

Becoming a Yogi on Mac ATT&CK with OceanLotus Postures

Objective by the Sea v4.0

Cat Self
Adam Pennington



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



- Past academic, defender, CTI analyst
- Former live sound engineer
- Part of ATT&CK since it was in Excel
- Lead MITRE ATT&CK

ATT&CK Is ...

MITRE ATT&CK® Knowledge Base



©2021 The MITRE Corporation. ALL RIGHTS RESERVED. Approved for public release. Distribution unlimited 21-00706-16



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***



ATT&CK FOR ENTERPRISE MATRIX

Reconnaissance 10 techniques	Resource Development 7 techniques	Initial Access 9 techniques	Execution 12 techniques	Persistence 19 techniques	Privilege Escalation 13 techniques	Defense Evasion 39 techniques	Credential Access 15 techniques	Discovery 27 techniques	Lateral Movement 9 techniques	Collection 17 techniques	Command and Control 16 techniques	Exfiltration 9 techniques	Impact 13 techniques
Active Scanning (2)	Acquire Infrastructure (6)	Drive-by Compromise	Command and Scripting Interpreter (9)	Account Manipulation (4)	Abuse Elevation Control Mechanism (4)	Abuse Elevation Control Mechanism (4)	Brute Force (4)	Account Discovery (4)	Exploitation of Remote Services	Archive Collected Data (3)	Application Layer Protocol (4)	Automated Exfiltration (1)	Account Access Removal
Gather Victim Host Information (4)	Compromise Accounts (2)	Exploit Public-Facing Application	Container Administration Command	BITS Jobs	Access Token Manipulation (5)	Access Token Manipulation (5)	Credentials from Password Stores (5)	Application Window Discovery	Internal Spearphishing	Audio Capture	Communication Through Removable Media	Data Transfer Size Limits	Data Destruction
Gather Victim Identity Information (3)	Compromise Infrastructure (6)	External Remote Services	Deploy Container	Boot or Logon Autostart Execution (14)	Boot or Logon Autostart Execution (14)	Build Image on Host	Exploitation for Credential Access	Browser Bookmark Discovery	Lateral Tool Transfer	Automated Collection	Data Encoding (2)	Exfiltration Over Alternative Protocol (3)	Data Encrypted for Impact
Gather Victim Network Information (5)	Develop Capabilities (4)	Hardware Additions	Exploitation for Client Execution	Boot or Logon Initialization Scripts (3)	Boot or Logon Initialization Scripts (5)	Deobfuscate/Decode Files or Information	Forced Authentication	Cloud Infrastructure Discovery	Remote Service Session Hijacking (2)	Clipboard Data	Data from Cloud Storage Object	Exfiltration Over C2 Channel	Data Manipulation (3)
Gather Victim Org Information (4)	Establish Accounts (2)	Phishing (3)	Inter-Process Communication (2)	Browser Extensions	Create or Modify System Process (4)	Deploy Container	Forge Web Credentials (2)	Cloud Service Dashboard	Remote Services (6)	Data from Configuration Repository (2)	Dynamic Resolution (3)	Exfiltration Over Other Network Medium (1)	Defacement (2)
Phishing for Information (3)	Obtain Capabilities (6)	Replication Through Removable Media	Native API	Compromise Client Software Binary	Domain Policy Modification (2)	Direct Volume Access	Input Capture (4)	Cloud Service Discovery	Replication Through Removable Media	Encrypted Channel (2)	Exfiltration Over Physical Medium (1)	Disk Wipe (2)	Endpoint Denial of Service (4)
Search Closed Sources (2)	Stage Capabilities (3)	Supply Chain Compromise (3)	Scheduled Task/Job (7)	Create Account (3)	Escape to Host	Domain Policy Modification (2)	Man-in-the-Middle (2)	Container and Resource Discovery	Software Deployment Tools	Fallback Channels	Exfiltration Over Web Service (2)	Firmware Corruption	Inhibit System Recovery
Search Open Technical Databases (5)		Trusted Relationship	Shared Modules	Create or Modify System Process (4)	Event Triggered Execution (15)	Execution Guardrails (1)	Modify Authentication Process (4)	File and Directory Permissions Modification (2)	Taint Shared Content	Ingress Tool Transfer	Network Denial of Service (2)	Resource Hijacking	Service Stop
Search Open Websites/Domains (2)		Valid Accounts (4)	Software Deployment Tools	Event Triggered Execution (15)	Hijack Execution Flow (11)	Exploitation for Defense Evasion	Network Sniffing	Hide Artifacts (7)	Data from Local System	Multi-Stage Channels	Scheduled Transfer	System Shutdown/Reboot	
Search Victim-Owned Websites			System Services (2)	External Remote Services	Implant Internal Image	File and Directory Permissions Modification (2)	OS Credential Dumping (6)	Hijack Execution Flow (11)	Data from Network Shared Drive	Non-Application Layer Protocol	Transfer Data to Cloud Account		
			User Execution (3)	Hijack Execution Flow (11)	Modify Authentication Process (4)	Hide Artifacts (7)	Steal Application Access Token	Impair Defenses (7)	Data from Removable Media	Non-Standard Port			
			Windows Management Instrumentation	Scheduled Task/Job (7)	Office Application Startup (6)	Indicator Removal on Host (4)	Steal or Forge Kerberos Tickets (4)	Indirect Command Execution	Data Staged (2)	Protocol Tunneling			
				Valid Accounts (4)	Pre-OS Boot (3)	Modify Authentication Process (4)	Steal Web Session Cookie	Masquerading (3)	Email Collection (3)	Proxy (4)			
					Scheduled Task/Job (7)	Modify Authentication Process (4)	Two-Factor Authentication Interception	Modify Cloud Compute Infrastructure (4)	Input Capture (4)	Remote Access Software			
					Server Software Component (3)	Modify Cloud Compute Infrastructure (4)	Unsecured Credentials (7)	Modify Registry	Man in the Browser	Traffic Signaling (1)			
					Traffic Signaling (1)	Modify Registry		Modify System Image (2)	Web Service (3)				
					Valid Accounts (4)	Modify System Image (2)		Network Boundary Bridging (1)					
						Network Boundary Bridging (1)		Obfuscated Files or Information (5)					
						Obfuscated Files or Information (5)		Pre-OS Boot (3)					
						Pre-OS Boot (3)		Process Injection (11)					
						Process Injection (11)		Rogue Domain Controller					
						Rogue Domain Controller		Rootkit					
						Signed Binary Proxy Execution (11)		Signed Script Proxy Execution (1)					
						Signed Script Proxy Execution (1)		Subvert Trust Controls (6)					
						Template Injection		Traffic Signaling (1)					
						Trusted Developer Utilities Proxy Execution (1)		Trusted Developer Utilities Proxy Execution (1)					
						Unused/Unsupported Cloud Regions		Use Alternate Authentication Material (4)					
						Use Alternate Authentication Material (4)		Valid Accounts (4)					
						Virtualization/Sandbox Evasion (3)		Virtualization/Sandbox Evasion (3)					
						Weaken Encryption (2)		Weaken Encryption (2)					
						XSL Script Processing		XSL Script Processing					

ATT&CK FOR MACOS MATRIX

Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection	Command and Control	Exfiltration	Impact
7 techniques	7 techniques	14 techniques	10 techniques	18 techniques	12 techniques	19 techniques	6 techniques	13 techniques	16 techniques	8 techniques	13 techniques
Drive-by Compromise	II Command and Scripting Interpreter (5)	II Account Manipulation (1)	II Abuse Elevation Control Mechanism (3)	II Abuse Elevation Control Mechanism (3)	II Brute Force (4)	II Account Discovery (2)	Exploitation of Remote Services	II Archive Collected Data (3)	II Application Layer Protocol (4)	Automated Exfiltration	Account Access Removal
Exploit Public-Facing Application	Exploitation for Client Execution	II Boot or Logon Autostart Execution (3)	II Boot or Logon Autostart Execution (3)	Deobfuscate/Decode Files or Information	II Credentials from Password Stores (4)	Application Window Discovery	Internal Spearphishing	Audio Capture	Communication Through Removable Media	Data Transfer Size Limits	Data Destruction
Hardware Additions	Native API	II Boot or Logon Initialization Scripts (3)	II Boot or Logon Initialization Scripts (3)	II Execution Guardrails (1)	Exploitation for Credential Access	Browser Bookmark Discovery	Lateral Tool Transfer	Automated Collection	II Data Encoding (2)	II Exfiltration Over Alternative Protocol (3)	Data Encrypted for Impact
II Phishing (3)	II Scheduled Task/Job (2)	Browser Extensions	II Create or Modify System Process (2)	Exploitation for Defense Evasion	II Forge Web Credentials (1)	File and Directory Discovery	II Remote Service Session Hijacking (1)	Clipboard Data	II Data Obfuscation (3)	II Exfiltration Over C2 Channel	II Data Manipulation (3)
II Supply Chain Compromise (3)	Software Deployment Tools	Compromise Client Software Binary	II Event Triggered Execution (4)	II File and Directory Permissions Modification (1)	II Input Capture (3)	Network Service Scanning	II Remote Services (2)	Data from Information Repositories	II Dynamic Resolution (3)	II Endpoints Denial of Service (4)	II Defacement (2)
Trusted Relationship	II System Services (1)	II Create Account (2)	Exploitation for Privilege Escalation	II Hide Artifacts (6)	II Man-in-the-Middle (1)	Network Share Discovery	Software Deployment Tools	Data from Local System	II Encrypted Channel (2)	II Exfiltration Over Other Network Medium (1)	II Disk Wipe (2)
II Valid Accounts (3)	II User Execution (2)	II Create or Modify System Process (2)	II Hijack Execution Flow (2)	II Hijack Execution Flow (2)	II Modify Authentication Process (1)	Network Sniffing	Password Policy Discovery	Data from Network Shared Drive	Fallback Channels	II Exfiltration Over Physical Medium (1)	Firmware Corruption
		II Event Triggered Execution (4)	II Process Injection	II Indicator Removal on Host (4)	II OS Credential Dumping	II Permission Groups Discovery (2)	Peripheral Device Discovery	Data from Removable Media	Ingress Tool Transfer	II Exfiltration Over Web Service (2)	Inhibit System Recovery
		II Hijack Execution Flow (2)	II Scheduled Task/Job (2)	II Masquerading (5)	Steal Web Session Cookie	Process Discovery	II Software Discovery (1)	II Data Staged (2)	Multi-Stage Channels	Non-Application Layer Protocol	Resource Hijacking
		II Modify Authentication Process (1)	II Valid Accounts (3)	II Modify Authentication Process (1)	Two-Factor Authentication Interception	Remote System Discovery	System Information Discovery	II Input Capture (3)	Non-Standard Port	Protocol Tunneling	Service Stop
		II Scheduled Task/Job (2)		II Obfuscated Files or Information (5)	II Unsecured Credentials (3)	System Location Discovery	System Network Configuration Discovery (1)	II Man-in-the-Middle (1)	Proxy (4)		System Shutdown/Reboot
		II Server Software Component (1)		II Process Injection		System Network Connections Discovery	System Network Connections Discovery	Screen Capture	Remote Access Software		
		II Traffic Signaling (1)		II Rootkit		System Owner/User Discovery	Virtualization/Sandbox Evasion (3)	Video Capture	II Traffic Signaling (1)		
		II Valid Accounts (3)		II Subvert Trust Controls (4)					II Web Service (3)		
				II Traffic Signaling (1)							
				II Virtualization/Sandbox Evasion (3)							

ATT&CK TACTICS: THE ADVERSARY'S TECHNICAL GOALS

Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection	Command and Control	Exfiltration	Impact
7 techniques	7 techniques	14 techniques	10 techniques	18 techniques	12 techniques	19 techniques	6 techniques	13 techniques	16 techniques	8 techniques	13 techniques
Drive-by Compromise	II Command and Scripting Interpreter (5)	II Account Manipulation (1)	II Abuse Elevation Control Mechanism (3)	II Abuse Elevation Control Mechanism (3)	II Brute Force (4)	II Account Discovery (2)	Exploitation of Remote Services	II Archive Collected Data (3)	II Application Layer Protocol (4)	Automated Exfiltration	Account Access Removal
Exploit Public-Facing Application	Exploitation for Client Execution	II Boot or Logon Autostart Execution (3)	II Boot or Logon Autostart Execution (3)	Deobfuscate/Decode Files or Information	II Credentials from Password Stores (4)	Application Window Discovery	Internal Spearphishing	Audio Capture	Communication Through Removable Media	Data Transfer Size Limits	Data Destruction
Hardware Additions	Native API	II Boot or Logon Initialization Scripts (3)	II Boot or Logon Initialization Scripts (3)	II Execution Guardrails (1)	Exploitation for Credential Access	Browser Bookmark Discovery	Lateral Tool Transfer	Automated Collection	II Data Encoding (2)	II Exfiltration Over Alternative Protocol (3)	Data Encrypted for Impact
II Phishing (3)	II Scheduled Task/Job (2)	Browser Extensions	II Boot or Logon Initialization Scripts (3)	Exploitation for Defense Evasion	II Forge Web Credentials (1)	File and Directory Discovery	II Remote Service Session Hijacking (1)	Clipboard Data	II Data Obfuscation (3)	Exfiltration Over C2 Channel	II Data Manipulation (3)
II Supply Chain Compromise (3)	Software Deployment Tools	Compromise Client Software Binary	II Create or Modify System Process (2)	II File and Directory Permissions Modification (1)	II Input Capture (3)	Network Service Scanning	II Remote Services (2)	Data from Information Repositories	II Dynamic Resolution (3)	II Exfiltration Over Other Network Medium (1)	II Defacement (2)
Trusted Relationship	II System Services (1)	II Create Account (2)	II Event Triggered Execution (4)	II Hide Artifacts (6)	II Man-in-the-Middle (1)	Network Share Discovery	Software Deployment Tools	Data from Local System	II Encrypted Channel (2)	II Exfiltration Over Physical Medium (1)	II Disk Wipe (2)
II Valid Accounts (3)	II User Execution (2)	II Create or Modify System Process (2)	Exploitation for Privilege Escalation	II Hijack Execution Flow (2)	II Modify Authentication Process (1)	Network Sniffing		Data from Network Shared Drive	Fallback Channels	II Exfiltration Over Web Service (2)	Endpoint Denial of Service (4)
		II Event Triggered Execution (4)	II Hijack Execution Flow (2)	II Impair Defenses (4)	II Network Sniffing	OS Credential Dumping		Data from Removable Media	Ingress Tool Transfer	Resource Hijacking	Firmware Corruption
		II Hijack Execution Flow (2)	Process Injection	II Indicator Removal on Host (4)	Steal Web Session Cookie	Permission Groups Discovery (2)		II Data Staged (2)	Multi-Stage Channels	Scheduled Transfer	Inhibit System Recovery
		II Modify Authentication Process (1)	II Scheduled Task/Job (2)	II Masquerading (5)	Two-Factor Authentication Interception	Process Discovery	II Software Discovery (1)	II Input Capture (3)	Non-Application Layer Protocol		Network Denial of Service (2)
		II Scheduled Task/Job (3)	II Valid Accounts (3)	II Modify Authentication Process (1)	Rootkit	Remote System Discovery	System Information Discovery	II Man-in-the-Middle (1)	Non-Standard Port		Service Stop
				II Obfuscated Files or Information (5)	Subvert Trust	System Location Discovery		Screen	Protocol Tunneling		System Shutdown/Reboot

ATT&CK TECHNIQUE: HOW THE GOALS ARE ACHIEVED

Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Other
7 techniques	7 techniques	14 techniques	10 techniques	18 techniques	12 techniques	
Drive-by Compromise	Command and Scripting Interpreter (5)	Account Manipulation (1)	Abuse Elevation Control Mechanism (3)	Abuse Elevation Control Mechanism (3)	Brute Force (4)	A
Exploit Public-Facing Application		Boot or Logon Autostart Execution (3)		Deobfuscate/Decode Files or Information	Credentials from Password Stores (4)	App Disc
Hardware Additions	Exploitation for Client Execution	Hijack Execution Flow (2)	Boot or Logon Autostart Execution (3)	Execution Guardrails (1)	Exploitation for Credential Access	Bro Disc
Phishing (3)	Native API		Boot or Logon Initialization Scripts (3)	Exploitation for Defense Evasion		File Disc
Supply Chain Compromise (3)	Scheduled Task/Job (2)	Dynamic Linker Hijacking	Create or Modify System Process (2)	File and Directory Permissions Modification (1)	Forge Web Credentials (1)	Net Sca
Trusted Relationship	Software Deployment Tools	Dylib Hijacking		Hide Artifacts (6)	Input Capture (3)	Net
		Boot or Logon Initialization			Man-in-the-	Net

ATT&CK SUB-TECHNIQUE: MORE SPECIFIC TECHNIQUE

Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	
7 techniques	7 techniques	14 techniques	10 techniques	18 techniques	12 techniques	
Drive-by Compromise	Command and Scripting Interpreter (5)	Account Manipulation (1)	Abuse Elevation Control Mechanism (3)	Abuse Elevation Control Mechanism (3)	Brute Force (4)	A
Exploit Public-Facing Application		Boot or Logon Autostart Execution (3)		Deobfuscate/Decode Files or Information	Credentials from Password Stores (4)	
Hardware Additions	Exploitation for Client Execution	Hijack Execution Flow (2)	Boot or Logon Autostart Execution (3)	Execution Guardrails (1)	Exploitation for Credential Access	Bro
Phishing (3)	Native API	Dynamic Linker Hijacking	Boot or Logon Initialization Scripts (3)	Exploitation for Defense Evasion	Forge Web Credentials (1)	File
Supply Chain Compromise (3)	Scheduled Task/Job (2)		Create or Modify System Process (2)	File and Directory Permissions Modification (1)	Input Capture (3)	Net
Trusted Relationship	Software Deployment Tools	Dylib Hijacking		Hide Artifacts (6)	Man-in-the-	Net
		Boot or Logon Initialization				

INSIDE A TECHNIQUE

TECHNIQUES

- Enterprise
- Reconnaissance
- Resource Development
- Initial Access
- Execution
- Persistence
- Account Manipulation
- BITS Jobs
- Boot or Logon Autostart Execution
- Boot or Logon Initialization Scripts
- Browser Extensions
- Compromise Client Software Binary
- Create Account
- Create or Modify System Process
- Event Triggered Execution
- External Remote Services
- Hijack Execution Flow
- DLL Search Order Hijacking
- DLL Side-Loading
- Dylib Hijacking
- Executable Installer File Permissions Weakness
- Dynamic Linker Hijacking
- Path Interception by PATH Environment Variable
- Path Interception by Search Order Hijacking
- Path Interception by Unquoted Path
- Services File Permissions Weakness
- Services Registry Permissions Weakness
- COR_PROFILER
- Implant Internal Image
- Modify Authentication Process
- Office Application Startup
- Pre-OS Boot

Home > Techniques > Enterprise > Hijack Execution Flow > Dylib Hijacking

Hijack Execution Flow: Dylib Hijacking

Other sub-techniques of Hijack Execution Flow (11)

Adversaries may execute their own payloads by placing a malicious dynamic library (dylib) with an expected name in a path a victim application searches at runtime. The dynamic loader will try to find the dylibs based on the sequential order of the search paths. Paths to dylibs may be prefixed with `@rpath`, which allows developers to use relative paths to specify an array of search paths used at runtime based on the location of the executable. Additionally, if weak linking is used, such as the `LC_LOAD_WEAK_DYLIB` function, an application will still execute even if an expected dylib is not present. Weak linking enables developers to run an application on multiple macOS versions as new APIs are added.

Adversaries may gain execution by inserting malicious dylibs with the name of the missing dylib in the identified path.^{[1][2][3][4]} Dylibs are loaded into an application's address space allowing the malicious dylib to inherit the application's privilege level and resources. Based on the application, this could result in privilege escalation and uninhibited network access. This method may also evade detection from security products since the execution is masked under a legitimate process.^{[5][6][7]}

ID: T1574.004
Sub-technique of: T1574
① Tactics: Persistence, Privilege Escalation, Defense Evasion
① Platforms: macOS
① Data Sources: File: File Creation, File: File Modification, Module: Module Load
① Defense Bypassed: Application control
① CAPEC ID: CAPEC-471
Version: 2.0
Created: 16 March 2020
Last Modified: 27 April 2021

Version Permalink

Procedure Examples

ID	Name	Description
S0363	Empire	Empire has a dylib hijacker module that generates a malicious dylib given the path to a legitimate dylib of a vulnerable application. ^[8]

Mitigations

ID	Mitigation	Description
M1022	Restrict File and Directory Permissions	Set directory access controls to prevent file writes to the search paths for applications, both in the folders where applications are run from and the standard dylib folders.

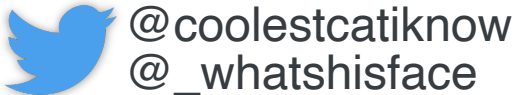
Detection

Monitor file systems for moving, renaming, replacing, or modifying dylibs. Changes in the set of dylibs that are loaded by a process (compared to past behavior) that do not correlate with known software, patches, etc., are suspicious. Check the system for multiple dylibs with the same name and monitor which versions have historically been loaded into a process.

Run path dependent libraries can include `LC_LOAD_DYLIB`, `LC_LOAD_WEAK_DYLIB`, and `LC_RPATH`. Other special keywords are recognized by the macOS loader are `@rpath`, `@loader_path`, and `@executable_path`.^[9] These loader instructions can be examined for individual binaries or frameworks using the `otool -l` command. Objective-See's Dylib Hijacking Scanner can be used to identify applications vulnerable to dylib hijacking.^{[1][3]}

References

- Patrick Wardle. (2019, July 2). Getting Root with Benign AppStore Apps. Retrieved March 31, 2021.
- Patrick Wardle. (2015, March 1). Dylib Hijacking on OS X. Retrieved March 29, 2021.
- Wardle, P., Ross, C. (2017, September 21). Empire Project Dylib Hijack Vulnerability Scanner. Retrieved April 1, 2021.
- Wardle, P., Ross, C. (2018, April 8). EmpireProject Create Dylib Hijacker. Retrieved April 1, 2021.
- Patrick Wardle. (2015). Writing Bad @\$ Malware for OS X. Retrieved July 10, 2017.
- Patrick Wardle. (2020, August 5). The Art of Mac Malware Volume 0x1: Analysis. Retrieved March 19, 2021.
- Amanda Rousseau. (2020, April 4). MacOS Dylib Injection Workshop. Retrieved March 29, 2021.
- Schroeder, W., Warner, J., Nelson, M. (n.d.). Github PowerShellEmpire. Retrieved April 28, 2016.
- Apple Inc.. (2012, July 7). Run-Path Dependent Libraries. Retrieved March 31, 2021.



TECHNIQUE DESCRIPTION

[Home](#) > [Techniques](#) > [Enterprise](#) > [Hijack Execution Flow](#) > [Dylib Hijacking](#)

Hijack Execution Flow: Dylib Hijacking

Other sub-techniques of Hijack Execution Flow (11)



Adversaries may execute their own payloads by placing a malicious dynamic library (dylib) with an expected name in a path a victim application searches at runtime. The dynamic loader will try to find the dylibs based on the sequential order of the search paths. Paths to dylibs may be prefixed with `@rpath`, which allows developers to use relative paths to specify an array of search paths used at runtime based on the location of the executable. Additionally, if weak linking is used, such as the `LC_LOAD_WEAK_DYLIB` function, an application will still execute even if an expected dylib is not present. Weak linking enables developers to run an application on multiple macOS versions as new APIs are added.

Adversaries may gain execution by inserting malicious dylibs with the name of the missing dylib in the identified path.^{[1][2][3][4]} Dylibs are loaded into an application's address space allowing the malicious dylib to inherit the application's privilege level and resources. Based on the application, this could result in privilege escalation and uninhibited network access. This method may also evade detection from security products since the execution is masked under a legitimate process.^{[5][6][7]}



TECHNIQUE EXTERNAL REPORTING

Home > Techniques > Enterprise > Hijack Execution Flow > Dylib Hijacking

Hijack Execution Flow: Dylib Hijacking

The Art of Mac Malware: Analysis
p. wardle

The (in)famous OSX.FlashBack.B [22] malware abused DYLD_INSERT_LIBRARIES to maintain persistence by targeting users' browsers:

"A DYLD_INSERT_LIBRARIES environment variable is also added to the targeted browsers as a launch point. This is done by inserting a LSEnvironment entry to the corresponding Info.plist of the browsers" [22]:

```
$ cat /Applications/Safari.app/Contents/Info.plist:  
...  
  
<key>LSEnvironment</key>  
<dict>  
  <key>DYLD_INSERT_LIBRARIES</key>  
  <string>/Applications/Safari.app/Contents/Resources/%payload_filename%</string>  
</dict>
```

*DYLD_INSERT_LIBRARIES persistence
(OSX.FlashBack.B)*

ic library (dylib) with an expected name in a path a victim
s based on the sequential order of the search paths. Paths
tive paths to specify an array of search paths used at
g is used, such as the LC_LOAD_WEAK_DYLIB function, an
inking enables developers to run an application on multiple

e of the missing dylib in the identified path.^{[1][2][3][4]} Dylibs are
nherit the application's privilege level and resources. Based
network access. This method may also evade detection from

security products since the execution is masked under a legitimate process^{[5][6][7]}

ATT&CK: TECHNIQUE METADATA

MITRE | ATT&CK

Home > Techniques > Enterprise > Hijack Execution Flow > Dll Hijacking

ID: T1574.004

Sub-technique of: T1574

- ① **Tactics:** Persistence, Privilege Escalation, Defense Evasion
- ① **Platforms:** macOS
- ① **Data Sources:** File: File Creation, File: File Modification, Module: Module Load
- ① **Defense Bypassed:** Application control
- ① **CAPEC ID:** CAPEC-471

Version: 2.0

Created: 16 March 2020

Last Modified: 27 April 2021

ID: T1574.004

Sub-technique of: T1574

- ① **Tactics:** Persistence, Privilege Escalation, Defense Evasion
- ① **Platforms:** macOS
- ① **Data Sources:** File: File Creation, File: File Modification, Module: Module Load
- ① **Defense Bypassed:** Application control
- ① **CAPEC ID:** CAPEC-471

Version: 2.0

Created: 16 March 2020

Last Modified: 27 April 2021

6. Patrick Wardle. (2020, August 5). The Art of Mac Malware Volume 0x1: Analysis. Retrieved March 19, 2021.

7. Amanda Rousseau. (2020, April 4). MacOS Dll Injection Workshop. Retrieved March 29, 2021.

8. Schroeder, W., Warner, J., Nelson, M. (n.d.). Github PowerShellEmpire. Retrieved April 28, 2016.

9. Apple Inc.. (2012, July 7). Run-Path Dependent Libraries. Retrieved March 31, 2021.

ATT&CK: PROCEDURES

MITRE ATT&CK

Techniques

Enterprise

Hijack Execution Flow: Dylib Hijacking

Procedure Examples

ID	Name	Description
S0363	Empire	Empire has a dylib hijacker module that generates a malicious dylib given the path to a legitimate dylib of a vulnerable application. ^[8]

Adversaries may gain execution by inserting malicious dylibs into the name of the missing dylib in the identified path.^{[1][2][3][4]} Dylibs are loaded into an application's address space allowing the malicious dylib to inherit the application's privilege level and resources. Based on the application, this could result in privilege escalation and uninhibited network access. This method may also evade detection from security products since the execution is masked under a legitimate process.^{[5][6][7]}

Version: 2.0
Created: 16 March 2020
Last Modified: 27 April 2021

Procedure Examples

ID	Name	Description
S0363	Empire	Empire has a dylib hijacker module that generates a malicious dylib given the path to a legitimate dylib of a vulnerable application. ^[8]

Mitigations

ID	Mitigation	Description
M1022	Restrict File and Directory Permissions	Set directory access controls to ensure applications are run from and to trusted locations.

Detection

Monitor file systems for moving, renaming, replacing, or modifying dylibs. Changes that do not correlate with known software, patches, etc., are suspicious. Check the system logs for any unusual activity.

Run path dependent libraries can include `LC_LOAD_DYLIB`, `LC_LOAD_WEAK_DYLIB`, `LC_LOAD_DYLINKER`, `LC_LOAD_DYLIB_INTO_DLL`, and `EXECUTABLE_PATH`.^[6] These loader instructions can be used to identify applications vulnerable to dylib hijacking. See the Dylib Hijacking Scanner can be used to identify applications vulnerable to dylib hijacking.

References

1. Patrick Wardle. (2019, July 2). Getting Root with Benign AppStore Apps. Retrieved March 31, 2021.
2. Patrick Wardle. (2015, March 1). Dylib Hijacking on OS X. Retrieved March 31, 2021.
3. Wardle, P., Ross, C. (2017, September 21). Empire Project Dylib Hijack Vulnerability Scanner. Retrieved April 1, 2021.
4. Wardle, P., Ross, C. (2018, April 8). EmpireProject Create Dylib Hijacker. Retrieved April 1, 2021.
5. Patrick Wardle. (2015). Writing Bad @\$\$ Malware for OS X. Retrieved July 2, 2019.

553 lines (401 sloc) | 17.4 KB

```
1 import base64
2 class Module:
3
4     def __init__(self, mainMenu, params=[]):
5
6         # metadata info about the module, not modified during runtime
7         self.info = {
8             # name for the module that will appear in module menus
9             'Name': 'CreateDylibHijacker',
10
11             # list of one or more authors for the module
12             'Authors': ['Patrick Wardle', 'Ross C. Wardle']
13         }
```



ATT&CK: DETECTION IDEAS

Detection

Monitor file systems for moving, renaming, replacing, or modifying dylibs. Changes in the set of dylibs that are loaded by a process (compared to past behavior) that do not correlate with known software, patches, etc., are suspicious. Check the system for multiple dylibs with the same name and monitor which versions have historically been loaded into a process.

Run path dependent libraries can include `LC_LOAD_DYLIB`, `LC_LOAD_WEAK_DYLIB`, and `LC_RPATH`. Other special keywords are recognized by the macOS loader are `@rpath`, `@loader_path`, and `@executable_path`.^[9] These loader instructions can be examined for individual binaries or frameworks using the `otool -l` command. Objective-See's Dylib Hijacking Scanner can be used to identify applications vulnerable to dylib hijacking.^{[1][3]}

master Empire / lib / modules / python / persistence / osx / CreateHijacker.py / <> Jump to

xorrior Fix for #1059. Fixed launcher string, which caused macho to crash. R... Latest commit efba9bd

3 contributors

553 lines (401 sloc) 17.4 KB

```
1 import base64
2 class Module:
3
4     def __init__(self, mainMenu, params=[]):
5
6         # metadata info about the module, not modified during runtime
7         self.info = {
8             # name for the module that will appear in module menus
9             'Name': 'CreateDylibHijacker',
10
11             # list of one or more authors for the module
12             'Author': ['@patrickwardle', '@xorrior'],
13
14             # more verbose multi-line description of the module
15             'Description': ('Configures and Empire dylib for use in a Dylib hijack, given the pat
16
```


ATT&CK: DETECTION IDEAS

Detection

Monitor file systems for moving, renaming, replacing, or modifying dylibs. Changes in the set of dylibs that are loaded by a process (compared to past behavior) that do not correlate with known software, patches, etc., are suspicious. Check the system for multiple dylibs with the same name and monitor which versions have historically been loaded into a process.

Run path dependent libraries can include `LC_LOAD_DYLIB`, `LC_LOAD_WEAK_DYLIB`, and `LC_RPATH`. Other special keywords are recognized by the macOS loader are `@rpath`, `@loader_path`, and `@executable_path`.^[9] These loader instructions can be examined for individual binaries or frameworks using the `otool -l` command. Objective-See's Dylib Hijacking Scanner can be used to identify applications vulnerable to dylib hijacking.^{[1][3]}

Getting Root with Benign AppStore Apps

July 2, 2019

In this guest blog post, "Objective by the Sea" speaker, [Csaba Fitzl](#) writes about an interesting way to get root via Apps from the official Mac App Store.

His research was originally presented at "Objective by the Sea" v2.0. Check out his slides, "[Getting Root with Benign AppStore Apps](#)".



ATT&CK: REFERENCES

MITRE | ATT&CK

Matrices | Tactics | Techniques | Mitigations | Groups | Software | Resources | Blog | Contribute | Search

TECHNIQUES

Enterprise
Reconnaissance
Resource
Development
Initial Access
Execution
Persistence

Hijack Execution Flow: Dylib Hijacking

Other sub-techniques of Hijack Execution Flow (11)

Adversaries may execute their own payloads by placing a malicious dynamic library (dylib) with an expected name in a path a victim application searches at runtime. The dynamic loader will try to find the dylibs based on the sequential order of the search paths. Paths to dylibs may be prefixed with `#xpaths`, which allows developers to use relative paths to specify an array of search paths used at runtime based on the location of the

ID: T1574.004
Sub-technique of: T1574
① Tactics: Persistence, Privilege Escalation, Defense Evasion
① Platforms: macOS

References

1. Patrick Wardle. (2019, July 2). Getting Root with Benign AppStore Apps. Retrieved March 31, 2021.
2. Patrick Wardle. (2015, March 1). Dylib Hijacking on OS X. Retrieved March 29, 2021.
3. Wardle, P., Ross, C. (2017, September 21). Empire Project Dylib Hijack Vulnerability Scanner. Retrieved April 1, 2021.
4. Wardle, P., Ross, C. (2018, April 8). EmpireProject Create Dylib Hijacker. Retrieved April 1, 2021.
5. Patrick Wardle. (2015). Writing Bad @\$ Malware for OS X. Retrieved July 10, 2017.
6. Patrick Wardle. (2020, August 5). The Art of Mac Malware Volume 0x1: Analysis. Retrieved March 19, 2021.
7. Amanda Rousseau. (2020, April 4). MacOS Dylib Injection Workshop. Retrieved March 29, 2021.
8. Schroeder, W., Warner, J., Nelson, M. (n.d.). Github PowerShellEmpire. Retrieved April 28, 2016.
9. Apple Inc.. (2012, July 7). Run-Path Dependent Libraries. Retrieved March 31, 2021.

Services
Hijack Execution Flow
DLL Search Order Hijacking
DLL Side-Loading
Dylib Hijacking
Executable Installer
File Permissions Weakness
Dynamic Linker Hijacking
Path Interception by PATH Environment Variable
Path Interception by Search Order
Path Interception by Unsanitized Path

M1022 Restrict File and Directory Permissions
Set directory access controls to prevent file writes to the search paths for applications, both in the folders where applications are run from and the standard dylib folders.

Detection

Monitor file systems for moving, renaming, replacing, or modifying dylibs. Changes in the set of dylibs that are loaded by a process (compared to past behavior) that do not correlate with known software, patches, etc., are suspicious. Check the system for multiple dylibs with the same name and monitor which versions have historically been loaded into a process.

Run path dependent libraries can include `LC_LOAD_DYLIB`, `LC_LOAD_WEAK_DYLIB`, and `LC_RPATH`. Other special keywords are recognized by the macOS loader are `#xpaths`, `#loader_path`, and `#executable_path`. These loader instructions can be examined for individual binaries or frameworks using the `otool -l` command. Objective-See's Dylib Hijacking Scanner can be used to identify applications vulnerable to dylib hijacking.

References

1. Patrick Wardle. (2019, July 2). Getting Root with Benign AppStore Apps. Retrieved March 31, 2021.
2. Patrick Wardle. (2015, March 1). Dylib Hijacking on OS X. Retrieved March 29, 2021.
3. Wardle, P., Ross, C. (2017, September 21). Empire Project Dylib Hijack Vulnerability Scanner. Retrieved April 1, 2021.
4. Wardle, P., Ross, C. (2018, April 8). EmpireProject Create Dylib Hijacker. Retrieved April 1, 2021.
5. Patrick Wardle. (2015). Writing Bad @\$ Malware for OS X. Retrieved July 10, 2017.
6. Patrick Wardle. (2020, August 5). The Art of Mac Malware Volume 0x1: Analysis. Retrieved March 19, 2021.
7. Amanda Rousseau. (2020, April 4). MacOS Dylib Injection Workshop. Retrieved March 29, 2021.
8. Schroeder, W., Warner, J., Nelson, M. (n.d.). Github PowerShellEmpire. Retrieved April 28, 2016.
9. Apple Inc.. (2012, July 7). Run-Path Dependent Libraries. Retrieved March 31, 2021.



WHAT'S DIFFERENT ABOUT ATT&CK FOR MAC?

- Built in hardware security (Notarization)
- Opt-in Programs (Gatekeeper, Sandbox)
- Mic drop hardware changes
- Local Admin for everyone!
- Exploitation verses behavior
- Lacking Documentation
- Limited reporting (especially on adversary behaviors)

INTRO TO OCEAN LOTUS

- 2012-present - believed to be the Vietnamese government
- MacOS, Windows, Android Spyware
- Human rights + Vietnamese interests



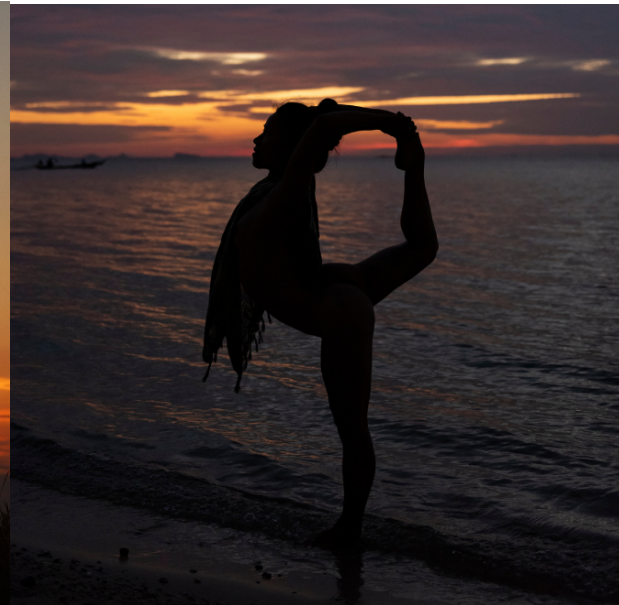
ATT&CK Postures



**Cyber Threat
Intelligence**



Threat Hunting



**Adversary
Emulation**



**Engineering &
Assessment**

Using ATT&CK for ...

Cyber Threat Intelligence



ATT&CK FOR CTI

- CTI in ATT&CK
- CTI from external reporting
- CTI from your own data



OCEANLOTUS IN ATT&CK



OceanLotus

[Matrices](#) [Tactics](#) [Techniques](#) [Mitigations](#) [Groups](#) [Software](#) [Resources](#) [Blog](#) [Contribute](#)

[Home](#) > [Groups](#) > [APT32](#)

APT32

APT32 is a threat group that has been active since at least 2014. The group has targeted multiple private sector industries as well as with foreign governments, dissidents, and journalists with a strong focus on Southeast Asian countries like Vietnam, the Philippines, Laos, and Cambodia. They have extensively used strategic web compromises to compromise victims. The group is believed to be Vietnam-based.^{[1][2][3]}

ID: G0050

① Associated Groups: SeaLotus, OceanLotus, APT-C-00

Contributors: Romain Dumont, ESET

Version: 2.4

Created: 14 December 2017

Last Modified: 20 April 2021

[Version](#) [Permalink](#)

Associated Group Descriptions

Name	Description
SeaLotus	[4]
OceanLotus	[1] [2][4][5][6]

ATT&CK TECHNIQUES FROM A GROUP PAGE



OceanLotus

Techniques Used

Domain	ID	Name	Use
Enterprise	T1087	.001 Account Discovery: Local Account	APT32 enumerated administrative users using the commands <code>localgroup administrators</code> . ^[7]
Enterprise	T1583	.001 Acquire Infrastructure: Domains	APT32 has set up and operated websites to gather information and deliver malware. ^[8]
		.006 Acquire Infrastructure: Web Services	APT32 has set up Dropbox, Amazon S3, and Google Drive to host malicious downloads. ^[8]
Enterprise	T1071	.001 Application Layer Protocol: Web Protocols	APT32 has used JavaScript that communicates over HTTP or HTTPS. The attacker controlled domains to download additional framework code. The group has also used downloaded encrypted payloads over HTTP.
		.003 Application Layer Protocol: Mail Protocols	APT32 has used email for C2 via an Office macro. ^{[4][7]}
Enterprise	T1560	Archive Collected Data	APT32's backdoor has used LZMA compression and RC4 encryption before exfiltration. ^[5]
Enterprise	T1547	.001 Boot or Logon Autostart Execution: Registry Run Keys / Startup Folder	APT32 established persistence using Registry Run keys, both to execute PowerShell and VBS scripts as well as to execute their backdoor. ^{[4][7][5]}
Enterprise	T1059	Command and Scripting Interpreter	APT32 has used COM scriptlets to download Cobalt Strike beacon.
		.001 PowerShell	APT32 has used PowerShell-based tools, PowerShell one-liners, and shellcode loaders for execution. ^{[1][4][7]}

ATT&CK® Navigator Layers ▾

Software

ID	Name	References	Techniques
S0099	Arp	[7]	System Network Configuration Discovery
S0154	Cobalt Strike	[1][2][4][7][8][6]	Abuse Elevation Control Mechanism: Bypass User Account Control, Access Token Manipulation: Token Impersonation/Theft, Access Token Manipulation: Parent PID Spoofing, Access Token Manipulation: Make and Impersonate Token, Account Discovery: Domain Account, Application Layer Protocol, Application Layer Protocol: DNS, Application Layer Protocol: Web Protocols, BITS Jobs, Command and Scripting Interpreter: Windows Command

References

1. Carr, N.. (2017, May 14). Cyber Espionage is Alive and Well: APT32 and the Threat to Global Corporations. Retrieved June 18, 2017.

2. Lassalle, D., et al. (2017, November 6). OceanLotus Blossoms: Mass Digital Surveillance and Attacks Targeting ASEAN, Asian Nations, the Media, Human Rights Groups, and Civil Society. Retrieved November 6, 2017.

3. Foltyn, T. (2018, March 13). OceanLotus ships new backdoor using old tricks. Retrieved May 22, 2018.

4. Adair, S. and Lancaster, T. (2020, November 6). OceanLotus: Extending Cyber Espionage Operations Through Fake Websites. Retrieved November 20, 2020.

5. Dumont, R.. (2019, April 9). OceanLotus: macOS malware update. Retrieved April 15, 2019.

6. Carr, N.. (2017, December 26). Nick Carr Status Update APT32 pubprn. Retrieved April 22, 2019.

7. Bohannon, D.. (2017, March 13). Invoke-Obfuscation - PowerShell Obfuscator. Retrieved June 18, 2017.

8. Adair, S. and Lancaster, T. (2020, November 6). OceanLotus: Extending Cyber Espionage Operations Through Fake Websites. Retrieved November 20, 2020.

9. Dumont, R.. (2019, April 9). OceanLotus: macOS malware update. Retrieved April 15, 2019.

10. Carr, N.. (2017, December 26). Nick Carr Status Update APT32 pubprn. Retrieved April 22, 2019.

11. Bohannon, D.. (2017, March 13). Invoke-Obfuscation - PowerShell Obfuscator. Retrieved June 18, 2017.

OceanLotus

OCEAN LOTUS TECHNIQUES OVERLAP WITH MAC ATT&CK

Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection	Command and Control	Exfiltration	Impact
Drive-by Compromise	Command and Scripting Interpreter	Account Manipulation	Abuse Elevation Control Mechanism	Abuse Elevation Control Mechanism	Brute Force	Account Discovery	Exploitation of Remote Services	Archive Collected Data	Application Layer Protocol	Automated Exfiltration	Account Access Removal
Exploit Public-Facing Application	AppleScript	Boot or Logon Autostart Execution	Boot or Logon Autostart Execution	Deobfuscate/Decode Files or Information	Credentials from Password Stores	Local Account	Internal Spearphishing	Audio Capture	Web Protocols	Data Transfer Size Limits	Data Destruction
Hardware Additions	Unix Shell	Kernel Modules and Extensions	Kernel Modules and Extensions	Execution Guardrails	Exploitation for Credential Access	Domain Account	Lateral Tool Transfer	Automated Collection	File Transfer Protocols	Exfiltration Over Alternative Protocol	Data Encrypted for Impact
Phishing	Visual Basic	Re-opened Applications	Re-opened Applications	Exploitation for Defense Evasion	Forge Web Credentials	Application Window Discovery	Remote Service Session Hijacking	Clipboard Data	Mail Protocols	Exfiltration Over Symmetric Encrypted Non-C2 Protocol	Data Manipulation
Spearphishing Attachment	Python	Plist Modification	Plist Modification	File and Directory Permissions Modification	Input Capture	Browser Bookmark Discovery	Remote Services	Data from Information Repositories	DNS	Exfiltration Over Asymmetric Encrypted Non-C2 Protocol	Defacement
Spearphishing Link	JavaScript	Boot or Logon Initialization Scripts	Boot or Logon Initialization Scripts	Linux and Mac File and Directory Permissions Modification	Keylogging	File and Directory Discovery	SSH	Data from Local System	Communication Through Removable Media	Exfiltration Over Encrypted/Obfuscated Non-C2 Protocol	Disk Wipe
Spearphishing via Service	Exploitation for Client Execution	Browser Extensions	Create or Modify System Process	Hide Artifacts	GUI Input Capture	Network Service Scanning	VNC	Data from Network Shared Drive	Data Encoding	Exfiltration Over C2 Channel	Endpoint Denial of Service
Supply Chain Compromise	Native API	Compromise Client Software Binary	Launch Agent	Hidden Files and Directories	Web Portal Capture	Network Share Discovery	Software Deployment Tools	Data from Removable Media	Data Obfuscation	Exfiltration Over Other Network Medium	Firmware Corruption
Trusted Relationship	Scheduled Task/Job	Create Account	Launch Daemon	Hidden Users	Man-in-the-Middle	Network Sniffing		Data Staged	Dynamic Resolution	Exfiltration Over Physical Medium	Inhibit System Recovery
Valid Accounts	Launchd	Create or Modify System Process	Event Triggered Execution	Hidden Window	Modify Authentication Process	Password Policy Discovery		Input Capture	Encrypted Channel	Exfiltration Over Web Service	Network Denial of Service
Default Accounts	Cron	Event Triggered Execution	Exploitation for Privilege Escalation	Hidden File System	Network Sniffing	Peripheral Device Discovery		Keylogging	Fallback Channels	Scheduled Transfer	Resource Hijacking
Domain Accounts	Software Deployment Tools	Hijack Execution Flow	Hijack Execution Flow	Run Virtual Instance	OS Credential Dumping	Permission Groups Discovery		GUI Input Capture	Ingress Tool Transfer		Service Stop
Local Accounts	System Services	Modify Authentication Process	Dynamic Linker Hijacking	VBA Stomping	Steal Web Session Cookie	Process Discovery		Web Portal Capture	Multi-Stage Channels		System Shutdown/Reboot
	Launchctl	Scheduled Task/Job	Dylib Hijacking	Hijack Execution Flow	Two-Factor Authentication Interception	Remote System Discovery		Man-in-the-Middle	Non-Application Layer Protocol		
	User Execution	Server Software Component	Process Injection	Impair Defenses	Unsecured Credentials	Software Discovery		Screen Capture	Protocol Tunneling		
	Malicious Link	Web Shell	Scheduled Task/Job	Indicator Removal on Host	Credentials in Files	System Information Discovery		Video Capture	Non-Standard Port		
	Malicious File	Traffic Signaling	Launchd	Clear Linux or Mac System Logs	Bash History	System Location Discovery			Proxy		
		Valid Accounts	Cron	Clear Command History	Private Keys	System Network Configuration Discovery			Remote Access Software		
		Default Accounts	Valid Accounts	File Deletion		System Network Connections Discovery			Traffic Signaling		
		Domain Accounts	Default Accounts	Timestomp		System Owner/User Discovery			Web Service		
		Local Accounts	Domain Accounts	Masquerading		Virtualization/Sandbox Evasion					
			Local Accounts	Invalid Code Signature							
				Right-to-Left Override							
				Rename System Utilities							
				Watch Legitimate Name or Location							
				Space after Filename							
				Modify Authentication Process							
				Obfuscated Files or Information							
				Binary Padding							
				Software Packing							
				Steganography							
				Compile After Delivery							
				Indicator Removal from Tools							
				Process Injection							
				Rootkit							
				Subvert Trust Controls							
				Traffic Signaling							
				Valid Accounts							
				Default Accounts							
				Domain Accounts							
				Local Accounts							
				Virtualization/Sandbox Evasion							

Generated from ATT&CK Navigator
bit.ly/attacknav

OCEAN LOTUS TECHNIQUES

Reconnaissance	Resource Development	Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection	Command and Control	Exfiltration	Impact
Active Scanning	Acquire Infrastructure	Drive-by Compromise	Command and Scripting Interpreter	Account Manipulation	Abuse Elevation Control Mechanism	Abuse Elevation Control Mechanism	Brute Force	Account Discovery	Exploitation of Remote Services	Archive Collected Data	Application Layer Protocol	Automated Exfiltration	Account Access Removal
Other Victim Host Information	Domain	Region Pkts-Rising Application	Powershell	BITS Jobs	Access Token Manipulation	Access Token Manipulation	Credentials from Password Stores	Local Account	Internal Spearphishing	Audio Capture	Web Protocols	Data Transfer Size Limits	Data Destruction
Gather Victim Identity Information	DNS Server	External Remote Services	AppleScript	Boot or Logon Autostart Execution	Registry Hiv Maps / Startup Folder	Registry Hiv Maps / Startup Folder	Exploitation for Credential Access	Domain Account	Lateral Tool Transfer	Automated Collection	File Transfer Protocols	Data Encryption for Impact	Data Encrypted for Impact
Credentials	Virtual Private Server	Hardware Additions	Windows Command Shell	Authentication Package	Time Providers	Time Providers	Forced Authentication	Email Account	Remote Service Session Hijacking	Clipboard Data	Mail Protocols	Data Manipulation	Data Manipulation
Email Addresses	Server	Phishing	Unix Shell	Visual Basic	Windows Helper DLL	Windows Helper DLL	Forge Web Credentials	Application Window Discovery	Remote Services	Data from Information Repositories	DNS	Defacement	Defacement
Employee Names	Notnet	Spearphishing Attachment	JavaScript	Time Providers	Kernel Modules and Extensions	Kernel Modules and Extensions	Input Capture	Browser Bookmark Discovery	Remote Desktop Protocol	Data from Local System	Communication Through Removable Media	Disk Wipe	Disk Wipe
Other Victim Network Information	Web Services	Spearphishing Link	Python	Windows Defender DLL	Execution Guardrails	Execution Guardrails	Keylogger	Domain Trust Discovery	Remote Windows Admin Shares	Data from Network Shared Drive	Exfiltration Over Removable Media	Endpoint Denial of Service	Endpoint Denial of Service
Gather Victim Org Information	Compromise Accounts	Spearphishing via Service	Exploitation Through Removable Media	Security Support Provider	Exploitation for Defense Evasion	Exploitation for Defense Evasion	Man-in-the-Middle	File and Directory Discovery	Unauthorized Command Prompt Shell	Data from Removable Media	Data Obfuscation	Firmware Corruption	Firmware Corruption
Phishing for Information	Compromise Infrastructure	Supply Chain Compromise	Exploitation Through Removable Media	Re-opened Applications	File and Directory Permission Modification	File and Directory Permission Modification	Network Sniffing	Network Service Scanning	SSH	Data Staged	Dynamic Resolution	Inhibit System Recovery	Inhibit System Recovery
Spearphishing Service	Develop Capabilities	Trusted Relationship	Valid Accounts	Re-opened Applications	Hidden File Attributes	Hidden File Attributes	OS Credential Dumping	Network Share Discovery	VNC	Email Collection	Encrypted Channel	Network Denial of Service	Network Denial of Service
Spearphishing Attachment	Establish Accounts	Valid Accounts	Default Accounts	LSASS Driver	NTFS File Attributes	NTFS File Attributes	Security Account Manager	Query Registry	Windows Remote Management	Input Capture	Fallback Channels	Resource Hijacking	Resource Hijacking
Spearphishing Link	Local User Accounts	Default Accounts	Scheduled Task/Job	LSASS Driver	Active Setup	Active Setup	NTDS	Remote System Discovery	Exploitation Through Removable Media	Keylogging	Reverse Tool Transfer	Service Stop	Service Stop
Search Closed Sources	Email Accounts	Domain Accounts	At (Windows)	Port Monitors	Run Virtual Instance	Run Virtual Instance	DCSync	Software Discovery	Communication Through Removable Media	GUI Input Capture	Multi-Stage Channels	System Shutdown/Reboot	System Shutdown/Reboot
Search Open Technical Databases	Obtain Capabilities	Local Accounts	Scheduled Task	Plist Modification	VBA Stomping	VBA Stomping	Proc Filesystem	System Information Discovery	Software Deployment Tools	Web Portal Capture	Non-Application Layer Protocol		
Search Open Websites/Domains	Stage Capabilities		At (Linux)	Print Processors	Hijack Execution Flow	Hijack Execution Flow	/etc/passwd and /etc/shadow	System Location Discovery	Taint Shared Content	Credential API Hooking	Protocol Tunneling		
Search Victim-Owned Websites	Upload Malware		Launchd	XDG Autostart Entries	Sections File Permissions Weakness	Sections File Permissions Weakness	Security Account Manager	System Network Configuration Discovery	Use Alternate Authentication Material	Man in the Browser	Remote Access Software		
	Upload Tool		Cron	Active Setup	Service Registry Permissions Weakness	Service Registry Permissions Weakness	LSA Secrets	System Owner/User Discovery	Pass the Hash	Man-in-the-Middle	Traffic Signaling		
	Install Digital Certificate		System Timers	Browser Extensions	Path Interception by Unquoted Path	Path Interception by Unquoted Path	Steal Web Session Cookie	System Service Discovery	Pass the Ticket	Proxy	Web Service		
	Drive-by Target		Shared Modules	Create Account	Path Interception by WMI Environment Variable	Path Interception by WMI Environment Variable	Unacquired Credentials	System Time Discovery					
	Link Target		Software Deployment Tools	Launch Agent	Path Interception by Search Order Hijacking	Path Interception by Search Order Hijacking	Credentials in Files	Virtualization/Sandbox Evasion					
			System Services	Launch Agent	Dynamic Linker Hijacking	Dynamic Linker Hijacking	Bash History						
			Launchctl	System Service	Dylib Hijacking	Dylib Hijacking	Private Keys						
			Service Execution	Windows Service	COR PROFILER	COR PROFILER	Group Policy Preferences						
			User Execution	Launch Daemon	Impair Defenses	Impair Defenses							
			Malicious Link	Event Triggered Execution	Indicator Removal on Host	Indicator Removal on Host							
			Malicious File	Hijack Execution Flow	Clear Windows Event Logs	Clear Windows Event Logs							
				Sections File Permissions Weakness	Clear Linux or Mac System Logs	Clear Linux or Mac System Logs							
				Service Registry Permissions Weakness	Clear Command History	Clear Command History							
				Path Interception by Unquoted Path	File Deletion	File Deletion							
				Path Interception by WMI Environment Variable	Network Share Connection Removal	Network Share Connection Removal							
				Path Interception by Search Order Hijacking	Timestamp	Timestamp							
				Dynamic Linker Hijacking	Indirect Command Execution	Indirect Command Execution							
				Dylib Hijacking	Invalid Code Signature	Invalid Code Signature							
				COR PROFILER	Right-to-Left Override	Right-to-Left Override							
				Modify Authentication Process	Rename System Utilities	Rename System Utilities							
				Office Application Startup	Marked Task or Service	Marked Task or Service							
				Pre-OS Boot	Marked Task or Service	Marked Task or Service							
				Scheduled Task/Job	Space after Filename	Space after Filename							
				At (Windows)	Modify Authentication Process	Modify Authentication Process							
				At (Linux)	Indirect Command Execution	Indirect Command Execution							
				Launchd	Invalid Code Signature	Invalid Code Signature							
				Cron	Right-to-Left Override	Right-to-Left Override							
				System Timers	Rename System Utilities	Rename System Utilities							
				Valid Accounts	Marked Task or Service	Marked Task or Service							
				Default Accounts	Space after Filename	Space after Filename							
				Local Accounts	Modify Authentication Process	Modify Authentication Process							
					Indirect Command Execution	Indirect Command Execution							
					Invalid Code Signature	Invalid Code Signature							
					Right-to-Left Override	Right-to-Left Override							
					Rename System Utilities	Rename System Utilities							
					Marked Task or Service	Marked Task or Service							
					Space after Filename	Space after Filename							
					Modify Authentication Process	Modify Authentication Process							
					Indirect Command Execution	Indirect Command Execution							
					Invalid Code Signature	Invalid Code Signature							
					Right-to-Left Override	Right-to-Left Override							
					Rename System Utilities	Rename System Utilities							
					Marked Task or Service	Marked Task or Service							
					Space after Filename	Space after Filename							
					Modify Authentication Process	Modify Authentication Process							
					Indirect Command Execution	Indirect Command Execution							
					Invalid Code Signature	Invalid Code Signature							
					Right-to-Left Override	Right-to-Left Override							
					Rename System Utilities	Rename System Utilities							
					Marked Task or Service	Marked Task or Service							
					Space after Filename	Space after Filename							
					Modify Authentication Process	Modify Authentication Process							
					Indirect Command Execution	Indirect Command Execution							
					Invalid Code Signature	Invalid Code Signature							
					Right-to-Left Override	Right-to-Left Override							
					Rename System Utilities	Rename System Utilities							
					Marked Task or Service	Marked Task or Service							
					Space after Filename	Space after Filename							
					Modify Authentication Process	Modify Authentication Process							
					Indirect Command Execution	Indirect Command Execution							
					Invalid Code Signature	Invalid Code Signature							
					Right-to-Left Override	Right-to-Left Override							
					Rename System Utilities	Rename System Utilities							
					Marked Task or Service	Marked Task or Service							
					Space after Filename	Space after Filename							
					Modify Authentication Process	Modify Authentication Process							
					Indirect Command Execution	Indirect Command Execution							
					Invalid Code Signature	Invalid Code Signature							
					Right-to-Left Override	Right-to-Left Override							
					Rename System Utilities	Rename System Utilities							
					Marked Task or Service	Marked Task or Service							
					Space after Filename	Space after Filename							
					Modify Authentication Process	Modify Authentication Process							
					Indirect Command Execution	Indirect Command Execution							
					Invalid Code Signature	Invalid Code Signature							
					Right-to-Left Override	Right-to-Left Override							
					Rename System Utilities	Rename System Utilities							
					Marked Task or Service	Marked Task or Service							
					Space after Filename	Space after Filename							
					Modify Authentication Process	Modify Authentication Process							
					Indirect Command Execution	Indirect Command Execution							
					Invalid Code Signature	Invalid Code Signature							
					Right-to-Left Override	Right-to-Left Override							
					Rename System Utilities	Rename System Utilities							
					Marked Task or Service	Marked Task or Service							
					Space after Filename	Space after Filename							
					Modify Authentication Process	Modify Authentication Process							
					Indirect Command Execution	Indirect Command Execution							
					Invalid Code Signature	Invalid Code Signature							
					Right-to-Left Override	Right-to-Left Override							
					Rename System Utilities	Rename System Utilities							
					Marked Task or Service	Marked Task or Service							
					Space after Filename	Space after Filename							
					Modify Authentication Process	Modify Authentication Process							
					Indirect Command Execution	Indirect Command Execution							
					Invalid Code Signature	Invalid Code Signature							
					Right-to-Left Override	Right-to-Left Override							
					Rename System Utilities	Rename System Utilities							
					Marked Task or Service	Marked Task or Service							
					Space after Filename	Space after Filename							
					Modify Authentication Process	Modify Authentication Process							
					Indirect Command Execution	Indirect Command Execution							
					Invalid Code Signature	Invalid Code Signature							
					Right-to-Left Override	Right-to-Left Override							
					Rename System Utilities	Rename System Utilities							
					Marked Task or Service	Marked Task or Service							
					Space after Filename	Space after Filename							
					Modify Authentication Process	Modify Authentication Process							
					Indirect Command Execution	Indirect Command Execution							
					Invalid Code Signature	Invalid Code Signature							
					Right-to-Left Override	Right-to-Left Override							
					Rename System Utilities	Rename System Utilities							
					Marked Task or Service	Marked Task or Service							
					Space after Filename	Space after Filename							
					Modify Authentication Process	Modify Authentication Process							
					Indirect Command Execution	Indirect Command Execution							
					Invalid Code Signature	Invalid Code Signature							
					Right-to-Left Override	Right-to-Left Override							
					Rename System Utilities	Rename System Utilities							
					Marked Task or Service	Marked Task or Service							
					Space after Filename	Space after Filename							
					Modify Authentication Process	Modify Authentication Process							
					Indirect Command Execution	Indirect Command Execution							
					Invalid Code Signature	Invalid Code Signature							
					Right-to-Left Override	Right-to-Left Override							
					Rename System Utilities	Rename System Utilities							
					Marked Task or Service	Marked Task or Service							
					Space after Filename	Space after Filename							
					Modify Authentication Process	Modify Authentication Process							
					Indirect Command Execution	Indirect Command Execution							
					Invalid Code Signature	Invalid Code Signature							
					Right-to-Left Override	Right-to-Left Override							
					Rename System Utilities	Rename System Utilities							
					Marked Task or Service	Marked Task or Service							
					Space after Filename	Space after Filename							
					Modify Authentication Process	Modify Authentication Process							
					Indirect Command Execution	Indirect Command Execution							
					Invalid Code Signature	Invalid Code Signature							
					Right-to-Left Override	Right-to-Left Override							
					Rename System Utilities	Rename System Utilities							
					Marked Task or Service	Marked Task or Service							
					Space after Filename	Space after Filename							
					Modify Authentication Process	Modify Authentication Process							
					Indirect Command Execution	Indirect Command Execution							
					Invalid Code Signature	Invalid Code Signature							
					Right-to-Left Override	Right-to-Left Override							

TECHNIQUES FROM OCEAN LOTUS' MALWARE

[illegible]

Dennis, Goopy, SOUNDBITE,
KOMPROGO, PHOREAL,
WINDSHIELD, OCEANLOTUS.D/F,
and Kerrdown

Generated from ATT&CK Navigator
bit.ly/attacknav

PULLING IT ALL TOGETHER

[illegible]

Initial Access

Drive-by Compromise

Exploit Public-Facing Application

External Remote Services

Hardware Additions

Phishing

Spearphishing Attachment

Spearphishing Link

Spearphishing via Service

Replication Through Removable Media

Supply Chain Compromise

Execution

Command and Scripting Interpreter

PowerShell

AppleScript

Windows Command Shell

Unix Shell

Visual Basic

Python

JavaScript

Network Device CLI

Container Administration Command

[illegible]

= Ocean Lotus

= Ocean Lotus' Software

= Both

Dennis, Goopy, SOUNDBITE,
KOMPROGO, PHOREAL,
WINDSHIELD, OCEANLOTUS.D/F,
and Kerrrdown

ADDING EXTERNAL REPORTING: MAPPING TO ATT&CK

Afterwards, the persistence file will be created in `/Library/LaunchDaemons/` or `~/Library/LaunchAgents/` folder. The

T1543.004 – Create or Modify System Process: Launch Daemon

randomly generated file date and time.

For the initial information packet, the backdoor also collects the following:

T1564.001 – Hide Artifacts: Hidden Files and Directories

T1070.006 – Indicator Removal on Host: Timestamp

```
sw_vers -productVersion
```

T1033 – System Owner/User Discovery

T1082 – System Information Discovery

Figure 15. OS version

Running `getpwnuid -> pw_name` `scutil - -get ComputerName` and `uname -m` will provide the following returns respectively:

ATT&CK MAPPING PROCESS

0. Find the behavior

1. Research the behavior

2. Identify the Tactic(s)

3. Identify the Technique(s)

4. Identify the Sub-Technique(s)

5. Compare notes



MITRE | ATT&CK®

Best Practices for MITRE ATT&CK® Mapping

<https://us-cert.cisa.gov/best-practices-mitre-attckr-mapping>



<https://www.cybrary.it/course/mitre-attack-defender-mad-attack-for-cyber-threat-intelligence>



LEVERAGING INTERNAL INTEL

- Map internal reporting
- GitHub resources
- In-depth technical products
- Red & Threat Hunting teams
- Honeypot Reports
- Binary analysis



```
14 0x00401d07 3 95 sym.my_ecb_crypt
15 0x00400ba0 1 6 sym.imp.__stack_chk_fail
16 0x00403480 1 9 sym._fini
17 0x0040274c 17 1998 sym._des_crypt
18 0x00401600 9 389 sym.crypt_1mb
19 0x00400f6d 23 582 sym.bypass_dir
20 0x00401785 9 379 sym.crypt_all
21 0x004033f0 4 101 sym.__libc_csu_init
22 0x00400e36 6 311 main
```

MAPPING DIRECTLY FROM A SAMPLE

T1059.004 – Command and Scripting Interpreter: Unix Shell

T1027 – Obfuscated Files or Information

T1553.001 – Subvert Trust Controls: Gatekeeper Bypass

T1083 – File and Directory Discovery

T1140 – Deobfuscate/Decode Files or Information

T1070.004 – Indicator Removal on Host: File Deletion

```
1  #!/bin/bash
2  NiIASKWgwKHzfjHn="$( cd "$( dirname "${BASH_SOURCE[0]}" )" >/dev/null 2>&1 &&
3  RLJQXaUXkiFodbEn="$( basename "${BASH_SOURCE[0]}" )"
4  asFaGDyzpKvtLaSb="<giant base64 removed for readability>"
5  TEMPPATH_IOP="Contents/Resources/configureDefault.def"
6  krcxhMaZjArWHDx0="ALL tim nha Chi Ngoc Canada.doc"
7  crkEVUWKhhHDpNy="cXzxXRFWYXstJJZX"
8  ls ~/Downloads
9  if [[ $? == 0 ]]; then
10 find ~ -name "*$RLJQXaUXkiFodbEn*" -exec xattr -d com.apple.quarantine {}
11 if [[ $NiIASKWgwKHzfjHn == *"AppTranslocation"* ]]; then
12 md5="$( md5 "$NiIASKWgwKHzfjHn/$RLJQXaUXkiFodbEn" | cut -d '=' -f 2 )"
13 A="$( dirname "$NiIASKWgwKHzfjHn/$RLJQXaUXkiFodbEn" )"
14 rh5="$( basename "${A}" )"
15 find ~ -type f -name "$RLJQXaUXkiFodbEn" -exec md5 {} + | grep $md5 | grep
16 else
17 AmLGEEGPFKiYFBxM="$( dirname "$NiIASKWgwKHzfjHn/$RLJQXaUXkiFodbEn" )"
18 FuTJofXeGGrBlR0x="$( dirname "$AmLGEEGPFKiYFBxM" )"
19 cp "$AmLGEEGPFKiYFBxM/$TEMPPATH_IOP" "/tmp/$krcxhMaZjArWHDx0" && op
20 echo $asFaGDyzpKvtLaSb | base64 -D > "$AmLGEEGPFKiYFBxM/$TEMPPATH_I
21 $TEMPPATH_IOP" & >/dev/null 2>&1
22 sleep 3 ; rm -rf "$AmLGEEGPFKiYFBxM" ; mv "/tmp/$krcxhMaZjArWHDx0" "$F
23 $krcxhMaZjArWHDx0" &
24 killall -9 find
```




Using ATT&CK for ...

Threat Hunting



WHAT WE KNOW

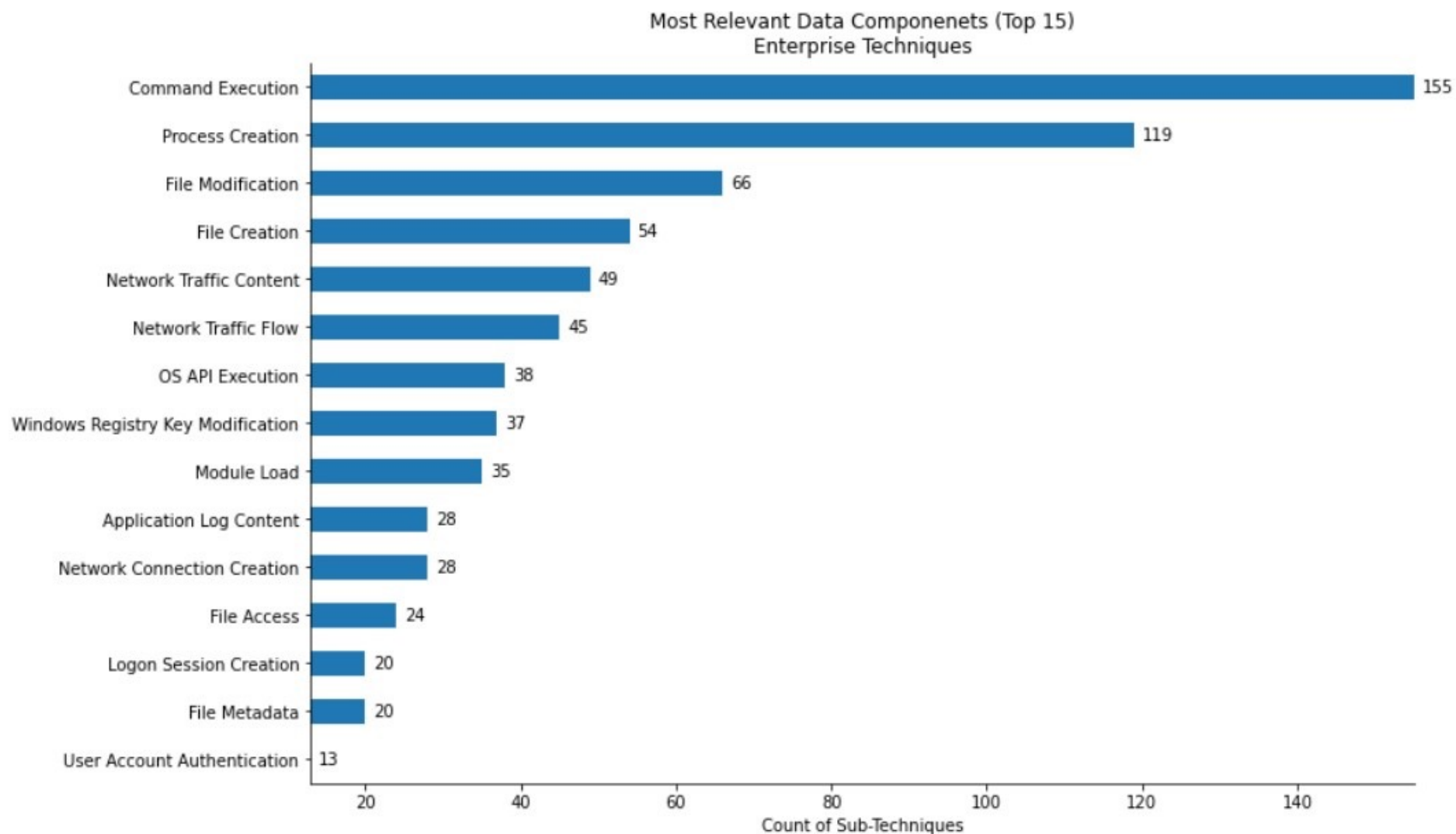
Dennis, Goopy, SOUNDBITE, KOMPROGO, PHOREAL, WINDSHIELD, OCEANLOTUS.D/F, and Kerndown

 = Ocean Lotus
 = Ocean Lotus' Software
 = Both

Dennis, Goopy, SOUNDBITE,
KOMPROGO, PHOREAL,
WINDSHIELD, OCEANLOTUS.D/F,
and Kerrdown

 = Ocean Lotus
 = Ocean Lotus' Software
 = Both

DATA SOURCES



WHAT DO I HUNT?

- Usage, bottleneck, data
- Auto-runs -> (tactic == persistence)
- Identify macOS techniques associated with OceanLotus

Execution
Command and Scripting Interpreter
PowerShell
AppleScript
Windows Command Shell
Unix Shell
Visual Basic
Python
JavaScript
Network Device CLI
Container Administration Command
Deploy Container
Exploitation for Client Execution
Inter-Process Communication
Native API
Scheduled Task/Job
At (Windows)
Scheduled Task
At (Linux)
Launchd
Cron
Systemd Timers
Container Orchestration Job
Shared Modules
Software Deployment Tools
System Services
Launchctl
Service Execution
User Execution
Malicious Link
Malicious File
Malicious Image
Windows Management Instrumentation

Persistence
Account Manipulation
BITS Jobs
Boot or Logon Autostart Execution
Registry Run Keys / Startup Folder
Authentication Package
Time Providers
Winlogon Helper DLL
Security Support Provider
Kernel Modules and Extensions
Re-opened Applications
LSASS Driver
Shortcut Modification
Port Monitors
Plist Modification
Print Processors
XDG Autostart Entries
Active Setup
Boot or Logon Initialization Scripts
Browser Extensions
Compromise Client Software Binary
Create Account
Create or Modify System Process
Launch Agent
Systemd Service
Windows Service
Launch Daemon
Event Triggered Execution
External Remote Services
Hijack Execution Flow
Implant Internal Image
Modify Authentication Process
Office Application Startup
Pre-OS Boot
Scheduled Task/Job
At (Windows)
Scheduled Task
At (Linux)
Launchd
Cron
Systemd Timers
Container Orchestration Job
Server Software Component
SQL Stored Procedures
Transport Agent
Web Shell
Traffic Signaling
Valid Accounts
Default Accounts
Domain Accounts
Local Accounts
Cloud Accounts

Privilege Escalation
Abuse Elevation Controls
Access Token Manipulation
Boot or Logon Autostart Execution
Registry Run Keys / Startup Folder
Authentication Package
Time Providers
Winlogon Helper DLL
Security Support Provider
Kernel Modules and Extensions
Re-opened Applications
LSASS Driver
Shortcut Modification
Port Monitors
Plist Modification
Print Processors
XDG Autostart Entries
Active Setup
Boot or Logon Initialization Scripts
Create or Modify System Process
Launch Agent
Systemd Service
Windows Service
Launch Daemon
Domain Policy Modification
Escape to Host
Event Triggered Execution
Exploitation for Privilege Escalation
Hijack Execution Flow
Services File Permissions
Executable Installer File
Services Registry Permissions
Path Interception by PATH
Path Interception by Search Path
DLL Search Order
DLL Side-Loading
Dynamic Linker Hijacking
Dylib Hijacking
COR_PROFILER
Process Injection
Dynamic-link Libraries
Portable Executable
Thread Execution
Asynchronous Process Creation
Thread Local Storage
Ptrace System Call
Proc Memory
Extra Window Memory
Process Doppelgänger
Process Hollowing
VDSO Hijacking

Persistence	Privilege Escalation	Defense Evasion	Credential Access
Account Manipulation	Abuse Elevation Control Mechanism	Abuse Elevation Control Mechanism	Brute Force
BITS Jobs	Access Token Manipulation	Access Token Manipulation	Credentials from Passwords
Boot or Logon Autostart Execution	Boot or Logon Autostart Execution	BITS Jobs	Exploitation for Credential Access
Registry Run Keys / Startup Folder	Registry Run Keys / Startup Folder	Build Image on Host	Forced Authentication
Authentication Package			
Time Providers			
Winlogon Helper DLL			
Security Support Providers			
Kernel Modules and Extensions			
Re-opened Applications			
LSASS Driver			
Shortcut Modification			
Port Monitors			
Plist Modification			
Print Processors			
XDG Autostart Entries			
Active Setup			
Boot or Logon Initialization			
Browser Extensions			
Compromise Client Software			
Create Account			
Create or Modify System Process			
Launch Agent			
Systemd Service			
Windows Service			
Launch Daemon			
Event Triggered Execution			
External Remote Services			
Hijack Execution Flow			
Implant Internal Image			
Modify Authentication Package			
Office Application Start			
Pre-OS Boot			
Scheduled Task/Job			
At (Windows)			
Scheduled Task			
At (Linux)			
Launchd			
Cron			

Create or Modify System Process: Launch Agent

Other sub-techniques of Create or Modify System Process (4)

Adversaries may create or modify launch agents to repeatedly execute malicious payloads as part of persistence. Per Apple's developer documentation, when a user logs in, a per-user launchd process is started which loads the parameters for each launch-on-demand user agent from the property list (plist) files found in `/System/Library/LaunchAgents`, `/Library/LaunchAgents`, and `$HOME/Library/LaunchAgents` [1] [2] [3]. These launch agents have property list files which point to the executables that will be launched [4].

Adversaries may install a new launch agent that can be configured to execute at login by using launchd or launchctl to load a plist into the appropriate directories [5] [6]. The agent name may be disguised by using a name from a related operating system or benign software. Launch Agents are created with user level privileges and are executed with the privileges of the user when they log in [7] [8]. They can be set up to execute when a specific user logs in (in the specific user's directory structure) or when any user logs in (which requires administrator privileges).

ID: T1543.001

Sub-technique of: T1543

Tactics: Persistence, Privilege Escalation

Platforms: macOS

Permissions

Required: Administrator, User

Data Sources: Command: Command Execution, File: File Creation, File: File Modification, Service: Service Creation, Service: Service Modification

Version: 1.0

Created: 17 January 2020

Last Modified: 25 March 2020

Version Permalink

Path Interception by Search Order Hijacking	Impair Defenses
DLL Search Order Hijacking	Disable or Modify Tools
DLL Side-Loading	Disable Windows Event Logging
Dynamic Linker Hijacking	Impair Command History Logging
Dylib Hijacking	Disable or Modify System Firewall
COR_PROFILER	Indicator Blocking

DATA SOURCES

ss: Launch Agent

(4)

ous payloads as part
er-user launchd
ser agent from the
/LaunchAgents, and
st files which point

at login by using
agent name may be
re. Launch Agents
he user when they
specific user's
ivileges).

ID: T1543.001

Sub-technique of: T1543

① Tactics: Persistence, Privilege Escalation

① Platforms: macOS

① Permissions

Required: Administrator, User

① Data Sources: Command: Command Execution, File: File Creation, File: File Modification, Service: Service Creation, Service: Service Modification

Version: 1.0

Created: 17 January 2020

Last Modified: 25 March 2020

[Version Permalink](#)



ATT&CK™

query / packs / incident-response.conf

adding platform tag incident-response pack (#4155) ✓

7 contributors



283 lines (283 sloc) | 13.5 KB

```
1 {
2   "queries": {
3     "launchd": {
4       "query" : "select * from launchd;",
5       "interval" : "3600",
6       "platform" : "darwin",
7       "version" : "1.4.5",
8       "description" : "Retrieves all the daemons that will run",
9       "value" : "Identify malware that uses this persistence me
10    },
11   "startup_items": {
12     "query" : "select * from startup_items;",
13     "interval" : "86400",
14     "platform" : "darwin",
```



BUILDING USE CASES

Procedure Examples

ID	Name	Description
S0482	Bundlore	Bundlore can persist via a LaunchAgent. ^[9]
S0274	Calisto	Calisto adds a .plist file to the /Library/LaunchAgents folder to maintain persistence. ^[10]
S0369	CoinTicker	CoinTicker creates user launch agents named .espl.plist and com.apple.[random string].plist to establish persistence. ^[11]
S0352	OSX_OCEANLOTUS.D	OSX_OCEANLOTUS.D can create a persistence file in the folder /Library/LaunchAgents. ^{[19][20]}
S0279	Proton	Proton persists via Launch Agent. ^[16]
S0595	ThiefQuest	ThiefQuest installs a launch item using an embedded encrypted launch agent property list template. The plist file is installed in the ~/Library/LaunchAgents/ folder and configured with the path to the persistent binary located in the ~/Library/ folder. ^[21]

BUILDING USE CASES

For a user other than root, it takes the MD5 hash of the structure returned by `getpwuid()` and breaks the hash down into segments `<first 8 chars of hash>-<next 16 chars of hash>-<last 8 chars of hash>`. This segmented MD5 hash is prepended with "0000-" then used as a directory in `~/Library/OpenSSL/` to store the executable file (see Figure 3). If the user is root, the executable is stored in the system wide library directory at `/Library/TimeMachine/bin/mtmfs`.

It is interesting to note that the executable and plist locations look like legitimate applications.

UID	plist Location	Executable Location
0	/Library/LaunchDaemons/com.apple.mtmfsd.plist	/Library/TimeMachine/bin/mtmfs
> 0	~/Library/LaunchAgents/com.apple.openssl.plist	~/Library/OpenSSL/0000-<segmented MD5 hash>/servicessl

Figure 3. plist and executable names and locations based on UID

S0276	Keydnep	Keydnep uses a Launch Agent to persist. ^[17]
S0162	Komplex	The Komplex trojan creates a persistent launch agent in the <code>\$HOME/Library/LaunchAgents/com.apple.updates</code> directory. ^[18]
S0282	MacSpy	MacSpy persists via a Launch Agent. ^[19]
S0198	NETWIRE	NETWIRE can use launch agents for persistence.
S0352	OSX_OCEANLOTUS.D	OSX_OCEANLOTUS.D can create a persistence file.
S0279	Proton	Proton persists via Launch Agent. ^[16]
S0595	ThiefQuest	ThiefQuest installs a launch item using an embedded plist file. The file is installed in the <code>~/Library/LaunchAgents/</code> folder, located in the <code>~/Library/</code> folder. ^[21]

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
  <dict>
    <key>Label</key>
    <string>com.apple.marcoagent.voiceinstallerd</string>
    <key>ProgramArguments</key>
    <array>
      <string>/Users/test/Library/User Photos/mount_devfs</string>
    </array>
    <key>RunAtLoad</key>
    <true/>
    <key>KeepAlive</key>
    <true/>
  </dict>
</plist>
```



Figure 8. Plist file `~/Library/LaunchAgents/com.apple.marcoagent.voiceinstallerd.plist`

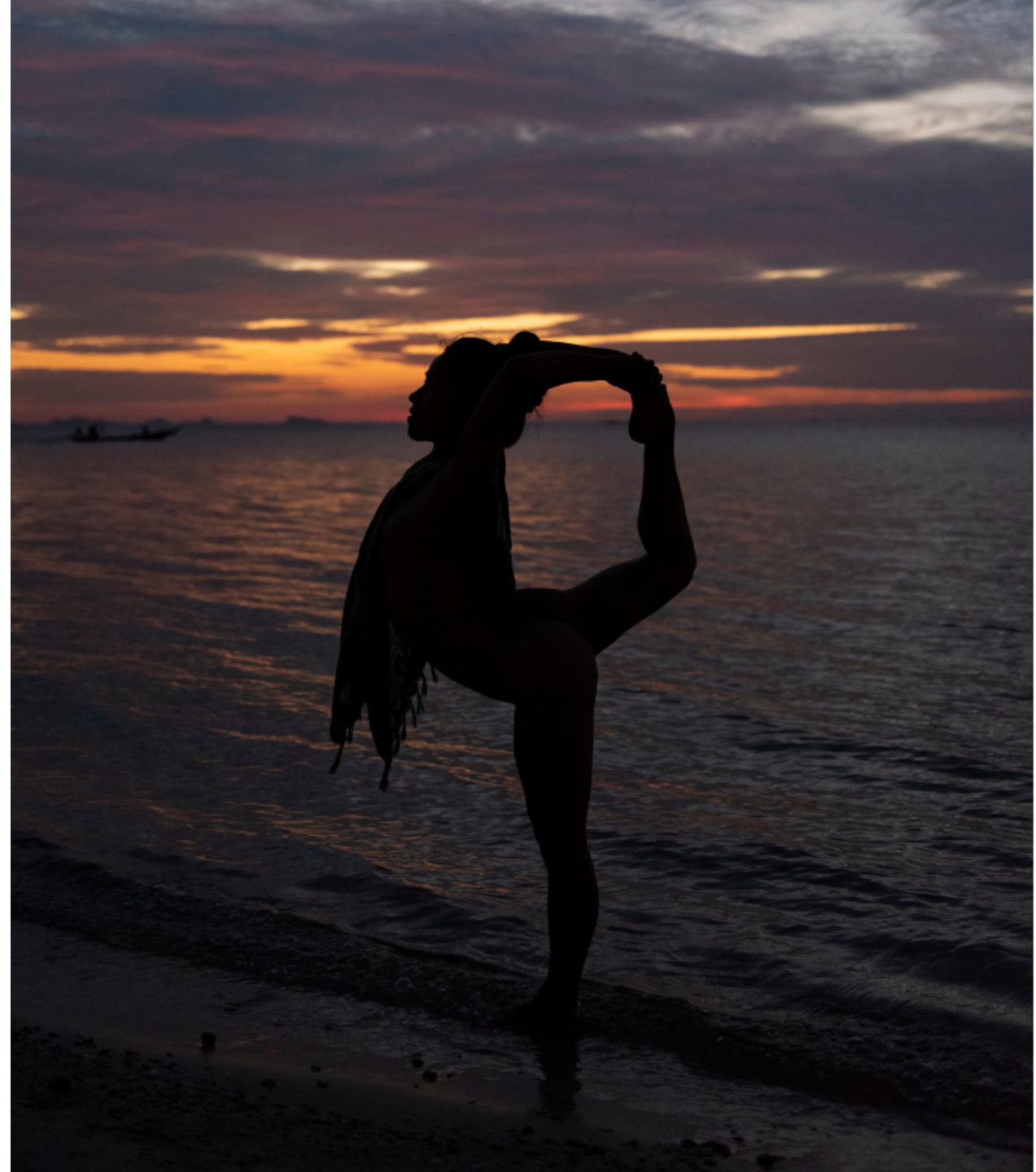
ADDITIONAL HUNTING RESOURCES

- [Hunting with ATT&CK -> MITRE - TTP-Based Hunting](#)
- Filippo Mottini – [osquery-attck](#)
- @Cyb3rWard0g's (Roberto Rodriguez) [Threat Hunter Playbook](#)
- David Bianco's [Threat Hunting Project](#)



Using ATT&CK for ...

ADVERSARY EMULATION



WHAT WE KNOW

[illegible]

Dennis, Goopy, SOUNDBITE,
KOMPROGO, PHOREAL,
WINDSHIELD, OCEANLOTUS.D/F,
and Kerrdown

 = Ocean Lotus
 = Ocean Lotus' Software
 = Both

ATT&CK IN EMULATION

- CTID Adversary Emulation

Plans

- Atomic Red Team

- MITRE CALDERA™

Atomic Test #1 - Launch Agent

Create a plist and execute it

Supported Platforms: macOS

auto_generated_guid: a5983dee-bf6c-4eaf-951c-dbc1a7b90900

Inputs:

Name	Description	Type	
plist_filename	filename	String	com.atomicredteam.plist
path_malicious_plist	Name of file to store in cron folder	String	\$PathToAtomsFolder/

Attack Commands: Run with **bash**! Elevation Required (e.g. root or admin)

```
if [ ! -d ~/Library/LaunchAgents ]; then mkdir ~/Library/LaunchAgents; fi;
sudo cp #{path_malicious_plist} ~/Library/LaunchAgents/#{plist_filename}
sudo launchctl load -w ~/Library/LaunchAgents/#{plist_filename}
```

Cleanup Commands:

```
sudo launchctl unload ~/Library/LaunchAgents/#{plist_filename}
sudo rm ~/Library/LaunchAgents/#{plist_filename}
```

10.B - Use VNC Persistence (T1021.005)

On your Ubuntu machine:

1. Setup an SSH tunnel to forward VNC through the Attack Platform

```
ssh <attacker>@192.168.0.4 -L 12345:<cfo_ip>:5900
```

Provide the <attacker> password when prompted.

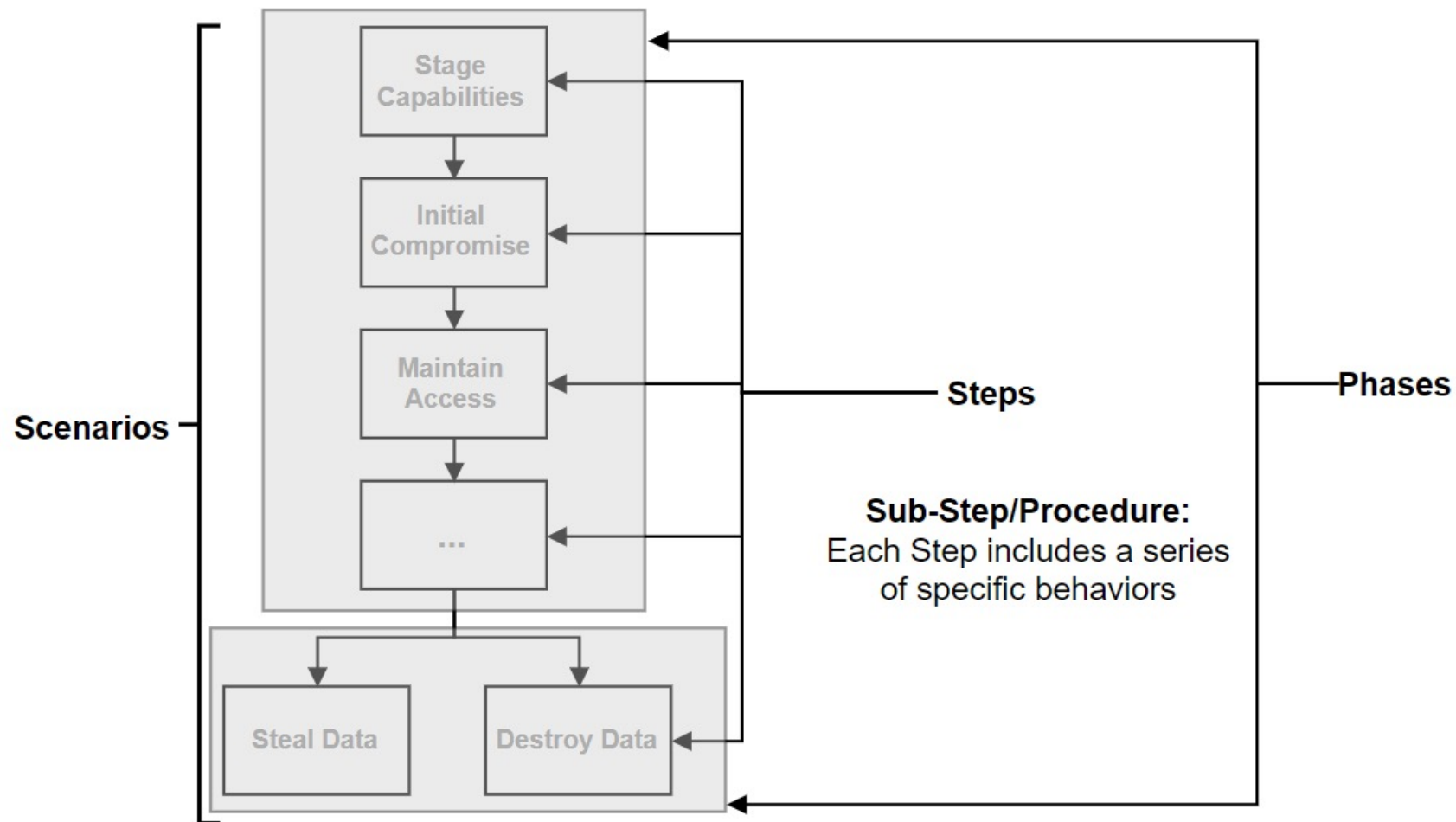
2. Open a VNC client
3. Set the target to 127.0.0.1:12345 and connect

CALDERA

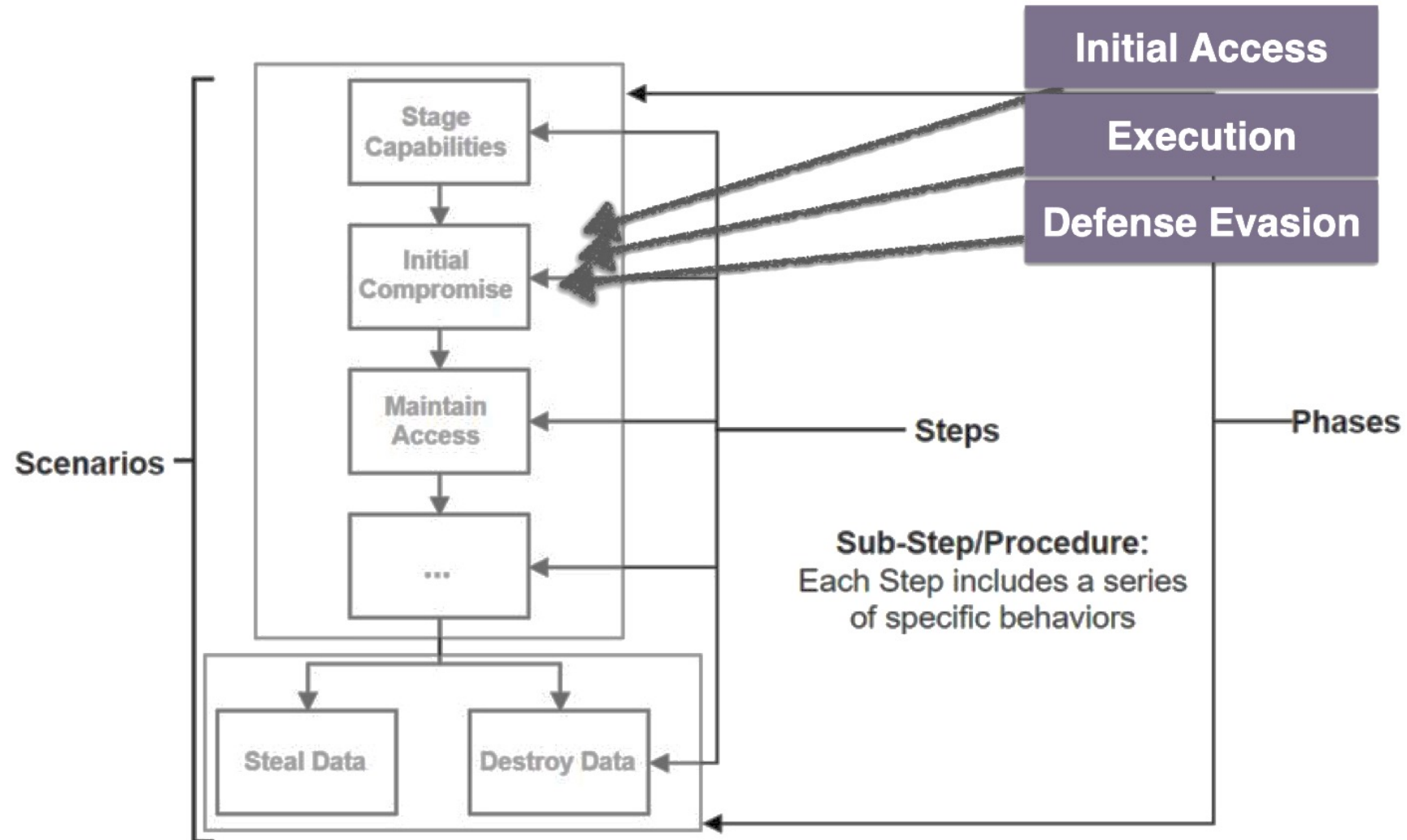
Welcome home. Go into the Agents tab to review your deployed agents.

BUILDING AN ADVERSARY EMULATION PLAN WITH ATT&CK

- Scenario
- Step
- Sub-Step/Procedure
- Objective



UNPACKING A STEP



BUILDING OFF CTI

T1059.004 – Command and Scripting Interpreter: Unix Shell

T1027 – Obfuscated Files or Information

T1553.001 – Subvert Trust Controls: Gatekeeper Bypass

T1083 – File and Directory Discovery

T1140 – Deobfuscate/Decode Files or Information

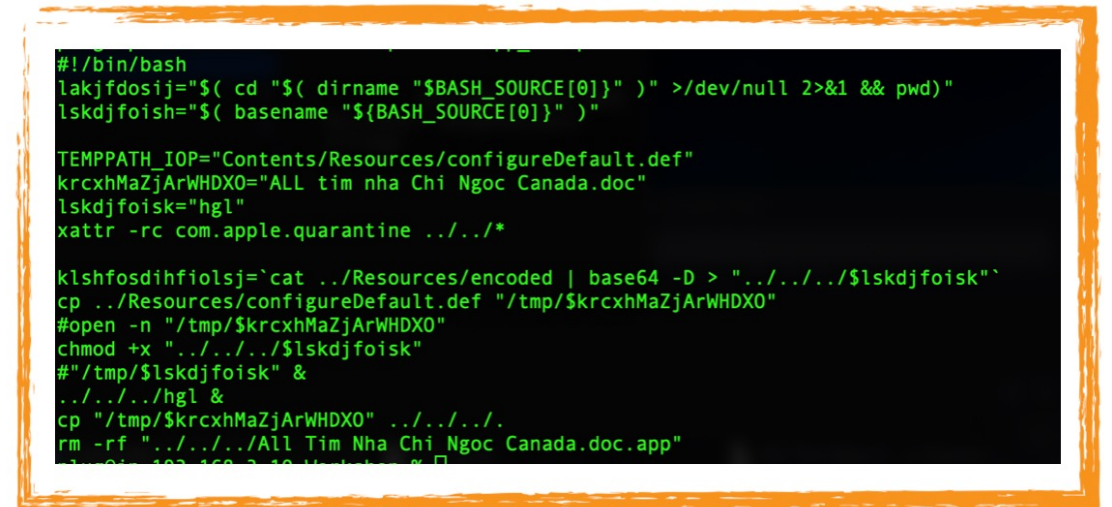
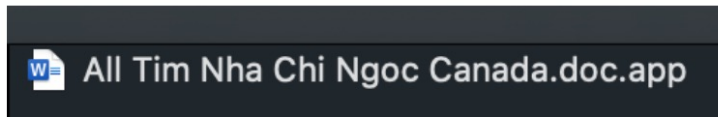
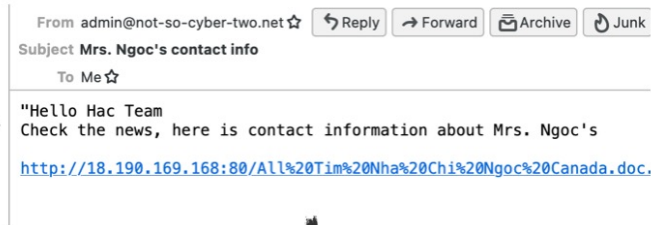
T1070.004 – Indicator Removal on Host: File Deletion

```
1  #!/bin/bash
2  NiIASKWgwKHzfjHn="$( cd "$( dirname "${BASH_SOURCE[0]}" )" >/dev/null 2>&1 &&
3  RLJQXaUXkiFodbEn="$( basename "${BASH_SOURCE[0]}" )"
4  asFaGDyzpKvtLaSb="<giant base64 removed for readability>"
5  TEMPPATH_IOP="Contents/Resources/configureDefault.def"
6  krcxhMaZjArWHDx0="ALL tim nha Chi Ngoc Canada.doc"
7  crkEVUWKhhHDpNy="cXzxXRFWYXstJJZX"
8  ls ~/Downloads
9  if [[ $? == 0 ]]; then
10 find ~ -name "*$RLJQXaUXkiFodbEn*" -exec xattr -d com.apple.quarantine {}
11 if [[ $NiIASKWgwKHzfjHn == *"AppTranslocation*" ]]; then
12 md5="$( md5 "$NiIASKWgwKHzfjHn/$RLJQXaUXkiFodbEn" | cut -d '=' -f 2 )"
13 A="$( dirname "$NiIASKWgwKHzfjHn/$RLJQXaUXkiFodbEn" )"
14 rh5="$( basename "${A}" )"
15 find ~ -type f -name "$RLJQXaUXkiFodbEn" -exec md5 {} + | grep $md5 | grep
16 else
17 AmLGEEGPFKiYFBxM="$( dirname "$NiIASKWgwKHzfjHn/$RLJQXaUXkiFodbEn" )"
18 FuTJofXeGGrBlR0x="$( dirname "$AmLGEEGPFKiYFBxM" )"
19 cp "$AmLGEEGPFKiYFBxM/$TEMPPATH_IOP" "/tmp/$krcxhMaZjArWHDx0" && op
20 echo $asFaGDyzpKvtLaSb | base64 -D > "$AmLGEEGPFKiYFBxM/$TEMPPATH_I
21 $TEMPPATH_IOP" & >/dev/null 2>&1
22 sleep 3 ; rm -rf "$AmLGEEGPFKiYFBxM" ; mv "/tmp/$krcxhMaZjArWHDx0" "$F
23 $krcxhMaZjArWHDx0" &
24 killall -9 find
```

TACTICS, TECHNIQUES & REPORTING

	Commonly Called	Technique	Reporting
Initial Access →	Click and Bait	Phishing: <u>Spearphishing</u> Attachment	2021 - A great summary - Click and Bait: Vietnamese Human Rights Defenders Targeted with Spyware Attacks 2014 - Vietnamese Malware Gets Very Personal 2016- OceanLotus OS X Malware Disguises Itself as Adobe Flash Update 2018 - New OceanLotus Backdoor Discovered Targeting macOS
Defense Evasion →	Ideograph/Homoglyphs	Masquerading: Match Legitimate Name or Location	2020 - Use of ideograph trick
Execution →	Shell Script	Command and Scripting Interpreter: Unix Shell	2020 - Shell Script use 2020 - Apt <u>Multi-Stage MacOS</u> Trojan
Defense Evasion →	Gatekeeper Bypass	Subvert Trust Controls: Gatekeeper Bypass	2020 - APT32 Multi-stage macOS Trojan Innovates on Crimeware Scripting Technique
Defense Evasion →	<u>Base 64</u> encoding (specified type of encoding)	Data Encoding: Standard Encoding	2018 - Obfuscated Macros 2020 - Obfuscated Payload 2020 - Apt <u>Multi-Stage MacOS</u> Trojan
Defense Evasion →	chmod +x	File and Directory Permissions: Linux and Mac File Directory Permissions	2020 - APT32 Multi-stage macOS Trojan Innovates on Crimeware Scripting Technique 2020 - Backdoor. <u>MacOS.OCEANLOTUS.F</u>
Defense Evasion →	clean up	Indicator Removal on Host: File Deletion	2018 - New MacOS Backdoor Linked to OceanLotus Found 2020 - New MacOS Backdoor Connected to OceanLotus Surfaces

COMPLETING THE PICTURE



Hunt for Red Apples: OceanLotus Edition

@plugoxr, @CptOfEvilMinion, @1njection, @wildphish, @CoolestCatiKnow

RABBIT HOLE OF PROCEDURES

Subvert Trust Controls: Gatekeeper Bypass

Other sub-techniques of Subvert Trust Controls (6)

xattr (/usr/bin/xattr)

Display and manipulate extended attributes. Used by malware and threat actors as a means to bypass **Gatekeeper** and **Notarization** checks on macOS. Incredibly, any process or user can remove the file attribute that is required for these checks to proceed without admin rights.

Common Arguments

```
xattr -d com.apple.quarantine
```

xattr -c

```
xattr -cr
```

ITW Examples

OceanLotus

```
find /Users/user -name *ALL tim nha Chi Ngoc Canada* -exec  
xattr -d com.apple.quarantine {} +
```



attributes that signify programs are from untrusted sources to subvert OS. When documents, applications, or programs are downloaded an extended `apple.quarantine` can be set on the file by the application performing the so known as a quarantine flag, is read by Apple's Gatekeeper defense program provides a prompt to the user to allow or deny execution. Gatekeeper also monitors dynamic libraries (dylibs) loaded outside the application folder on any quarantined `open` function. If the quarantine flag is set in macOS 10.15+, Gatekeeper also fetches the file from the Internet and sends a cryptographic hash to Apple's servers to check for validity for all

It-in system and not imposed by macOS. If an application opts-in, a file
et will be given a quarantine flag before being saved to disk. Any application or
to the file can change or strip the quarantine flag. With elevated permission
removed from any file. The presence of the `com.apple.quarantine` quarantine
e `xattr` command `xattr -l /path/to/examplefile`. Similarly, this attribute can
h all files in a folder using `xattr`, `sudo xattr -d com.apple.quarantine`

from USB flash drive, optical disk, external hard drive, or even from a drive
rk do not set this flag. Additionally, it is possible to avoid setting this flag using
n may bypass Gatekeeper. An application can load dylibs located outside of the
g the `NSCreateObjectFileImageFromMemory` API call which may bypass

RABBIT HOLE OPPORTUNITY

Subvert Trust Controls: Gatekeeper

Other sub-techniques of Subvert Trust Controls (6)

Adversaries may modify file attributes that signify programs are from Gatekeeper controls in macOS. When documents, applications, or programs are downloaded from the Internet, they are given a quarantine flag before being downloaded. This attribute, also known as a quarantine flag, is read by Gatekeeper when the file is run and provides a prompt to the user to allow or deny an application's usage of dynamic libraries (dylibs) loaded outside the application binary, often using the `dlopen` function. If the quarantine flag is set, Gatekeeper checks for a notarization ticket and sends a cryptographic hash to Apple to verify the unsigned executables.^{[1][2]}

The quarantine flag is an opt-in system and not imposed by macOS. Applications downloaded from the Internet will be given a quarantine flag before being downloaded. A user with write permissions to the file can change or strip the quarantine flag (using `sudo`), this attribute can be removed from any file. The presence of the quarantine flag can be checked with the `kattr` command `xattr -l /path/to/file`.

```
NSData *data = NSDataFromMemory(codeAddr, codeSize);  
NSObjectFileImageFromMemory(codeAddr, codeSize, &fileImage);
```

```
module = NSLinkModule(fileImage, "module",  
symbol = NSLookupSymbolInModule(module, "_execute");  
function = NSAddressOfSymbol(symbol);
```

Gatekeeper.^{[4][5][6][7]}

IN-MEMORY MACH-O LOADING

`dyld` supports in-memory loading/linking

```
//vars  
NSObjectFileImage fileImage = NULL;  
NSModule module = NULL;  
NSSymbol symbol = NULL;  
void (*function)(const char *message);  
  
//have an in-memory (file) image of a mach-O file to load/link  
// ->note: memory must be page-aligned and alloc'd via vm_alloc!  
  
//create object file image  
NSData *data = NSDataFromMemory(codeAddr, codeSize);  
NSObjectFileImageFromMemory(codeAddr, codeSize, &fileImage);  
  
//link module  
module = NSLinkModule(fileImage, "<anything>", NSLINKMODULE_OPTION_PRIVATE);  
  
//lookup exported symbol (function)  
symbol = NSLookupSymbolInModule(module, "_" "HelloBlackHat");  
  
//get exported function's address  
function = NSAddressOfSymbol(symbol);  
  
//invoke exported function  
function("thanks for being so offensive ;)");
```

loading a mach-O file from memory

Using ATT&CK for ...

Assessment and Engineering



CONNECTING TEAMS

- **Interview your detection/ops team, CTI team, and red team**
 - CTI: What threats do we face and which techniques should we prioritize?
 - Threat Hunting/Detection: What techniques can we cover, and which can't we?
 - Adversary Emulation: What have we validated?
- **Examine your tools, documentation, and analytics**
 - Tools: Which data sources can we collect?
 - Documentation: Do policies and procedures help us with techniques?
 - Analytics: What techniques can it detect? How much procedure coverage?
- **Look at how your overall technique coverage fares**
 - Are there gaps in either visibility or validation?

PRIORITY TECHNIQUES FROM THREAT INTEL




Initial Access 7 techniques	Execution 7 techniques	Persistence 14 techniques	Privilege Escalation 10 techniques	Defense Evasion 18 techniques	Credential Access 12 techniques	Discovery 19 techniques	Lateral Movement 6 techniques	Collection 13 techniques	Command and Control 16 techniques	Exfiltration 8 techniques	Impact 13 techniques
Drive-by Compromise	II Command and Scripting Interpreter	II Account Manipulation (0/1)	II Abuse Elevation Control Mechanism (0/3)	II Abuse Elevation Control Mechanism (0/3)	II Brute Force (0/4)	II Account Discovery (1/2)	Exploitation of Remote Services	II Archive Collected Data	II Application Layer Protocol (3/4)	II Automated Exfiltration (0/0)	Account Access Removal
Exploit Public-Facing Application	II AppleScript	II Boot or Logon Autostart Execution (0/3)	II Boot or Logon Autostart Execution (0/3)	II Deobfuscate/Decode Files or Information	II Credentials from Password Stores (0/4)	II Application Window Discovery	Internal Spearphishing	Audio Capture	Communication Through Removable Media	Data Transfer Size Limits	Data Destruction
Hardware Additions	II JavaScript	II Boot or Logon Initialization Scripts (0/3)	II Boot or Logon Initialization Scripts (0/3)	II Execution Guardrails (0/1)	II Exploitation for Credential Access	II Browser Bookmark Discovery	Lateral Tool Transfer	Automated Collection	II Data Encoding (1/2)	II Exfiltration Over Alternative Protocol (1/3)	Data Encrypted for Impact
II Phishing (2/3)	II Python	II Browser Extensions	II Create or Modify System Process (2/2)	II Exploitation for Defense Evasion	II Forge Web Credentials (0/1)	II File and Directory Discovery	II Remote Service Session Hijacking (0/1)	II Data from Information Repositories (0/0)	II Data Obfuscation (0/3)	II Exfiltration Over C2 Channel	II Data Manipulation (0/3)
II Supply Chain Compromise (0/3)	II Unix Shell	II Compromise Client Software Binary	II Launch Agent	II File and Directory Permissions Modification (1/1)	II Input Capture (1/3)	II Network Service Scanning	II Remote Services (0/2)	II Data from Local System	II Dynamic Resolution (0/3)	II Exfiltration Over Other Network Medium (0/1)	II Defacement (0/2)
Trusted Relationship	II Visual Basic	II Create Account (0/2)	II Launch Daemon	II Hide Artifacts (2/6)	II Man-in-the-Middle (0/1)	II Network Share Discovery	II Software Deployment Tools	Data from Network Shared Drive	II Encrypted Channel (0/2)	II Exfiltration Over Physical Medium (0/1)	II Disk Wipe (0/2)
II Valid Accounts (1/3)	II Exploitation for Client Execution	II Event Triggered Execution (0/4)	II Exploitation for Privilege Escalation	II Hijack Execution Flow (0/2)	II Modify Authentication Process (0/1)	II Network Sniffing		Data from Removable Media	II Input Capture (1/3)	II Exfiltration Over Web Service (0/2)	II Endpoint Denial of Service (0/4)
	II Native API	II Hijack Execution Flow (0/2)	II Hijack Execution Flow (0/2)	II Impair Defenses (1/4)	II Network Sniffing	II Password Policy Discovery		II Data Staged (0/2)	II Multi-Stage Channels	Scheduled Transfer	Firmware Corruption
	II Scheduled Task/Job (0/2)	II Launch Agent	II Launch Daemon	II Indicator Removal on Host	II OS Credential Dumping (0/0)	II Permission Groups Discovery (0/2)		II Input Capture (1/3)	II Non-Application Layer Protocol		Inhibit System Recovery
	II Software Deployment Tools	II Event Triggered Execution (0/4)	II Event Triggered Execution (0/4)	II Masquerading	II Steal Web Session Cookie	II Process Discovery		II Man-in-the-Middle (0/1)	II Non-Standard Port		Resource Hijacking
	II System Services (0/1)	II Hijack Execution Flow (0/2)	II Hijack Execution Flow (0/2)	II Modify Authentication Process (0/1)	II Two-Factor Authentication Interception	II Remote System Discovery		Screen Capture	II Protocol Tunneling		Service Stop
	II User Execution (2/2)	II Modify Authentication Process (0/1)	II Modify Authentication Process (0/1)	II Obfuscated Files or Information	II Unsecured Credentials (0/3)	II Software Discovery (0/1)		Video Capture	II Proxy (0/4)		System Shutdown/Reboot
		II Scheduled Task/Job (0/2)	II Scheduled Task/Job (0/2)	II Process Injection (0/0)		II System Information Discovery			II Remote Access Software		
		II Valid Accounts (1/3)	II Valid Accounts (1/3)	II Rootkit		II System Location Discovery			II Traffic Signaling (0/1)		
				II Subvert Trust Controls (1/4)		II System Network Configuration Discovery (0/1)			II Web Service (0/3)		
				II Code Signing		II System Network Connections Discovery					
				II Code Signing Policy Modification		II System Owner/User Discovery					
				II Gatekeeper Bypass		II Virtualization/Sandbox Evasion (1/3)					
				II Install Root Certificate							
				II Traffic Signaling (0/1)							

VISIBILITY TO HUNTING/DETECTION

Initial Access 7 techniques	Execution 7 techniques	Persistence 14 techniques	Privilege Escalation 10 techniques	Defense Evasion 18 techniques	Credential Access 12 techniques	Discovery 19 techniques	Lateral Movement 6 techniques	Collection 13 techniques	Command and Control 16 techniques	Exfiltration 8 techniques	Impact 13 techniques
Drive-by Compromise	II Command and Scripting Interpreter (1/5)	II Account Manipulation (0/1)	II Abuse Elevation Control Mechanism (0/3)	II Abuse Elevation Control Mechanism (0/3)	II Brute Force (0/4)	II Account Discovery (0/2)	Exploitation of Remote Services	II Archive Collected Data (0/3)	II Application Layer Protocol (0/4)	II Automated Exfiltration (0/0)	Account Access Removal
Exploit Public-Facing Application	AppleScript	II Boot or Logon Autostart Execution (0/3)	II Boot or Logon Autostart Execution (0/3)	Deobfuscate/Decode Files or Information	II Credentials from Password Stores (0/4)	Application Window Discovery	Internal Spearphishing	Audio Capture	Communication Through Removable Media	Data Transfer Size Limits	Data Destruction
Hardware Additions	JavaScript	II Boot or Logon Initialization Scripts (0/3)	II Boot or Logon Initialization Scripts (0/3)	II Execution Guardrails (0/1)	Exploitation for Credential Access	Browser Bookmark Discovery	Lateral Tool Transfer	Automated Collection	Clipboard Data	II Exfiltration Over Alternative Protocol (0/3)	Data Encrypted for Impact
II Phishing (0/3)	Python	Browser Extensions	II Create or Modify System Process (2/2)	Exploitation for Defense Evasion	II Forge Web Credentials (0/1)	File and Directory Discovery	II Remote Service Session Hijacking (0/1)	II Data from Information Repositories (0/0)	II Data Encoding (0/2)	II Exfiltration Over C2 Channel	II Data Manipulation (0/3)
II Supply Chain Compromise (0/3)	Unix Shell	Compromise Client Software Binary	II Launch Agent	II File and Directory Permissions Modification (0/1)	II Input Capture (0/3)	Network Service Scanning	II Remote Services (0/2)	II Data from Local System	II Data Obfuscation (0/3)	II Exfiltration Over Other Network Medium (0/1)	II Defacement (0/2)
Trusted Relationship	Visual Basic	II Create Account (0/2)	II Launch Daemon	II Hide Artifacts (0/6)	II Man-in-the-Middle (0/1)	Network Share Discovery	Software Deployment Tools	Data from Network Shared Drive	II Dynamic Resolution (0/3)	II Exfiltration Over Physical Medium (0/1)	II Disk Wipe (0/2)
II Valid Accounts (0/3)	Exploitation for Client Execution	II Create or Modify System Process (2/2)	II Event Triggered Execution (0/4)	II Hijack Execution Flow (0/2)	II Modify Authentication Process (0/1)	Network Sniffing	II OS Credential Dumping (0/0)	Data from Removable Media	II Encrypted Channel (0/2)	II Exfiltration Over Web Service (0/2)	II Endpoint Denial of Service (0/4)
	Native API	II Launch Agent	Exploitation for Privilege Escalation	II Impair Defenses (0/4)	II Network Sniffing	Permission Groups Discovery (0/2)	II Steal Web Session Cookie	II Data Staged (0/2)	Fallback Channels	Scheduled Transfer	Firmware Corruption
	II Scheduled Task/Job (0/2)	II Launch Daemon	II Hijack Execution Flow (0/2)	II Indicator Removal on Host (0/4)	II OS Credential Dumping (0/0)	Process Discovery	Two-Factor Authentication Interception	II Input Capture (0/3)	Ingress Tool Transfer		Inhibit System Recovery
	Software Deployment Tools	II Event Triggered Execution (0/4)	II Process Injection (0/0)	II Masquerading (0/5)	II Steal Web Session Cookie	Remote System Discovery	II Unsecured Credentials (0/3)	II Man-in-the-Middle (0/1)	Multi-Stage Channels		Network Denial of Service (0/2)
	II System Services (0/1)	II Hijack Execution Flow (0/2)	II Scheduled Task/Job (0/2)	II Modify Authentication Process (0/1)	II Two-Factor Authentication Interception	II Software Discovery (0/1)		Screen Capture	Non-Application Layer Protocol		Resource Hijacking
	II User Execution (0/2)	II Modify Authentication Process (0/1)	II Valid Accounts (0/3)	II Obfuscated Files or Information (0/5)	II Two-Factor Authentication Interception	System Information Discovery		Video Capture	Non-Standard Port		Service Stop
		II Scheduled Task/Job (0/2)		II Process Injection (0/0)	II Unsecured Credentials (0/3)	System Location Discovery			Protocol Tunneling		System Shutdown/Reboot
		II Server Software Component (0/1)		Rootkit		System Network Configuration Discovery (0/1)			II Proxy (0/4)		
		II Traffic Signaling (0/1)		II Subvert Trust Controls (0/4)		System Network Connections Discovery			Remote Access Software		
		II Valid Accounts (0/3)		Code Signing		System Owner/User Discovery			II Traffic Signaling (0/1)		
				Code Signing Policy Modification		II Virtualization/Sandbox Evasion (0/3)			II Web Service (0/3)		
				Gatekeeper Bypass							
				Install Root Certificate							
				II Traffic Signaling (0/1)							

OVERLAP BETWEEN VISIBILITY AND INTEL




Initial Access 7 techniques	Execution 7 techniques	Persistence 14 techniques	Privilege Escalation 10 techniques	Defense Evasion 18 techniques	Credential Access 12 techniques	Discovery 19 techniques	Lateral Movement 6 techniques	Collection 13 techniques	Command and Control 16 techniques	Exfiltration 8 techniques	Impact 13 techniques
Drive-by Compromise	II Command and Scripting Interpreter	II Account Manipulation (0/1)	II Abuse Elevation Control Mechanism (0/3)	II Abuse Elevation Control Mechanism (0/3)	II Brute Force (0/4)	II Account Discovery (1/2)	Exploitation of Remote Services	II Archive Collected Data	II Application Layer Protocol (3/4)	II Automated Exfiltration (0/0)	Account Access Removal
Exploit Public-Facing Application	II Exploitation for Client Execution	II Boot or Logon Autostart Execution (0/3)	II Boot or Logon Autostart Execution (0/3)	II Deobfuscate/Decode Files or Information	II Credentials from Password Stores (0/4)	II Application Window Discovery	II Internal Spearphishing	II Audio Capture	II Communication Through Removable Media	II Data Transfer Size Limits	Data Destruction
Hardware Additions	II Native API	II Boot or Logon Initialization Scripts (0/3)	II Boot or Logon Initialization Scripts (0/3)	II Execution Guardrails (0/1)	II Exploitation for Credential Access	II Browser Bookmark Discovery	II Lateral Tool Transfer	II Automated Collection	II Clipboard Data	II Exfiltration Over Alternative Protocol (1/3)	Data Encrypted for Impact
II Phishing (2/3)	II Scheduled Task/Job (0/2)	II Browser Extensions	II Create or Modify System Process (2/2)	II File and Directory Permissions Modification (1/1)	II Forge Web Credentials (0/1)	II File and Directory Discovery	II Remote Service Session Hijacking (0/1)	II Data from Information Repositories (0/0)	II Data Encoding (1/2)	II Data Manipulation (0/3)	II Data Manipulation (0/3)
II Supply Chain Compromise (0/3)	II Software Deployment Tools	II Compromise Client Software Binary	II Launch Agent	II Hide Artifacts (2/6)	II Input Capture (1/3)	II Network Service Scanning	II Remote Services (0/2)	II Data from Local System	II Data Obfuscation (0/3)	II Exfiltration Over C2 Channel	II Defacement (0/2)
Trusted Relationship	II System Services (0/1)	II Create Account (0/2)	II Launch Daemon	II Hijack Execution Flow (0/2)	II Man-in-the-Middle (0/1)	II Network Share Discovery	II Software Deployment Tools	II Data from Network Shared Drive	II Dynamic Resolution (0/3)	II Exfiltration Over Other Network Medium (0/1)	II Disk Wipe (0/2)
II Valid Accounts (1/3)	II User Execution (2/2)	II Create or Modify System Process (2/2)	II Event Triggered Execution (0/4)	II Impair Defenses (1/4)	II Modify Authentication Process (0/1)	II Network Sniffing		II Data from Removable Media	II Encrypted Channel (0/2)	II Exfiltration Over Physical Medium (0/1)	II Endpoint Denial of Service (0/4)
		II Event Triggered Execution (0/4)	II Hijack Execution Flow (0/2)	II Indicator Removal on Host	II Network Sniffing	II Permission Groups Discovery (0/2)		II Data Staged (0/2)	II Fallback Channels	II Exfiltration Over Web Service (0/2)	Firmware Corruption
		II Hijack Execution Flow (0/2)	II Hijack Execution Flow (0/2)	II Masquerading	II OS Credential Dumping (0/0)	II Process Discovery		II Input Capture (1/3)	II Ingress Tool Transfer	II Scheduled Transfer	Inhibit System Recovery
		II Modify Authentication Process (0/1)	II Process Injection (0/0)	II Modify Authentication Process (0/1)	II Steal Web Session Cookie	II Remote System Discovery		II Man-in-the-Middle (0/1)	II Multi-Stage Channels		II Network Denial of Service (0/2)
		II Scheduled Task/Job (0/2)	II Scheduled Task/Job (0/2)	II Obfuscated Files or Information	II Two-Factor Authentication Interception	II Software Discovery (0/1)		II Screen Capture	II Non-Application Layer Protocol		Resource Hijacking
		II Server Software Component (1/1)	II Valid Accounts (1/3)	II Process Injection (0/0)	II Unsecured Credentials (0/3)	II System Information Discovery		II Video Capture	II Non-Standard Port		Service Stop
		II Traffic Signaling (0/1)		II Rootkit		II System Location Discovery			II Protocol Tunneling		System Shutdown/Reboot
		II Valid Accounts (1/3)		II Subvert Trust Controls (1/4)		II System Network Configuration Discovery (0/1)			II Proxy (0/4)		
				II Code Signing		II System Network Connections Discovery			II Remote Access Software		
				II Code Signing Policy Modification		II System Owner/User Discovery			II Traffic Signaling (0/1)		
				II Gatekeeper Bypass		II Virtualization/Sandbox Evasion (1/3)			II Web Service (0/3)		
				II Install Root Certificate							
				II Traffic Signaling (0/1)							

 = CTI
 = Visible
 = Both



VALIDATED BY ADVERARY EMULATION

Initial Access 7 techniques	Execution 7 techniques	Persistence 14 techniques	Privilege Escalation 10 techniques	Defense Evasion 18 techniques	Credential Access 12 techniques	Discovery 19 techniques	Lateral Movement 6 techniques	Collection 13 techniques	Command and Control 16 techniques	Exfiltration 8 techniques	Impact 13 techniques
Drive-by Compromise	II Command and Scripting Interpreter	II Account Manipulation (0/1)	II Abuse Elevation Control Mechanism (0/3)	II Abuse Elevation Control Mechanism (0/3)	II Brute Force (0/4)	II Account Discovery (1/2)	Exploitation of Remote Services	II Archive Collected Data	II Application Layer Protocol (3/4)	II Automated Exfiltration (0/0)	Account Access Removal
Exploit Public-Facing Application	II Exploitation for Client Execution	II Boot or Logon Autostart Execution (0/3)	II Boot or Logon Autostart Execution (0/3)	II Deobfuscate/Decode Files or Information	II Credentials from Password Stores (0/4)	II Application Window Discovery	II Internal Spearphishing	II Audio Capture	II Communication Through Removable Media	II Data Transfer Size Limits	Data Destruction
Hardware Additions	II Native API	II Boot or Logon Initialization Scripts (0/3)	II Boot or Logon Initialization Scripts (0/3)	II Execution Guardrails (0/1)	II Exploitation for Credential Access	II Browser Bookmark Discovery	II Lateral Tool Transfer	II Automated Collection	II Data Encoding (1/2)	II Exfiltration Over Alternative Protocol (1/3)	Data Encrypted for Impact
II Phishing (2/3)	II Scheduled Task/Job (0/3)	II Browser Extensions	II Create or Modify System Process (0/3)	II Exploitation for Defense Evasion	II Forge Web Credentials (0/1)	II File and Directory Permissions Modification (1/1)	II Remote Service Session Hijacking (0/1)	II Data from Information Repositories (0/0)	II Data Obfuscation (0/3)	II Exfiltration Over C2 Channel	II Data Manipulation (0/3)
II Supply Chain Compromise (0/3)	II Software Deployment Tools (0/3)	II Compromise Client Software Binary	II Launch Agent (0/3)	II File and Directory Permissions Modification (1/1)	II Input Capture (1/3)	II Network Service Scanning	II Remote Services (0/2)	II Data from Local System	II Dynamic Resolution (0/3)	II Exfiltration Over Other Network Medium (0/1)	II Defacement (0/2)
Trusted Relationship	II System Services (0/1)	II Create Account (0/2)	II Launch Daemon (0/2)	II Hide Artifacts (2/6)	II Man-in-the-Middle (0/1)	II Network Share Discovery	II Software Deployment Tools (0/2)	II Data from Network Shared Drive	II Encrypted Channel (0/2)	II Exfiltration Over Physical Medium (0/1)	II Disk Wipe (0/2)
II Valid Accounts (1/3)	II User Execution (2/2)	II Create or Modify System Process (2/2)	II Event Triggered Execution (0/4)	II Hijack Execution Flow (0/2)	II Modify Authentication Process (0/1)	II Network Sniffing	II Permission Groups Discovery (0/2)	II Data from Removable Media	II Fallback Channels	II Exfiltration Over Web Service (0/2)	II Endpoint Denial of Service (0/4)
		II Event Triggered Execution (0/4)	II Hijack Execution Flow (0/2)	II Impair Defenses (1/4)	II Network Sniffing	II Password Policy Discovery	II OS Credential Dumping (0/0)	II Data Staged (0/2)	II Ingress Tool Transfer	II Scheduled Transfer	Firmware Corruption
		II Hijack Execution Flow (0/2)	II Process Injection (0/0)	II Indicator Removal on Host	II Steal Web Session Cookie	II Peripheral Device Discovery	II Process Discovery	II Input Capture (1/3)	II Multi-Stage Channels		Inhibit System Recovery
		II Modify Authentication Process (0/1)	II Scheduled Task/Job (0/2)	II Masquerading	II Two-Factor Authentication Interception	II Software Discovery (0/1)	II Remote System Discovery	II Man-in-the-Middle (0/1)	II Non-Application Layer Protocol		II Network Denial of Service (0/2)
		II Scheduled Task/Job (0/2)	II Valid Accounts (1/3)	II Modify Authentication Process (0/1)	II Unsecured Credentials (0/3)	II System Information Discovery	II Software Discovery (0/1)	II Screen Capture	II Non-Standard Port		Resource Hijacking
		II Server Software Component (1/1)		II Obfuscated Files or Information		II System Location Discovery	II System Network Configuration Discovery (0/1)	II Video Capture	II Protocol Tunneling		Service Stop
		II Traffic Signaling (0/1)		II Process Injection (0/0)		II System Network Connections Discovery	II System Owner/User Discovery		II Proxy (0/4)		System Shutdown/Reboot
		II Valid Accounts (1/3)		II Rootkit		II System Owner/User Discovery	II Virtualization/Sandbox Evasion (1/3)		II Remote Access Software		
				II Subvert Trust Controls (1/4)					II Traffic Signaling (0/1)		
				II Code Signing					II Web Service (0/3)		
				II Code Signing Policy Modification							
				II Gatekeeper Bypass							
				II Install Root Certificate							
				II Traffic Signaling (0/1)							

 = CTI
 = Visible
 = Both

MITRE

©2021 The MITRE Corporation. ALL RIGHTS RESERVED. Approved for public release. Distribution unlimited 21-00706-16



= Validated via Adv Emu



@coolestcatiknow
@_whatshisface

DECIDE WHERE TO IMPROVE

Initial Access 7 techniques	Execution 7 techniques	Persistence 14 techniques	Privilege Escalation 10 techniques	Defense Evasion 18 techniques	Credential Access 12 techniques	Discovery 19 techniques	Lateral Movement 6 techniques	Collection 13 techniques	Command and Control 16 techniques	Exfiltration 8 techniques	Impact 13 techniques
Drive-by Compromise	Command and Scripting Interpreter	Account Manipulation (0/1)	Abuse Elevation Control Mechanism (0/3)	Abuse Elevation Control Mechanism (0/3)	Brute Force (0/4)	Account Discovery	Exploitation of Remote Services	Archive Collected Data	Application Layer Protocol (3/4)	Automated Exfiltration (0/0)	Account Access Removal
Exploit Public-Facing Application	Exploitation of Client Execution	Boot or Logon Autostart Execution (0/3)	Boot or Logon Autostart Execution (0/3)	Deobfuscate/Decode Files or Information	Credentials from Password Stores (0/4)	Application Window Discovery	Internal Spearphishing	Audio Capture	Communication Through Removable Media	Data Transfer Size Limits	Data Destruction
Hardware Additions	Native API	Boot or Logon Initialization	Boot or Logon	Execution Guardrails (0/1)	Exploitation for Credential Access	Browser Bookmark Discovery	Lateral Tool Transfer	Automated Collection	Data Encoding (1/2)	Exfiltration Over Alternative Protocol (1/3)	Data Encrypted for Impact
Phishing (2/3)				Exploitation for Defense		File and Directory Discovery	Remote Service Hijacking (0/1)	Clipboard Data	Data Obfuscation (0/3)	Exfiltration Over Channel	Data Manipulation (0/3)
						Network Service Scanning	Remote Services (0/2)	Data from Information Repositories (0/0)	Dynamic Resolution (0/3)	Exfiltration Over Other Network Medium (0/1)	Defacement (0/2)
						Network Share Discovery	Software Deployment Tools	Data from Local System	Encrypted Channel (0/2)	Exfiltration Over Physical	Disk Wipe (0/2)
						Network Sniffing		Data from Network Shared Drive	Fallback Channels		Endpoint Denial of Service (0/4)
						Password Policy Discovery					Firmware Corruption

Command and Scripting Interpreter, OS Credential Dumping, and Gatekeeper Bypass are used by a threat actor we've prioritized.

Focus on Data Encrypted for Impact and Exploitation of Remote Services as they can have significant impact to operations

Existing logs can be used to detect File and Directory Discovery and Browser Extensions making analytic development easier.

Launch Agent, Launch Daemon, and Launchctl are popular techniques and improvement can give a big return on investment.

■ = CTI
 ■ = Visible
 ■ = Both

BE INTENTIONAL ABOUT IMPROVEMENTS

- **Think about the best way to mitigate each gap**
 - Maybe it's a new detection or data source
 - Maybe it's a mitigation, new group policy, or new user training
 - Maybe the gap shouldn't be closed, and risk should be accepted
- **Validate any changes using adversary emulation**

BE INTENTIONAL ABOUT IMPROVEMENTS

Initial Access 7 techniques	Execution 7 techniques	Persistence 14 techniques	Privilege Escalation 10 techniques	Defense Evasion 18 techniques	Credential Access 12 techniques	Discovery 19 techniques	Lateral Movement 6 techniques	Collection 13 techniques	Command and Control 16 techniques	Exfiltration 8 techniques	Impact 13 techniques
Drive-by Compromise	Command and Scripting Interpreter	Account Manipulation (0/1)	Abuse Elevation Control	Abuse Elevation Control Mechanism	Brute Force (0/4)	Account Discovery (1/2)	Exploitation of Remote Services	Archive Collected Data	Application Layer Protocol (3/4)	Automated Exfiltration (0/0)	Account Access Removal
Exploit Public-Facing Application	Exploitation for Client Execution	Boot or Logon Autostart Execution (0/3)	Local or Remote Administration	File and Directory Permissions Modification (1/1)	Forge Web Credentials (0/1)	Network Service Scanning	Remote Service Session Hijacking (0/1)	Data from Information Repositories (0/0)	Communication High Removable (0/1)	Data Transfer Size Limits	Data Destruction
Hardware Additions	Native API	Boot or Logon Initialization Scripts (0/3)	Local or Remote Administration	Hide Artifacts (0/2)	Input Capture (1/3)	Network Share Discovery	Remote Services (0/2)	Data from Local System	Encoding (1/2)	Exfiltration Over Alternative Protocol (1/3)	Data Encrypted for Impact
Phishing (2/3)	Scheduled Task/Job (0/2)	Browser Extensions	Local or Remote Administration	Impair Defenses (1/4)	Man-in-the-Middle (0/1)	Network Sniffing	Software Deployment Tools	Data from Network Shared Drive	Dynamic Resolution (0/3)	Exfiltration Over C2 Channel	Data Manipulation (0/3)
Compromise Valid Accounts (1/3)	Software Deployment Tools	Compromise Client Software Binary	Local or Remote Administration	Indicator Removal on Host (0/6)	Modify Authentication Process (0/1)	Password Policy Discovery		Data from Removable Media	Encrypted Channel (0/2)	Exfiltration Over Other Network Medium (0/1)	Defacement (0/2)
	System Services (0/1)	Create Account (0/2)	Local or Remote Administration	Masquerading (0/1)	OS Credential Dumping (0/0)	Peripheral Device Discovery		Data Staged (0/2)	Fallback Channels	Exfiltration Over Physical Medium (0/1)	Disk Wipe (0/2)
	User Execution (0/2)	Create or Modify System Process (0/2)	Local or Remote Administration	Modify Authentication Process (0/1)	Steal Web Session Cookie (0/0)	Permission Groups Discovery (0/2)		Input Capture (0/1)	Ingress Tool Transfer	Exfiltration Over Service (0/2)	Endpoint Denial of Service (0/1)
		Event Triggered Execution (0/4)	Local or Remote Administration	Process (0/1)		Process Discovery (0/1)		Man-in-the-Middle (0/1)	Multi-Stage Channels	Scheduled Transfer	Firmware Corruption (0/0)
		Hijack Execution Flow (0/2)	Local or Remote Administration	Process (0/1)		Remote System Discovery (0/1)		Screen Capture (0/1)	Non-Application Layer Protocol		Inhibit System Recovery (0/0)
			Local or Remote Administration	Process (0/1)		Software Discovery (0/1)		Video Capture (0/1)	Non-Standard Port		Network Denial of Service (0/2)
			Local or Remote Administration	Process (0/1)					Protocol Tunneling		Resource Hijacking (0/0)
			Local or Remote Administration	Process (0/1)					Proxy (0/4)		Service Stop (0/0)
			Local or Remote Administration	Process (0/1)					Remote Access Software		System Shutdown/Reboot (0/0)
			Local or Remote Administration	Process (0/1)					Traffic Signaling (0/1)		
			Local or Remote Administration	Process (0/1)					Web Service (0/3)		

None of our existing tools have visibility into Hijack Execution Flow so we'll need to obtain something new

We'll tackle Phishing and User Execution with new user training

Firmware Corruption are beyond our capability and resources to stop or detect, so we'll accept the risk

TAKEAWAYS

- A common language for conversations between teams
- A stewarded community driven resource
- A relevant resource for macOS
- A place to start



HELPFUL RESOURCES

- Medium Blogs (mitre-attack)
- David Bianco's ThreatHunting.net
- @Cyb3rWard0g's Open Threat Research Forge (OTRF)
- Katie Nickels Getting Started with ATT&CK & Cyber Threat Intelligence Self Study Plan



Cat Self

@coolestcatiknow

Adam Pennington

@_whatshisface

attack@mitre.org

