Endpoint Protection

Right of Boom

CIS Control 10 - Malware Defenses

Prevent or control the installation, spread, and execution of malicious applications, code, or scripts on enterprise assets.

☑	10.1	Deploy and Maintain Anti-Malware Software	Devices	●	●	●
☑	10.2	Configure Automatic Anti-Malware Signature Updates	Devices	●	●	●
☑	10.3	Disable Autorun and Autoplay for Removable Media	Devices	●	●	●
☑	10.4	Configure Automatic Anti-Malware Scanning of Removable Media	Devices		●	●
☑	10.5	Enable Anti-Exploitation Features	Devices		●	●
☑	10.6	Centrally Manage Anti-Malware Software	Devices		●	●
☑	10.7	Use Behavior-Based Anti-Malware Software	Devices		●	●

MITRE

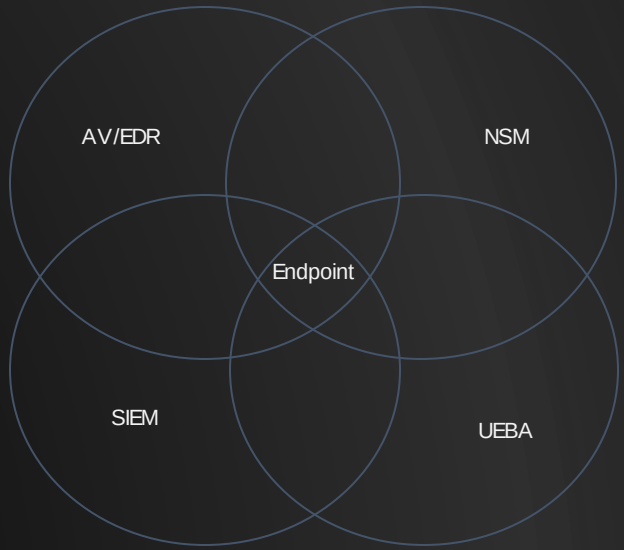
ATT&CK Matrix for Enterprise

Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection	Command and Control	Exfiltration	Impact
Drive-by Compromise	AppleScript	.bash_profile and .bashrc	Access Token Manipulation	Access Token Manipulation	Account Manipulation	Account Discovery	AppleScript	Audio Capture	Commonly Used Port	Automated Exfiltration	Account Access Removal
Exploit Public-Facing Application	CMSTP	Accessibility Features	Accessibility Features	Binary Padding	Bash History	Application Window Discovery	Application Deployment Software	Automated Collection	Communication Through Removable Media	Data Compressed	Data Destruction
External Remote Services	Command-Line Interface	Account Manipulation	AppCert DLLs	BITS Jobs	Brute Force	Browser Bookmark Discovery	Component Object Model and Distributed COM	Clipboard Data	Connection Proxy	Data Encrypted	Data Encrypted for Impact
Hardware Additions	Compiled HTML File	AppCert DLLs	AppInit DLLs	Bypass User Account Control	Credential Dumping	Domain Trust Discovery	Exploitation of Remote Services	Data from Information Repositories	Custom Command and Control Protocol	Data Transfer Size Limits	Defacement
Replication Through Removable Media	Component Object Model and Distributed COM	AppInit DLLs	Application Shimming	Clear Command History	Credentials from Web Browsers	File and Directory Discovery	Internal Spearphishing	Data from Local System	Custom Cryptographic Protocol	Exfiltration Over Alternative Protocol	Disk Content Wipe
Spearphishing Attachment	Control Panel Items	Application Shimming	Bypass User Account Control	CMSTP	Credentials in Files	Network Service Scanning	Logon Scripts	Data from Network Shared Drive	Data Encoding	Exfiltration Over Command and Control Channel	Disk Structure Wipe
Spearphishing Link	Dynamic Data Exchange	Authentication Package	DLL Search Order Hijacking	Code Signing	Credentials in Registry	Network Share Discovery	Pass the Hash	Data from Removable Media	Data Obfuscation	Exfiltration Over Other Network Medium	Endpoint Denial of Service
Spearphishing via Service	Execution through API	BITS Jobs	Dylib Hijacking	Compile After Delivery	Exploitation for Credential Access	Network Sniffing	Pass the Ticket	Data Staged	Domain Fronting	Exfiltration Over Physical Medium	Firmware Corruption
Supply Chain Compromise	Execution through Module Load	Bootkit	Elevated Execution with Prompt	Compiled HTML File	Forced Authentication	Password Policy Discovery	Remote Desktop Protocol	Email Collection	Domain Generation Algorithms	Scheduled Transfer	Inhibit System Recovery
Trusted Relationship	Exploitation for Client Execution	Browser Extensions	Emond	Component Firmware	Hooking	Peripheral Device Discovery	Remote File Copy	Input Capture	Fallback Channels		Network Denial of Service
Valid Accounts	Graphical User Interface	Change Default File Association	Exploitation for Privilege Escalation	Component Object Model Hijacking	Input Capture	Permission Groups Discovery	Remote Services	Man in the Browser	Multi-hop Proxy		Resource Hijacking
	InstallUtil	Component Firmware	Extra Window Memory Injection	Connection Proxy	Input Prompt	Process Discovery	Replication Through Removable Media	Screen Capture	Multi-Stage Channels		Runtime Data Manipulation
	Launchctl	Component Object Model Hijacking	File System Permissions Weakness	Control Panel Items	Kerberoasting	Query Registry	Shared Webroot	Video Capture	Multiband Communication		Service Stop
	Local Job Scheduling	Create Account	Hooking	DCShadow	Keychain	Remote System Discovery	SSH Hijacking		Multilayer Encryption		Stored Data Manipulation
	LSASS Driver	DLL Search Order Hijacking	Image File Execution Options Injection	Deobfuscate/Decode Files or Information	LLMNR/NBT-NS Poisoning and Relay	Security Software Discovery	Taint Shared Content		Port Knocking		System Shutdown/Reboot
	Mshta	Dylib Hijacking	Launch Daemon	Disabling Security Tools	Network Sniffing	Software Discovery	Third-party Software		Remote Access Tools		Transmitted Data Manipulation
	PowerShell	Emond	New Service	DLL Search Order Hijacking	Password Filter DLL	System Information Discovery	Windows Admin Shares		Remote File Copy		
	Regsvcs/Regasm	External Remote Services	Parent PID Spoofing	DLL Side-Loading	Private Keys	System Network Configuration Discovery	Windows Remote Management		Standard Application Layer Protocol		
						System Network			Standard Cryptographic		

Advanced Endpoint Protection

Overlapping Fields of View

- The key is overlapping fields of visibility
- Endpoint
- SIEM/UEBA
- Network Monitoring
- Sandboxing
- Internal Segmentation



Get Ready to Pay!

- Traditional Denylist AV is pretty much garbage
- With advanced Endpoint Detection and Response (EDR) we have the ability to look at the system more holistically
- What process begats which connection and other processes?
- A special note on Artificial intelligence

BlackBerry CYLANCE

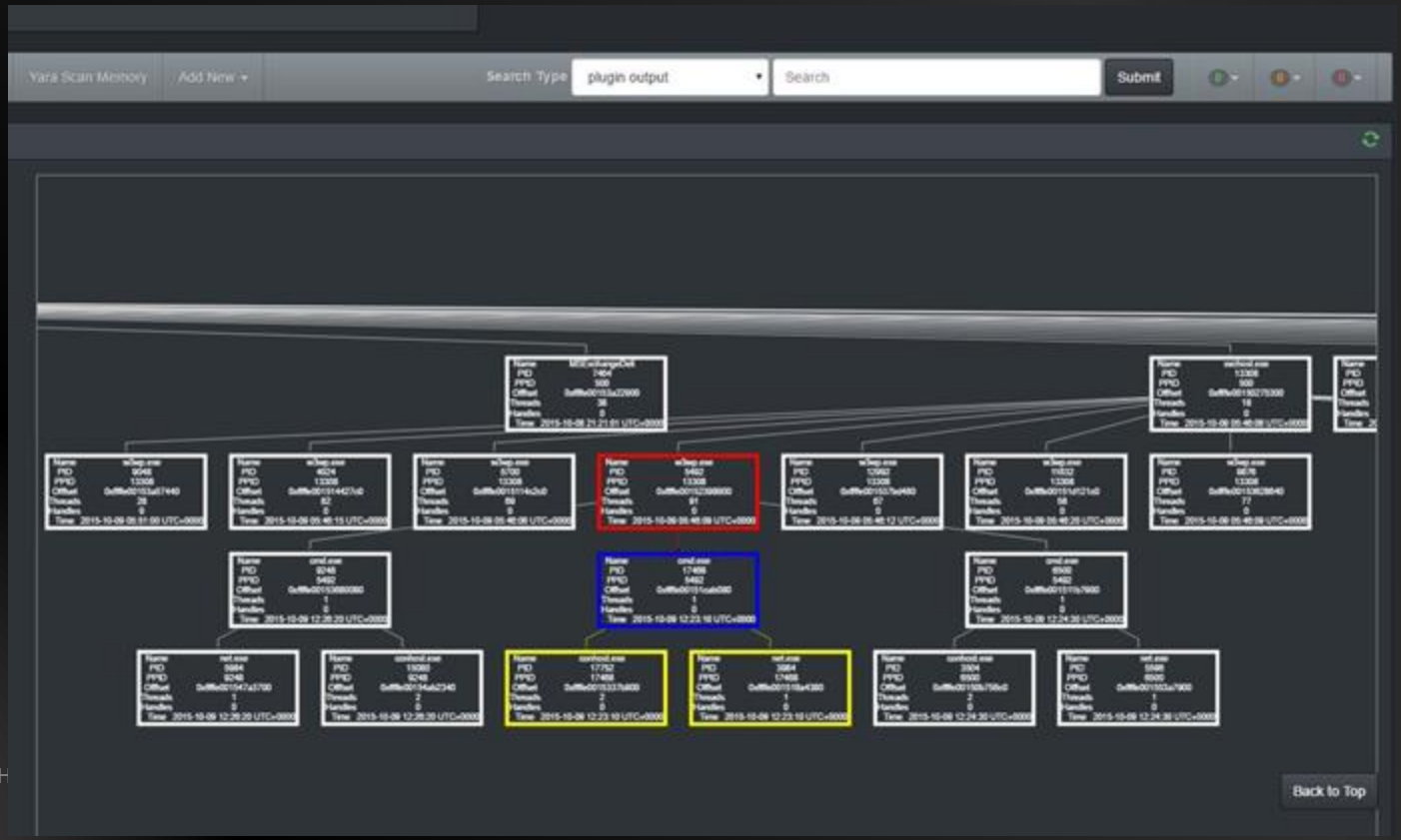
Carbon Black.

ENDGAME.

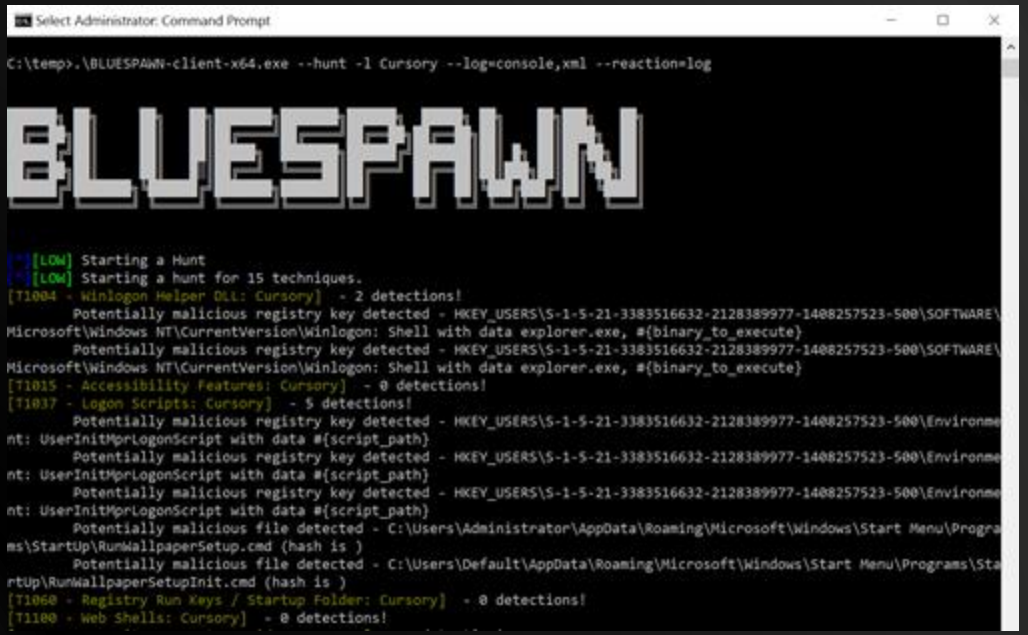
SentinelOne

CROWDSTRIKE

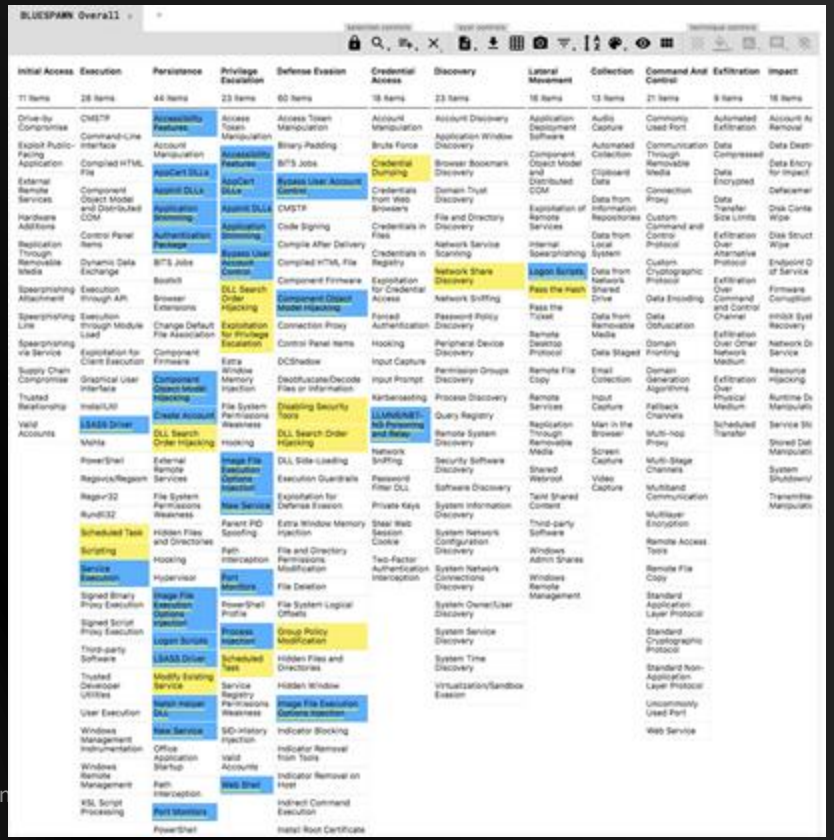
Incident Response



Bluespawn


Select Administrator: Command Prompt
C:\temp>.\BLUESPAWN-client-x64.exe --hunt -l Cursory --log=console,xml --reaction=log
BLUESPAWN
[*][LOW] Starting a Hunt
[*][LOW] Starting a hunt for 15 techniques.
[T1004 - Winlogon Helper DLL: Cursory] - 2 detections!
Potentially malicious registry key detected - HKEY_USERS\S-1-5-21-3383516632-2128389977-1408257523-500\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon: Shell with data explorer.exe, #{binary_to_execute}
Potentially malicious registry key detected - HKEY_USERS\S-1-5-21-3383516632-2128389977-1408257523-500\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon: Shell with data explorer.exe, #{binary_to_execute}
[T1015 - Accessibility Features: Cursory] - 0 detections!
[T1037 - Logon Scripts: Cursory] - 5 detections!
Potentially malicious registry key detected - HKEY_USERS\S-1-5-21-3383516632-2128389977-1408257523-500\Environment: UserInitMprLogonScript with data #{script_path}
Potentially malicious registry key detected - HKEY_USERS\S-1-5-21-3383516632-2128389977-1408257523-500\Environment: UserInitMprLogonScript with data #{script_path}
Potentially malicious registry key detected - HKEY_USERS\S-1-5-21-3383516632-2128389977-1408257523-500\Environment: UserInitMprLogonScript with data #{script_path}
Potentially malicious file detected - C:\Users\Administrator\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\StartUp\RunWallpaperSetup.cmd (hash is)
Potentially malicious file detected - C:\Users\Default\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\StartUp\RunWallpaperSetupInit.cmd (hash is)
[T1060 - Registry Run Keys / Startup Folder: Cursory] - 0 detections!
[T1100 - Web Shells: Cursory] - 0 detections!

MITRE Coverage

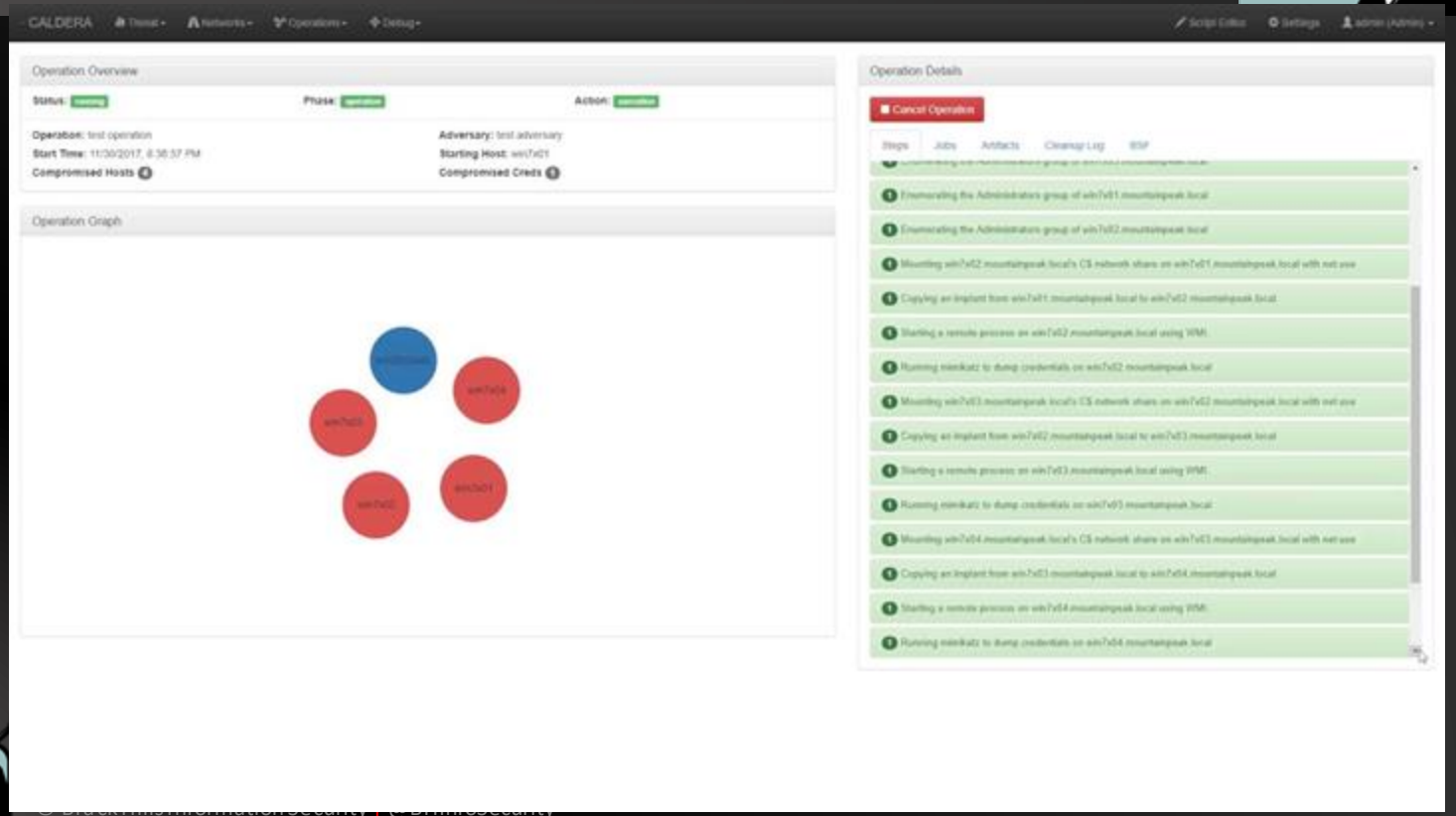


Threat Emulation

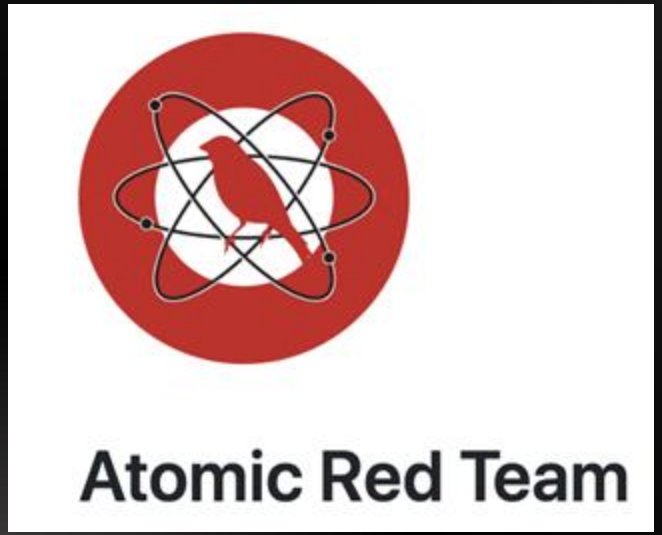
- Don’t just think of vulnerabilities as missing patches and misconfigurations on systems
- Think post exploitation
- What happens after an attacker gains access to a system
- There are a number of free tools that will automate parts of this process
- Currently, would take a bit of tuning and trial and error
- The collected data is invaluable

Question:
What is the Difference Between Emulation and Simulation?

Open Source Tool Example: Caldera



Open Source Tool Example: Atomic Red Team



Execute All Attacks for a Given Technique

```
Invoke-AtomicTest T1117
```

Speficy a Process Timeout

```
Invoke-AtomicTest T1117 -TimeoutSeconds 15
```

If the attack commands do not exit (return) within in the specified `-TimeoutSeconds`, the process and it's children will be forcefully terminated. The default value of `-TimeoutSeconds` is 120. This allows the `Invoke-AtomicTest` script to move on to the next test.

Execute All Tests

This is not recommended but you can execute all Atomic tests in your atomics folder with the follwing:

```
Invoke-AtomicTest All
```

Execute All Tests from a Specific Directory

Specify a custom path to your atomics folder, example C:\AtomicRedTeam\atomics

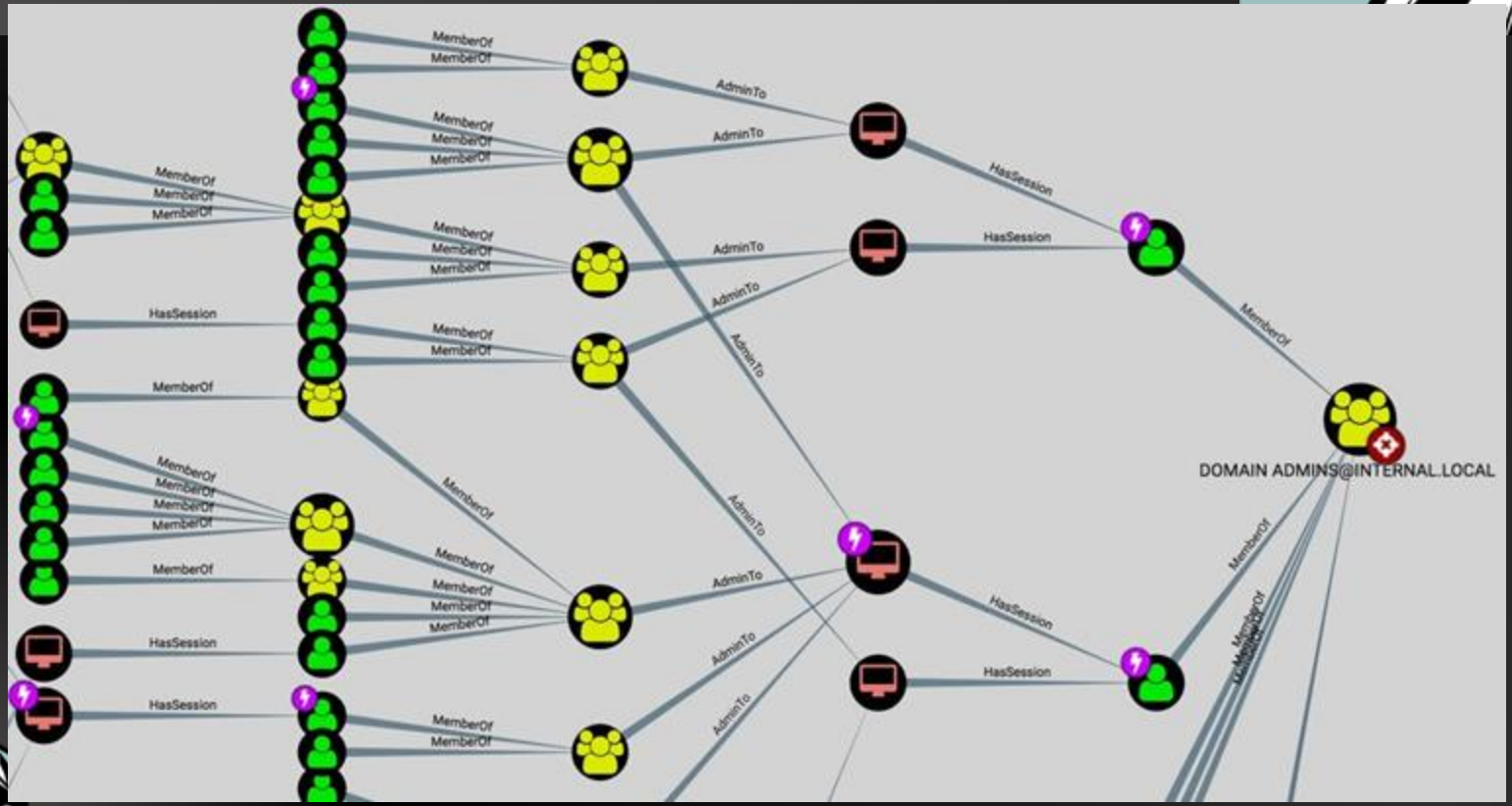
```
Invoke-AtomicTest All -PathToAtomicsFolder C:\AtomicRedTeam\atomics
```

```
PS C:\AtomicRedTeam> Invoke-AtomicTest T1117 -TestNumbers 1 -ShowDetails
PathToAtomicsFolder = C:\AtomicRedTeam\atomics

[********BEGIN TEST*******]
Technique: Regsvr32 T1117
Atomic Test Name: Regsvr32 local COM scriptlet execution
Atomic Test Number: 1
Description: Regsvr32.exe is a command-line program used to register and unregister OLE controls.
Upon execution, calc.exe will be launched.
Attack Commands:
Executor: command_prompt
ElevationRequired: False
Command:
regsvr32.exe /s /u /i:#{filename} scrobj.dll
Command (with inputs):
regsvr32.exe /s /u /i:C:\AtomicRedTeam\atomics\T1117\src\RegSvr32.sct scrobj.dll
Dependencies:
Description: Regsvr32.exe must exist on disk at specified location (C:\AtomicRedTeam\atomics\T1117
\src\RegSvr32.sct)
Check Prereq Command:
if (Test-Path #{filename}) {exit 0} else {exit 1}
Check Prereq Command (with inputs):
if (Test-Path C:\AtomicRedTeam\atomics\T1117\src\RegSvr32.sct) {exit 0} else {exit 1}
Get Prereq Command:
New-Item -Type Directory (split-path #{filename}) -ErrorAction ignore | Out-Null
Invoke-WebRequest "https://github.com/redcanaryco/atomic-red-team/raw/master/atomics/T1117/src/Reg
Svr32.sct" -OutFile "#{filename}"
Get Prereq Command (with inputs):
New-Item -Type Directory (split-path C:\AtomicRedTeam\atomics\T1117\src\RegSvr32.sct) -ErrorAction
 ignore | Out-Null
Invoke-WebRequest "https://github.com/redcanaryco/atomic-red-team/raw/master/atomics/T1117/src/Reg
Svr32.sct" -OutFile "C:\AtomicRedTeam\atomics\T1117\src\RegSvr32.sct"
[!!!!!!!!END TEST!!!!!!!]
```

Question:
How Many of You Feel Comfortable Running Tools Like Atomic Red Team?

Open Source Tool Example: Bloodhound



Threat Emulation Warning

- One of the traps of the MITRE framework and threat emulation is we train or systems to detect specific attacks
- Most of the attacks in Atomic Red Team and MITRE are representations of classes of attacks
- We are seeing vendors simply detect those attacks
  - More on this later!
- A few modifications and you can easily bypass detection

Commercial Offerings

ATTACKIQ

SCYTHE

XM CYBER

Everyone's a Winner!

MITRE | ATT&CK® Evaluations

Evaluations ▾ Tools ▾ Resources ▾ Get Evaluated

Home > APT3

APT3 Emulation

ATT&CK Evaluations 2018

RESULTS

Call For Participation → Evaluating → Preparing → Published

ATT&CK Description

APT3 is a China-based threat group that researchers have attributed to China's Ministry of State Security. [1] [2] This group is responsible for the campaigns known as Operation Clandestine Fox, Operation Clandestine Wolf, and Operation Double Tap. [1] [3] As of June 2015, the group appears to have shifted from targeting primarily US victims to primarily political organizations in Hong Kong. [4]

Emulation Notes

APT3 relies on harvesting credentials, issuing on-keyboard commands (versus Windows API calls), and using programs already trusted by the operating system ("living off the land"). Similarly, they are not known to do elaborate scripting techniques, leverage exploits after initial access, or use anti-EDR capabilities such as rootkits or bootkits.

Participants

Initial Cohort

Carbon Black. · CROWDSTRIKE · elastic · GoSecure · Windows Defender ATP · RSA · SentinelOne

Rolling Admission

cybereason · F-Secure · FIREEYE · McAfee Together is power. · paloalto CORTEX XDR

Scenario Overview

Two scenarios emulate publicly reported APT3/Gothic Panda tradecraft and operational flows. In both scenarios, access is established on the target victim. The scenario then proceeds into local/remote discovery, elevation of privileges, grabbing available credentials, then finally lateral movement within the breached network before collecting and exfiltrating sensitive data. Both scenarios include executing previously established persistence mechanisms executed after a simulated time lapse.

Red Team tooling is what primarily distinguishes the two scenarios. Cobalt Strike was used to execute the first scenario, while PowerShell Empire was used to execute the second. Using two different toolsets resulted in diversity and an observable variance in the emulation of the APT3/Gothic Panda behaviors.

Information Security
CELEBRATING 10 YEARS
2008-2018

Detection Categories

Main Detection Types

- None
- Telemetry
- MSSP
- General
- Tactic
- Technique

Modifier Detection Types

- Alert
- Correlated
- Delayed
- Host Interrogation
- Residual Artifact
- Configuration Change

Or not?

README.md

attack-eval-scoring

This project represented my attempts at analyzing the results of round 1 of the MITRE Enterprise ATT&CK Evaluation. With the release of round 2 results, please check out my new project: https://github.com/joshzelonis/EnterpriseAPT29Eval

For my initial blog post on the subject, check out: https://go.forrester.com/blogs/measuring-vendor-efficacy-using-the-mitre-attck-evaluation/

simple_score.py

In parsing the results, I found 56 ATT&CK techniques were measured with 136 procedures for doing so. This is a quick script for applying the scale on a procedure (or per step) basis. There were many instances where there were multiple detections for a single procedure/step which would skew any counting method that did not take this into effect.

coverage.py

This script generates two key metrics for understanding vendor performance. The first of which is a coverage score which gives insight into the percentage of ATT&CK techniques the solution was able to generate any type of detection against. This can be viewed as a high water mark for how the product could be used to generate detections. The second metric is a correlation metric which is the percentage of detections that had a tainted modifier. This is useful for understanding how the product reduces work for SOC analysts.

kill_chain_analysis.py

There were 10 different stages of attack measured from initial compromise to execution of persistence across two scenarios. One may argue that the most critical capability is being able to alert on an aversary at each stage of an intrusion. This script analyzes and breaks out how each vendor performed at each stage of these scenarios on the same 1-3-5 scale used by simple_score.py

“Simple” Score

```
john@pop-os:~/attack-eval-scoring$ python3 simple_score.py
./data/McAfee.1.APT3.1_Results.json - 268
./data/CarbonBlack.1.APT3.1_Results.json - 259
./data/Cybereason.1.APT3.1_Results.json - 285
./data/Microsoft.1.APT3.1_Results.json - 195
./data/PaloAltoNetworks.1.APT3.1_Results.json - 329
./data/GoSecure.1.APT3.1_Results.json - 108
./data/RSA.1.APT3.1_Results.json - 78
./data/F-Secure.1.APT3.1_Results.json - 376
./data/Endgame.1.APT3.1_Results.json - 225
./data/FireEye.1.APT3.1_Results.json - 288
./data/CrowdStrike.1.APT3.1_Results.json - 269
./data/SentinelOne.1.APT3.1_Results.json - 123
```

Misses

```
john@pop-os:~/attack-eval-scoring$ python3 total_misses.py
./data/McAfee.1.APT3.1_Results.json - 38
./data/CarbonBlack.1.APT3.1_Results.json - 34
./data/Cybereason.1.APT3.1_Results.json - 24
./data/Microsoft.1.APT3.1_Results.json - 23
./data/PaloAltoNetworks.1.APT3.1_Results.json - 9
./data/GoSecure.1.APT3.1_Results.json - 28
./data/RSA.1.APT3.1_Results.json - 49
./data/F-Secure.1.APT3.1_Results.json - 14
./data/Endgame.1.APT3.1_Results.json - 14
./data/FireEye.1.APT3.1_Results.json - 32
./data/CrowdStrike.1.APT3.1_Results.json - 22
./data/SentinelOne.1.APT3.1_Results.json - 35
```

Penetration Testing

Right of Boom

CIS Control 18 - Penetration Testing

Test the effectiveness and resiliency of enterprise assets through identifying and exploiting weaknesses in controls (people, processes, and technology), and simulating the objectives and actions of an attacker.

☑	18.1	Establish and Maintain a Penetration Testing Program	N/A	●	●	
☑	18.2	Perform Periodic External Penetration Tests	Network	●	●	
☑	18.3	Remediate Penetration Test Findings	Network	●	●	
☑	18.4	Validate Security Measures	Network		●	
☑	18.5	Perform Periodic Internal Penetration Tests	N/A		●	

Right of Boom

LAB: EDR with Bluespawn