH20093?locale=en_US
https://developer.apple.com/documentation/security/keychain_services/keychains
https://blog.malwarebytes.com/cybercrime/2017/09/keychain-vulnerability-in-macos/
https://www.patreon.com/posts/https://osquery.readthedocs.io/en/stable/installation/install-osx/mr-steal-yo-14556409
https://osquery.readthedocs.io/en/stable/installation/install-osx/
https://github.com/google/santa
http://www.irongeek.com/i.php?page=videos/derbycon7/s30-macos-host-monitoring-the-open-source-way-michael-george
https://www.ansible.com/
https://arstechnica.com/information-technology/2017/11/macos-bug-lets-you-log-in-as-admin-with-no-password-required/
https://blog.rapid7.com/2016/05/09/introduction-to-osquery-for-threat-detection-dfir/
https://developer.apple.com/library/content/documentation/MacOSX/Conceptual/BPSystemStartup/Chapters/CustomLogin.html
https://jordanpotti.com/20
https://support.apple.com/en-us/HT20224418/02/16/elk-osquery-kolide-fleet-love/
https://www.macworld.com/article/3029088/macs/a-primer-in-profile-manager-os-x-and-ios-payloads-for-devices.html
https://objechttps://derflounder.wordpreshttps://blog.malwarebytes.com/threat-analysis/2017/12/interesting-disguise-employed-by-new-mac-malware/s.com/2012/11/20/clearing-the-quarantine-extended-attribute-from-downloaded-applications/
http://newosxbook.com/articles/DYLD.html
https://www.fireeye.com/blog/threat-research/2017/05/cyber-espionage-apt32.html

tive-see.com/blog/blog_0x24.html

https://www.intego.com/mac-security-blog/osxshlayer-new-mac-malware-comes-out-of-its-shell/
https://whttps://jameshfisher.com/2017/08/22/inspecting-mach-o-files.htmlww.alienvault.com/blogs/labs-research/oceanlotus-for-os-x-an-application-bundle-pretending-to-be-an-adobe-flash-update