ATT&CKing the Status Quo: Threat-Based Adversary Emulation with MITRE ATT&CK™

Katie Nickels

Cody Thomas

SANS Threat Hunting & Incident Response Summit

September 6, 2018

MITRE

How *we* define threat hunting

“Human act of looking for badness that is not yet detected successfully.”

-Sergio Caltagirone

Problem: I need a threat to hunt for!

Solution: Create one by emulating real adversaries.

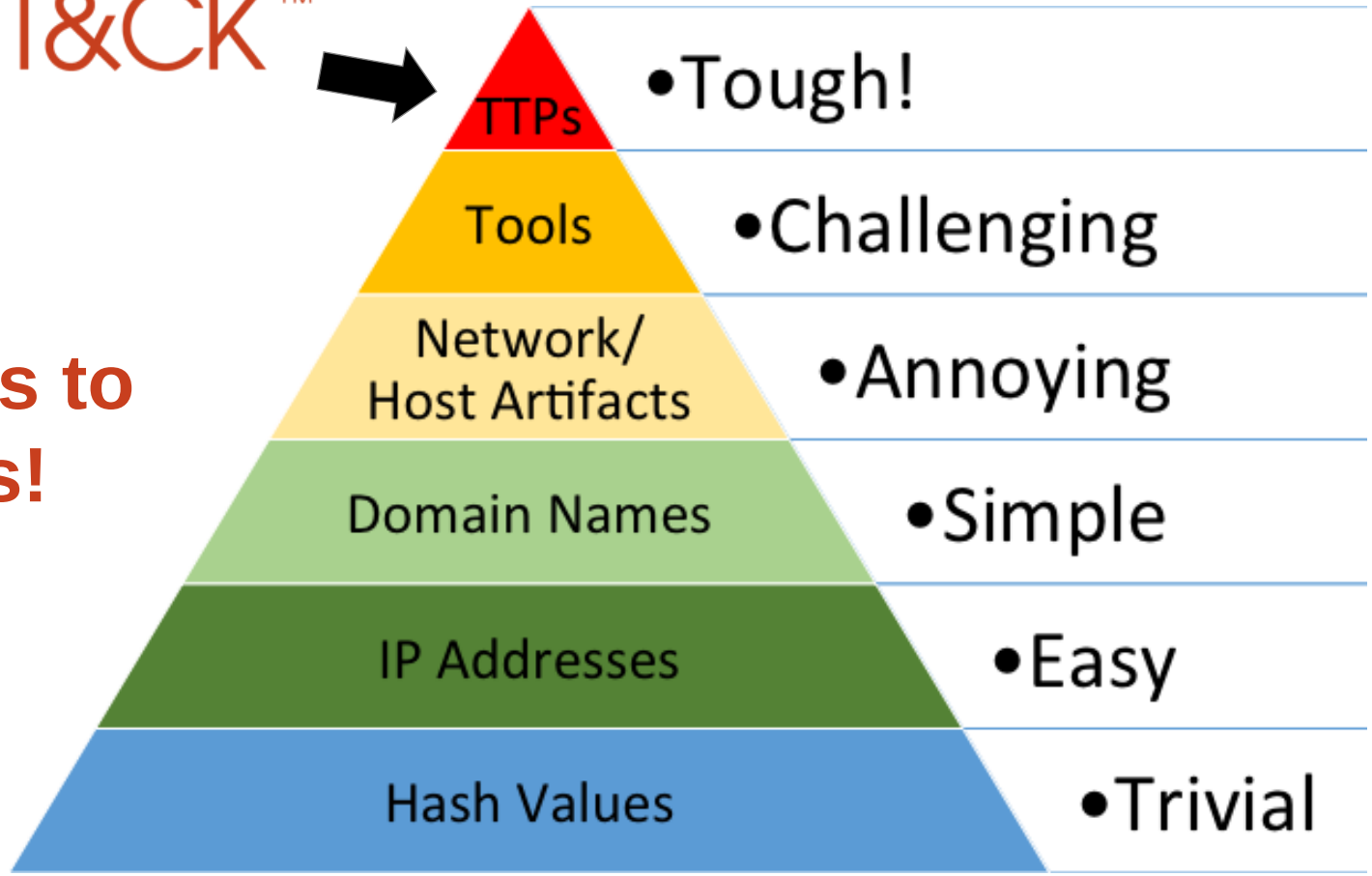
Tough questions for defenders

- How do I organize threat hunting?
- How do I know that my hunting techniques will work?
- Do I have a chance at detecting APT28?
- Is the data I'm collecting useful?
- Do I have overlapping tool coverage?
- Will this *shiny new* product from vendor XYZ help my organization's defenses?

The difficult task of detecting TTPs

ATT&CK™

Also applies to red teamers!



Source: David Bianco, https://detect-respond.blogspot.com/2013/03/the-pyramid-of-pain.html

David Bianco's Pyramid of Pain

MITRE

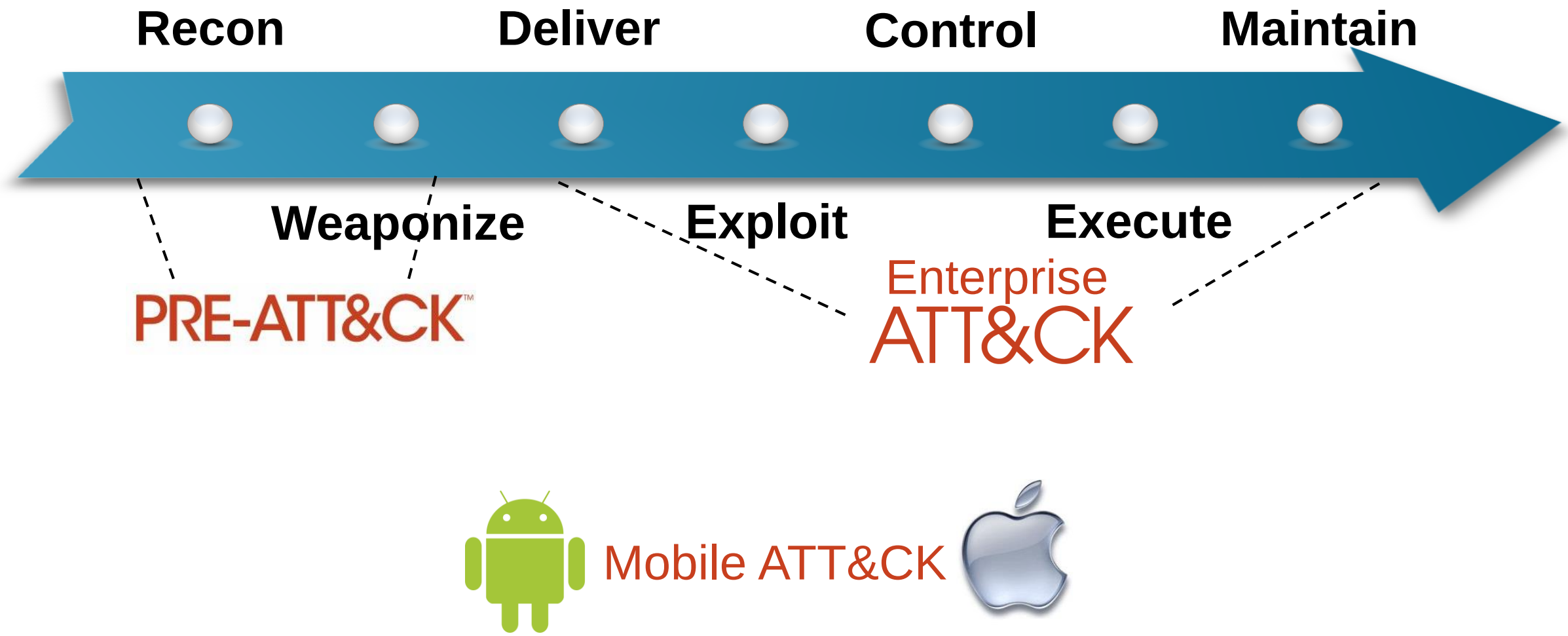
What is ATT&CK?

A knowledge base of adversary behavior

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

MITRE

Zooming in on the Adversary Lifecycle



MITRE

What is ATT&CK, really?

Tactics: the adversary's technical goals

Techniques: how the goals are achieved

Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection	Exfiltration	Command & Control
Hardware Additions	Scheduled Task			Binary Padding	Credentials in Registry	Browser Bookmark Discovery	Exploitation of Remote Services	Data from Information Repositories	Exfiltration Over Physical Medium	Remote Access Tools
Trusted Relationship	LSASS Driver		Extra Window Memory Injection		Exploitation for Credential Access			Video Capture		Port Knocking
Supply Chain Compromise	Local Job Scheduling		Access Token Manipulation			Network Share	Distributed Component		Exfiltration Over Command and Control Channel	Multi-hop Proxy
	Trap									Domain Fronting
Spearphishing Attachment	Launchctl									Data Encoding
	Signed Binary Proxy Execution								Data Encrypted	Remote File Copy
Exploit Public-Facing Application	User Execution								Automated Exfiltration	Multi-Stage Channels
Replication Through Removable Media	Exploitation for Client Execution								Exfiltration Over Other Network Medium	Web Service
										Standard Non-Application Layer Protocol
Spearphishing via Service	CMSTP								Exfiltration Over Alternative Protocol	
	Dynamic Data Exchange									Connection Proxy
Spearphishing Link	Mshta								Data Transfer Size Limits	Multilayer Encryption
Drive-by Compromise	AppleScript								Data Compressed	Standard Application Layer Protocol
Valid Accounts	Source								Scheduled Transfer	
	Space after Filename									Commonly Used Port
	Execution through Module Load									Standard Cryptographic Protocol
	Regsvcs/Regasm									Custom Cryptographic Protocol
	InstallUtil									Data Obfuscation
	Regsvr32									Custom Command and Control Protocol
	Execution through API									Communication Through Removable Media
	PowerShell									Multiband Communication
	Rundll32	Kernel Modules and Extensions								Fallback Channels
	Third-party Software									Uncommonly Used Port
	Scripting	Port Knocking								
	Graphical User Interface	SIP and Trust Provider Hijacking								
	Command-Line Interface									
		Screensaver	Exploitation for Privilege Escalation	Hidden Window		Scanning				

Main page
Help
Contribute
References
Data Drilldown
Using the API

Tactics
Initial Access
Execution
Persistence
Privilege Escalation
Defense Evasion
Credential Access
Discovery
Lateral Movement
Collection
Exfiltration
Command and Control

Techniques
Technique Matrix
All Techniques
Windows
Linux
macOS
Add a Technique

Groups
All Groups
Add a Group

Software
All Software

Scheduled Task

Utilities such as at and schtasks, along with the Windows Task Scheduler, can be used to schedule programs or scripts to be executed at a date and time. A task can also be scheduled on a remote system, provided the proper authentication is met to use RPC and file and printer sharing is turned on. Scheduling a task on a remote system typically required being a member of the Administrators group on the the remote system.[1]

An adversary may use task scheduling to execute programs at system startup or on a scheduled basis for persistence, to conduct remote Execution as part of Lateral Movement, to gain SYSTEM privileges, or to run a process under the context of a specified account.

Contents [hide]
1 Examples
2 Mitigation

Scheduled Task
Technique

ID	T1053
Tactic	Execution, Persistence, Privilege Escalation
Platform	Windows
Permissions Required	User, Administrator, SYSTEM
Effective Permissions	User, Administrator, SYSTEM
Data Sources	File monitoring, Process command-line parameters, Process monitoring, Windows event logs

Leo Loobeek, @leoloobeek, Alain Homewood, Insomnia Security

Procedures – Specific technique implementation

Examples

- APT18 actors used the native at Windows task scheduler tool to use scheduled tasks for execution on a victim network.[2]
- APT29 used named and hijacked scheduled tasks to establish persistence.[3]
- An APT3 downloader creates persistence by creating the following scheduled task: `schtasks /create /tn "mysc" /tr C:\Users\Public\test.exe /sc ONLOGON /ru "System"`.[4]
- APT32 has used scheduled tasks to persist on victim systems.[5]
- BRONZE BUTLER has used at and schtasks to register a scheduled task to execute malware during lateral movement.[6]
- Dragonfly 2.0 used scheduled tasks to automatically log out of created accounts every 8 hours as well as to execute tools to

MITRE

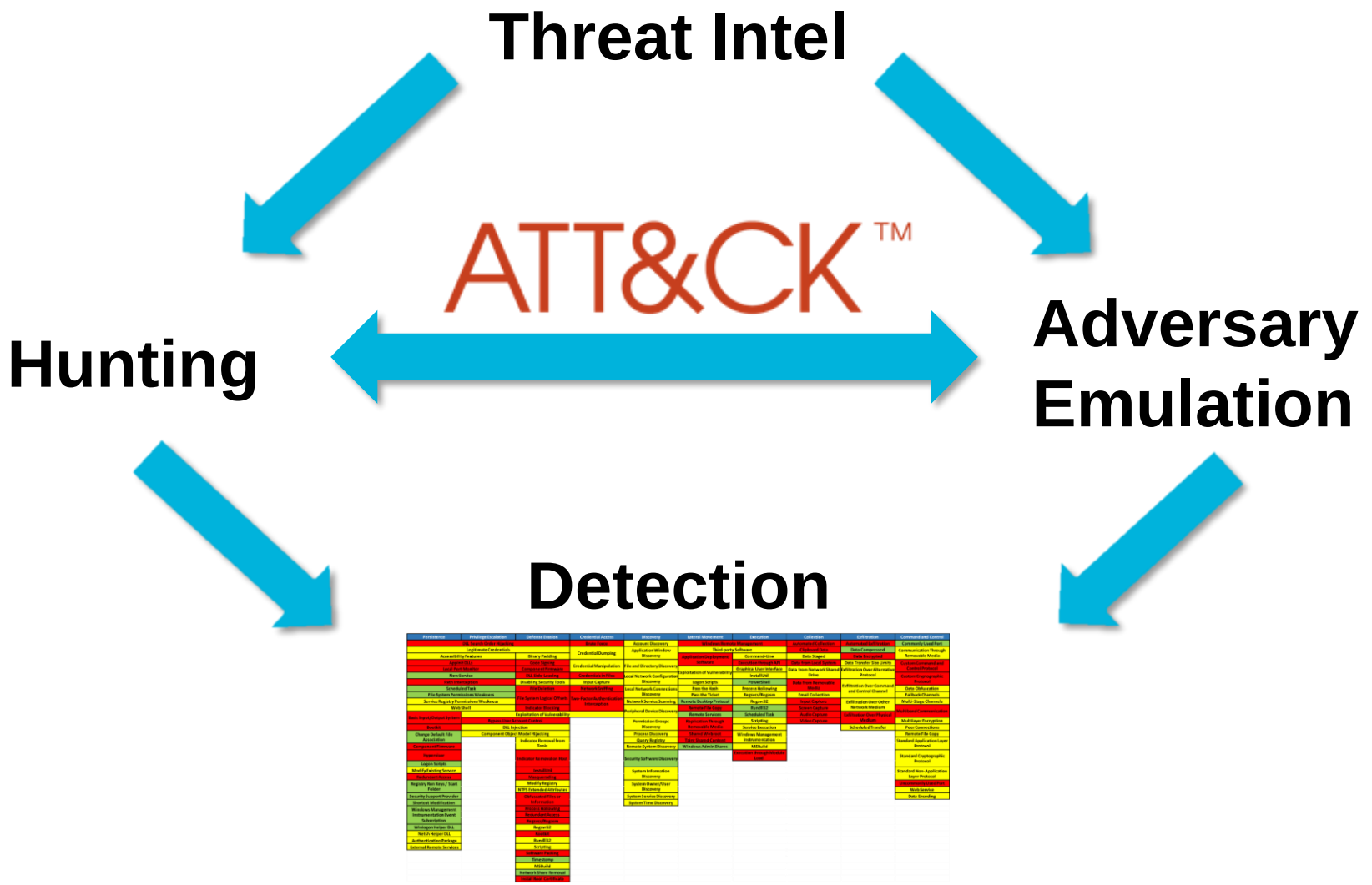
Example Technique: New Service

Description:	When operating systems boot up, they can start programs or applications called services that perform background system functions. […] Adversaries may install a new service which will be executed at startup by directly modifying the registry or by using tools. [1]
Platform:	Windows
Permissions required:	Administrator, SYSTEM
Effective permissions:	SYSTEM
Detection:	• Monitor service creation through changes in the Registry and common utilities using command-line invocation • …
Mitigation:	• Limit privileges of user accounts and remediate Privilege Escalation vectors • …
Data sources:	Windows registry, process monitoring, command-line parameters
Examples:	Carbanak, Lazarus Group, TinyZBot, Duqu, CozyCar, CosmicDuke, hcdLoader, …
References:	1. Microsoft. (n.d.). Services. Retrieved June 7, 2016.

Example Group: APT28

Description:	**APT28** is a threat group that has been attributed to the Russian government.[1 2 3 4] This group reportedly compromised the Democratic National Committee in April 2016.[5]	
Aliases:	Sednit, Sofacy, Pawn Storm, Fancy Bear, STRONTIUM, Tsar Team, Threat Group-4127, TG-4127 [1 2 3 4 5 6 7]	
Techniques:	• Data Obfuscation [1] • Connection Proxy [1 8] • Standard Application Layer Protocol [1] • Remote File Copy [8 9] • Rundll32 [8 9]	• Indicator Removal on Host [5] • Timestomp [5] • Credential Dumping [10] • Screen Capture [10 11] • Bootkit [7] *and more…*
Software:	CHOPSTICK, JHUHUGIT, ADVSTORESHELL, XTunnel, Mimikatz, HIDEDRV, USBStealer, CORESHELL, OLDBAIT, XAgentOSX, Komplex, Responder, Forfiles, Winexe, certutil [1 3 6]	
References:	1. FireEye. (2015). APT28: A WINDOW INTO RUSSIA'S CYBER ESPIONAGE OPERATIONS?. Retrieved August 19, 2015. …	

How to use it: threat-informed defense, but for real



MITRE

What is adversary emulation?

- **AKA: Threat-based red teaming**
- **Adversary emulation**
  - Emulate the techniques of an adversary that's most likely to target your environment
  - Focus on the behaviors of those techniques instead of specific implementations

https://giphy.com/explore/hackerman

https://tenor.com/view/hackerman-transformation-kung-fury-kung-fury-gif-7263543

MITRE

Step 1: Choose an adversary and gather threat intel

- **Identify the adversary you want to emulate**
  - Consider who's targeting you and gaps you're trying to assess
- **Gather data about that adversary**
  - Look for post-exploit information
  - Consider their tools, aliases, and campaigns
  - Think about the time frame

MITRE

Choosing an adversary based on gaps

Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection	Exfiltration	Command And Control
Drive-by Compromise	AppleScript	.bash_profile and .bashrc	Access Token Manipulation	Access Token Manipulation	Account Manipulation	Account Discovery	AppleScript	Audio Capture	Automated Exfiltration	Commonly Used Port
Exploit Public-Facing Application	CMSTP	Accessibility Features	Accessibility Features	Binary Padding	Bash History	Application Window Discovery	Application Deployment Software	Automated Collection	Data Compressed	Communication Through Removable Media
Hardware Additions	Command-Line Interface	AppCert DLLs	AppCert DLLs	BITS Jobs	Brute Force	Browser Bookmark Discover	Distributed Component Object Model	Clipboard Data	Data Encrypted	Connection Proxy
Replication Through Removable Media	Control Panel Items	AppInit DLLs	AppInit DLLs	Bypass User Account Contr	Credential Dumping	File and Directory Discovery	Exploitation of Remote Services	Data from Information Repositories	Data Transfer Size Limits	Custom Command and Control Protocol
Spearphishing Attachment	Dynamic Data Exchange	Application Shimming	Application Shimming	Clear Command History	Credentials in Files	Network Service Scanning	Logon Scripts	Data from Local System	Exfiltration Over Alternative Protocol	Custom Cryptographic Protocol
Spearphishing Link	Execution through API	Authentication Package	Bypass User Account Contr	CMSTP	Credentials in Registry	Network Share Discovery	Pass the Hash	Data from Network Shared Drive	Exfiltration Over Command and Control Channel	Data Encoding
Spearphishing via Service	Execution through Module Load	BITS Jobs	DLL Search Order Hijacking	Code Signing	Exploitation for Credential Access	Password Policy Discovery	Pass the Ticket	Data from Removable Media	Exfiltration Over Other Network Medium	Data Obfuscation
Supply Chain Compromise	Exploitation for Client Execution	Bootkit	Dylib Hijacking	Component Firmware	Forced Authentication	Peripheral Device Discovery	Remote Desktop Protocol	Data Staged	Exfiltration Over Physical Medium	Domain Fronting
Trusted Relationship	Graphical User Interface	Browser Extensions	Exploitation for Privilege Escalation	Component Object Model Hijacking	Hooking	Permission Groups Discove	Remote File Copy	Email Collection	Scheduled Transfer	Fallback Channels
Valid Accounts	InstallUtil	Change Default File Association	Extra Window Memory Injection	Control Panel Items	Input Capture	Process Discovery	Remote Services	Input Capture		Multi-hop Proxy
	Launchctl	Component Firmware	File System Permissions Weakness	DCShadow	Input Prompt	Query Registry	Replication Through Removable Media	Man in the Browser		Multi-Stage Channels
	Local Job Scheduling	Component Object Model Hijacking	Hooking	Deobfuscate/Decode Files o Information	Kerberoasting	Remote System Discovery	Shared Webroot	Screen Capture		Multiband Communication
	LSASS Driver	Create Account	Image File Execution Option Injection	Disabling Security Tools	Keychain	Security Software Discovery	SSH Hijacking	Video Capture		Multilayer Encryption
	Mshta	DLL Search Order Hijacking	Launch Daemon	DLL Search Order Hijacking	LLMNR/NBT-NS Poisoning	System Information Discove	Taint Shared Content			Port Knocking
	PowerShell	Dylib Hijacking	New Service	DLL Side-Loading	Network Sniffing	System Network Configuration Discovery	Third-party Software			Remote Access Tools
	Regsvcs/Regasm	External Remote Services	Path Interception	Exploitation for Defense Evasion	Password Filter DLL	System Network Connection Discovery	Windows Admin Shares			Remote File Copy
	Regsvr32	File System Permissions Weakness	Plist Modification	Extra Window Memory Injection	Private Keys	System Owner/User Discovery	Windows Remote Management			Standard Application Layer Protocol
	Rundll32	Hidden Files and Directories	Port Monitors	File Deletion	Replication Through Removable Media	System Service Discovery				Standard Cryptographic Protocol
	Scheduled Task	Hooking	Process Injection	File System Logical Offsets	Securityd Memory	System Time Discovery				Standard Non-Application Layer Protocol
	Scripting	Hypervisor	Scheduled Task	Gatekeeper Bypass	Two-Factor Authentication Interception					Uncommonly Used Port
	Service Execution	Image File Execution Option Injection	Service Registry Permissior Weakness	Hidden Files and Directories						Web Service
	Signed Binary Proxy Execution	Kernel Modules and Extensions	Setuid and Setgid	Hidden Users						
	Signed Script Proxy Execution	Launch Agent	SID-History Injection	Hidden Window						
	Source	Launch Daemon	Startup Items	HISTCONTROL						
	Space after Filename	Launchctl	Sudo	Image File Execution Options Injection						
	Third-party Software	LC_LOAD_DYLIB Addition	Sudo Caching	Indicator Blocking						
	Trap	Local Job Scheduling	Valid Accounts	Indicator Removal from Tools						
	Trusted Developer Utilities	Login Item	Web Shell	Indicator Removal on Host						
	User Execution	Logon Scripts		Indirect Command Execution						
	Windows Management Instrumentation	LSASS Driver		Install Root Certificate						
	Windows Remote Management	Modify Existing Service		InstallUtil						
		Netsh Helper DLL		Launchctl						
		New Service		LC_MAIN Hijacking						
		Office Application Startup		Masquerading						

Notional gaps in defenses

MITRE

Choosing an adversary based on gaps

Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection	Exfiltration	Command And Control
Drive-by Compromise	AppleScript	.bash_profile and .bashrc	Access Token Manipulation	Access Token Manipulation	Account Manipulation	Account Discovery	AppleScript	Audio Capture	Automated Exfiltration	Commonly Used Port
Exploit Public-Facing Application	CMSTP	Accessibility Features	Accessibility Features	Binary Padding	Bash History	Application Window Discovery	Application Deployment Software	Automated Collection	Data Compressed	Communication Through Removable Media
Hardware Additions	Command-Line Interface	AppCert DLLs	AppCert DLLs	BITS Jobs	Brute Force	Browser Bookmark Discovery	Distributed Component Object Model	Clipboard Data	Data Encrypted	Connection Proxy
Replication Through Removable Media	Control Panel Items	AppInit DLLs	AppInit DLLs	Bypass User Account Control	Credential Dumping	File and Directory Discovery	Exploitation of Remote Services	Data from Information Repositories	Data Transfer Size Limits	Custom Command and Control Protocol
Spearphishing Attachment	Dynamic Data Exchange	Application Shimming	Application Shimming	Clear Command History	Credentials in Files	Network Service Scanning	Logon Scripts	Data from Local System	Exfiltration Over Alternative Protocol	Custom Cryptographic Protocol
Spearphishing Link	Execution through API	Authentication Package	Bypass User Account Control	CMSTP	Credentials in Registry	Network Share Discovery	Pass the Hash	Data from Network Shared Drive	Exfiltration Over Command and Control Channel	Data Encoding
Spearphishing via Service	Execution through Module Load	BITS Jobs	DLL Search Order Hijacking	Code Signing	Exploitation for Credential Access	Password Policy Discovery	Pass the Ticket	Data from Removable Media	Exfiltration Over Other Network Medium	Data Obfuscation
Supply Chain Compromise	Exploitation for Client Execution	Bootkit	Dylib Hijacking	Component Firmware	Forced Authentication	Peripheral Device Discovery	Remote Desktop Protocol	Data Staged	Exfiltration Over Physical Medium	Domain Fronting
Trusted Relationship	Graphical User Interface	Browser Extensions	Exploitation for Privilege Escalation	Component Object Model Hijacking	Hooking	Permission Groups Discovery	Remote File Copy	Email Collection	Scheduled Transfer	Fallback Channels
Valid Accounts	InstallUtil	Change Default File Association	Extra Window Memory Injection	Control Panel Items	Input Capture	Process Discovery	Remote Services	Input Capture		Multi-hop Proxy
	Launchctl	Component Firmware	File System Permissions Weakness	DCShadow	Input Prompt	Query Registry	Replication Through Removable Media	Man in the Browser		Multi-Stage Channels
	Local Job Scheduling	Component Object Model Hijacking	Hooking	Deobfuscate/Decode Files or Information	Kerberoasting	Remote System Discovery	Shared Webroot	Screen Capture		Multiband Communication
	LSASS Driver	Create Account	Image File Execution Options Injection	Disabling Security Tools	Keychain	Security Software Discovery	SSH Hijacking	Video Capture		Multilayer Encryption
	Mshta	DLL Search Order Hijacking	Launch Daemon	DLL Search Order Hijacking	LLMNR/NBT-NS Poisoning	System Information Discovery	Taint Shared Content			Port Knocking
	PowerShell	Dylib Hijacking	New Service	DLL Side-Loading	Network Sniffing	System Network Configuration Discovery	Third-party Software			Remote Access Tools
	Regsvcs/Regasm	External Remote Services	Path Interception	Exploitation for Defense Evasion	Password Filter DLL	System Network Connections Discovery	Windows Admin Shares			Remote File Copy
	Regsvr32	File System Permissions Weakness	Plist Modification	Extra Window Memory Injection	Private Keys	System Owner/User Discovery	Windows Remote Management			Standard Application Layer Protocol
	Rundll32	Hidden Files and Directories	Port Monitors	File Deletion	Replication Through Removable Media	System Service Discovery				Standard Cryptographic Protocol
	Scheduled Task	Hooking	Process Injection	File System Logical Offsets	Securityd Memory	System Time Discovery				Standard Non-Application Layer Protocol
	Scripting	Hypervisor	Scheduled Task	Gatekeeper Bypass	Two-Factor Authentication Interception					Uncommonly Used Port
	Service Execution	Image File Execution Options Injection	Service Registry Permissions Weakness	Hidden Files and Directories						Web Service
	Signed Binary Proxy Execution	Kernel Modules and Extensions	Setuid and Setgid	Hidden Users						
	Signed Script Proxy Execution	Launch Agent	SID-History Injection	Hidden Window						
	Source	Launch Daemon	Startup Items	HISTCONTROL						
	Space after Filename	Launchctl	Sudo	Image File Execution Options Injection						
	Third-party Software	LC_LOAD_DYLIB Addition	Sudo Caching	Indicator Blocking						
	Trap	Local Job Scheduling	Valid Accounts	Indicator Removal from Tools						
	Trusted Developer Utilities	Login Item	Web Shell	Indicator Removal on Host						
	User Execution	Logon Scripts		Indirect Command Execution						
	Windows Management Instrumentation	LSASS Driver		Install Root Certificate						
	Windows Remote Management	Modify Existing Service		InstallUtil						
		Netsh Helper DLL		Launchctl						
		New Service		LC_MAIN Hijacking						
		Office Application Startup		Masquerading						

APT29 techniques
(based only on open source reporting)

MITRE

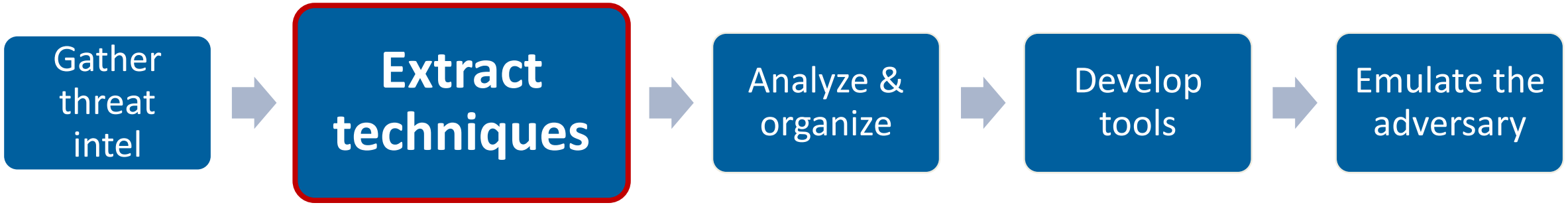
Choosing an adversary based on gaps

Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection	Exfiltration	Command And Control
Drive-by Compromise	AppleScript	.bash_profile and .bashrc	Access Token Manipulation	Access Token Manipulation	Account Manipulation	Account Discovery	AppleScript	Audio Capture	Automated Exfiltration	Commonly Used Port
Exploit Public-Facing Application	CMSTP	Accessibility Features	Accessibility Features	Binary Padding	Bash History	Application Window Discovery	Application Deployment Software	Automated Collection	Data Compressed	Communication Through Removable Media
Hardware Additions	Command-Line Interface	AppCert DLLs	AppCert DLLs	BITS Jobs	Brute Force	Browser Bookmark Discovery	Distributed Component Object Model	Clipboard Data	Data Encrypted	Connection Proxy
Replication Through Removable Media	Control Panel Items	AppInit DLLs	AppInit DLLs	Bypass User Account Control	Credential Dumping	File and Directory Discovery	Exploitation of Remote Services	Data from Information Repositories	Data Transfer Size Limits	Custom Command and Control Protocol
Spearphishing Attachment	Dynamic Data Exchange	Application Shimming	Application Shimming	Clear Command History	Credentials in Files	Network Service Scanning	Logon Scripts	Data from Local System	Exfiltration Over Alternative Protocol	Custom Cryptographic Protocol
Spearphishing Link	Execution through API	Authentication Package	Bypass User Account Control	CMSTP	Credentials in Registry	Network Share Discovery	Pass the Hash	Data from Network Shared Drive	Exfiltration Over Command and Control Channel	Data Encoding
Spearphishing via Service	Execution through Module Load	BITS Jobs	DLL Search Order Hijacking	Code Signing	Exploitation for Credential Access	Password Policy Discovery	Pass the Ticket	Data from Removable Media	Exfiltration Over Other Network Medium	Data Obfuscation
Supply Chain Compromise	Exploitation for Client Execution	Bootkit	Dylib Hijacking	Component Firmware	Forced Authentication	Peripheral Device Discovery	Remote Desktop Protocol	Data Staged	Exfiltration Over Physical Medium	Domain Fronting
Trusted Relationship	Graphical User Interface	Browser Extensions	Exploitation for Privilege Escalation	Component Object Model Hijacking	Hooking	Permission Groups Discovery	Remote File Copy	Email Collection	Scheduled Transfer	Fallback Channels
Valid Accounts	InstallUtil	Change Default File Association	Extra Window Memory Injection	Control Panel Items	Input Capture	Process Discovery	Remote Services	Input Capture		Multi-hop Proxy
	Launchctl	Component Firmware	File System Permissions Weakness	DCShadow	Input Prompt	Query Registry	Replication Through Removable Media	Man in the Browser		Multi-Stage Channels
	Local Job Scheduling	Component Object Model Hijacking	Hooking	Deobfuscate/Decode Files or Information	Kerberoasting	Remote System Discovery	Shared Webroot	Screen Capture		Multiband Communication
	LSASS Driver	Create Account	Image File Execution Options Injection	Disabling Security Tools	Keychain	Security Software Discovery	SSH Hijacking	Video Capture		Multilayer Encryption
	Mshta	DLL Search Order Hijacking	Launch Daemon	DLL Search Order Hijacking	LLMNR/NBT-NS Poisoning	System Information Discovery	Taint Shared Content			Port Knocking
	PowerShell	Dylib Hijacking	New Service	DLL Side-Loading	Network Sniffing	System Network Configuration Discovery	Third-party Software			Remote Access Tools
	Regsvcs/Regasm	External Remote Services	Path Interception	Exploitation for Defense Evasion	Password Filter DLL	System Network Connections Discovery	Windows Admin Shares			Remote File Copy
	Regsvr32	File System Permissions Weakness	Plist Modification	Extra Window Memory Injection	Private Keys	System Owner/User Discovery	Windows Remote Management			Standard Application Layer Protocol
	Rundll32	Hidden Files and Directories	Port Monitors	File Deletion	Replication Through Removable Media	System Service Discovery				Standard Cryptographic Protocol
	Scheduled Task	Hooking	Process Injection	File System Logical Offsets	Securityd Memory	System Time Discovery				Standard Non-Application Layer Protocol
	Scripting	Hypervisor	Scheduled Task	Gatekeeper Bypass	Two-Factor Authentication Interception					Uncommonly Used Port
	Service Execution	Image File Execution Options Injection	Service Registry Permissions Weakness	Hidden Files and Directories						Web Service
	Signed Binary Proxy Execution	Kernel Modules and Extensions	Setuid and Setgid	Hidden Users						
	Signed Script Proxy Execution	Launch Agent	SID-History Injection	Hidden Window						
	Source	Launch Daemon	Startup Items	HISTCONTROL						
	Space after Filename	Launchctl	Sudo	Image File Execution Options Injection						
	Third-party Software	LC_LOAD_DYLIB Addition	Sudo Caching	Indicator Blocking						
	Trap	Local Job Scheduling	Valid Accounts	Indicator Removal from Tools						
	Trusted Developer Utilities	Login Item	Web Shell	Indicator Removal on Host						
	User Execution	Logon Scripts		Indirect Command Execution						
	Windows Management Instrumentation	LSASS Driver		Install Root Certificate						
	Windows Remote Management	Modify Existing Service		InstallUtil						
		Netsh Helper DLL		Launchctl						
		New Service		LC_MAIN Hijacking						
		Office Application Startup		Masquerading						

Purple = APT29 techniques that can test our gaps

MITRE

Step 2: Extract ATT&CK techniques from reports

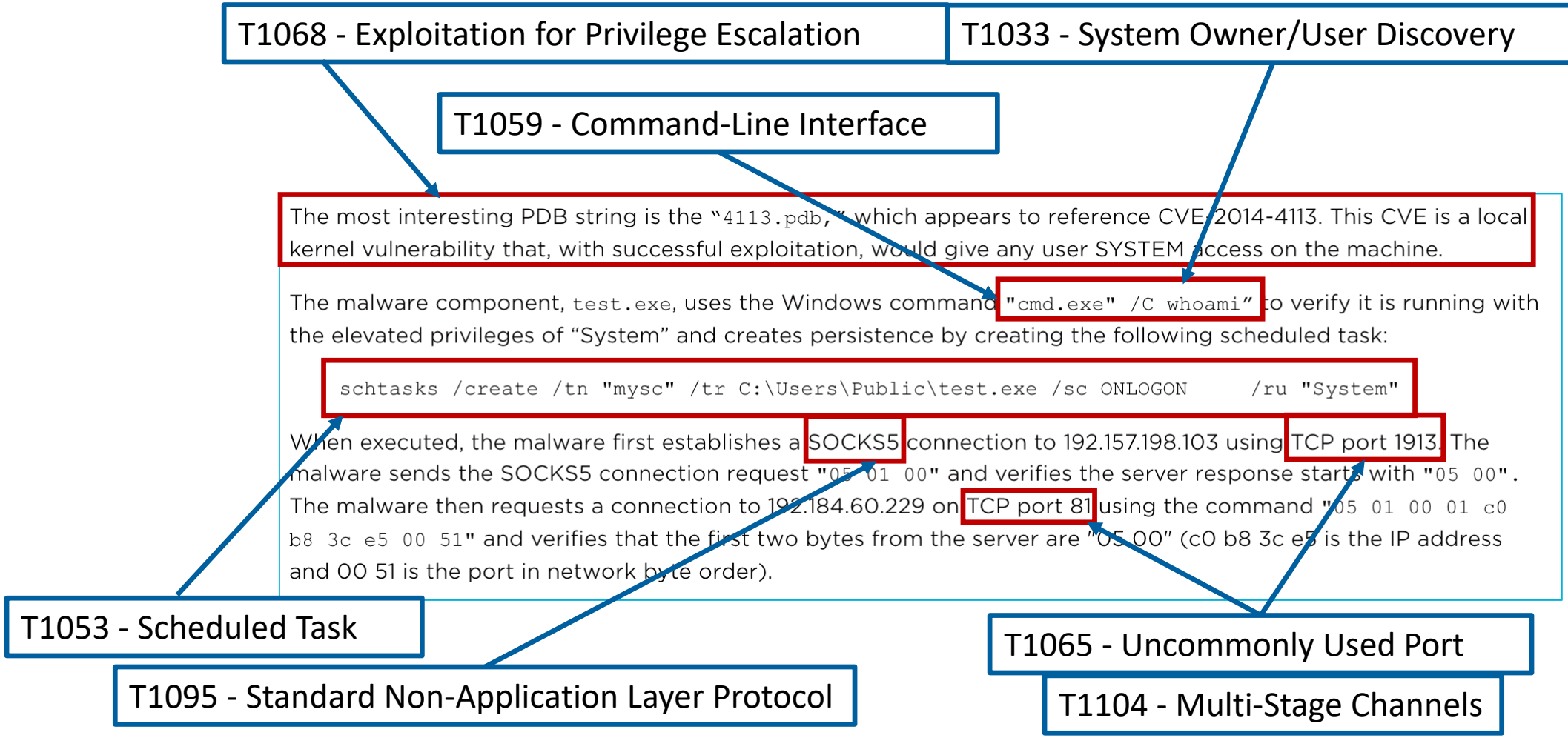


- **Look for behaviors**
- **Store the info in a structured way**
- **Have the threat intel originator do it**
- **Start at the tactic level**
- **Use ATT&CK website examples**
- **Work as a team**

MITRE

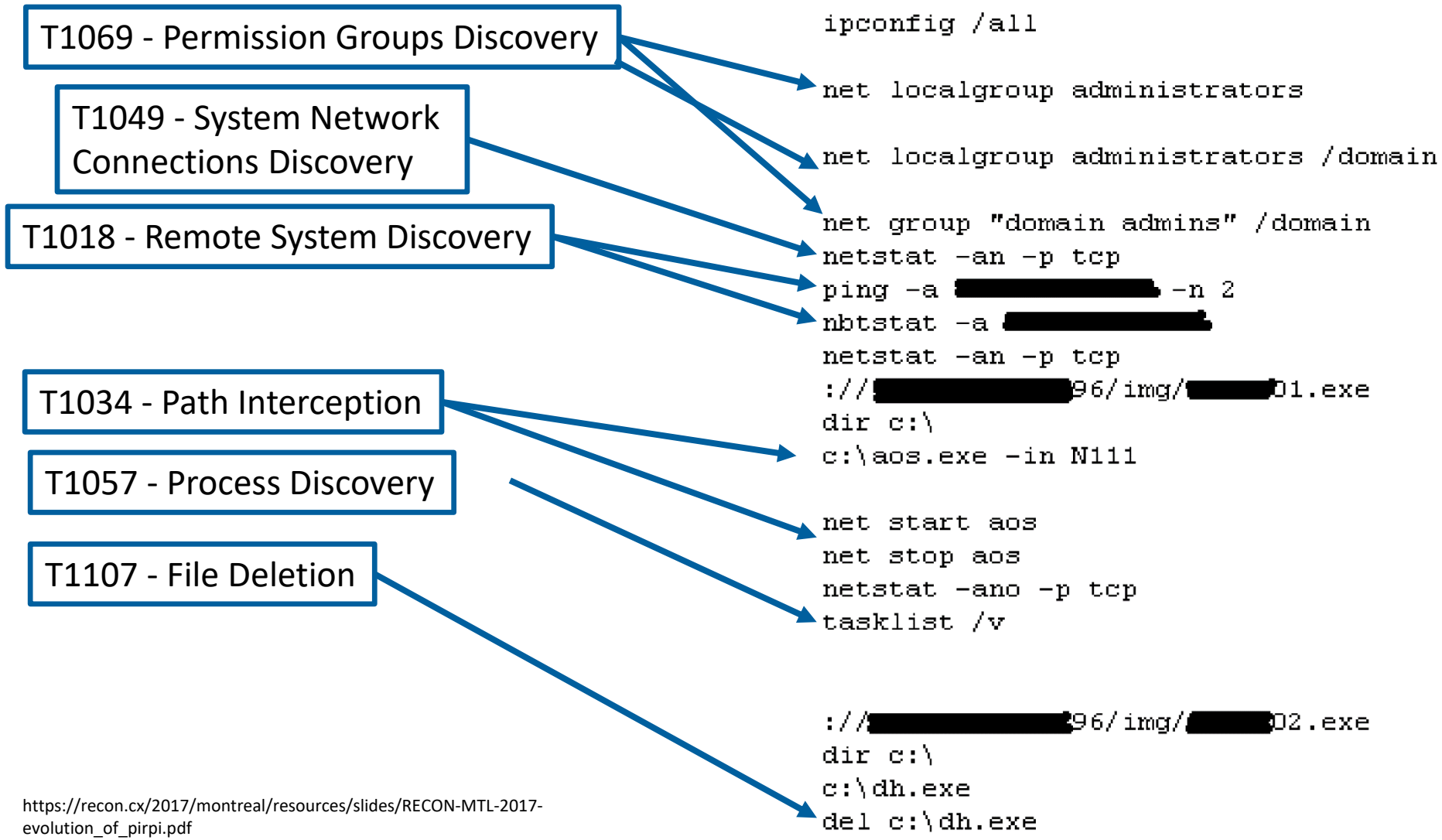
How to extract ATT&CK techniques

https://www.fireeye.com/blog/threat-research/2014/11/operation_doubletap.html



MITRE

How to extract ATT&CK techniques



https://recon.cx/2017/montreal/resources/slides/RECON-MTL-2017-evolution_of_pirpi.pdf

MITRE

Step 3: Analyze and organize techniques and intel

- **Establish the adversary's goal**
- **Consider adversary M.O.**
- **Think about the why, what, and how**
  - In ATT&CK: Tactic, Technique, Procedure

MITRE

Analyze intel for adversary M.O.

Buckeye seems to target file and print servers, which makes it likely the group is looking to steal documents. This, coupled with
mized tools, and the types of organizations being targeted would suggest
up.

Buckeye seems to target file and print servers, which makes it likely the group is looking to steal documents

https://www.symantec.com/connect/blogs/buckeye-cyberespionage-group-shifts-gaze-us-hong-kong

The APT group responsible for this exploit has been the first group to have access to a select number of browser-based 0-day exploits (e.g. IE, Firefox, and Flash) in the past. They are extremely proficient at lateral movement and are difficult to track, as they typically do not reuse
backdoors including one known as Pirpi that we pr
Internet Explorer 6, 7, and 8 dropped the Pirpi payl

They are extremely proficient at lateral movement … and typically do not reuse command and control infrastructure

https://www.fireeye.com/blog/threat-research/2014/04/new-zero-day-exploit-targeting-internet-explorer-versions-9-through-11-identified-in-targeted-attacks.html

First, consider the fact that the rarsfx archive is created 5-6 months before this attack; next examine the insertion times of the differe
few minutes, but by days. This attacker lik
before this attack.

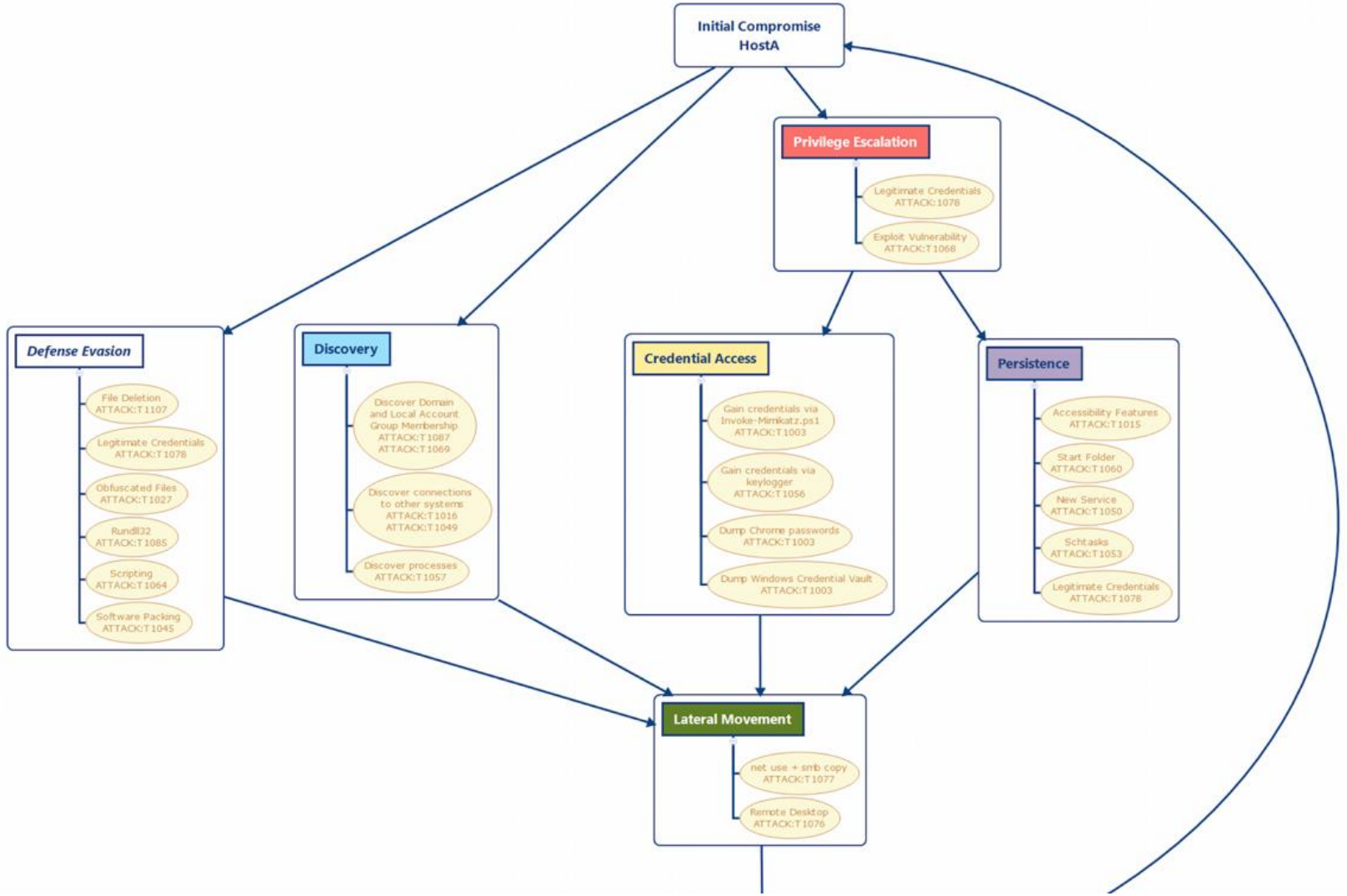
The rarsfx archive is created 5-6 months before this attack … used the same rarsfx archive with other payloads before this attack.

https://www.lastline.com/labsblog/an-analysis-of-plugx-malware/

MITRE

Organize intel into technique flow

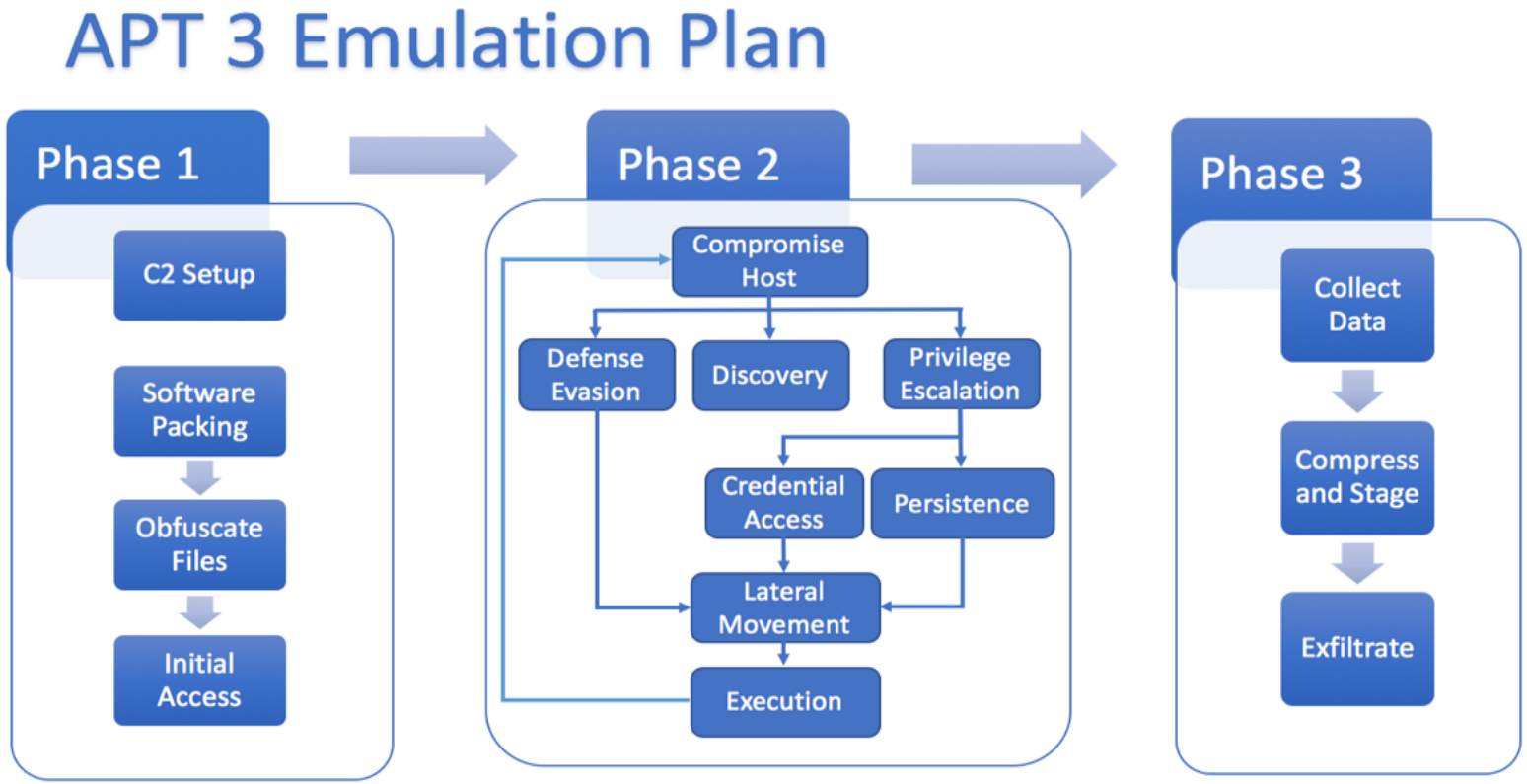
- **Provide order to techniques**
  - Not going to be perfect
  - Techniques have their own required ordering
  - Feeds the emulation plan



MITRE

Organize technique flow into plan phases

- This is the hardest part of the puzzle
- No plan will be perfect, so approximate where needed
- This isn't a replay of an incident - variation is OK

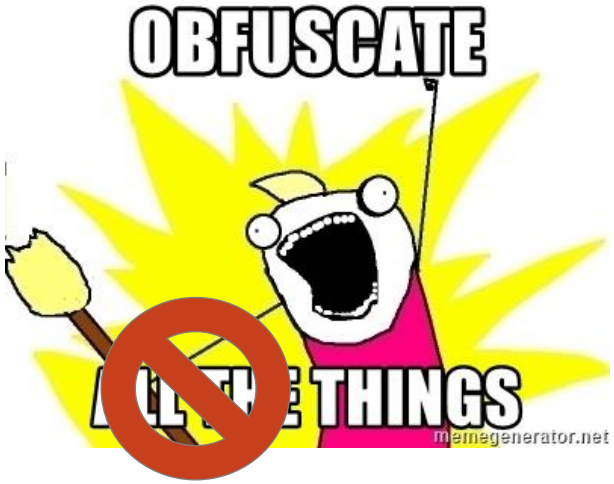


MITRE

Step 4: Develop tools to emulate behaviors

Gather threat intel → Extract techniques → Analyze & organize → **Develop tools** → Emulate the adversary

- **What are the COTS / Open Source tools available?**
  - Can you exhibit the right behaviors with these tools?
  - Can you extend or modify them?
- **Do you need to develop something specific?**
  - Delivery mechanisms, Command and Control, Capabilities
- **Create payloads "inspired by" the adversary's tradecraft**
  - Modify IoCs and behaviors if possible
  - Obfuscate with purpose, NOT all the things – "over-obfuscation" is itself suspicious!



MITRE

What is behavioral emulation for TTPs?

- **Performing adversary techniques with variations**
  - Adversary created “C:\aos.exe” for Priv Esc via path interception
    - You intercept any service path that runs under higher privileges
  - Adversary used “PSExec” for Lateral Movement
    - You do it manually with “sc.exe” or via PowerShell
  - Adversary runs “whoami” for Discovery
    - You do it with environment variables “%USERDOMAIN%\%USERNAME%”

Defining your toolset

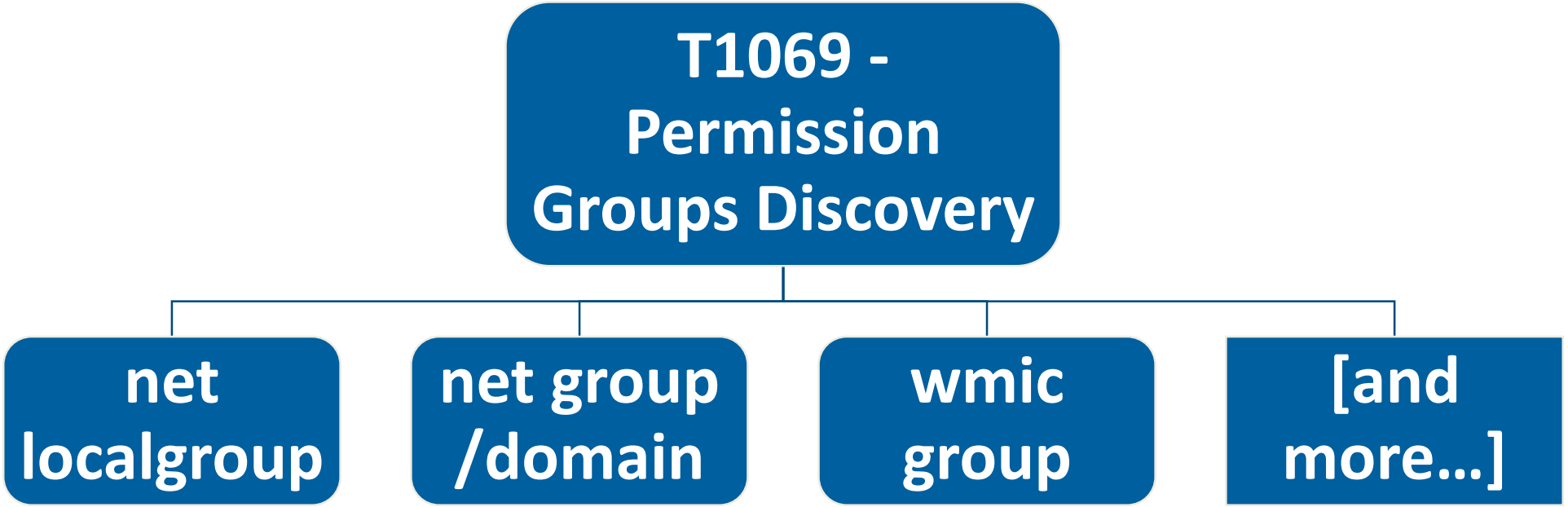
- **Don’t limit yourself to a single environment or tool**
  - Python, PowerShell, Command-Line, Custom Binary, etc
- **Do stay within the behavior boundaries**

Table 2 Pirpi Functions and Emulation

Pirpi Function	Windows Built-in	Cobalt Strike/Beacon	Metasploit/Meterpreter	ATT&CK Technique
List processes	tasklist	ps, shell qprocess *	ps	T1057 - Process Discovery
Download file	ftp	download [filename]	Download [filename]	T1041 - Exfiltration over Command and Control Channel

Create an Adversary Emulation Field Manual

- Provides multiple implementations across toolsets
- Provides offensive command-line examples
- Create this as you go, and use for reference later



MITRE

Step 5: Emulate the adversary

- **Set up infrastructure and test**
  - Set up C2 servers & redirector, buy domains, test, install
- **Emulate the adversary!**
  - Follow the adversary M.O.
  - “Domain Admin” most likely isn’t your goal
  - Keep the “speed of the adversary” in mind
    - Low and slow vs smash and grab

MITRE

In summary…

- Test your hunting capabilities with adversary emulation
- Use threat intelligence to drive your emulation
- Move toward a threat-based defense

Links

- **ATT&CK**
  - https://attack.mitre.org
  - github.com/mitre/cti
  - cti-taxii.mitre.org
- **ATT&CK Navigator**
  - https://github.com/mitre/attack-navigator
  - https://mitre.github.io/attack-navigator/enterprise/
- **Adversary Emulation Plans**
  - https://attack.mitre.org/wiki/Adversary_Emulation_Plans
- **CALDERA: Automated Adversary Emulation**
  - https://github.com/mitre/caldera
- **Cyber Analytic Repository (CAR)**
  - https://car.mitre.org

MITRE

ATT&CK

attack.mitre.org

attack@mitre.org
@MITREattack

Katie Nickels

@likethecoins

Cody Thomas

@its_a_feature_

MITRE