Adversary Village @ DEFCON 32

AN ARTIST GUIDE TO ADVERSARY FORGERY

COPY CAT

MITRE

ATT&CK™

- Former Artist
- Military Intelligence Veteran
- Red Teamer, Threat Hunter @Target
- Lead macOS & Linux ATT&CK

  @MITRE

WHAT ARE WE DOING TODAY?

Read through a report and reproduce what the MITRE ATT&CK technique the report is describing

ADVERSARY EMULATION IS…

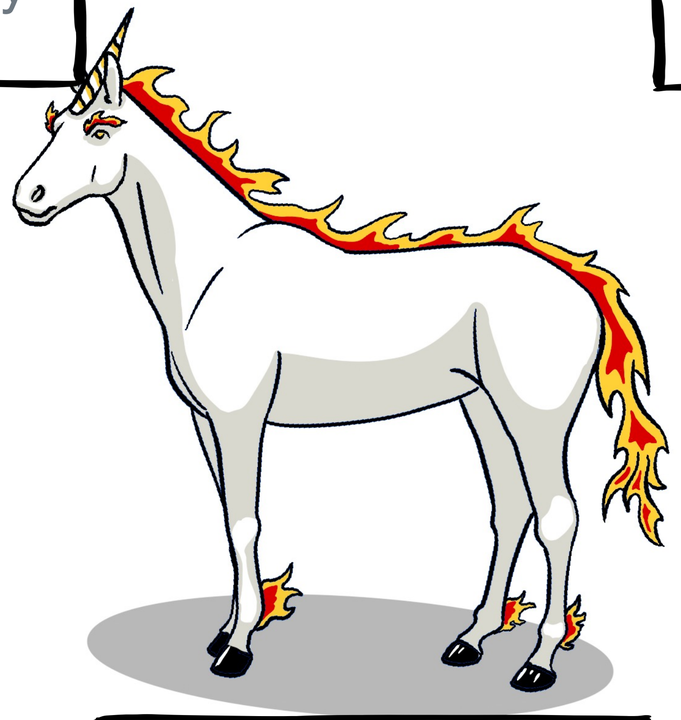
Charney, Noah. *The Art of Forgery: the Minds, Motives and Methods of Master Forgers.* Phaidon, 2015.

Cast of Characters

Ninja Panda & threads

Drunken Monkey Master

Kung Fu Master

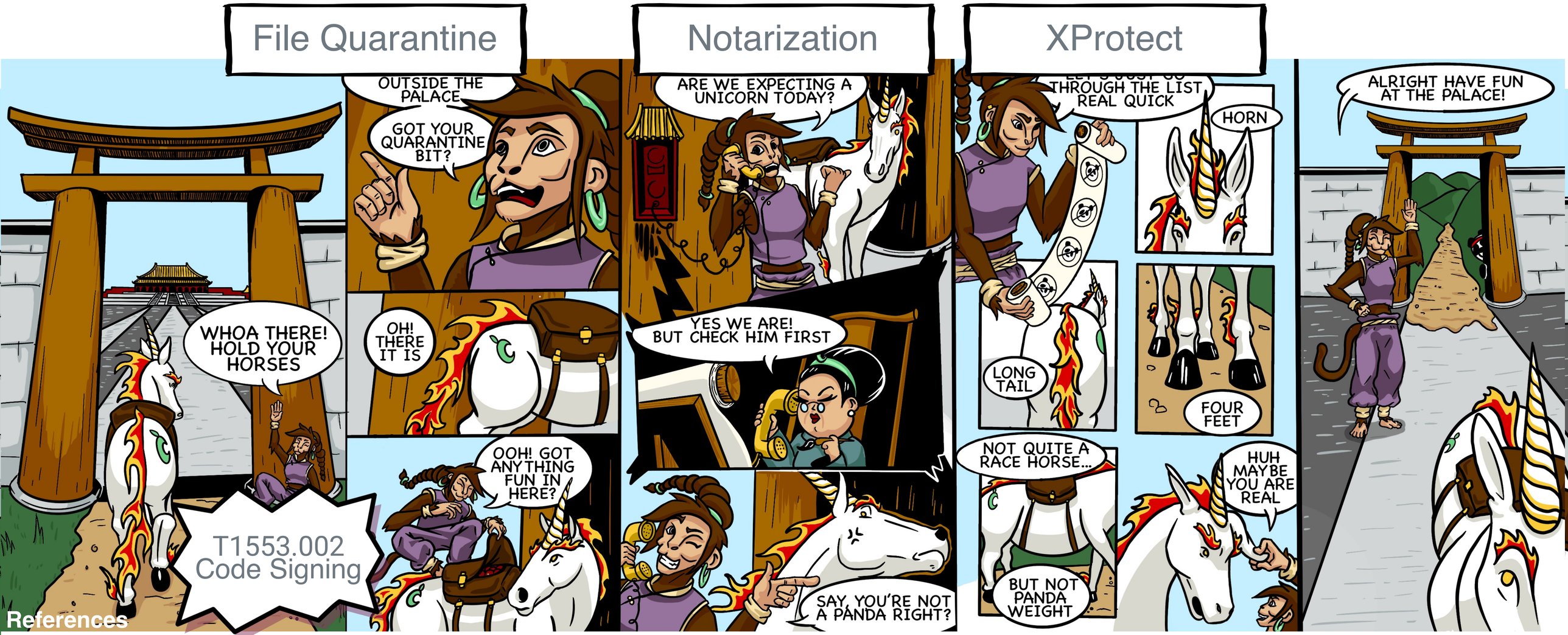


A Pure Unicorn App

Helper Daemons

Artwork powered by
@MiscreantsHQ
@coolestcatiknow

COPY CAT: AN ARTIST GUIDE TO ADVERSARY FORGERY



Quarantine process, data locations

Code Signing process, data locations

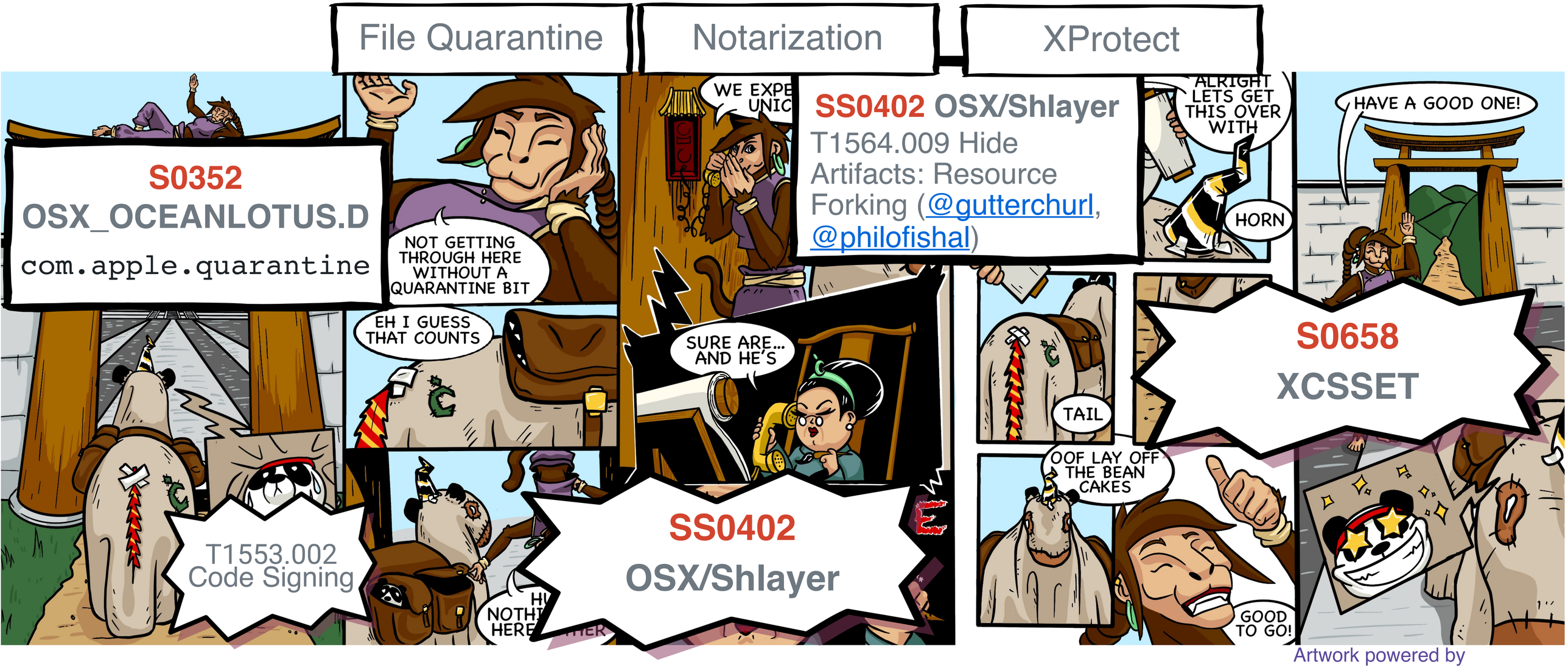
Notarization process, data locations

Xprotect process, data locations

Artwork powered by @MiscreantsHQ @coolestcatiknow

MITRE

#OBTS v5.0: "ATT&CKing Pandas: Drawing out ATT&CK Techniques in the Wild" - Cat Self



Artwork powered by
@MiscreantsHQ
@coolestcatiknow

#OBTS v5.0: "ATT&CKing Pandas: Drawing out ATT&CK Techniques in the Wild" - Cat Self

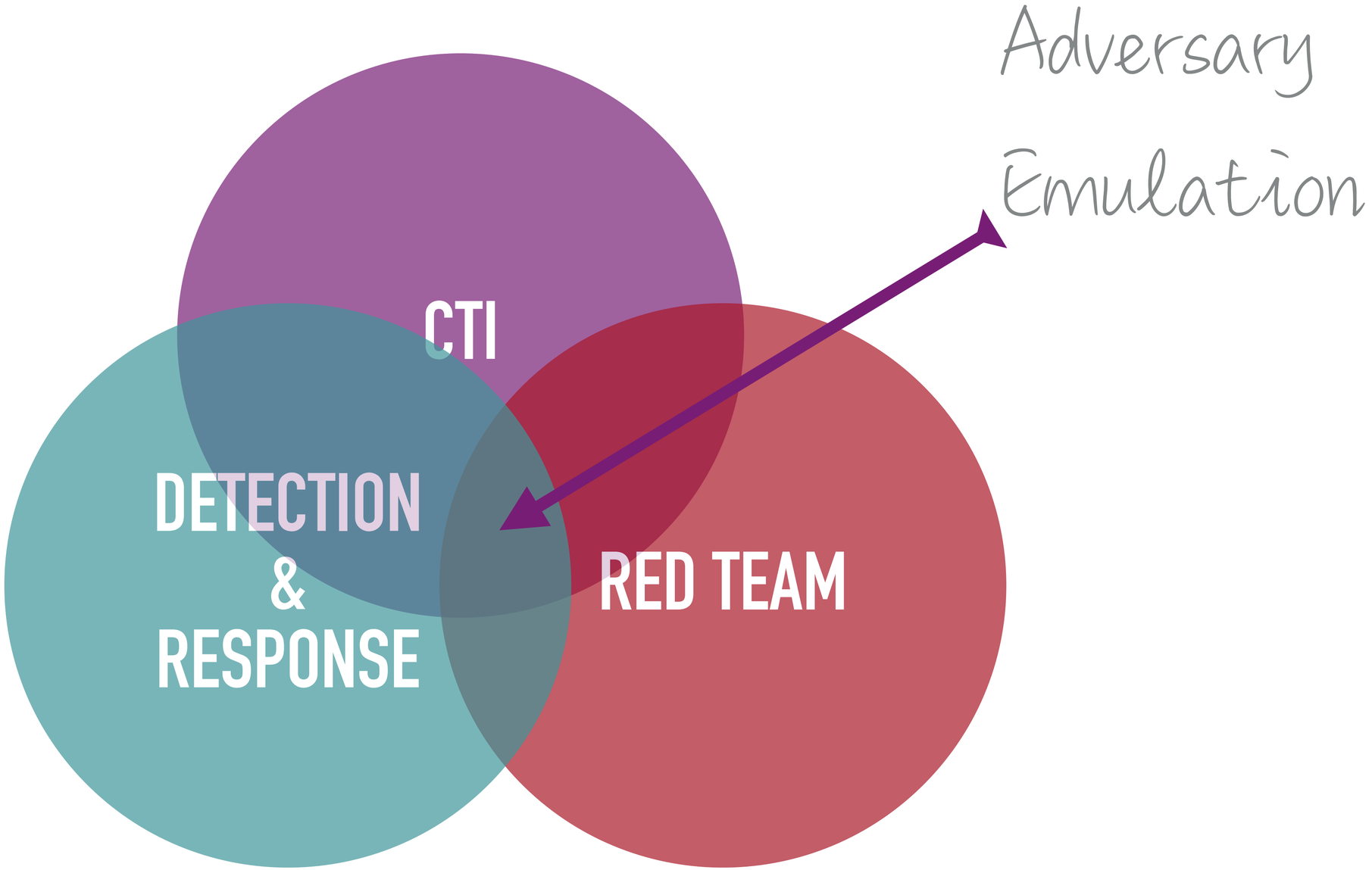
MITRE

WHAT'S INVOLVED?

People

Process

Technology



https://www.ibm.com/downloads/cas/QEBYPND1

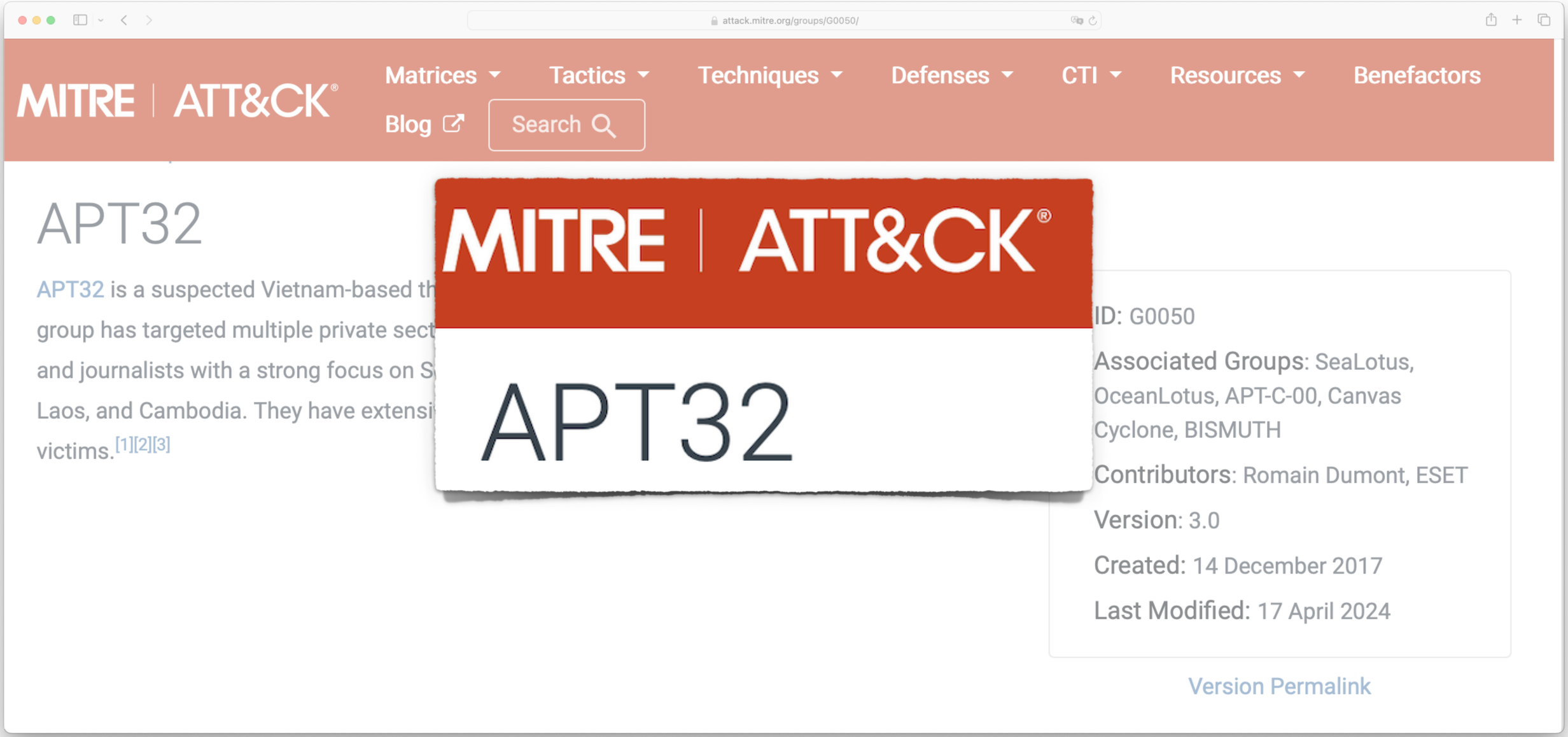
MITRE

DIFFERENT WAYS TO USE AE

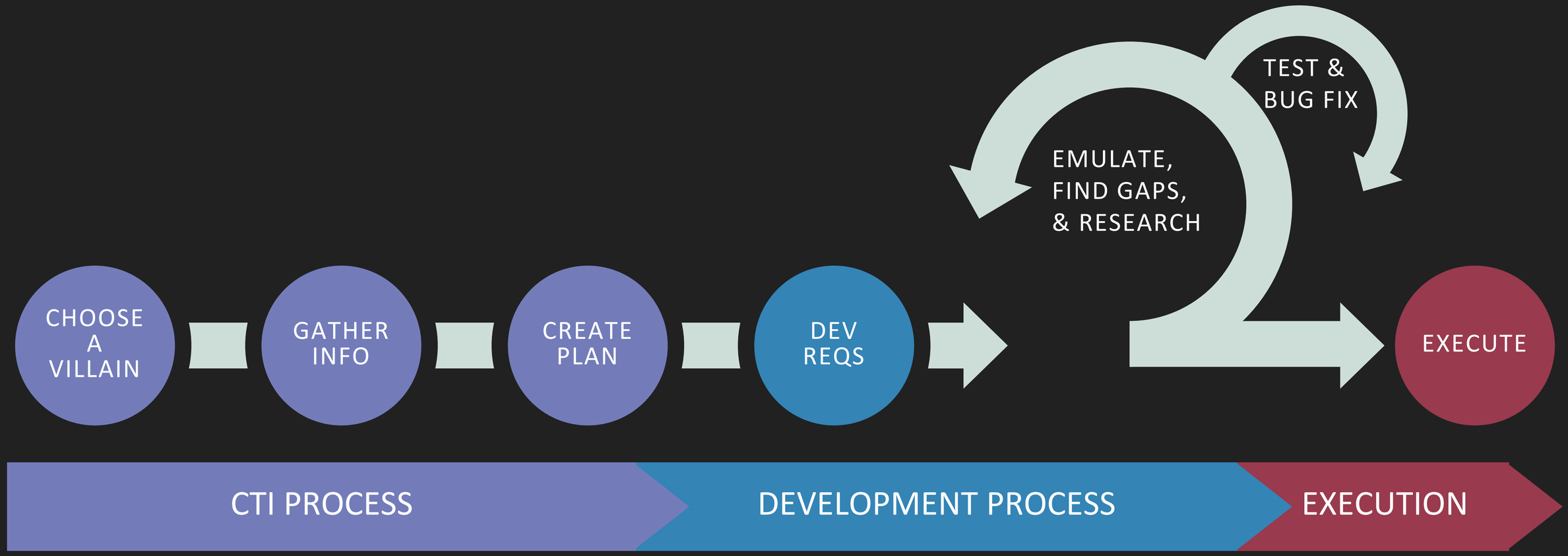
Atomic Testing	Micro Emulation	Full Emulation
Emulate single technique	**Emulate compound behaviors across 2–3 techniques**	**Emulate adversary operation**
Executable in **seconds**	Executable in **seconds**	Executable in **hours**
E.g., Atomic Red test for T1003.001 - LSASS Memory	*E.g., Fork & Run Process Injection*	*E.g., FIN6 adversary emulation plan*
Easy to automate	Easy to automate	Easy to automate
Validate atomic analytics	Validate atomic analytics	Validate atomic analytics
Validate chain analytics	Validate chain analytics	Validate chain analytics
Evaluate SOC against a specific set of TTPs	Evaluate SOC against a specific set of TTPs	Evaluate SOC against a specific set of TTPs
Evaluate SOC holistically against specific groups	Evaluate SOC holistically against specific groups	Evaluate SOC holistically against specific groups

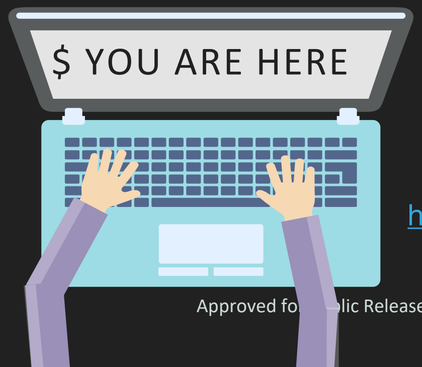
Skoog, Ingrid. "Ahhh, This Emulation Is Just Right: Introducing Micro Emulation Plans." *Medium*, MITRE-Engenuity, 15 Sept. 2022, medium.com/mitre-engenuity/ahhh-this-emulation-is-just-right-introducing-micro-emulation-plans-7bf4c26451d3.

MITRE



BECOMING A DARK KNIGHT





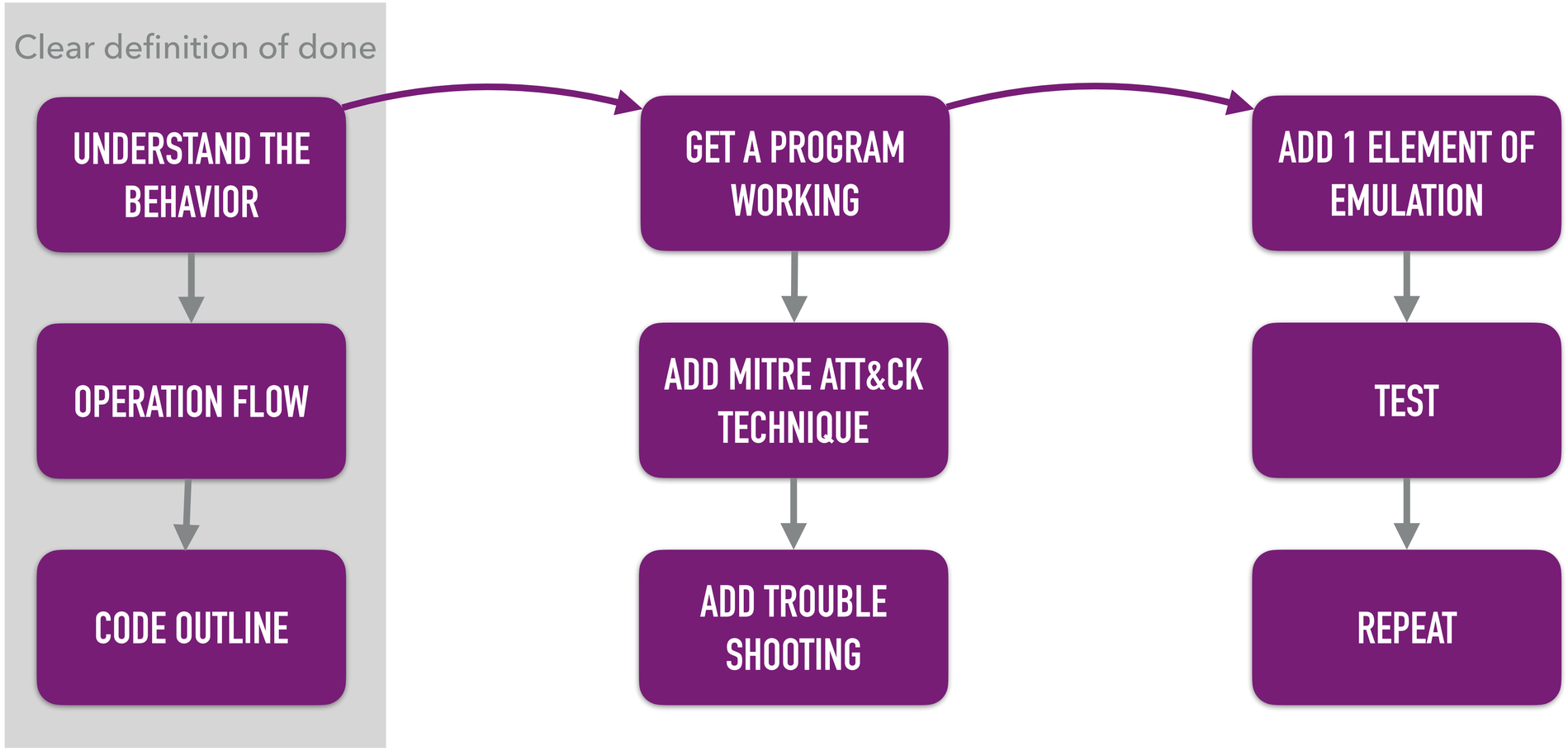
https://www.blackhat.com/us-23/briefings/schedule/#becoming-a-dark-knight-adversary-emulation-demonstration-for-attck-evaluations-33209

Cat @coolestcatiknow

Kate @phish4answers

MITRE ENGENUITY

PROCESS BREAKDOWN



MITRE

EXERCISE: FIND WHERE THE MITRE ATT&CK TECHNIQUE SHARED MODULES T1129 IS DESCRIBED IN THE REPORTS.

- Review the MITRE ATT&CK technique T1129's procedures. Look for dlopen & dlsym function calls to narrow it down.
- In reporting, look for words used to describe T1129 Shared Modules…*plugins*, *shared libs*, *dynamic libraries*, *library*, & *load*. Look for these words to isolate where the behavior is described in the reporting. *Mappings at the bottom of the report != 100% correct.*

Reports

1. https://www.trendmicro.com/en_us/research/20/k/new-macos-backdoor-connected-to-oceanlotus-surfaces.html
2. https://unit42.paloaltonetworks.com/unit42-new-improved-macos-backdoor-oceanlotus/
3. https://www.trendmicro.com/en_us/research/18/d/new-macos-backdoor-linked-to-oceanlotus-found.html
4. https://www.welivesecurity.com/2019/04/09/oceanlotus-macos-malware-update/

MITRE

EVALUATING CTI REPORTS

EXPLICIT: "THE GOOD"

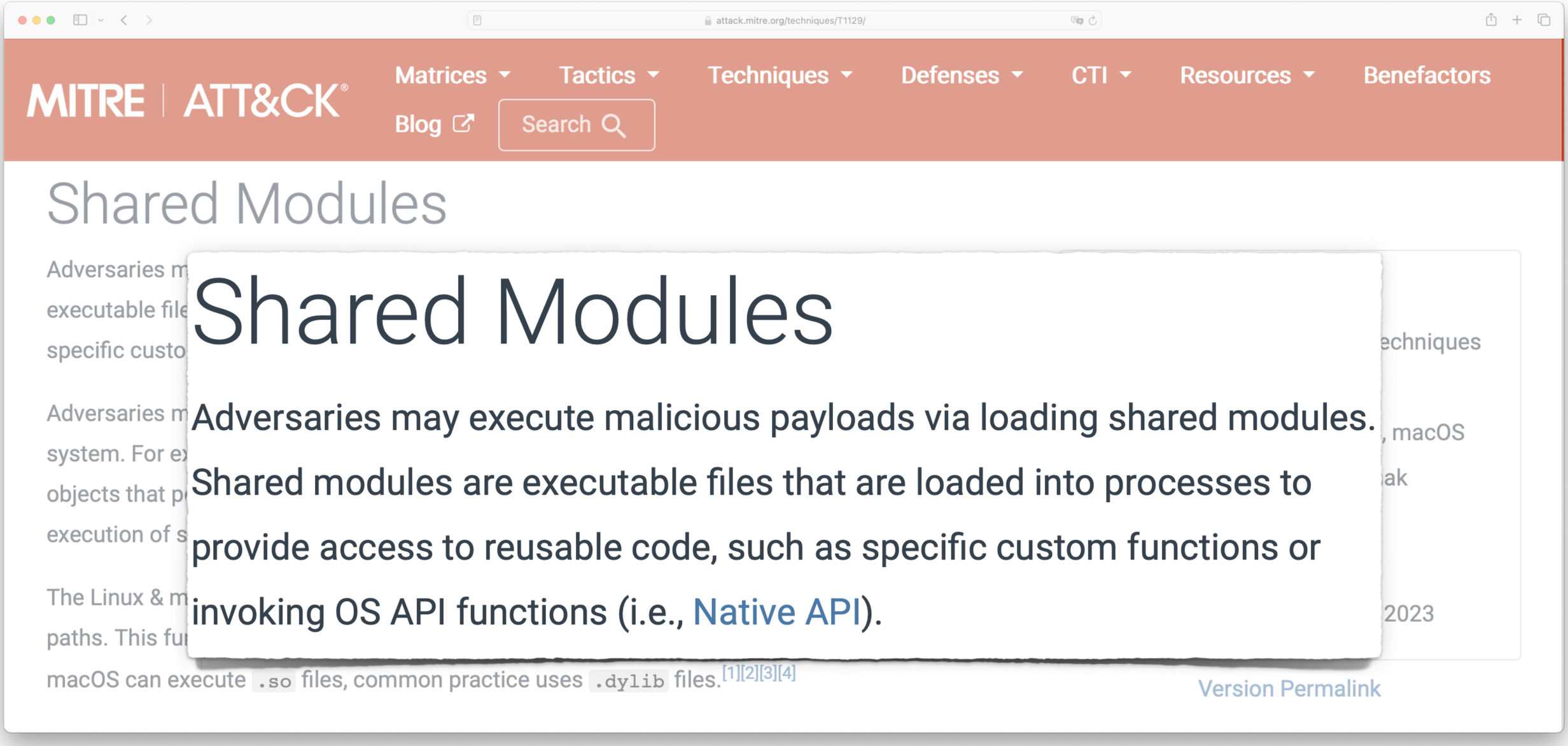
- Code/scripts
- C2 communication analysis
- Other artifacts (file paths, registry keys, etc.)

IMPLICIT: "THE GREAT"

- Lateral Movement
- Adversary actions on objectives
- Environment details

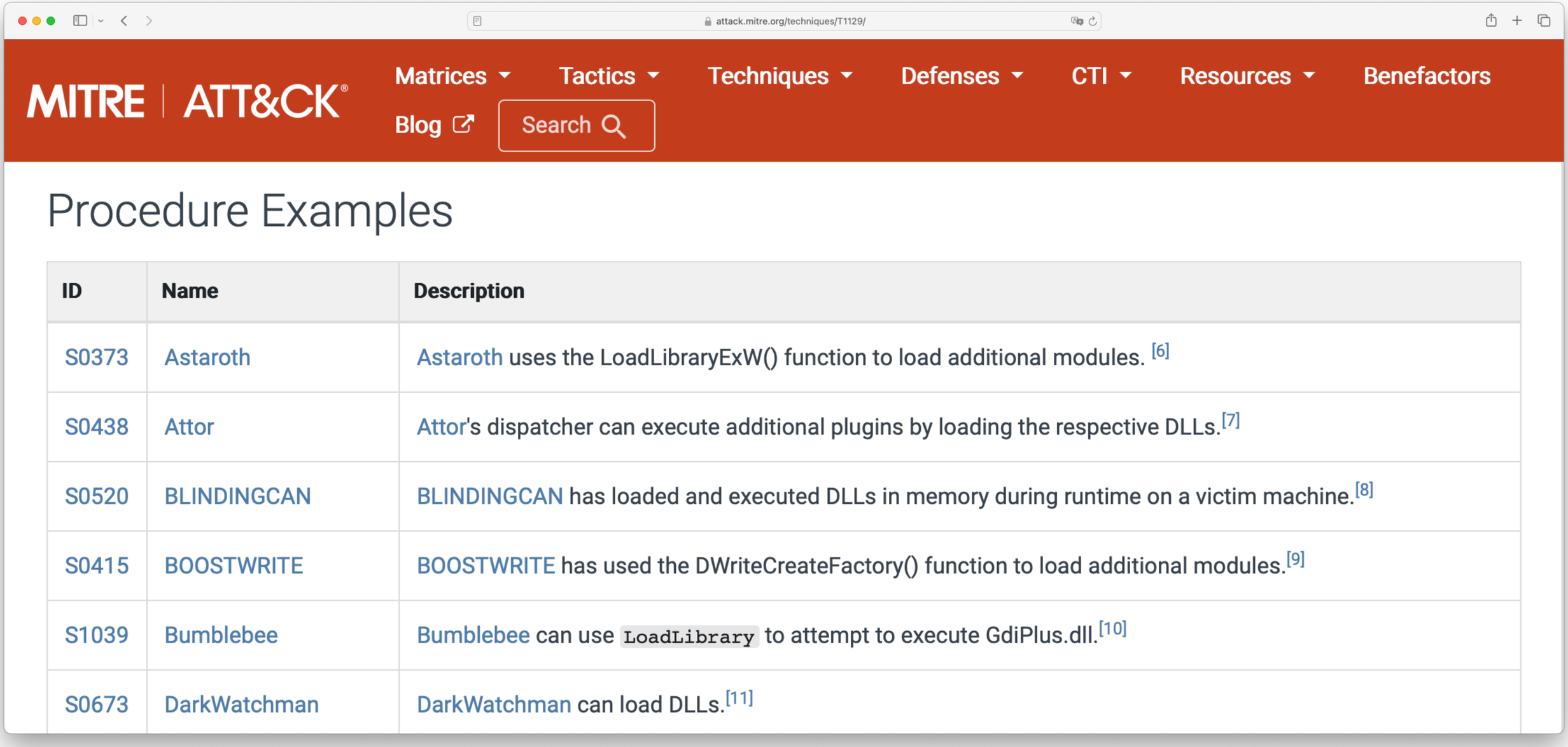
Cat @coolestcatiknow

Kate @phish4answers



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).



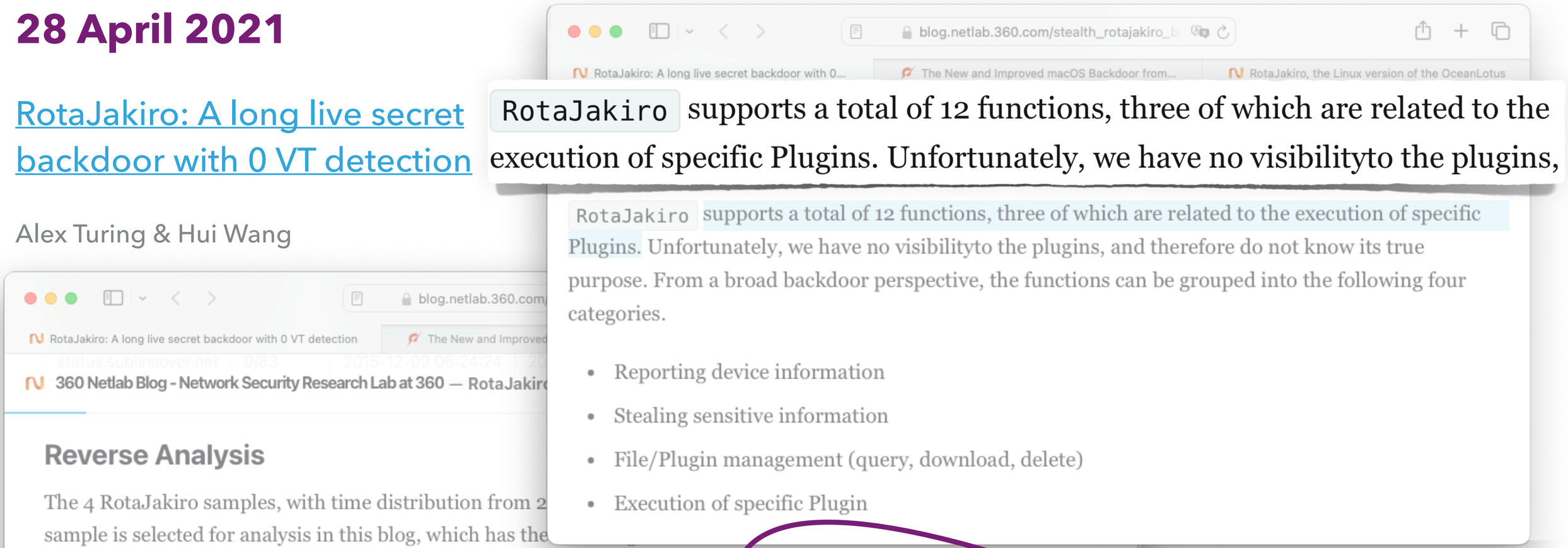
Procedure Examples

ID	Name	Description
S0373	Astaroth	Astaroth uses the LoadLibraryExW() function to load additional modules. [6]
S0438	Attor	Attor's dispatcher can execute additional plugins by loading the respective DLLs.[7]
S0520	BLINDINGCAN	BLINDINGCAN has loaded and executed DLLs in memory during runtime on a victim machine.[8]
S0415	BOOSTWRITE	BOOSTWRITE has used the DWriteCreateFactory() function to load additional modules.[9]
S1039	Bumblebee	Bumblebee can use `LoadLibrary` to attempt to execute GdiPlus.dll.[10]
S0673	DarkWatchman	DarkWatchman can load DLLs.[11]

28 April 2021

[RotaJakiro: A long live secret backdoor with 0 VT detection]

Alex Turing & Hui Wang



28 April 2021

[RotaJakiro: A long live secret backdoor with 0 VT detection]

Alex Turing & Hui Wang

`RotaJakiro` supports a total of 12 functions, three of which are related to the execution of specific Plugins. Unfortunately, we have no visibilityto the plugins,

`RotaJakiro` supports a total of 12 functions, three of which are related to the execution of specific Plugins. Unfortunately, we have no visibilityto the plugins, and therefore do not know its true purpose. From a broad backdoor perspective, the functions can be grouped into the following four categories.

- Reporting device information
- Stealing sensitive information
- File/Plugin management (query, download, delete)
- Execution of specific Plugin

RotaJakiro: A long live secret backdoor with 0 VT detection

360 Netlab Blog - Network Security Research Lab at 360 — RotaJakiro

Reverse Analysis

The 4 RotaJakiro samples, with time distribution from 2

sample is selected for analysis

```
MD5:64f6cfe44ba08b0babdd390423
ELF 64-bit LSB executable, x86-
Packer:No
```

`At the coding level` , Rot

protocols to counteract the bi

`At the functional level`

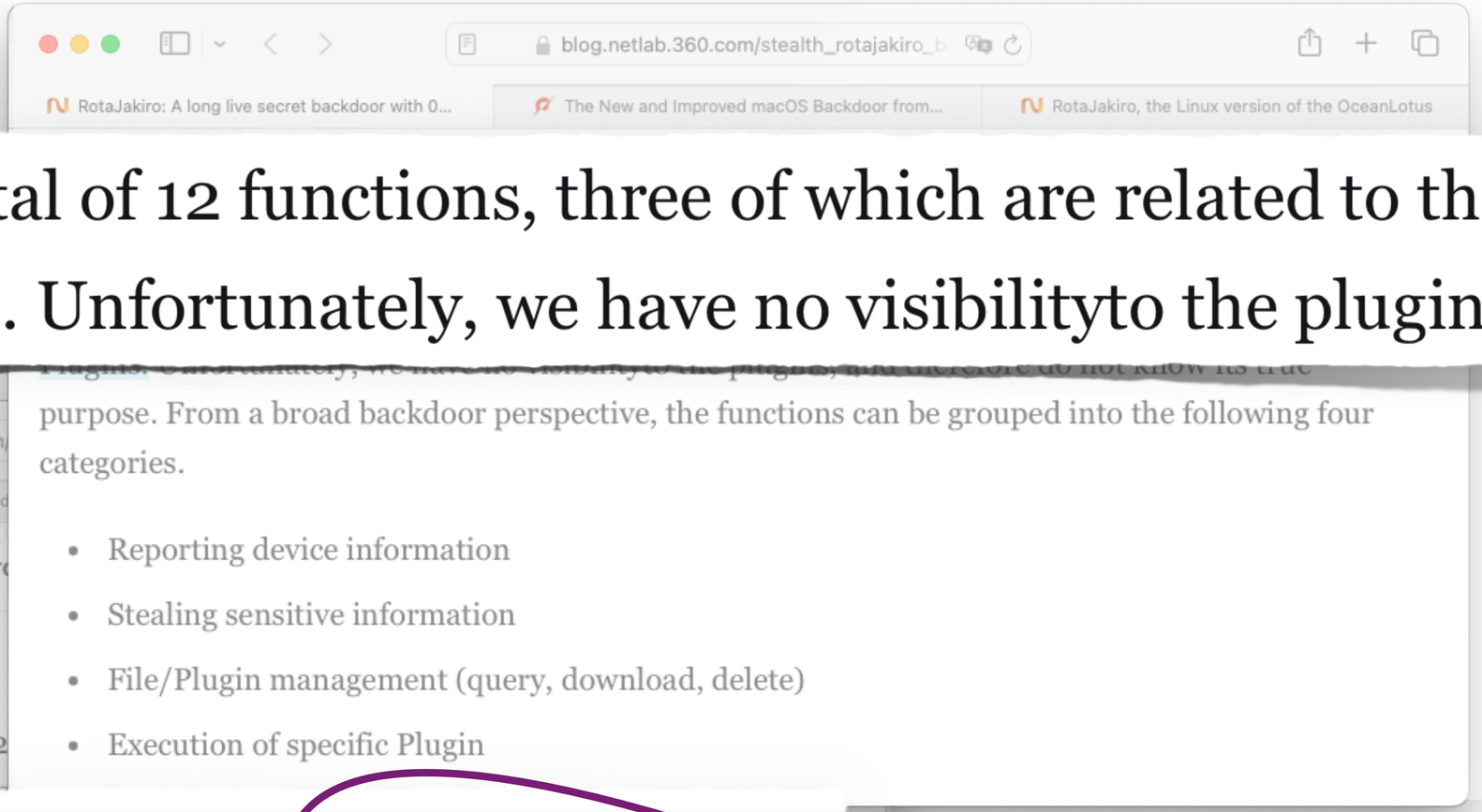
different execution policies for different accounts, then decrypts the relevant sensitive resources using AES& ROTATE

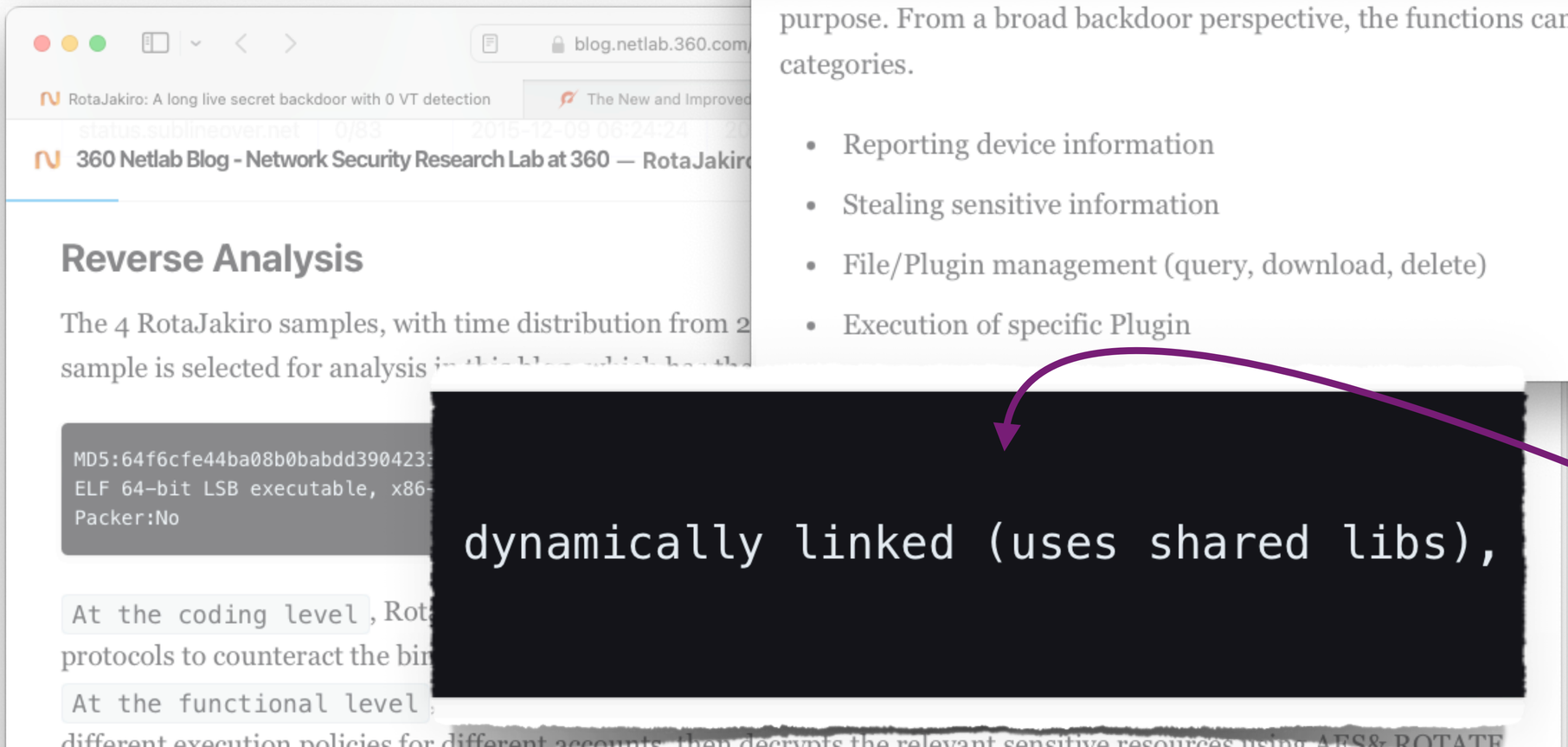
```
dynamically linked (uses shared libs),
```

EXPLICIT

28 April 2021

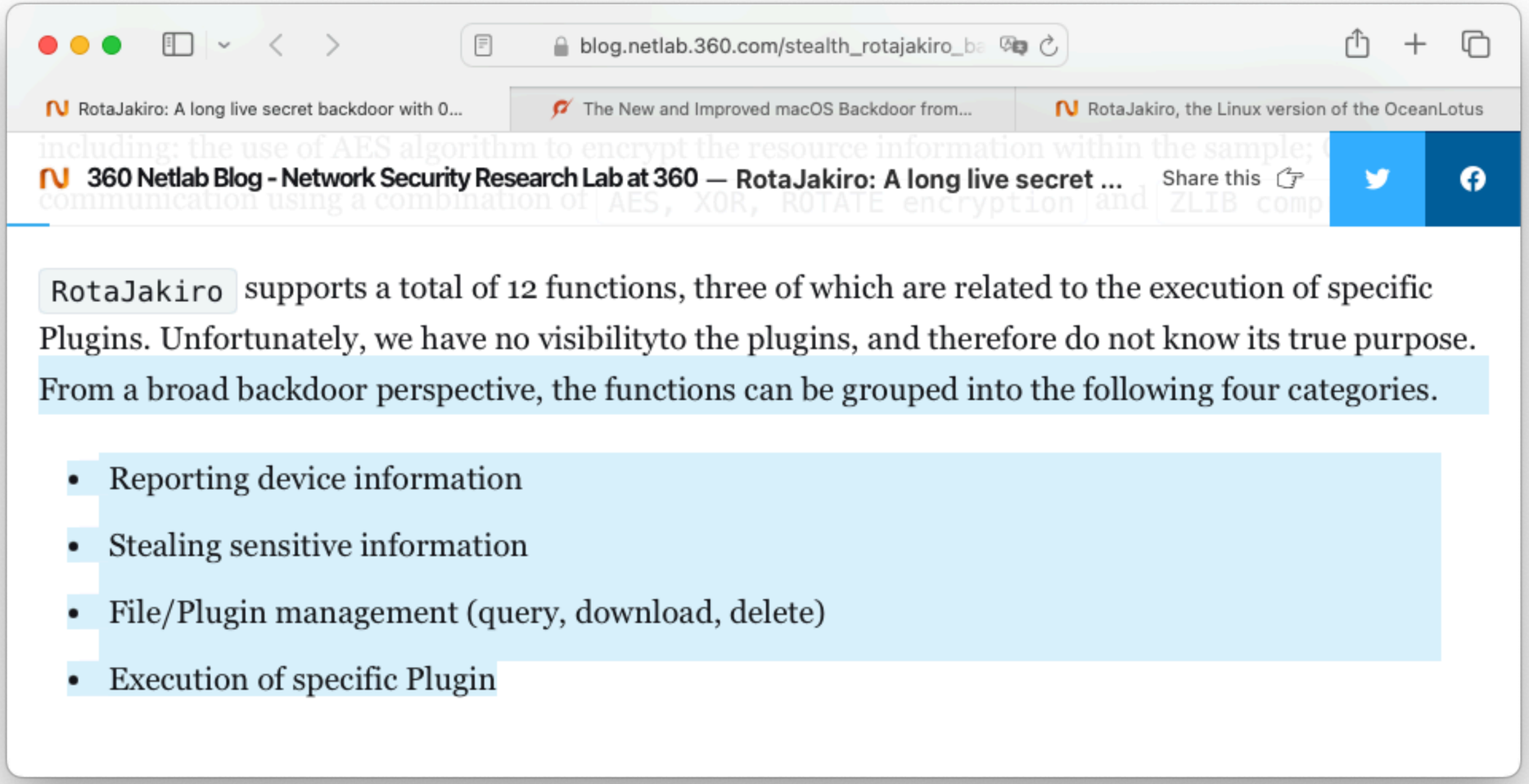
`RotaJakiro` supports a total of 12 functions, three of which are related to the execution of specific Plugins. Unfortunately, we have no visibilityto the plugins,






```
dynamically linked (uses shared libs),
```

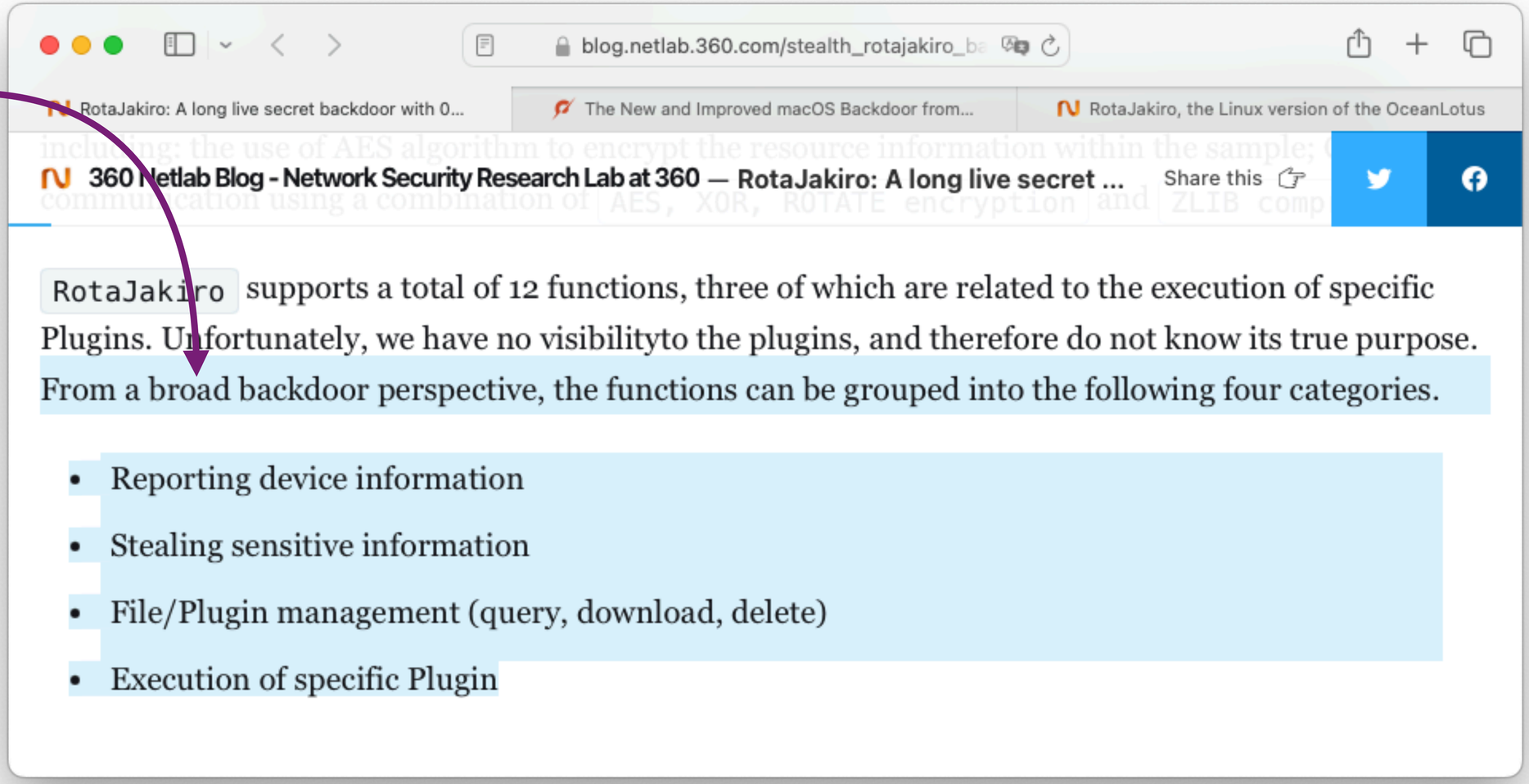
EXPLICIT



MITRE

COPY CAT: AN ARTIST GUIDE TO ADVERSARY FORGERY

IMPLICIT



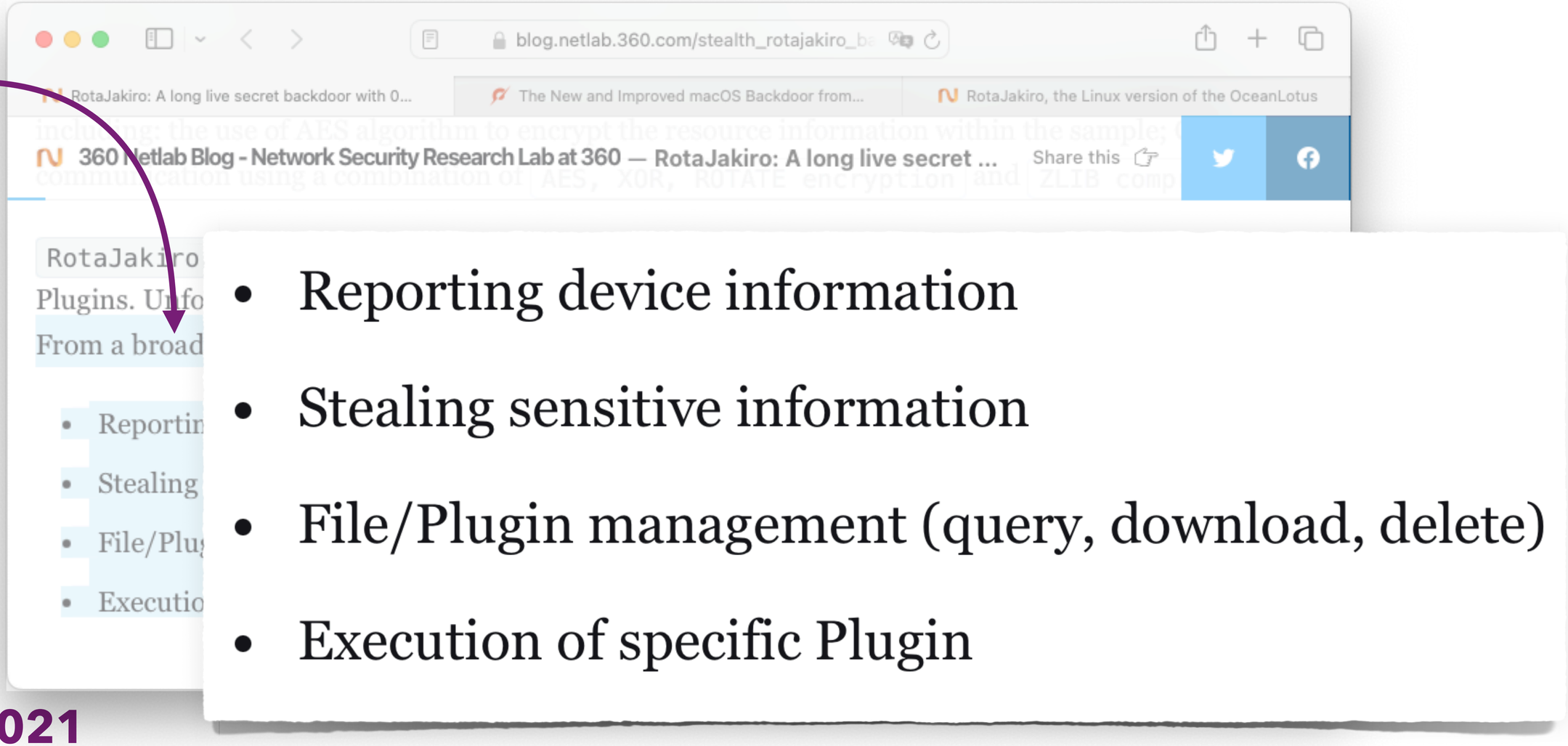
28 April 2021

[RotaJakiro: A long live secret backdoor with 0 VT detection]

Alex Turing & Hui Wang

MITRE

IMPLICIT



28 April 2021

[RotaJakiro: A long live secret backdoor with 0 VT detection]

Alex Turing & Hui Wang

MITRE

EXERCISE: FIND WHERE THE MITRE ATT&CK TECHNIQUE SHARED MODULES T1129 IS DESCRIBED IN THE REPORTS.

- Review the MITRE ATT&CK technique T1129's procedures. Look for dlopen & dlsym function calls to narrow it down.
- In reporting, look for words used to describe T1129 Shared Modules…*plugins*, *shared libs*, *dynamic libraries*, *library*, & *load*. Look for these words to isolate where the behavior is described in the reporting. *Mappings at the bottom of the report != 100% correct.*

Reports

1. https://www.trendmicro.com/en_us/research/20/k/new-macos-backdoor-connected-to-oceanlotus-surfaces.html
2. https://unit42.paloaltonetworks.com/unit42-new-improved-macos-backdoor-oceanlotus/
3. https://www.trendmicro.com/en_us/research/18/d/new-macos-backdoor-linked-to-oceanlotus-found.html
4. https://www.welivesecurity.com/2019/04/09/oceanlotus-macos-malware-update/

MITRE

22 JUNE 2017 – HERNANDEZ & TSECHANSKY – NEW AND IMPROVED MACOS BACKDOOR FROM OCEANLOTUS

EXPLICIT

unit42.paloaltonetworks.com/unit42-new-improved-macos-backdoor-

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 commands load a dynamic library using *dlopen()* and obtains a function pointer to execute within that shared library using *dlsym()*. Unfortunately, we do

same number of arguments with the first being a fairly large constant similar to the command constants, (see Figure 12) and the backdoor has a function for receiving files, it is possible that these functions correspond to a shared library that the server uploads to the victim host. This means that additional functionality can be added to this backdoor by loading modules directly from the C2 server.

MITRE

22 JUNE 2017 – HERNANDEZ & TSECHANSKY – NEW AND IMPROVED MACOS BACKDOOR FROM OCEANLOTUS

unit42.paloaltonetworks.com/unit42-new-improved-macos-backdoor-

Commands 0x25D5082, 0x1B25503, 0x1532E65

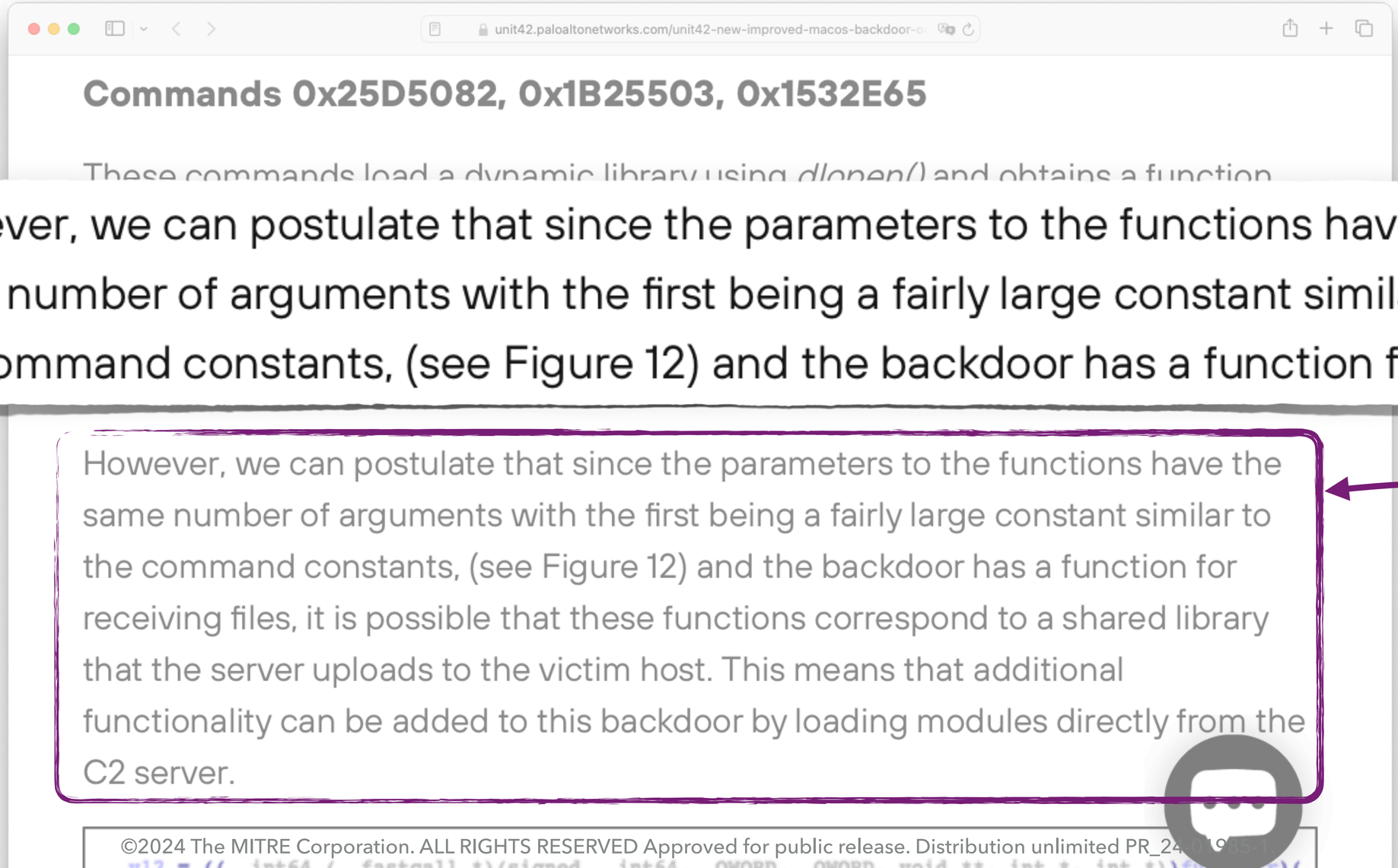
These commands load a dynamic library using *dlopen()* and obtains a function pointer to execute within that shared library using *dlsym()*. Unfortunately, we do not know which dynamic libraries or functions are used for each command since these are server supplied and we were not able to capture any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

However, we can postulate that since the parameters to the functions have the same number of arguments with the first being a fairly large constant similar to the command constants, (see Figure 12) and the backdoor has a function for receiving files, it is possible that these functions correspond to a shared library that the server uploads to the victim host. This means that additional functionality can be added to this backdoor by loading modules directly from the C2 server.

IMPLICIT

MITRE

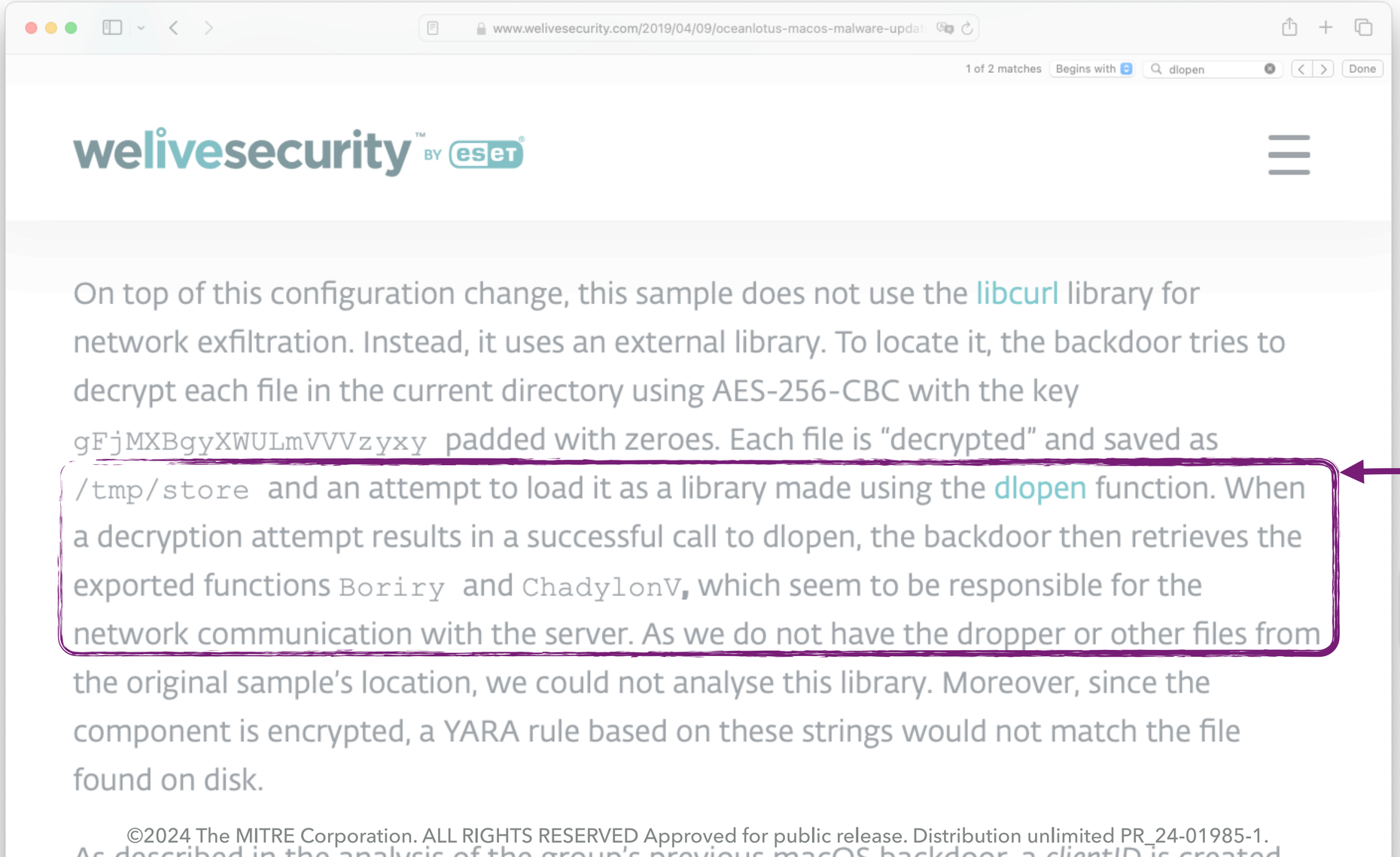
22 JUNE 2017 – HERNANDEZ & TSECHANSKY – NEW AND IMPROVED MACOS BACKDOOR FROM OCEANLOTUS



IMPLICIT

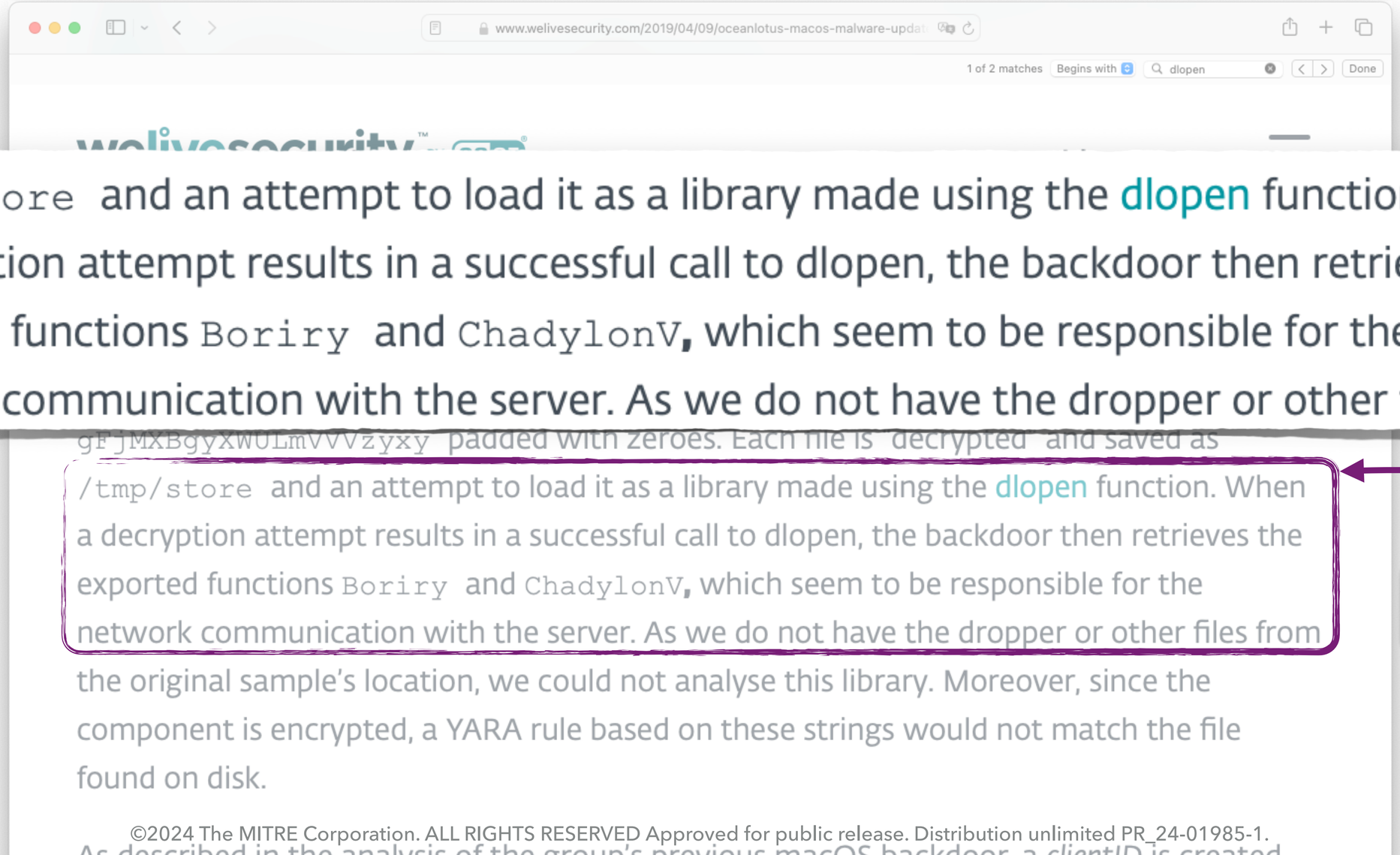
MITRE

22 JUNE 2017 – HERNANDEZ & TSECHANSKY – NEW AND IMPROVED MACOS BACKDOOR FROM OCEANLOTUS



MITRE

22 JUNE 2017 – HERNANDEZ & TSECHANSKY – NEW AND IMPROVED MACOS BACKDOOR FROM OCEANLOTUS



MITRE

ANSWER: MITRE ATT&CK TECHNIQUE

Find the macOS version of this technique from the following reports

Reports

1. https://www.trendmicro.com/en_us/research/20/k/new-macos-backdoor-connected-to-oceanlotus-surfaces.html
2. https://unit42.paloaltonetworks.com/unit42-new-improved-macos-backdoor-oceanlotus/
3. https://www.trendmicro.com/en_us/research/18/d/new-macos-backdoor-linked-to-oceanlotus-found.html
4. https://www.welivesecurity.com/2019/04/09/oceanlotus-macos-malware-update/

MITRE

1934–1996

ERIC HEBBORN

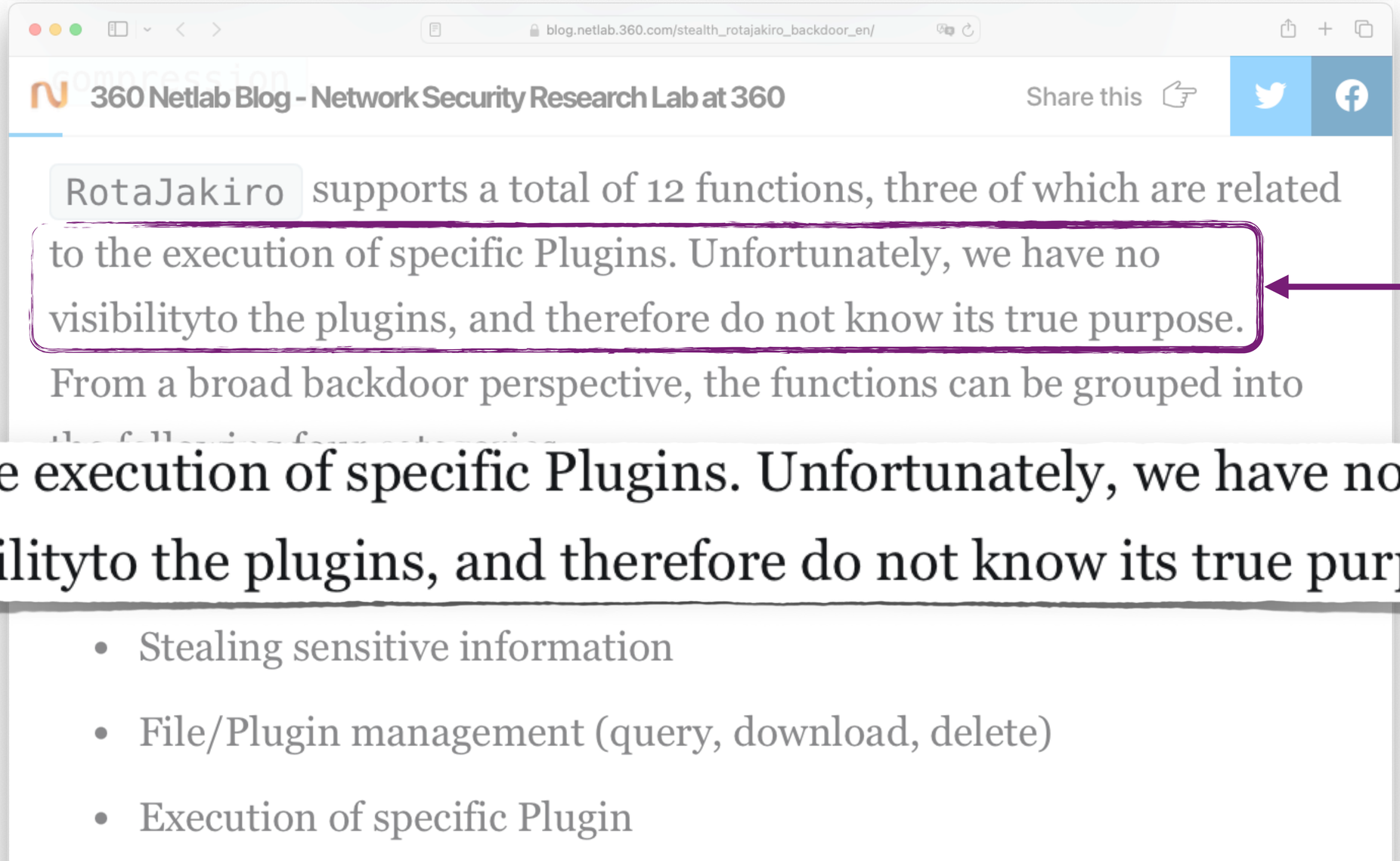
Kuesel, Christy. "The Prolific Forger." *Art Sy*, 10 Oct. 2019, www.artsy.net/article/artsy-editorial-prolific-forger-fake-masters-fooled-art.

EXERCISE: WRITE OUT IN PSEUDO CODE OR DIAGRAMS AN OUTLINE OF A PROGRAM TO EXECUTE THE T1129 SHARED MODULES MITRE ATT&CK TECHNIQUE

- What components need to be built?
- Think through the logic flow of what is needed to perform the T1129 & how to confirm it was executed
- Hint: .dylib is the macOS version of a .so file, how would you use dlopen & dlsym using .dylib files?

MITRE

GAP ANALYSIS



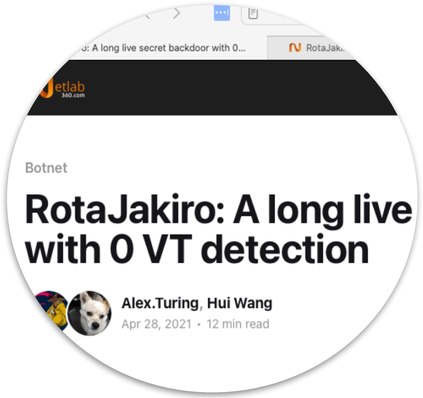
MITRE

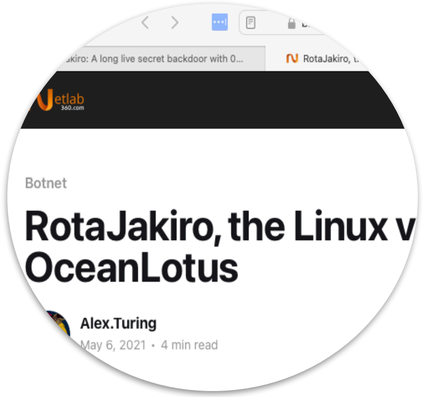
REPORTING TIMELINE

6 May 2021

Alex Turing

[RotaJakiro, the Linux version of the OceanLotus]





28 April 2021

Alex Turing & Hui Wang

[RotaJakiro: A long live secret backdoor with 0 VT detection]

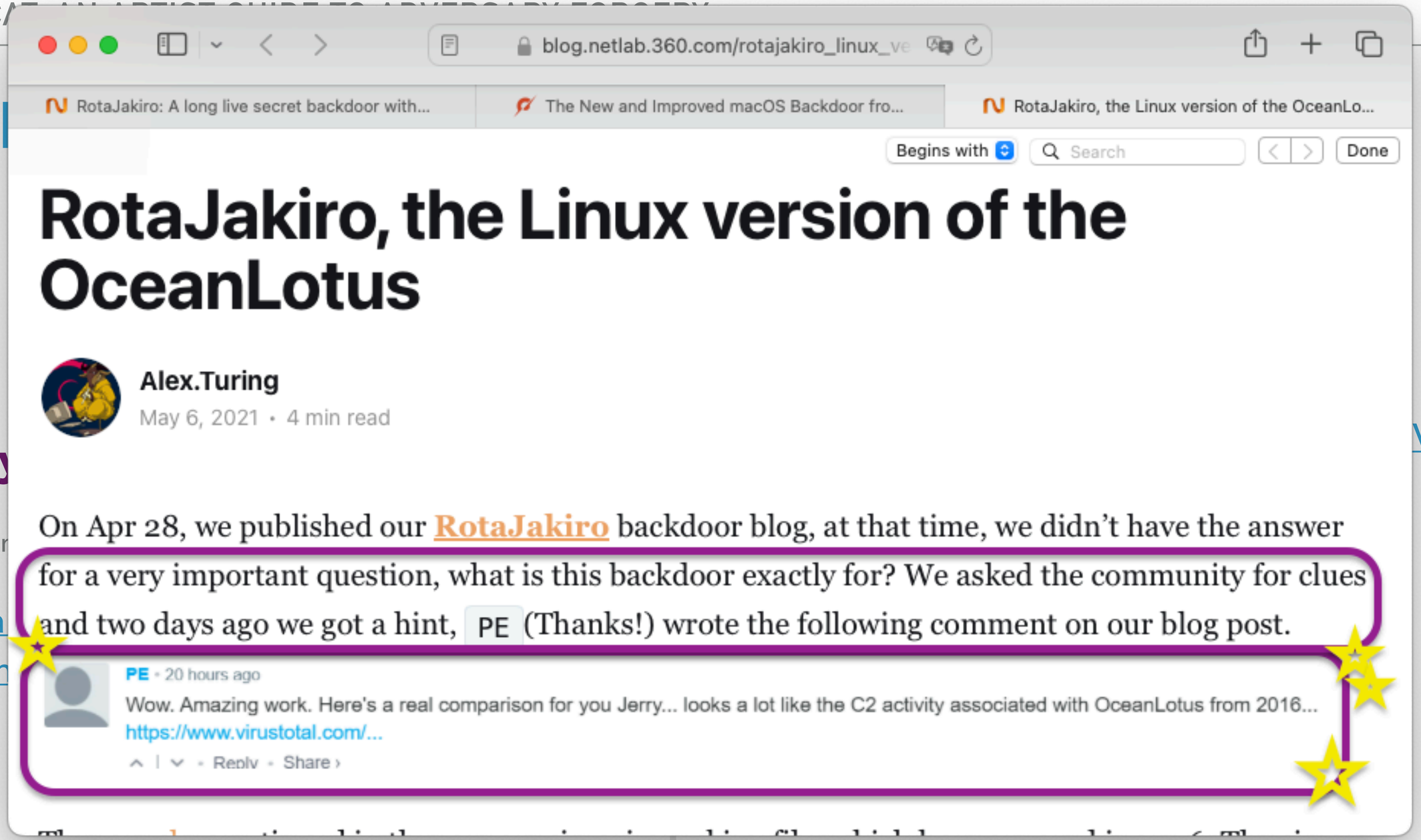
REPOR

6 May

Alex Turin

RotaJa
version

VT



MITRE

REPORTING TIMELINE

22 June 2017

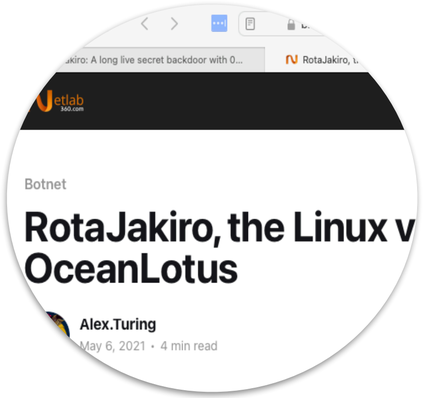
Erie Hernandez & Danny Tsechansky

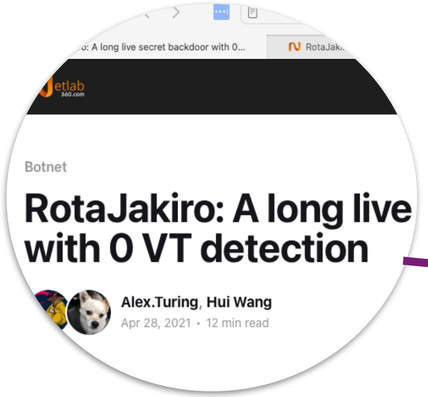
[The New and Improved macOS Backdoor from OceanLotus]

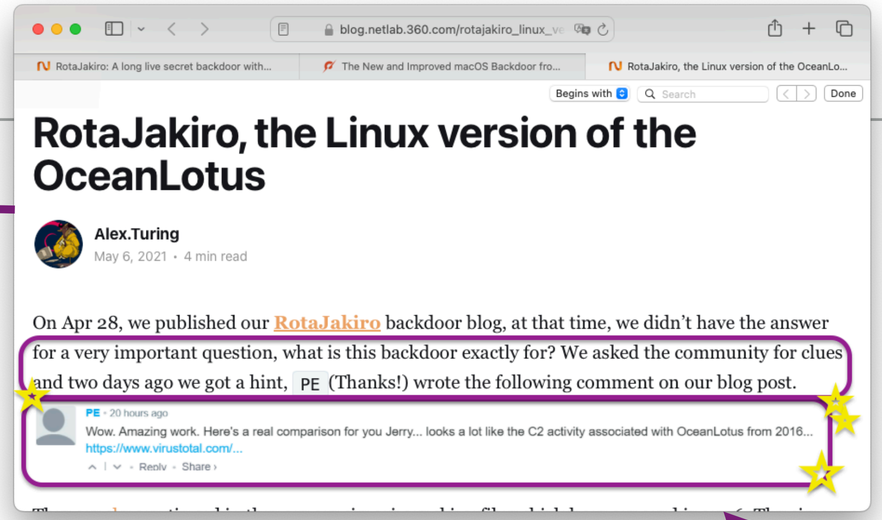
6 May 2021

Alex Turing

[RotaJakiro, the Linux version of the OceanLotus]







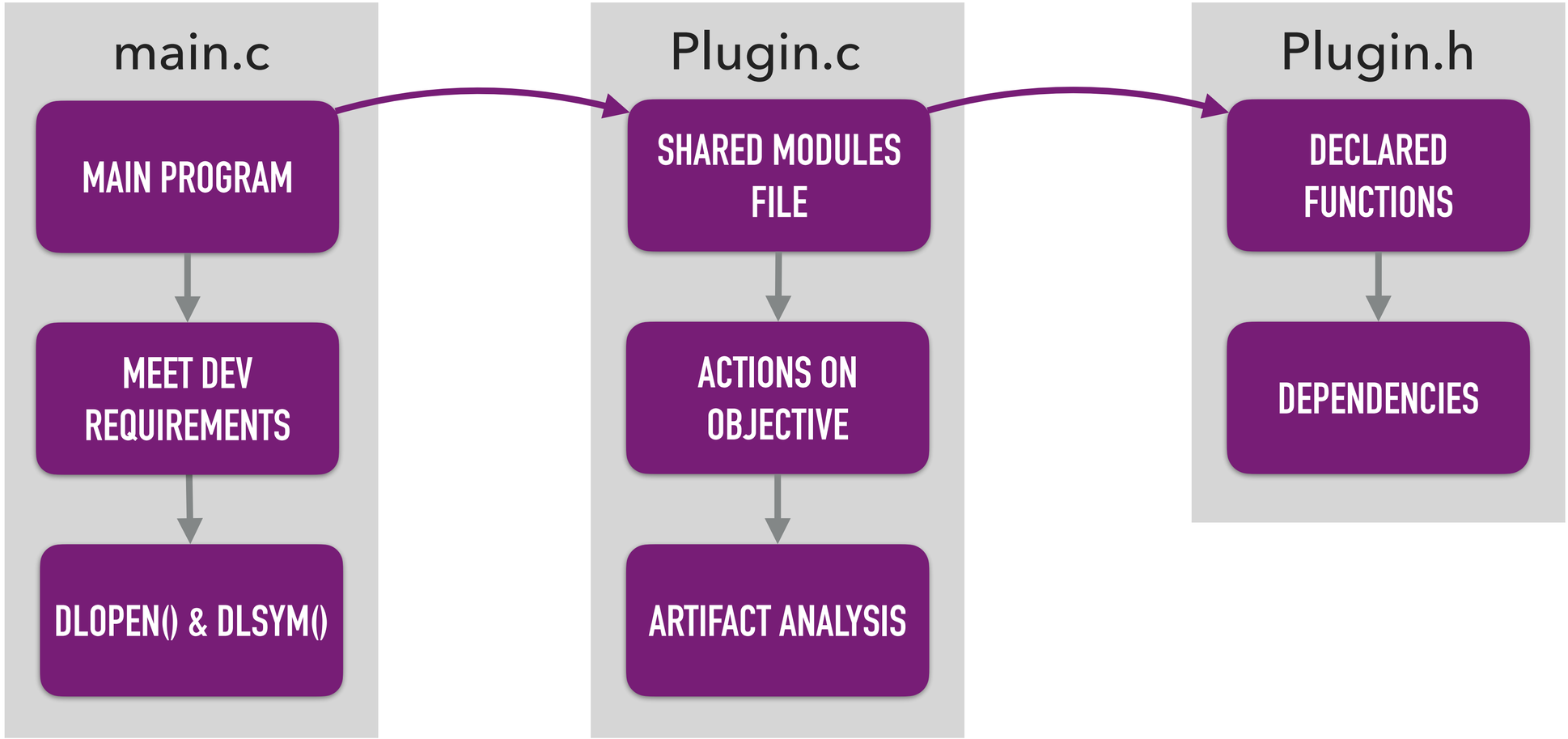
28 April 2021

Alex Turing & Hui Wang

[RotaJakiro: A long live secret backdoor with 0 VT detection]

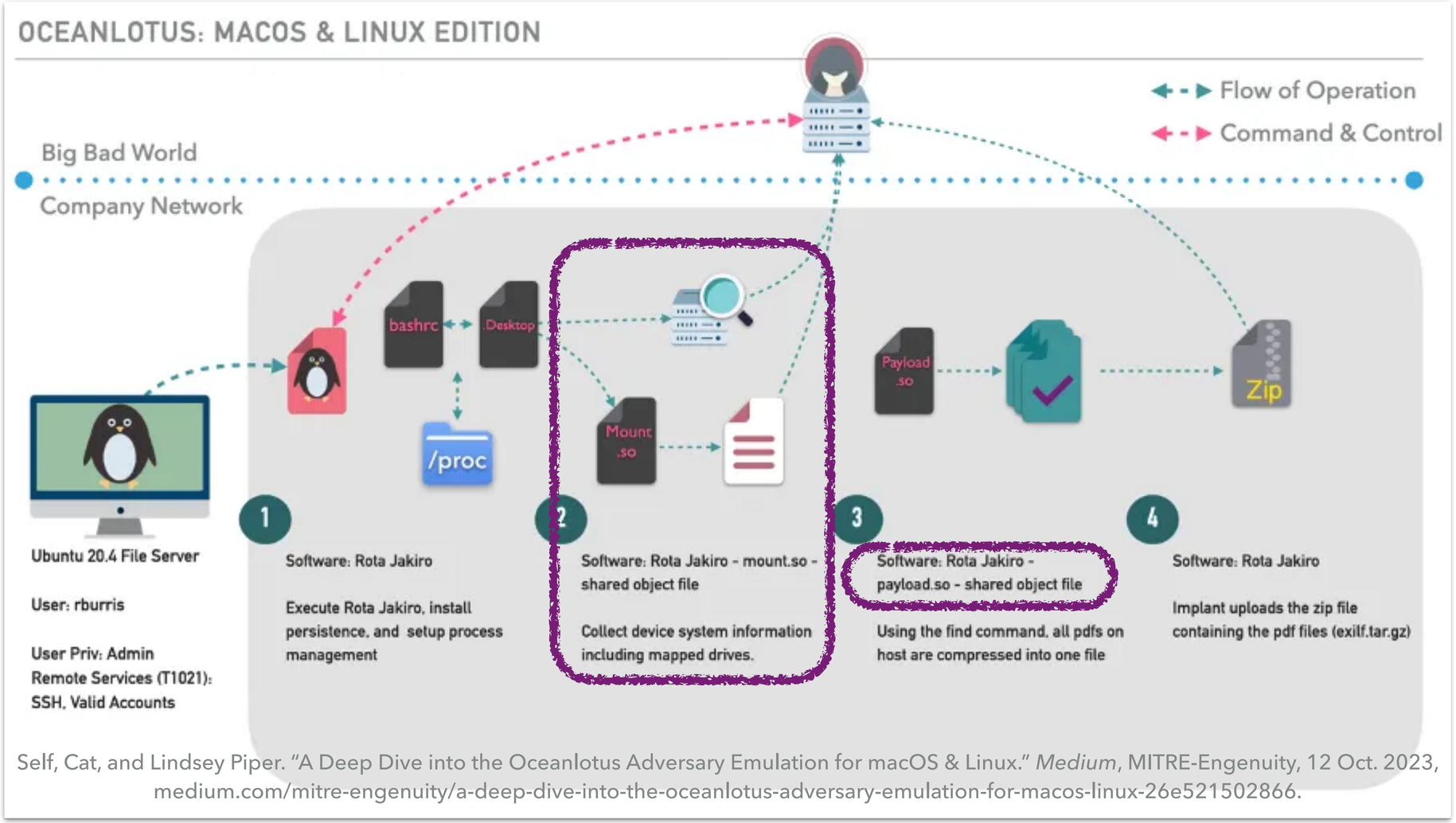
MITRE

LINUX PROGRAMING OUTLINE



MITRE

PROCEDURE FLOW



Self, Cat, and Lindsey Piper. "A Deep Dive into the Oceanlotus Adversary Emulation for macOS & Linux." *Medium*, MITRE-Engenuity, 12 Oct. 2023, medium.com/mitre-engenuity/a-deep-dive-into-the-oceanlotus-adversary-emulation-for-macos-linux-26e521502866.

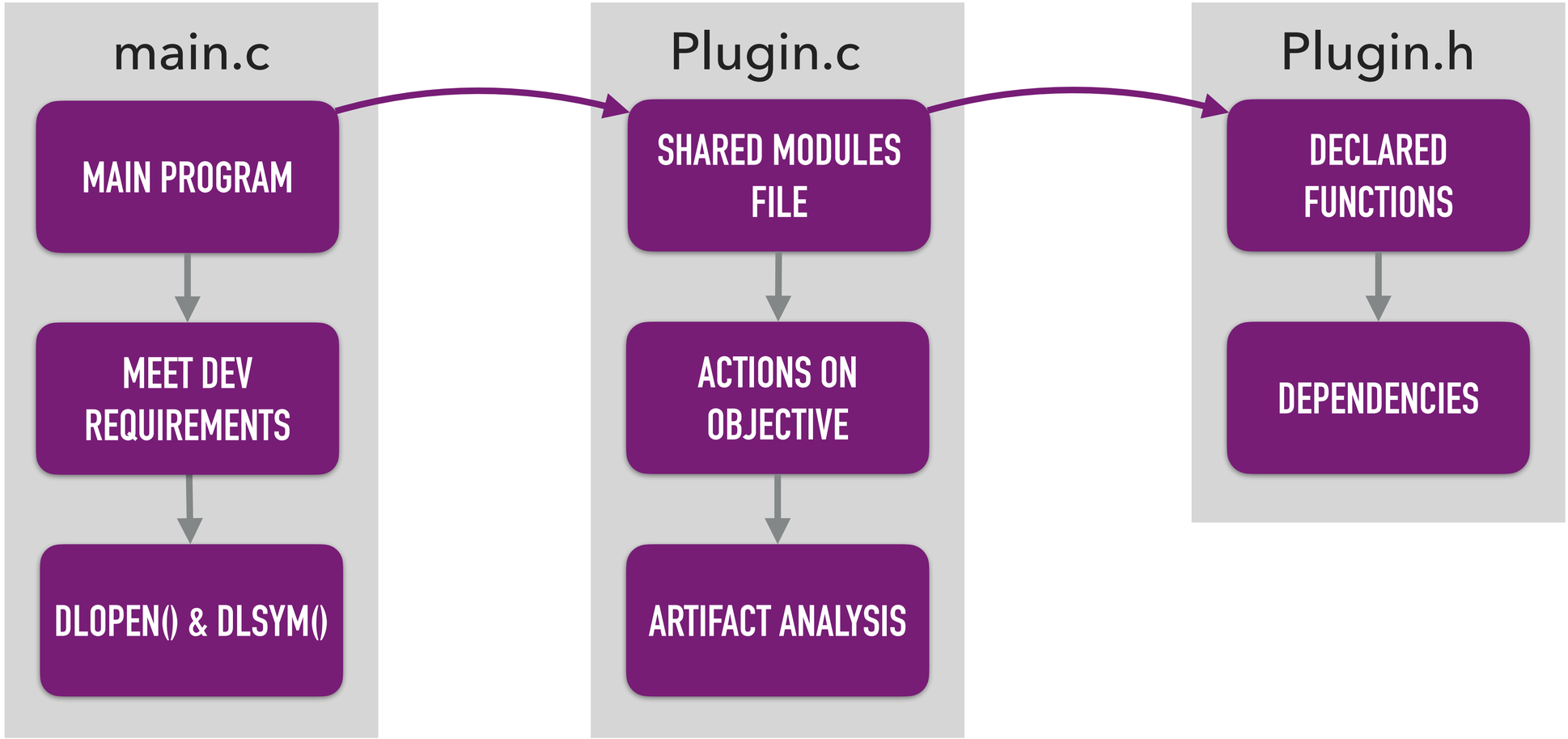
MITRE

EXERCISE: WRITE OUT IN PSEUDO CODE OR DIAGRAMS AN OUTLINE OF A PROGRAM TO EXECUTE THE T1129 SHARED MODULES MITRE ATT&CK TECHNIQUE

- What components need to be built?
- Think through the logic flow of what is needed to perform the T1129 & how to confirm it was executed
- Hint: .dylib is the macOS version of a .so file, how would you use dlopen & dlsym using .dylib files?

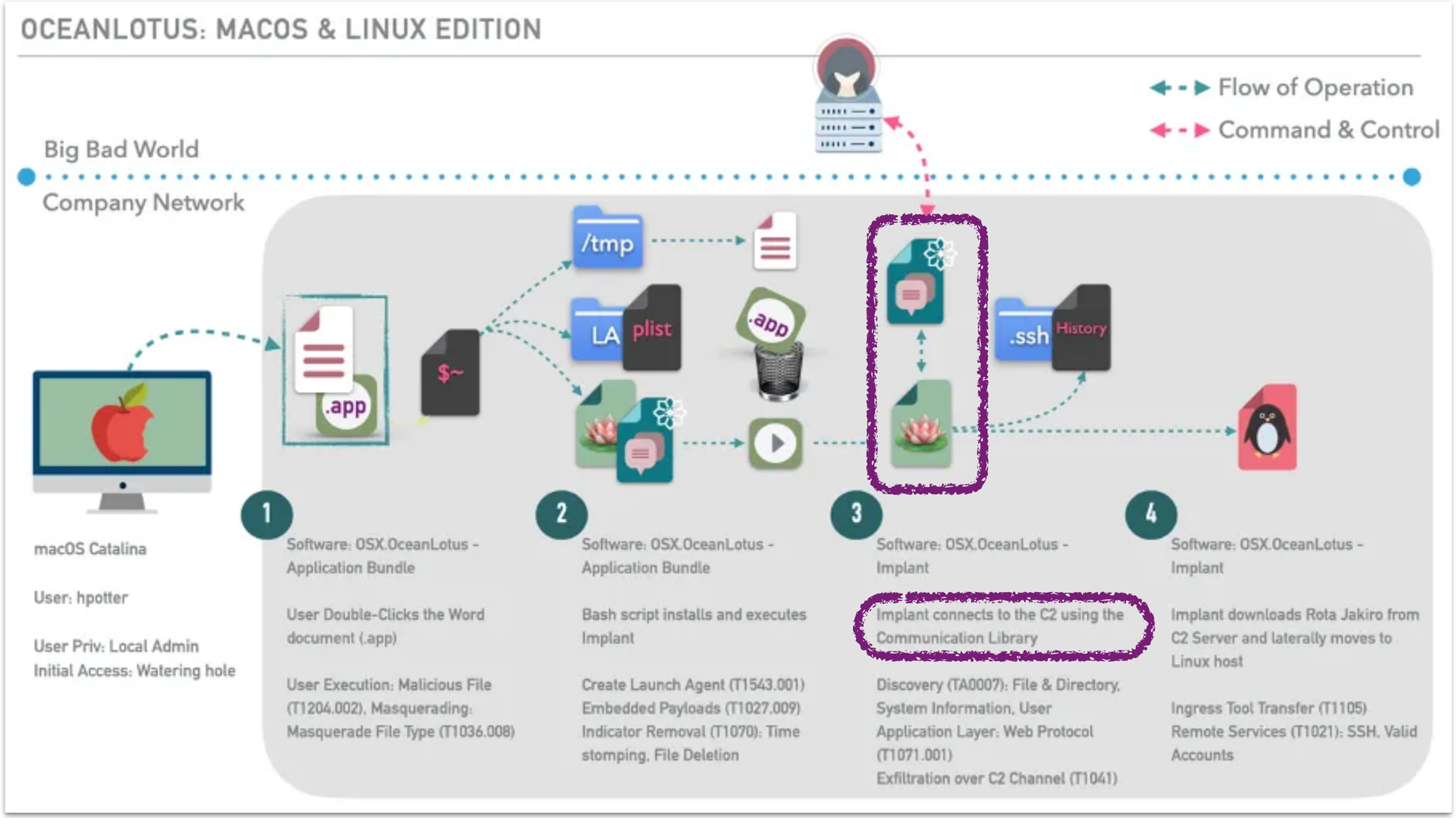
MITRE

MACOS PROGRAMING OUTLINE



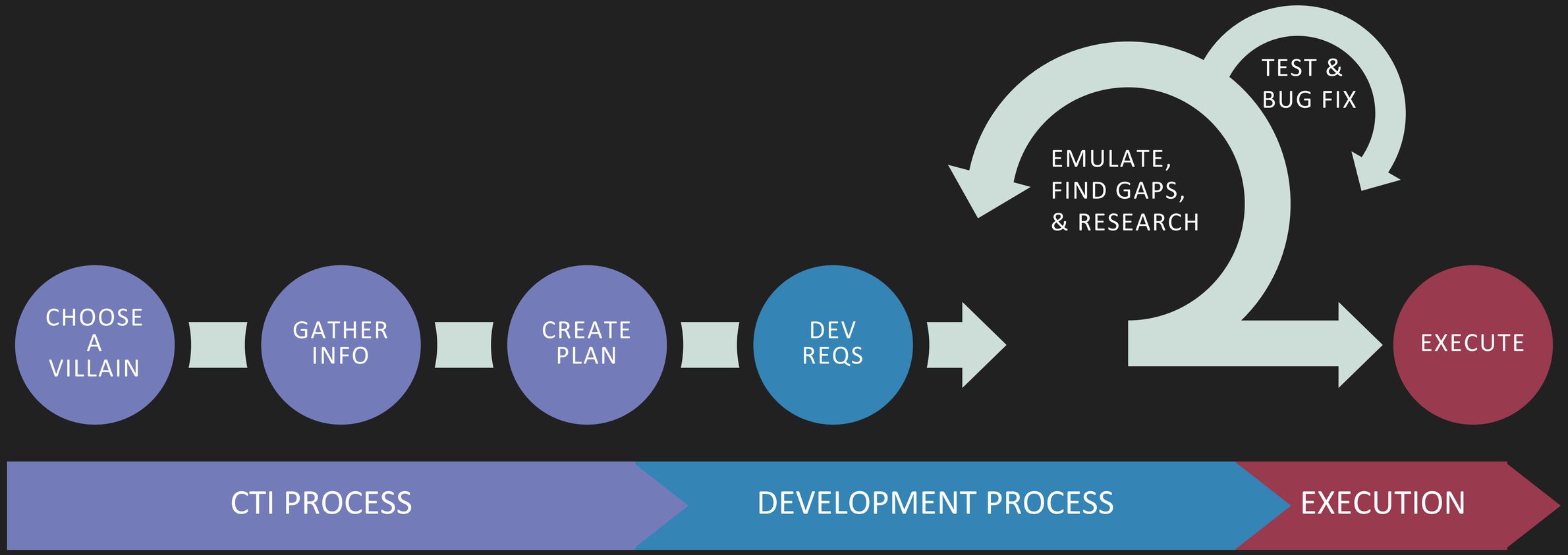
MITRE

MACOS PROCEDURE FLOW



Self, Cat, and Lindsey Piper. "A Deep Dive into the Oceanlotus Adversary Emulation for macOS & Linux." *Medium*, MITRE-Engenuity, 12 Oct. 2023, medium.com/mitre-engenuity/a-deep-dive-into-the-oceanlotus-adversary-emulation-for-macos-linux-26e521502866.

BECOMING A DARK KNIGHT



Cat @coolestcatiknow

Kate @phish4answers

EXERCISE: CODE A PROCEDURE FOR MACOS

Build out a small program that executes the shared module technique.

Hint: Start by recreating the Linux example, then modify to use a dylib

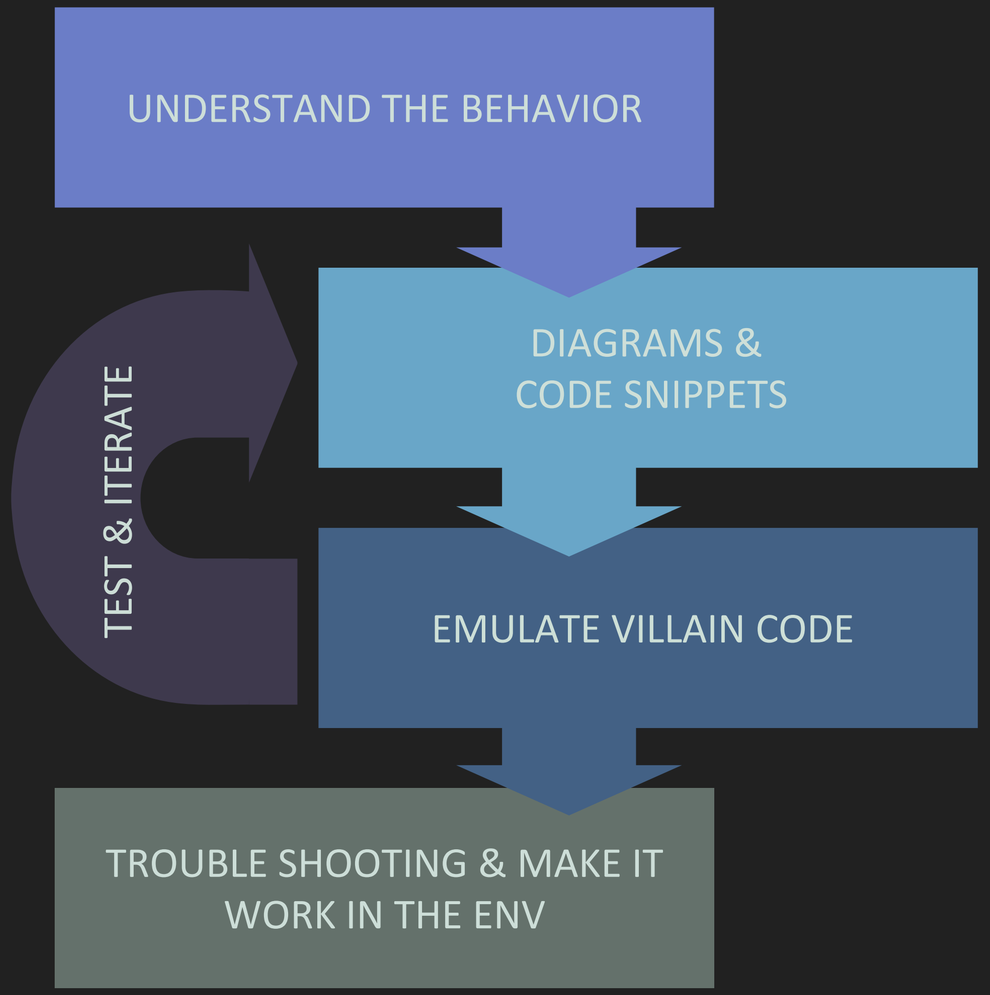
Useful Resources

1. https://github.com/center-for-threat-informed-defense/adversary_emulation_library/tree/master/ocean_lotus/Resources/OSX.OceanLotus
2. https://unit42.paloaltonetworks.com/unit42-new-improved-macos-backdoor-oceanlotus/
3. https://www.welivesecurity.com/2019/04/09/oceanlotus-macos-malware-update/

MITRE

EMULATE, FIND GAPS, & RESEARCH

BREAKING DOWN EACH STEP AKA RABBIT HOLE PROCESS

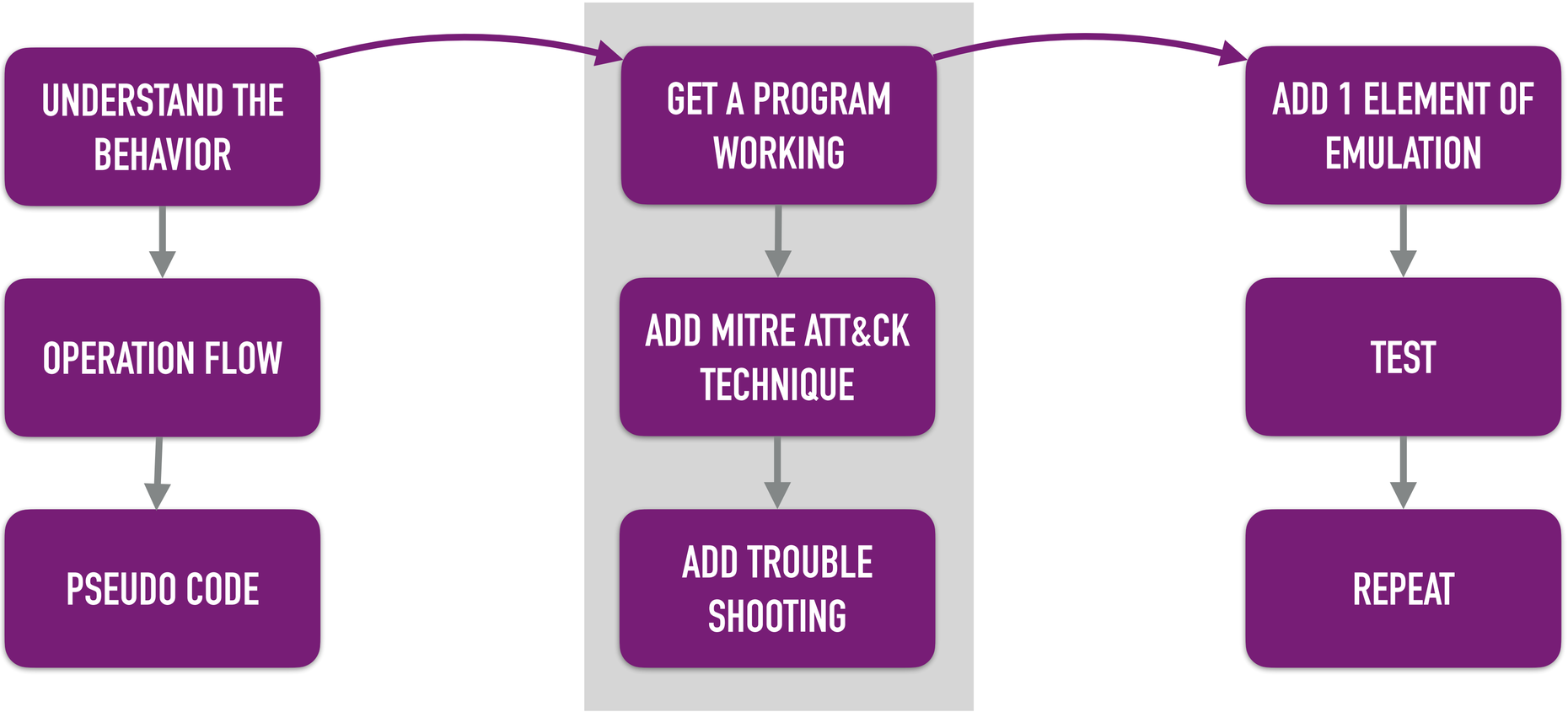


Cat @coolestcatiknow

Kate @phish4answers

MITRE ENGENUITY

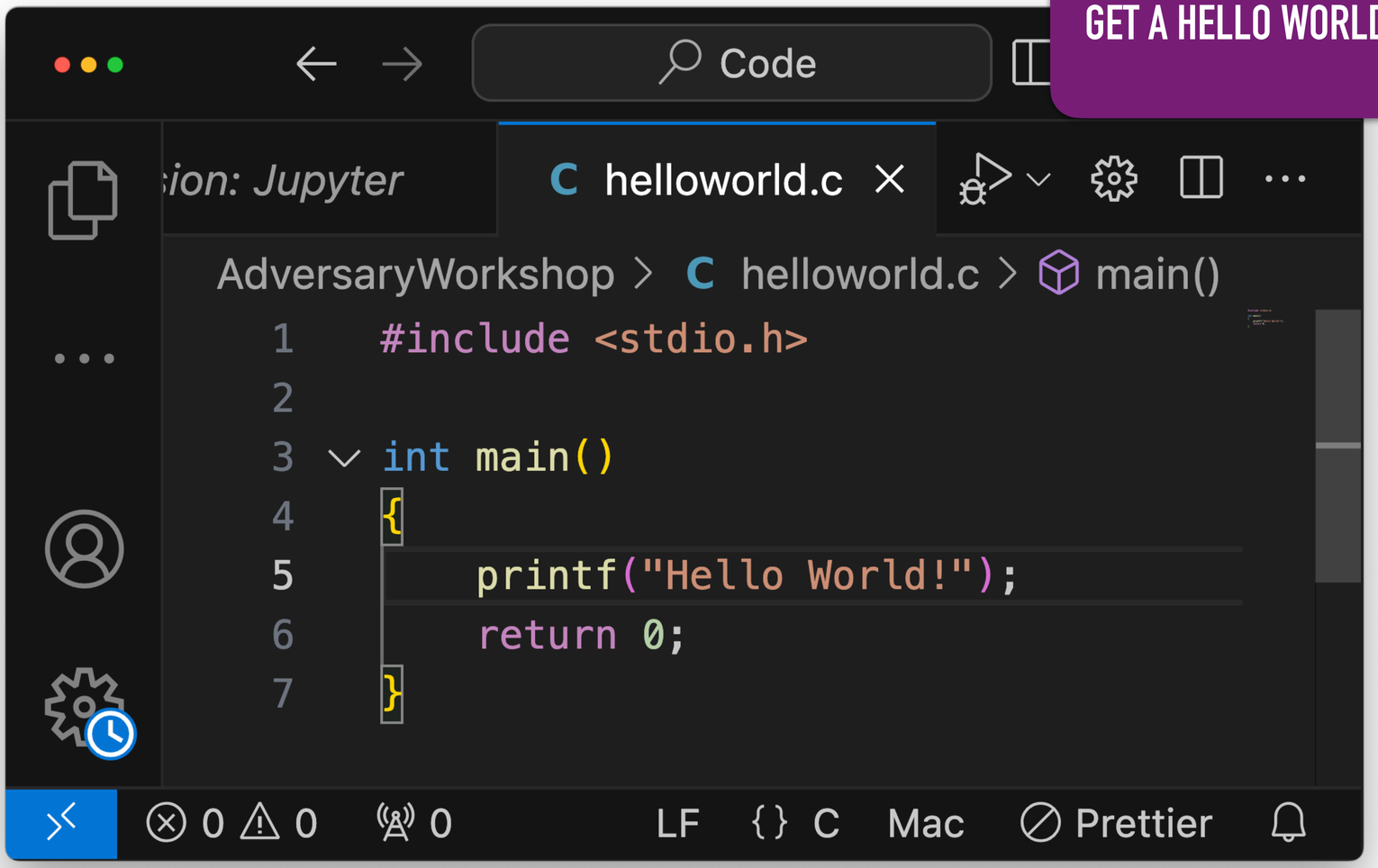
LINUX PROGRAMING PROCESS



MITRE

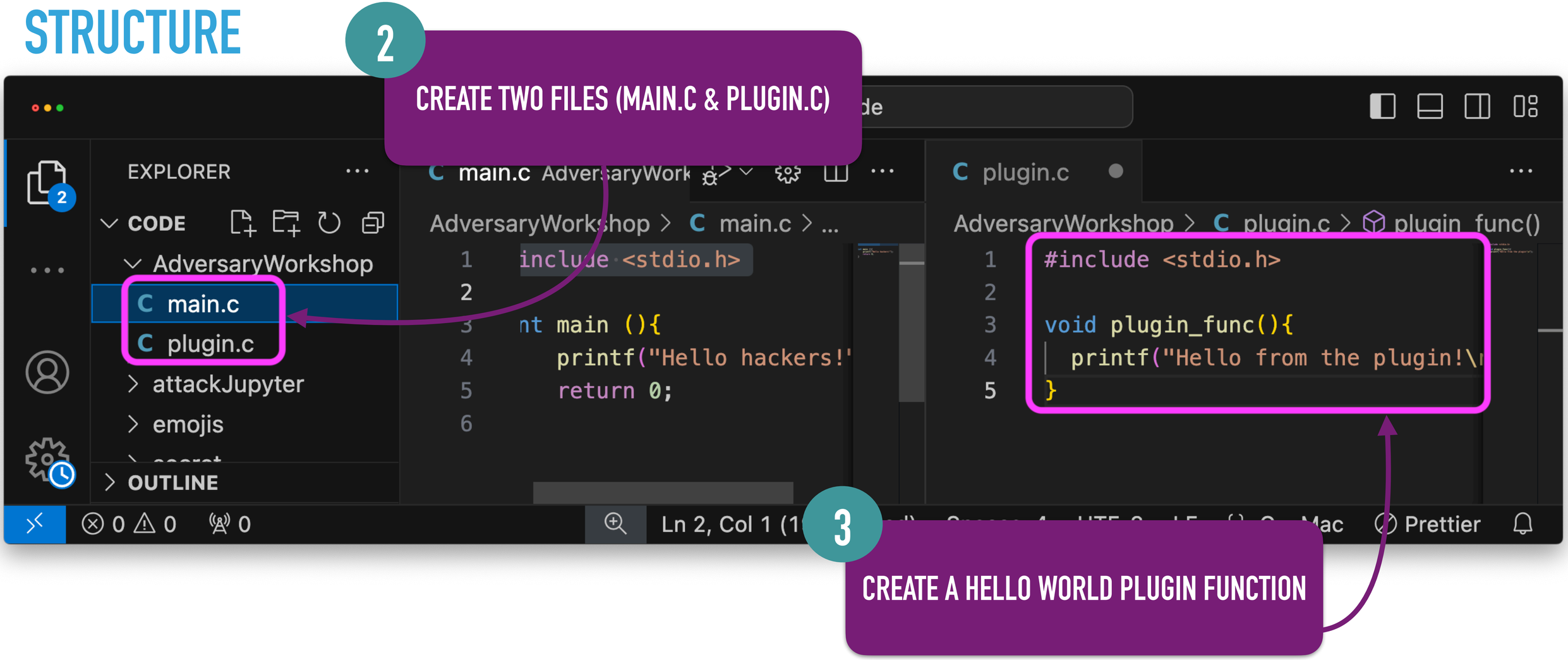
BUILD A HELLO WORLD PROGRAM

1 GET A HELLO WORLD PROGRAM WORKING



MITRE

STRUCTURE

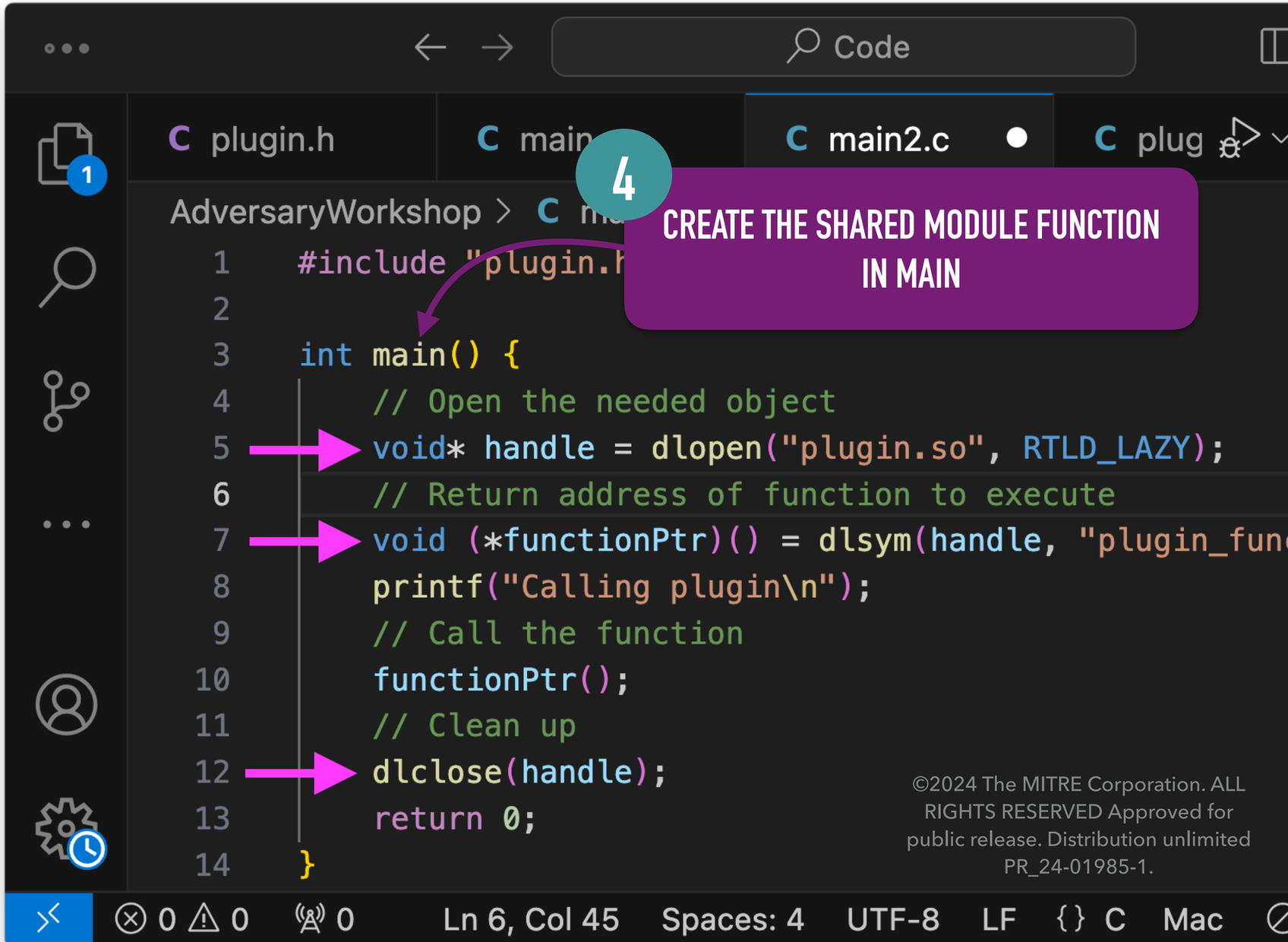


Fisher, James. “How to Make Plugins with `Dlopen`.” Jameshfisher.com, 24 Aug. 2017, jameshfisher.com/2017/08/24/dlopen/. Accessed 2 Aug. 2024.

MITRE

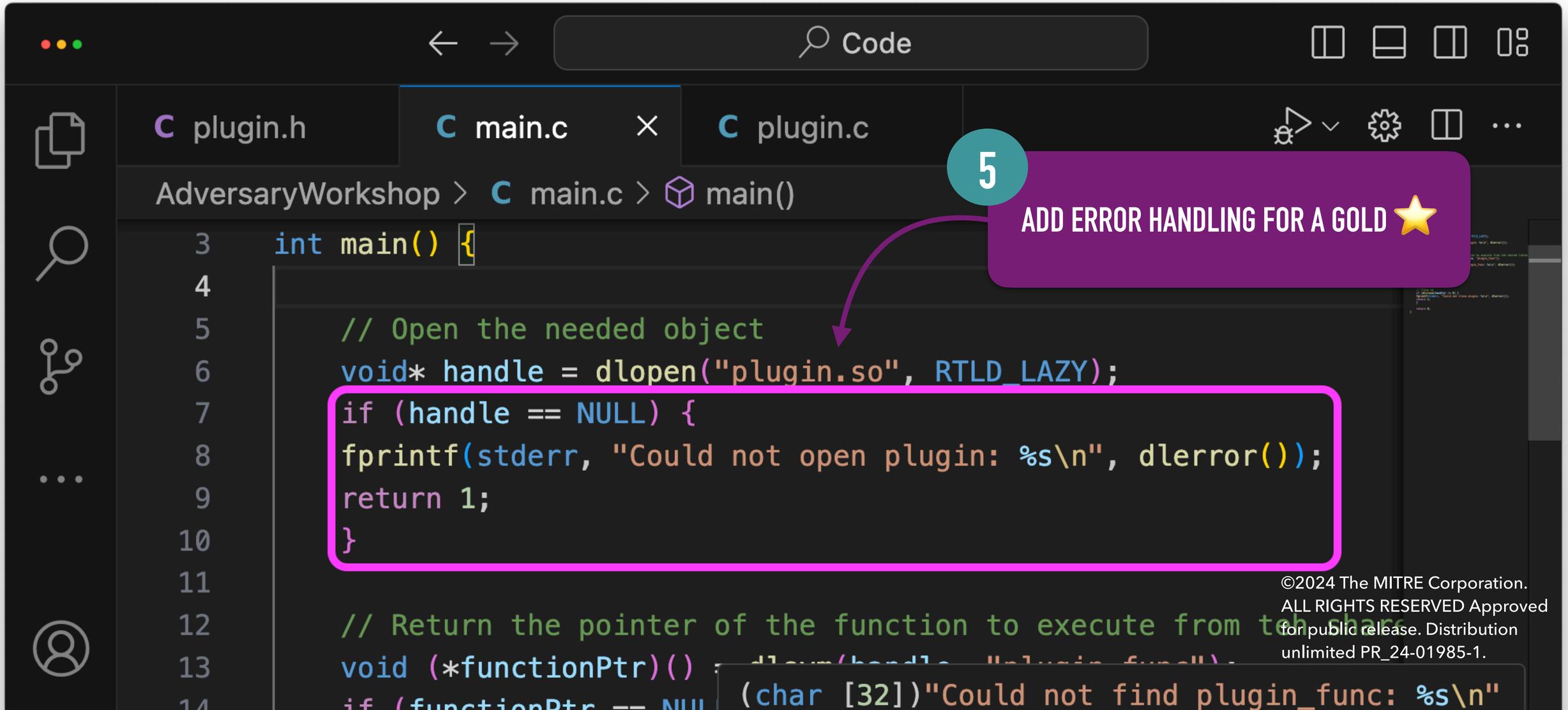
MAIN FUNCTIONALITY

- Use manuals to understand program logic - dlopen() & dlsym()
- Keep it simple - then build on



MITRE

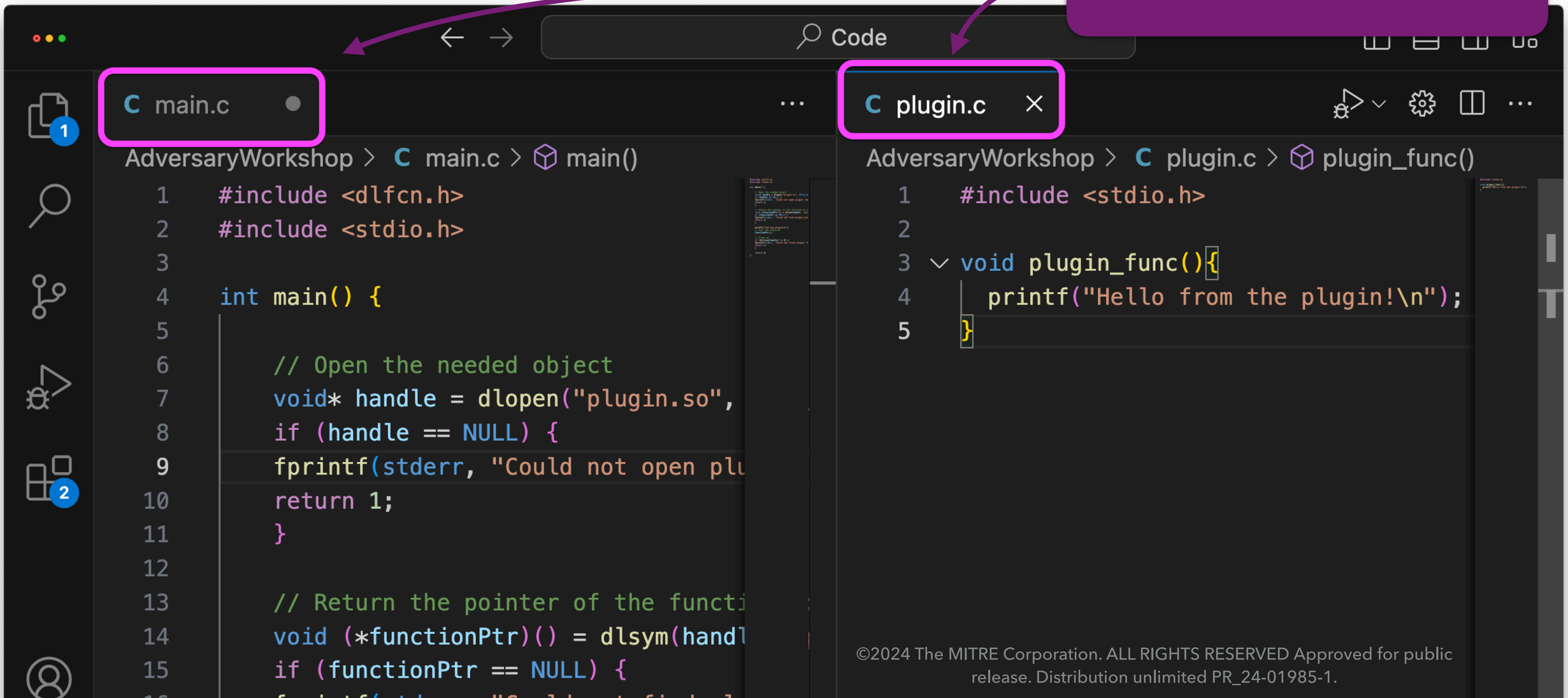
ERROR HANDLING



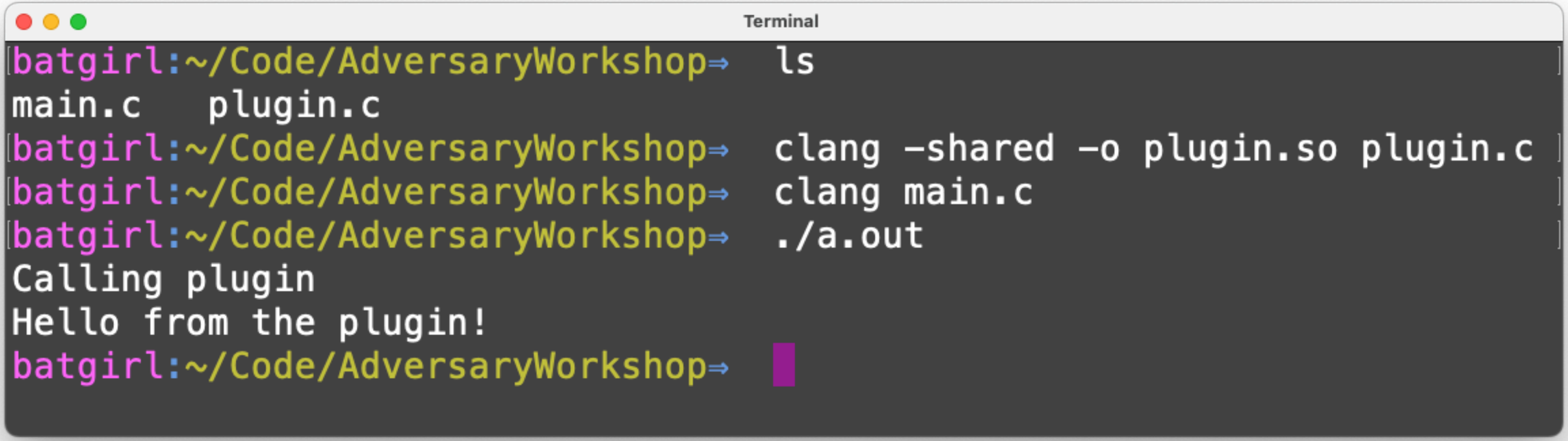
WHAT YOU SHOULD HAVE AT THIS POINT

6

TWO .C FILES

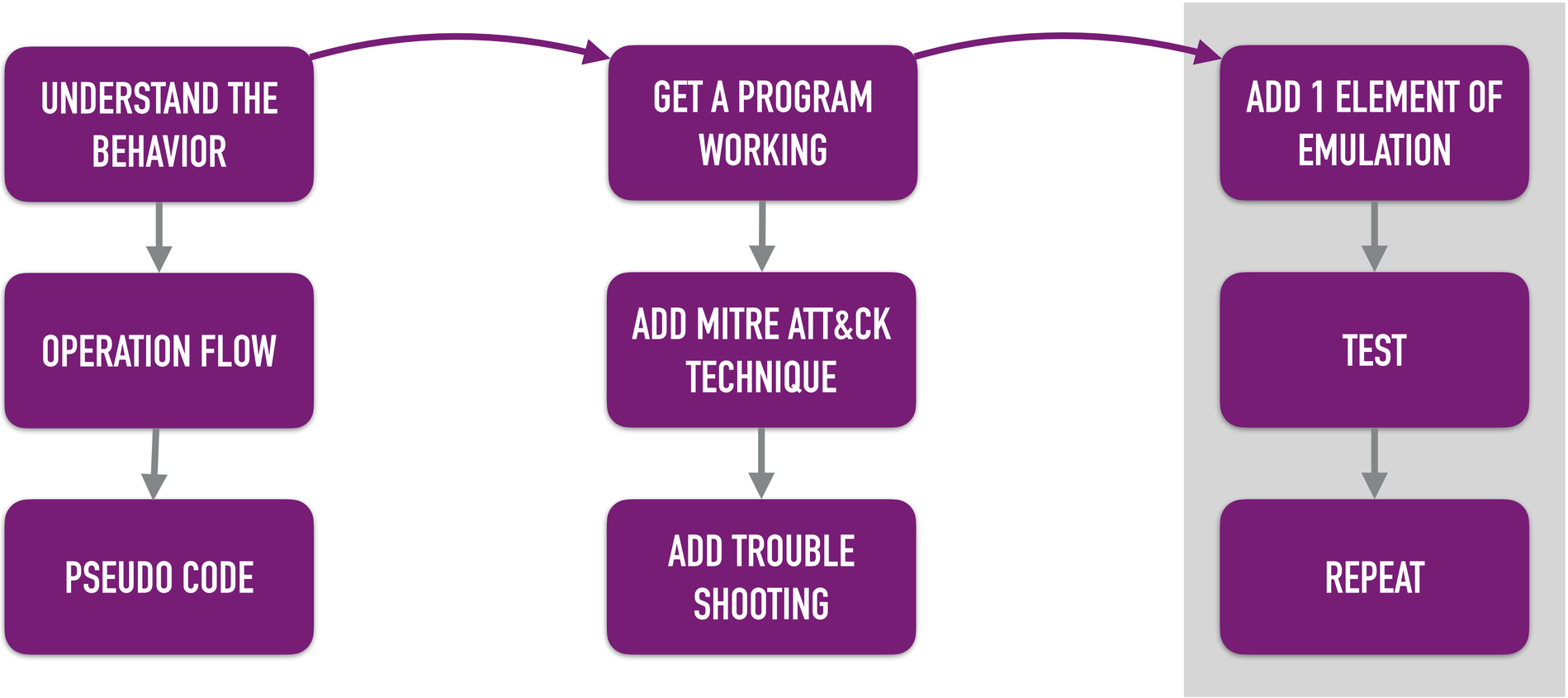


GET IT WORKING



MITRE

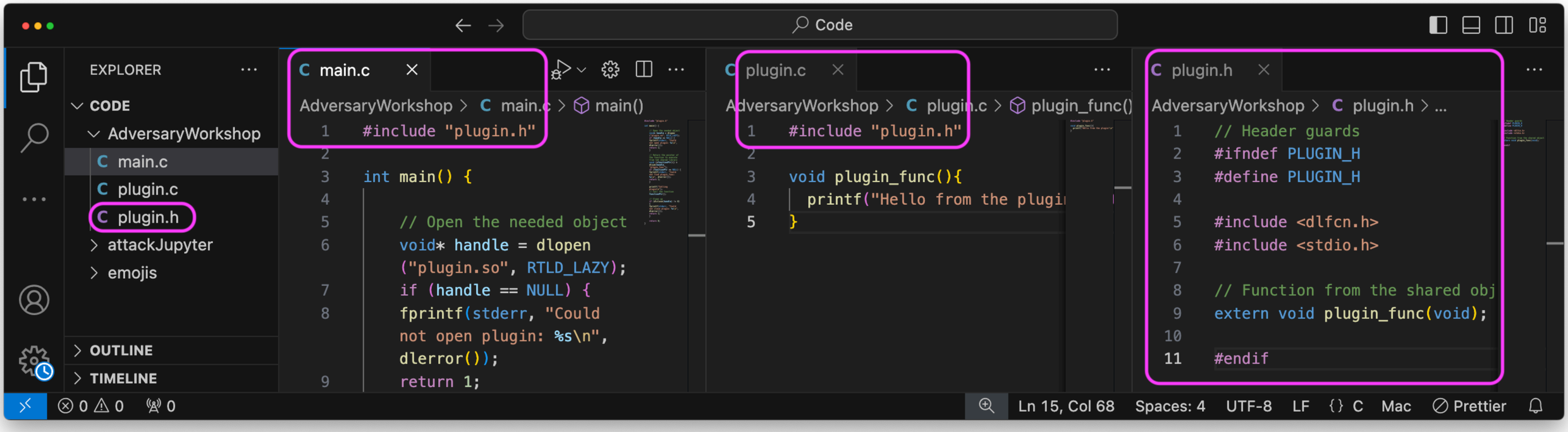
LINUX PROGRAMING PROCESS



MITRE

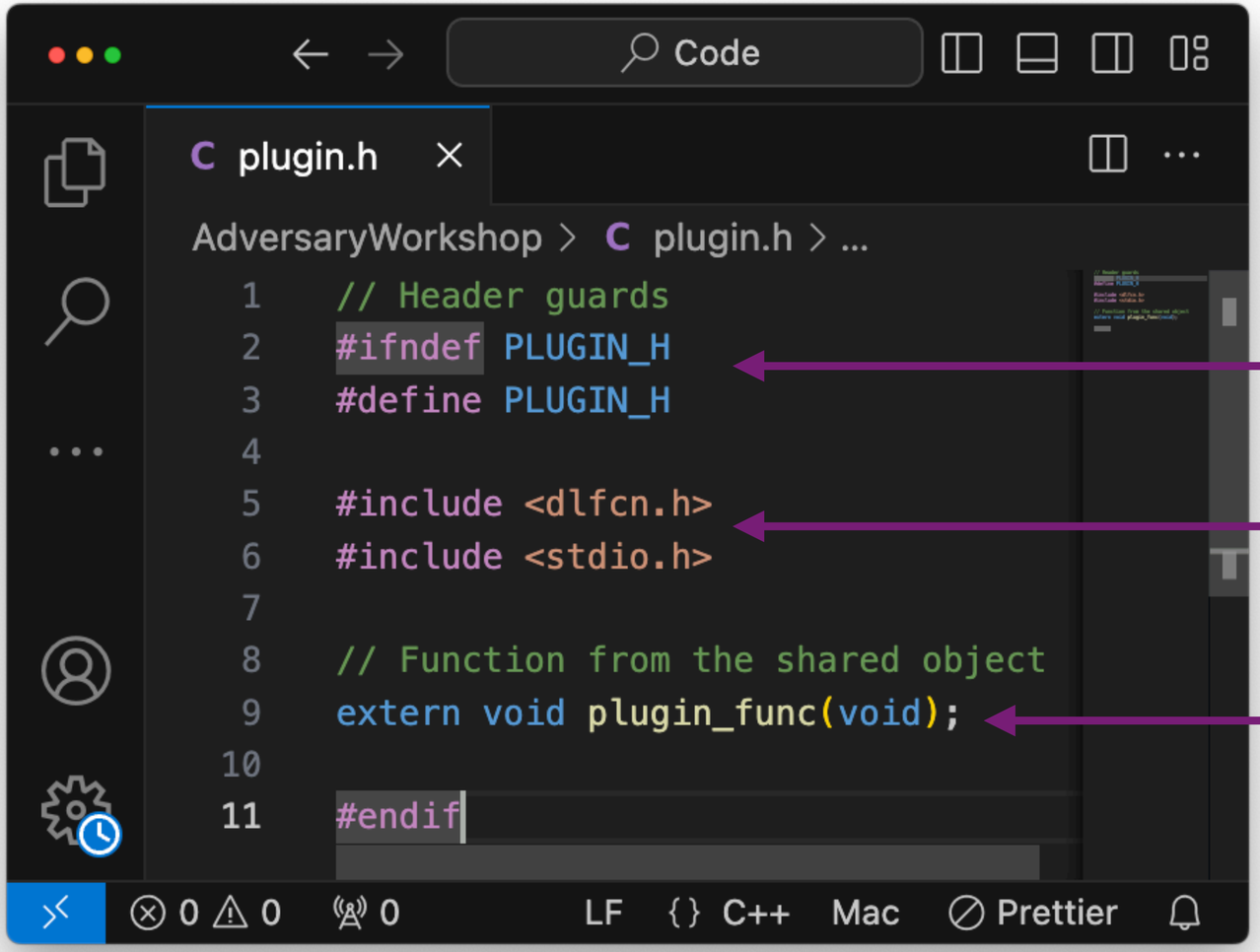
BREAK OUT THE SHARED LIBRARY WITH A HEADER FILE

CREATE A HEADER FILE → MOVE LIBRARIES → ADD HEADER GUARDS → EXPORT THE FUNCTION



MITRE

WHAT YOU SHOULD HAVE AT THIS POINT



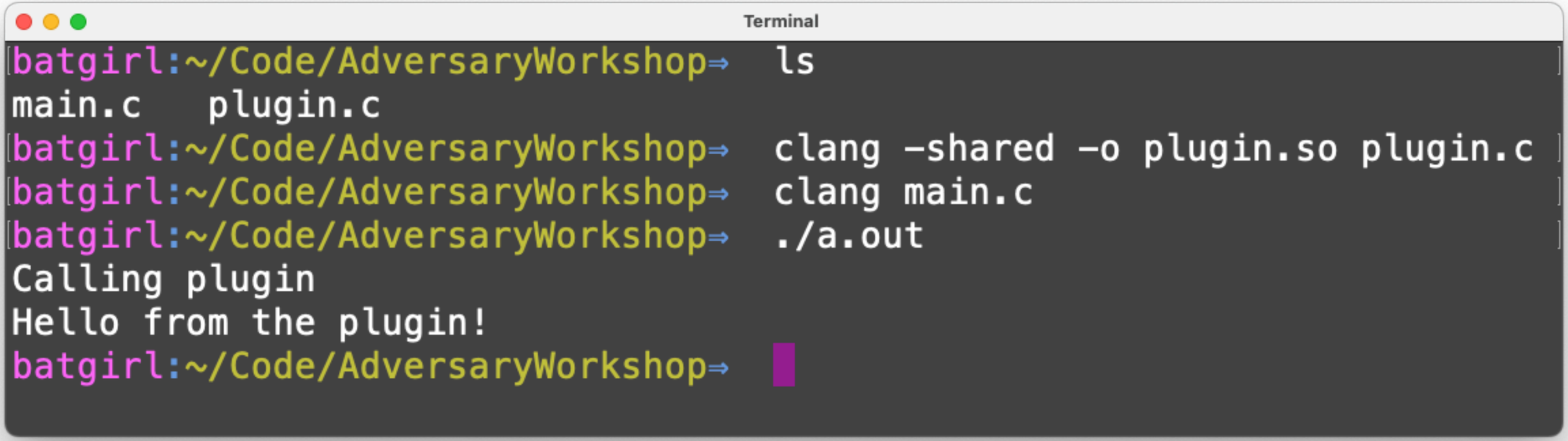
HEADER GUARDS

DEPENDENCIES

FUNCTION

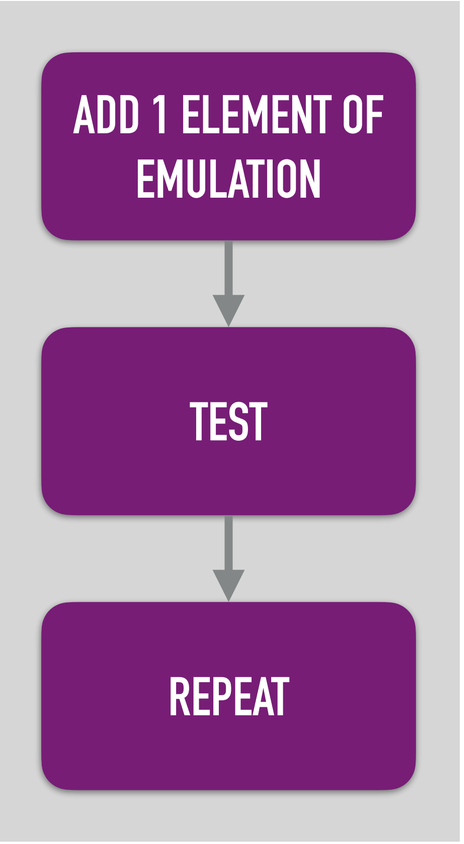
MITRE

GET IT WORKING



MITRE

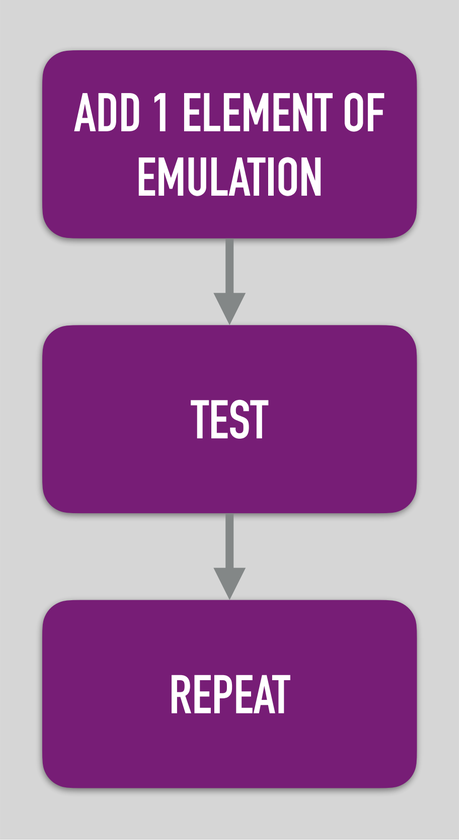
BUILD THE SHARED MODULE



https://github.com/center-for-threat-informed-defense/adversary_emulation_library/blob/master/ocean_lotus/Resources/rota/src/c2_commands.c#L153

```c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>

void create_dir() {
    system("mkdir -p /tmp/.rota;");
}

void copy_pdfs() {
    system("find /home/ -name *.pdf -exec cp {} /tmp/.rota \\; 2>/dev/null");
}

void tar() {
        system("cd /tmp/ && tar -czvf rota.tar.gz /tmp/.rota");

}

// this is loaded via dlsym.
// the argument to the rota_c2_run_plugin_1 command
extern void update(void) {
    create_dir();
    copy_pdfs();
    sleep(10);
```

BUILD THE SHARED MODULE



github.com/center-for-threat-i

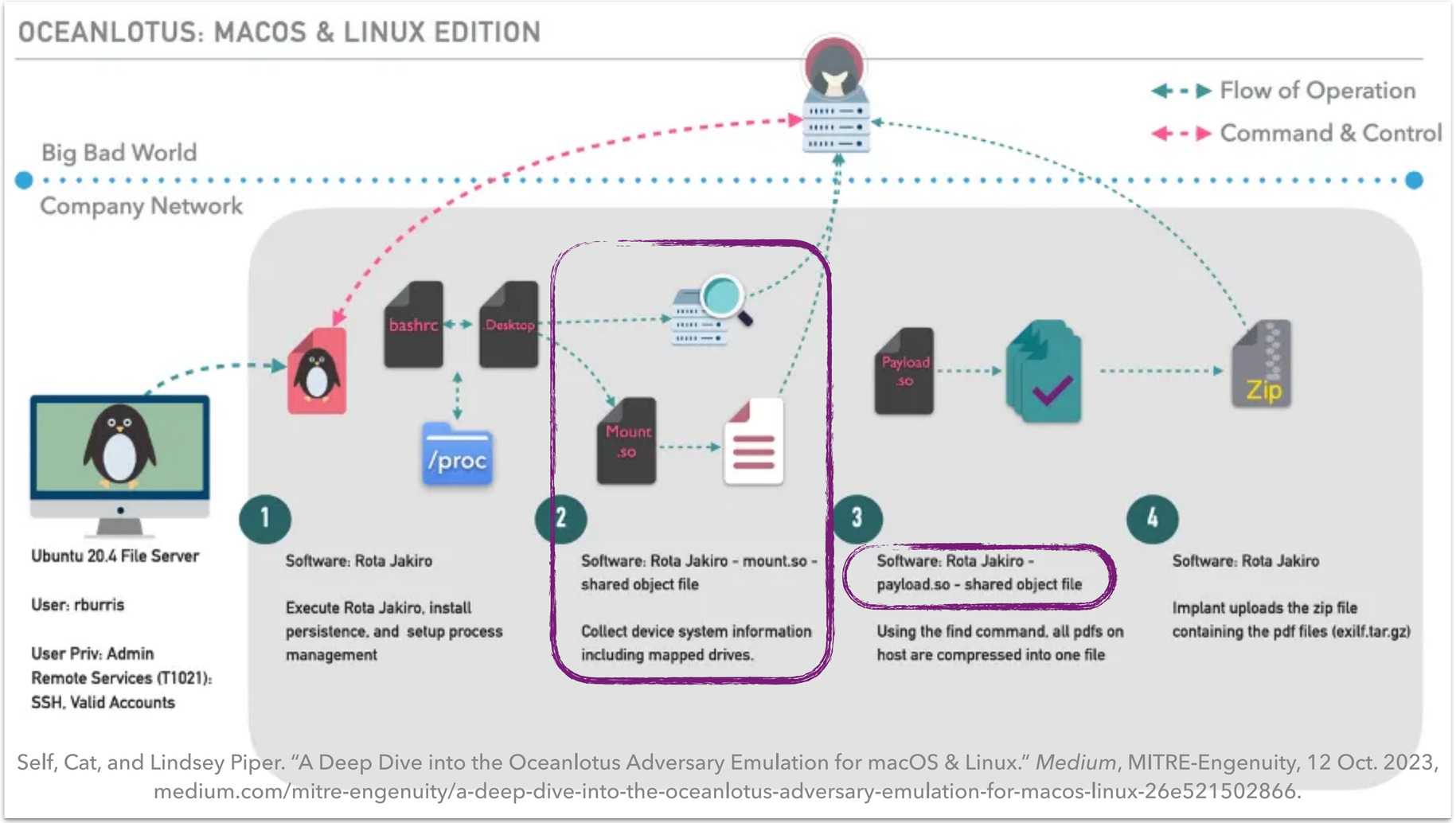
1 of 13 matches Begins with

master adversary_emulation_library / ocean_lotus / Resour / c2_commands.c

Code Blame https://github.com/center-for-threat-informed-defense/adversary_emulation_library/blob/master/ocean_lotus/Resources/rota/src/c2_commands.c#L153

```
145
146     void c2_run_plugin_1(char *funcName) {
147
148         // SO expected to be uploaded via rota_c2_upload_fi
149         char *soPath = "./local_rota_file.so";
150         void *handle = dlopen(soPath, RTLD_LAZY);
151         void (*func_ptr)() = dlsym(handle, funcName);
152         // execution of shared object
153         func_ptr();
154         dlclose(handle);
155     }
```

PROCEDURE FLOW



Self, Cat, and Lindsey Piper. "A Deep Dive into the Oceanlotus Adversary Emulation for macOS & Linux." *Medium*, MITRE-Engenuity, 12 Oct. 2023, medium.com/mitre-engenuity/a-deep-dive-into-the-oceanlotus-adversary-emulation-for-macos-linux-26e521502866.

MITRE

EXERCISE: CODE A PROCEDURE FOR MACOS

Build out a small program that executes the shared module technique.

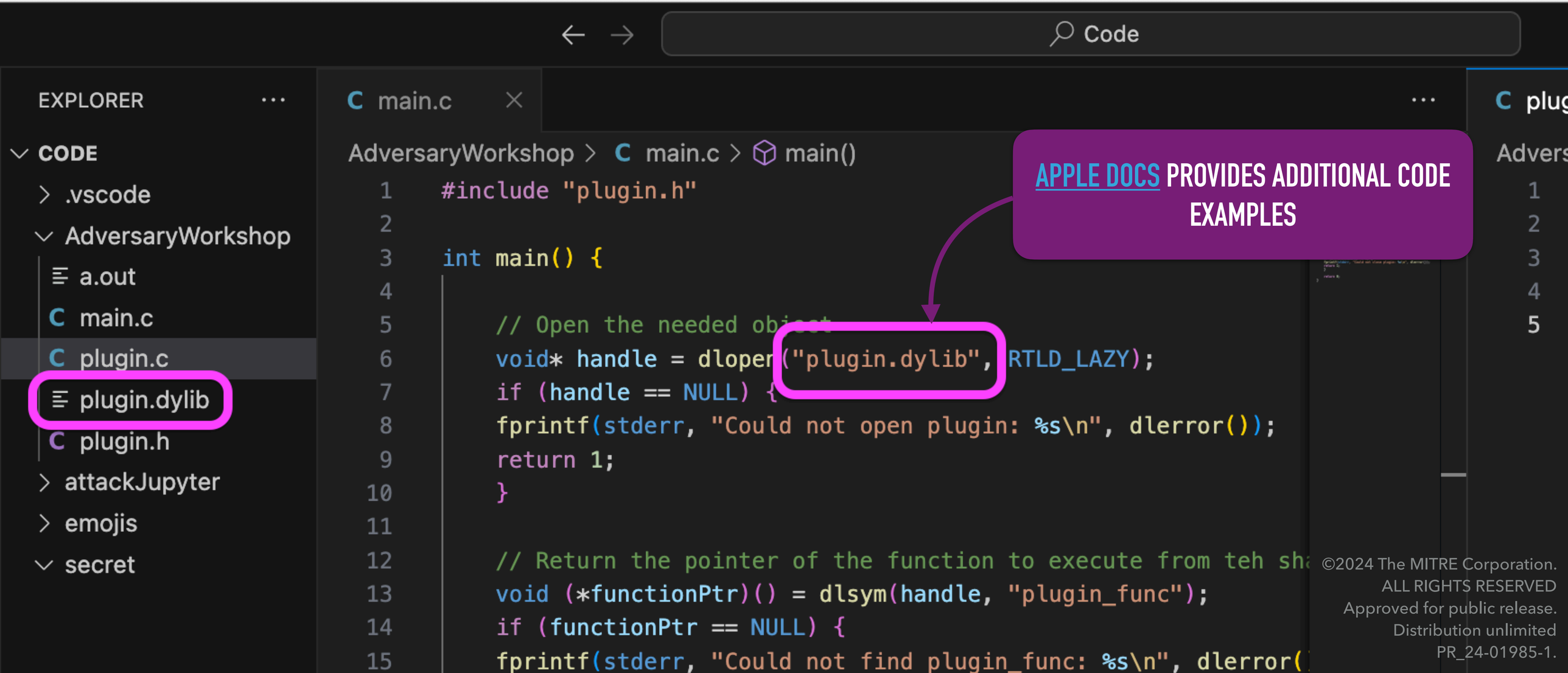
Hint: Start by recreating the Linux example, then modify to use a dylib

Useful Resources

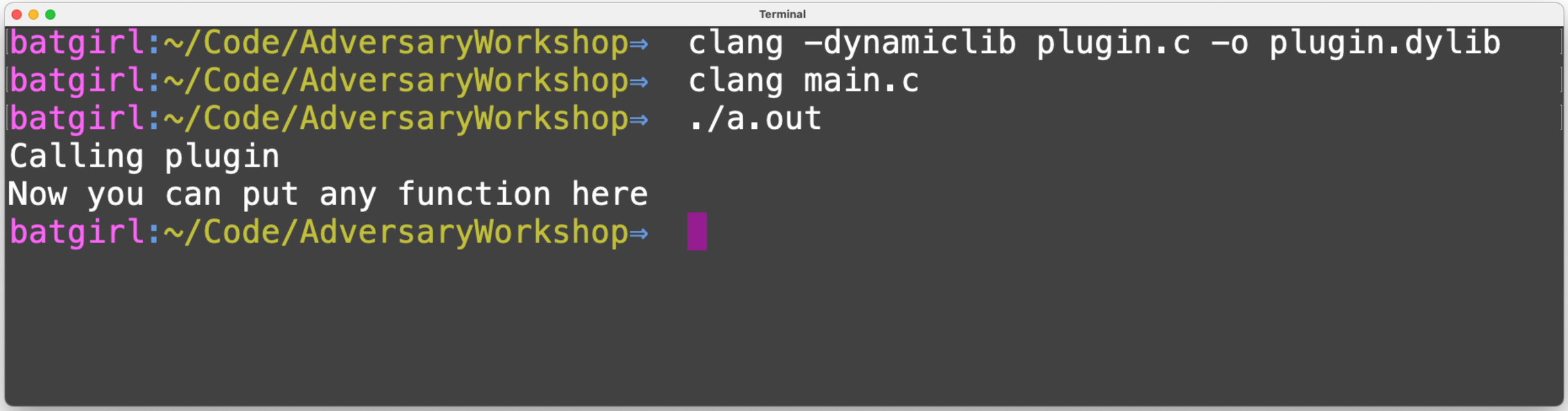
1. https://github.com/center-for-threat-informed-defense/adversary_emulation_library/tree/master/ocean_lotus/Resources/OSX.OceanLotus
2. https://unit42.paloaltonetworks.com/unit42-new-improved-macos-backdoor-oceanlotus/
3. https://www.welivesecurity.com/2019/04/09/oceanlotus-macos-malware-update/

MITRE

SAME EXACT STEPS AS LINUX EXCEPT…

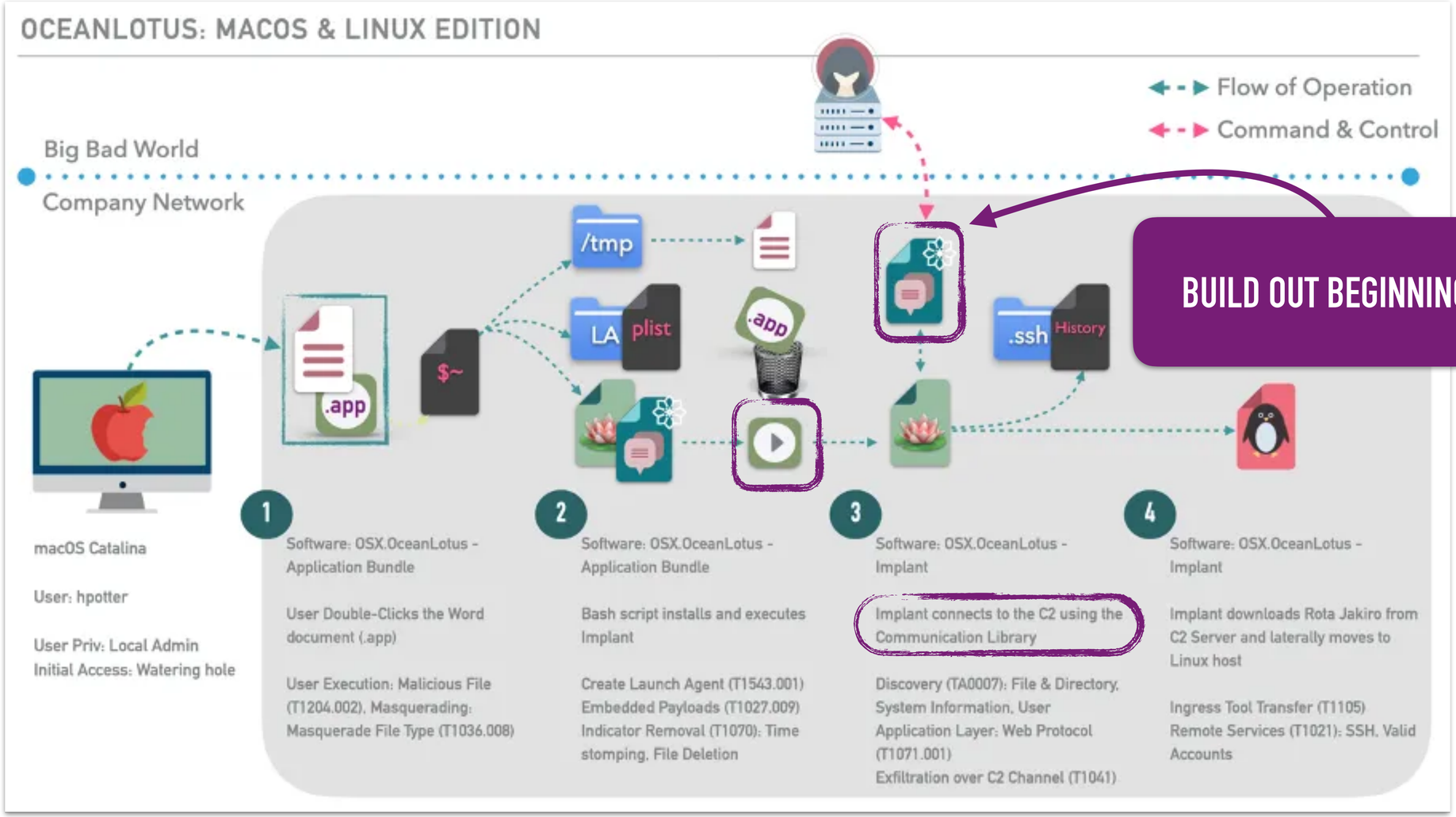


TEST IT IN YOUR MACOS VM



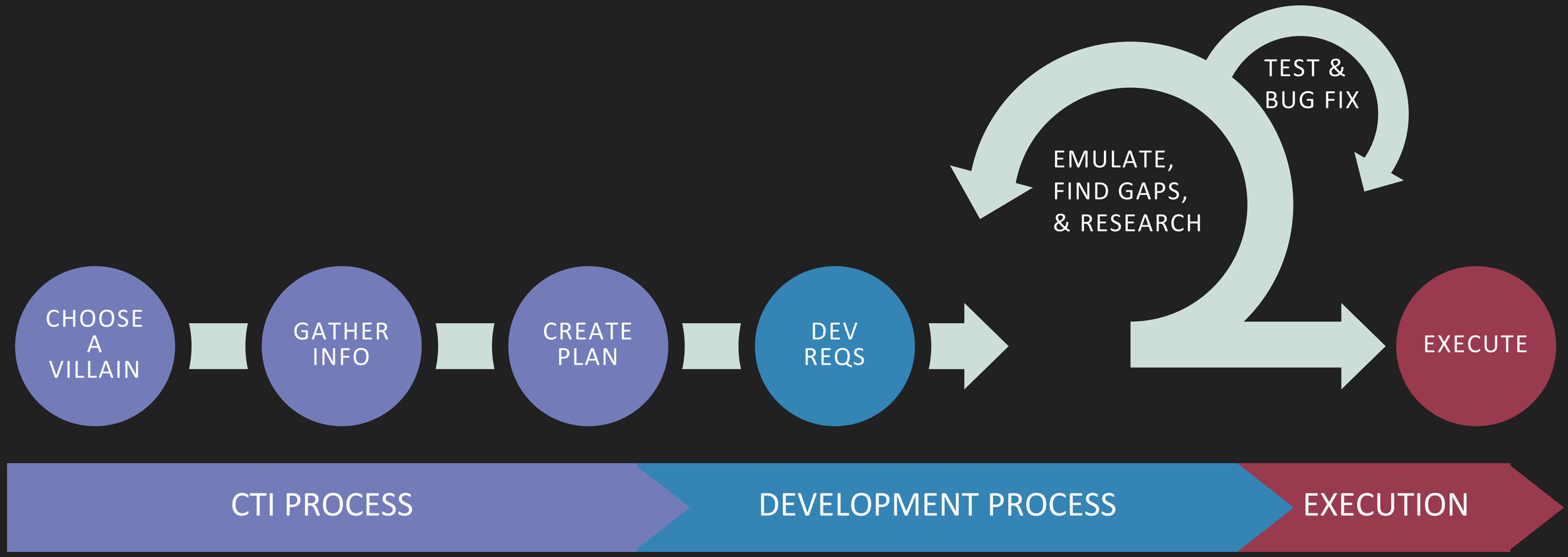
MITRE

MACOS PROCEDURE FLOW



Self, Cat, and Lindsey Piper. "A Deep Dive into the Oceanlotus Adversary Emulation for macOS & Linux." *Medium*, MITRE-Engenuity, 12 Oct. 2023, medium.com/mitre-engenuity/a-deep-dive-into-the-oceanlotus-adversary-emulation-for-macos-linux-26e521502866.

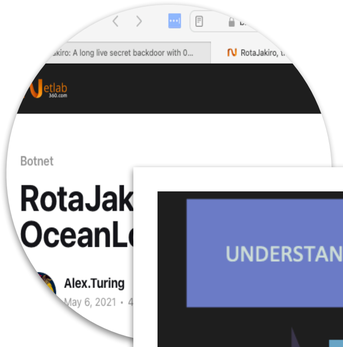
BECOMING A DARK KNIGHT

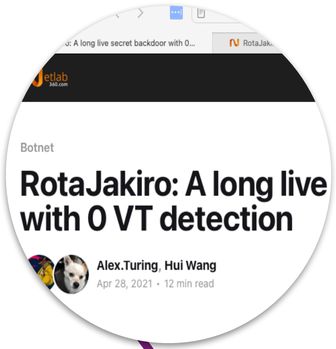


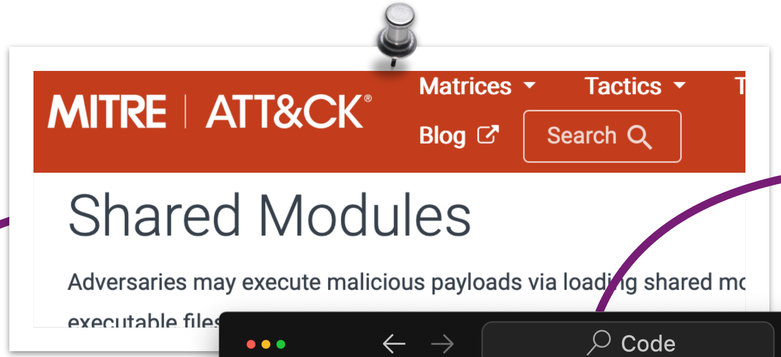
Cat @coolestcatiknow

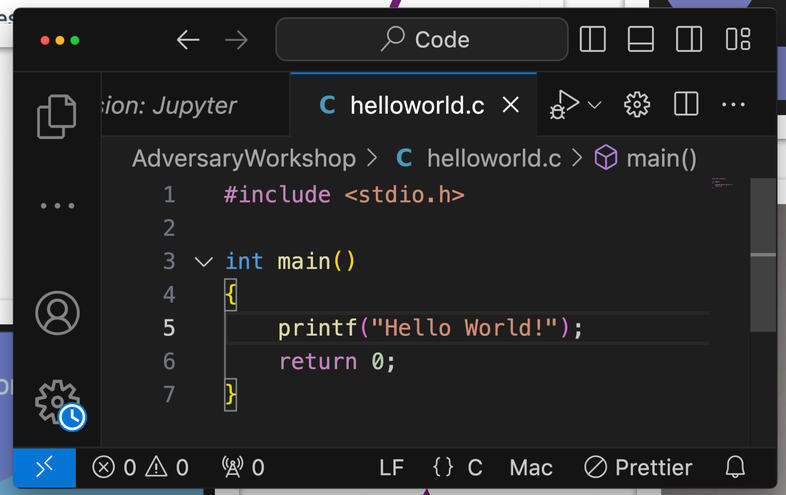
Kate @phish4answers

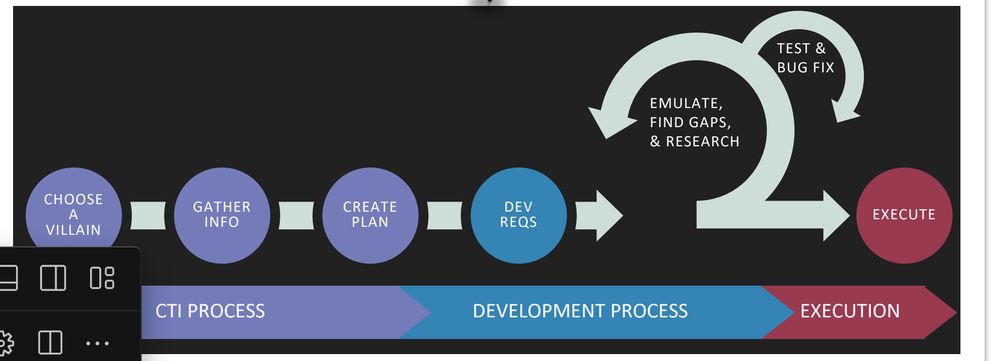
WHAT WE COVERED











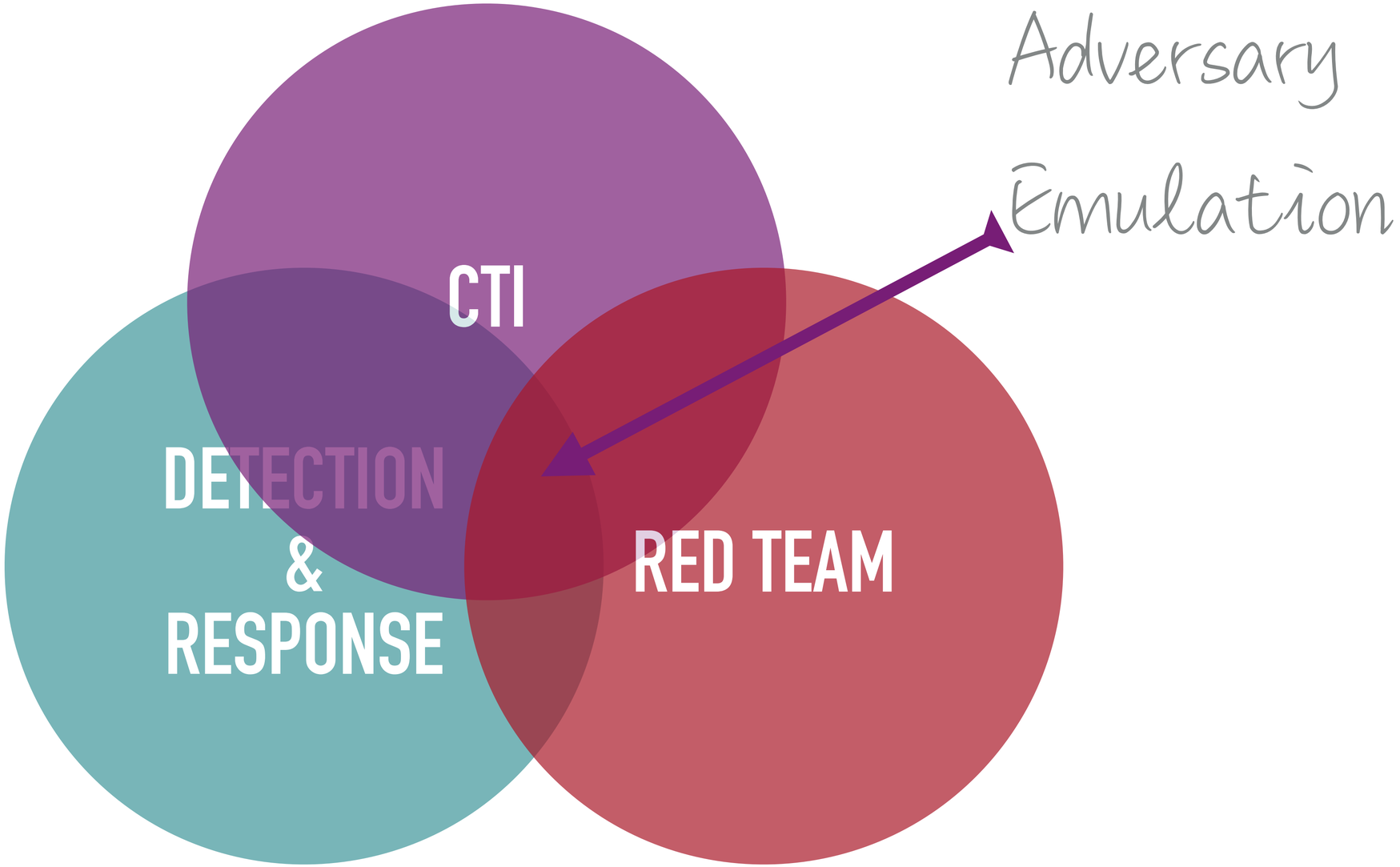
MITRE

WHY USE AE?

People

Process

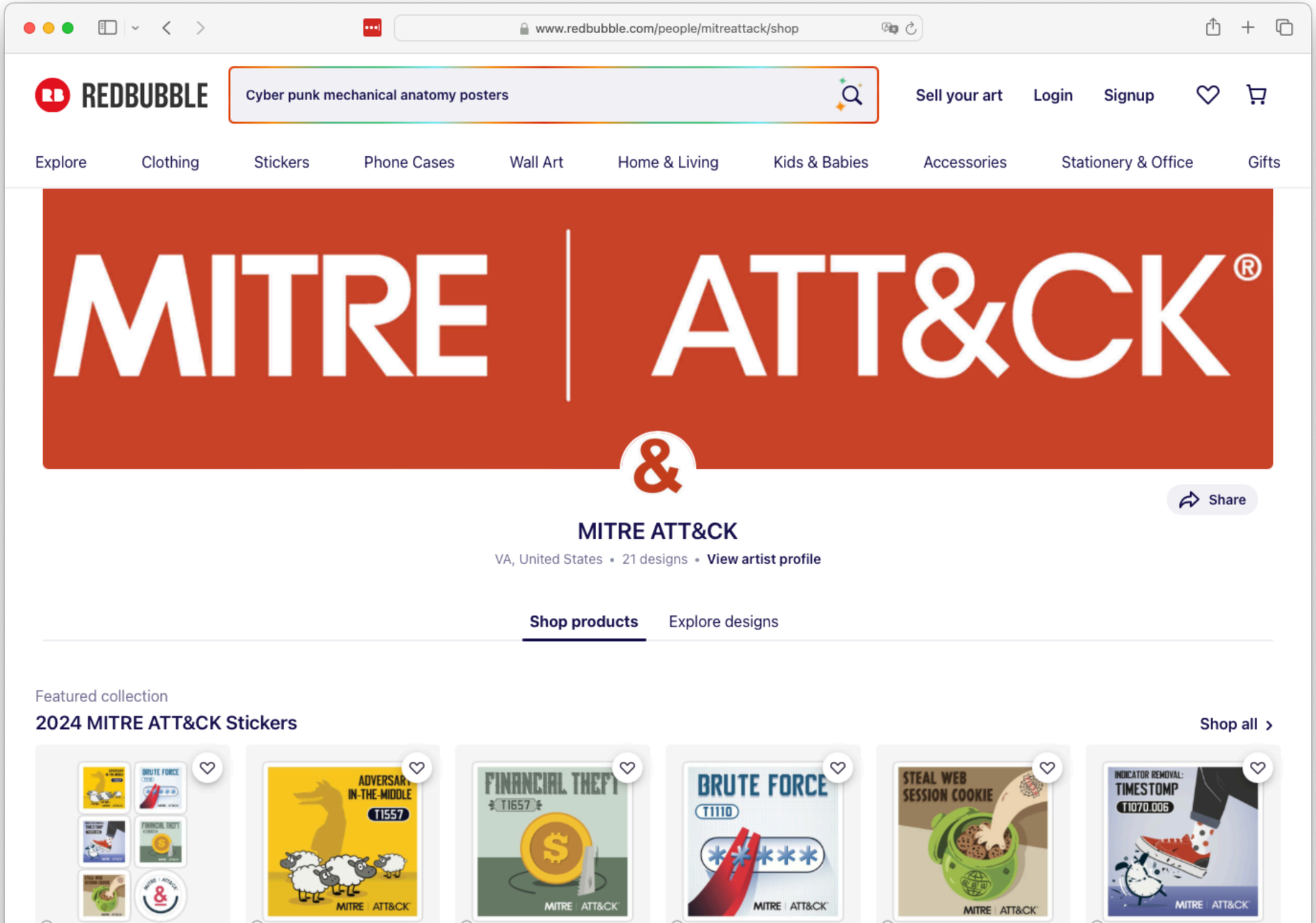
Technology



https://www.ibm.com/downloads/cas/QEBYPND1

MITRE

SWAG



MITRE

ATT&CK™?

Cat Self

attack@mitre.org

@coolestcatiknow

linkedin.com/ln/coolestcatiknow

www.catalystcode.org