#ThreatThursday - APT33

scythe.io/library/threatthursday-apt33

<< All Posts

Jorge Orchilles

June 18, 2020

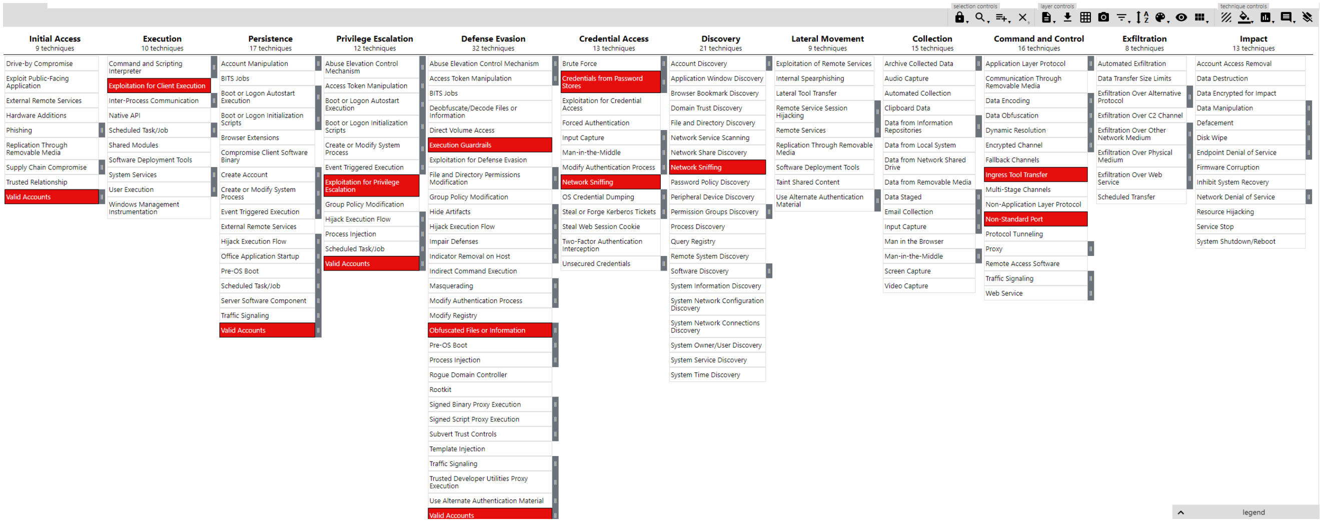
This week on #ThreatThursday we look at an Iranian Threat Actor, APT33 or Elfin. We introduce the MITRE ATT&CK Beta with sub-techniques, create and share an adversary emulation plan for APT33 on Github, show how to execute PowerShell (both powershell.exe and unmanaged PowerShell) through SCYTHE and show how to perform lateral movement within the SCYTHE user interface as well as on the command line. As usual, we cover how to detect attacks from APT33. We hope you enjoy it!

Cyber Threat Intelligence

If you read #ThreatThursday on APT19 you saw how to leverage MITRE ATT&CK for Cyber Threat Intelligence and map it with ATT&CK Navigator. APT33 is documented on the MITRE ATT&CK site so we do not need to extract TTPs from Cyber Threat Intelligence like we did with Buhtrap. Although, we do recommend reading through the CTI as you may get details about the procedures used by the threat actor. Here are a few reports on APT33:

Keeping with the theme of covering new topics, let's explore the beta of ATT&CK Navigator that has sub-techniques. It is available on: https://mitre-attack.github.io/attack-navigator/beta/enterprise/

We select APT33 as we showed in the APT19 #ThreatThursday and see something similar to Figure 1:



Initial Access
9 techniques

- Drive-by Compromise
- Exploit Public-Facing Application
- External Remote Services
- Hardware Additions
- Phishing
- Replication Through Removable Media
- Supply Chain Compromise
- Trusted Relationship
- Valid Accounts

Execution
10 techniques

- Command and Scripting Interpreter
- Exploitation for Client Execution
- Inter-Process Communication
- Native API
- Scheduled Task/Job
- Shared Modules
- Software Deployment Tools
- System Services
- User Execution
- Windows Management Instrumentation

Persistence
17 techniques

- 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
- Office Application Startup
- Pre-OS Boot
- Scheduled Task/Job
- Server Software Component
- Traffic Signaling
- Valid Accounts

Privilege Escalation
12 techniques

- Abuse Elevation Control Mechanism
- Access Token Manipulation
- Boot or Logon Autostart Execution
- Boot or Logon Initialization Scripts
- Create or Modify System Process
- Event Triggered Execution
- Exploitation for Privilege Escalation
- Group Policy Modification
- Hijack Execution Flow
- Process Injection
- Scheduled Task/Job
- Valid Accounts

Defense Evasion
32 techniques

- Abuse Elevation Control Mechanism
- Access Token Manipulation
- BITS Jobs
- Deobfuscate/Decode Files or Information
- Direct Volume Access
- Execution Guardrails
- Exploitation for Defense Evasion
- File and Directory Permissions Modification
- Group Policy Modification
- Hide Artifacts
- Hijack Execution Flow
- Impair Defenses
- Indicator Removal on Host
- Indirect Command Execution
- Masquerading
- Modify Authentication Process
- Modify Registry
- Obfuscated Files or Information
- Pre-OS Boot
- Process Injection
- Rogue Domain Controller
- Rootkit
- Signed Binary Proxy Execution
- Signed Script Proxy Execution
- Subvert Trust Controls
- Template Injection
- Traffic Signaling
- Trusted Developer Utilities Proxy Execution
- Use Alternate Authentication Material
- Valid Accounts

Credential Access
13 techniques

- Brute Force
- Credentials from Password Stores
- Exploitation for Credential Access
- Forced Authentication
- Input Capture
- Man-in-the-Middle
- Modify Authentication Process
- Network Sniffing
- OS Credential Dumping
- Steal or Forge Kerberos Tickets
- Steal Web Session Cookie
- Two-Factor Authentication Interception
- Unsecured Credentials

Discovery
21 techniques

- Account Discovery
- Application Window Discovery
- Browser Bookmark Discovery
- Domain Trust Discovery
- File and Directory Discovery
- Network Service Scanning
- Network Share Discovery
- Network Sniffing
- Password Policy Discovery
- Peripheral Device Discovery
- Permission Groups Discovery
- Process Discovery
- Query Registry
- Remote System Discovery
- Software Discovery
- System Information Discovery
- System Network Configuration Discovery
- System Network Connections Discovery
- System Owner/User Discovery
- System Service Discovery
- System Time Discovery

Lateral Movement
9 techniques

- Exploitation of Remote Services
- Internal Spearphishing
- Lateral Tool Transfer
- Remote Service Session Hijacking
- Remote Services
- Replication Through Removable Media
- Software Deployment Tools
- Taint Shared Content
- Use Alternate Authentication Material

Collection
15 techniques

- Archive Collected Data
- Audio Capture
- Automated Collection
- Clipboard Data
- Data from Information Repositories
- Data from Local System
- Data from Network Shared Drive
- Data from Removable Media
- Data Staged
- Email Collection
- Input Capture
- Man in the Browser
- Man-in-the-Middle
- Screen Capture
- Video Capture

Command and Control
16 techniques

- Application Layer Protocol
- Communication Through Removable Media
- Data Encoding
- Data Obfuscation
- Dynamic Resolution
- Encrypted Channel
- Fallback Channels
- Ingress Tool Transfer
- Multi-Stage Channels
- Non-Application Layer Protocol
- Non-Standard Port
- Protocol Tunneling
- Proxy
- Remote Access Software
- Traffic Signaling
- Web Service

Exfiltration
8 techniques

- Automated Exfiltration
- Data Transfer Size Limits
- Exfiltration Over Alternative Protocol
- Exfiltration Over C2 Channel
- Exfiltration Over Other Network Medium
- Exfiltration Over Physical Medium
- Exfiltration Over Web Service
- Scheduled Transfer

Impact
13 techniques

- Account Access Removal
- Data Destruction
- Data Encrypted for Impact
- Data Manipulation
- Defacement
- Disk Wipe
- Endpoint Denial of Service
- Firmware Corruption
- Inhibit System Recovery
- Network Denial of Service
- Resource Hijacking
- Service Stop
- System Shutdown/Reboot

Figure 1

APT33 Threat Profile

Reading through the CTI sources above (feel free to read other sources) and Navigator, we can extract the TTPs and create a Threat Profile for APT33:

Tactic	Description
Description	APT33 is a suspected Iranian threat group that has carried out operations since at least 2013. The group has targeted organizations in the United States, Saudi Arabia, and South Korea, in multiple industries including governments, research, chemical, engineering, manufacturing, consulting, finance, telecoms, and several other sectors.
Objective	Establishing persistent access to partners and suppliers of targets. Mounting supply chain attacks
Command and Control	T1043 - Commonly Used Port: Port 80 and 443 T1071 - Standard Application Layer Protocol: HTTP and HTTPS T1032 - Standard Cryptographic Protocol T1065 - Uncommonly Used Port: Ports 808 and 880
Initial Access	T1192 – Spear phishing Link T1110 - Brute Force T1078 - Valid Accounts

Execution	T1204 - User Execution T1203 - Exploitation for Client Execution
Defense Evasion	T1132 - Data Encoding T1480 - Execution Guardrails: Kill dates in payload T1027 - Obfuscated Files or Information T1086 – PowerShell
Discovery	T1040 - Network Sniffing
Privilege Escalation	T1068 - Exploitation for Privilege Escalation
Persistence	T1060 - Registry Run Keys / Startup Folder T1053 - Scheduled Task
Credential Access	T1003 - Credential Dumping: Publicly available tools like Mimikatz
Exfiltration	T1002 - Data Compressed T1048 - Exfiltration Over Alternative Protocol

Adversary Emulation Plan

We have covered how to automate most of these TTPs in previous #ThreatThursday so download the adversary emulation plan from our Community Threat Github and import it to SCYTHE to begin testing. A best practice we encource while using SCYTHE is to avoid automating the escalation of privilege or persistence as every new instance will trigger the automated actions resulting in your escalated shell trying to escalate again or your persistent shell trying to persist again.

This week, we will cover how to emulate PowerShell and how to move laterally through the SCYTHE user interface. Below is an explanation of what is done followed by a video.

PowerShell for Discovery

PowerShell is a very common technique (T1086) used by adversaries and it will be a sub-technique in the new version of MITRE ATT&CK (T1059.001 in beta). PowerShell may be executed through powershell.exe or through unmanaged PowerShell. It may be worth testing both of these methods against your target organization. In SCYTHE, we prefer to use unmanaged PowerShell which can be loaded with: *“loader --load upsh”*. Then to execute any PowerShell command you use “upsh --cmd <PowerShell Command>”. In this case we want to determine the anti-virus that is running and the Domain Controller/DNS Server:

To run the same from powershell.exe (not unmanaged) use the following run commands (as shown in Figure 2):

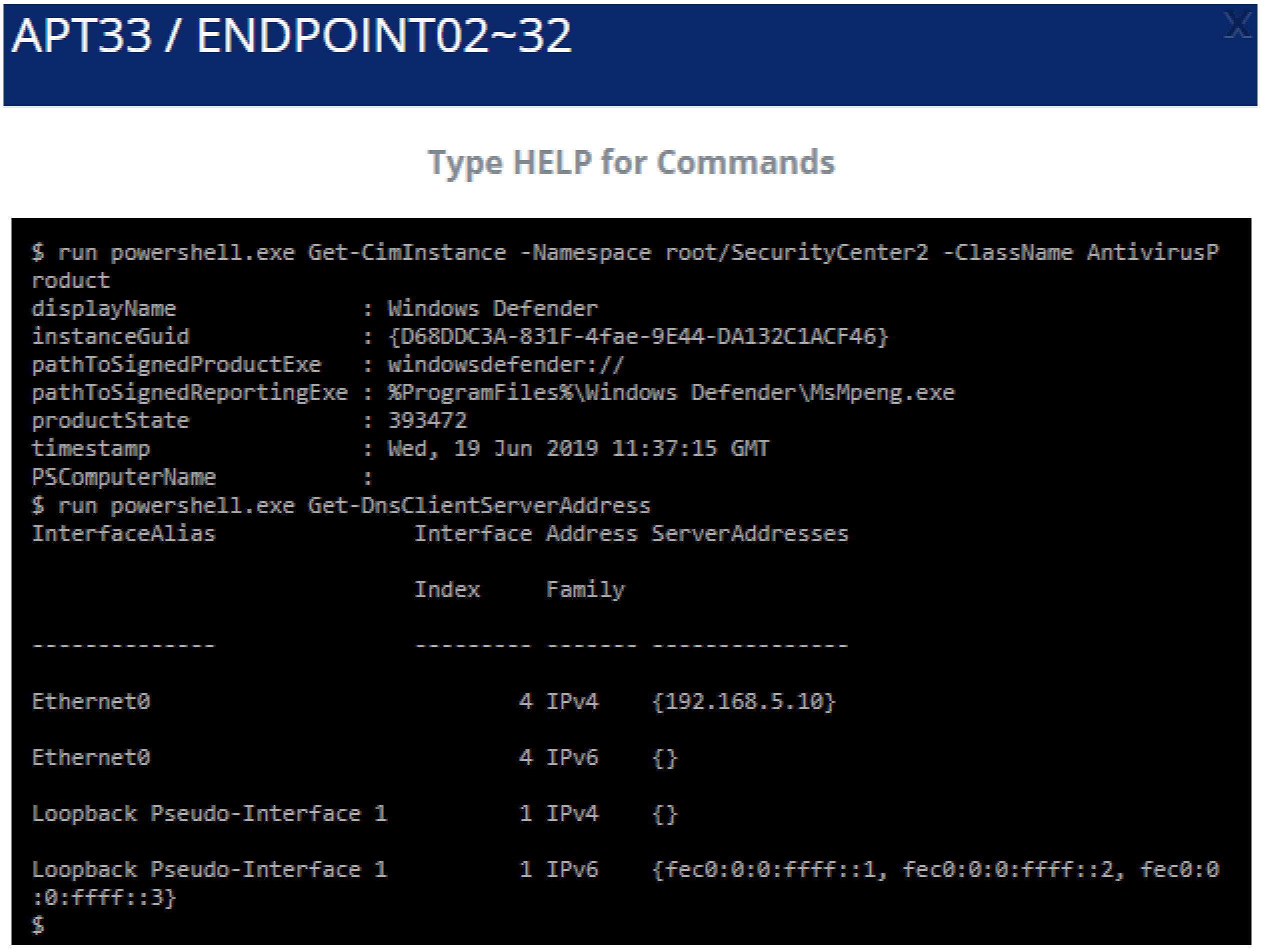
```
upsh --cmd Get-CimInstance -Namespace root/SecurityCenter2 -ClassName AntivirusProduct
```

```
upsh --cmd Get-DnsClientServerAddress
```

To run the same from powershell.exe (not unmanaged) use the following run commands (as shown in Figure 2):

```
run powershell.exe Get-CimInstance -Namespace root/SecurityCenter2 -ClassName AntivirusProduct
```

```
run powershell.exe Get-DnsClientServerAddress
```



```
$ run powershell.exe Get-CimInstance -Namespace root/SecurityCenter2 -ClassName AntivirusP
roduct
displayName              : Windows Defender
instanceGuid             : {D68DDC3A-831F-4fae-9E44-DA132C1ACF46}
pathToSignedProductExe   : windowsdefender://
pathToSignedReportingExe : %ProgramFiles%\Windows Defender\MsMpeng.exe
productState             : 393472
timestamp                : Wed, 19 Jun 2019 11:37:15 GMT
PSComputerName           :
$ run powershell.exe Get-DnsClientServerAddress
InterfaceAlias               Interface Address ServerAddresses

                             Index     Family

--------------               --------- ------- ---------------

Ethernet0                            4 IPv4    {192.168.5.10}

Ethernet0                            4 IPv6    {}

Loopback Pseudo-Interface 1          1 IPv4    {}

Loopback Pseudo-Interface 1          1 IPv6    {fec0:0:0:ffff::1, fec0:0:0:ffff::2, fec0:0
:0:ffff::3}
$
```

Figure 2

Lateral Movement

Now we will show how easy it is to move laterally within the user interface of SCYTHE. From the Campaign window, select “More actions…” - “Observe Campaign”. In the new window, click the system you want to use to move laterally from. Selecting it will show two new blue buttons as shown in Figure 3.

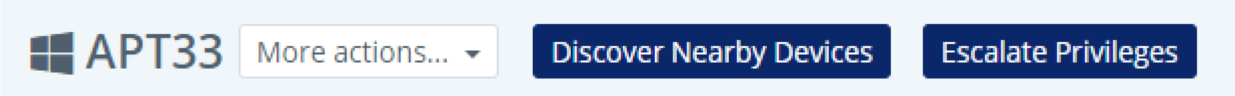


Figure 3

Click “Discover Nearby Devices” and give it a bit to discover and return the results. On the bottom of this screen, you can see the status. Once complete, SCYTHE will show you an image of the nearby devices as seen in Figure 4.

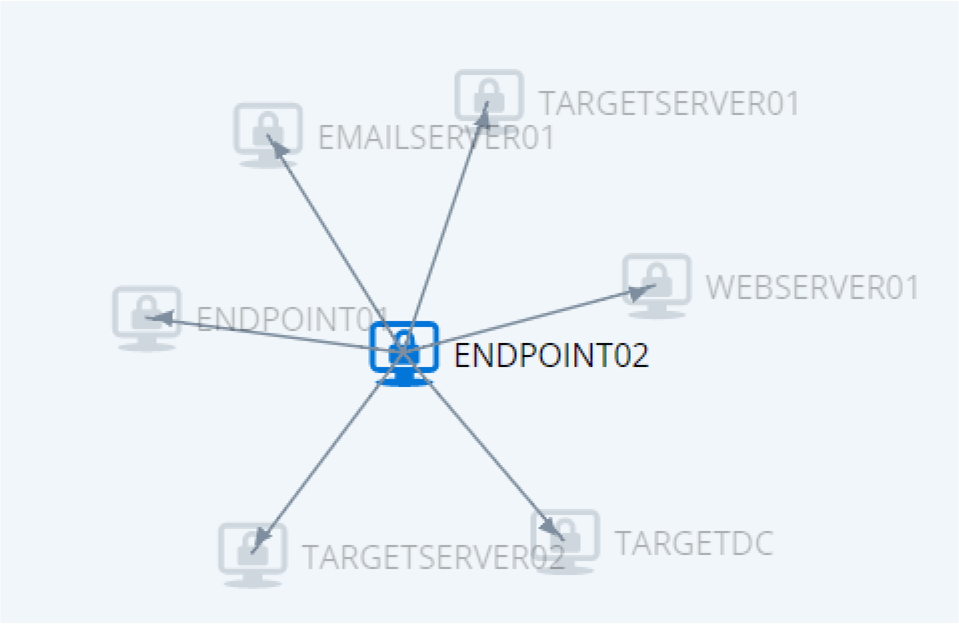


Figure 4

In this case, ENDPOINT02 has an SMB connection open to TARGETSERVER01. We know this because we ran the “run net use” command in the APT33 emulation. Click on TARGETSERVER01 and select the new blue button “Inject into Neighbor”. You will be prompted for credentials but since the SMB session has already been established, you can click “No credentials”. Give it a moment and if successful, you will see the icon turn blue. Go back to the Device List by selecting it under “More actions…”

To understand what just happened, you can click the device that you pivoted from, in our example it is ENDPOINT02. Click on the persist module and take a look at the request:

persist --hostname TARGETSERVER01 --name SCYTHEC --display SCYTHEC --description SCYTHE Client --path \\TARGETSERVER01\c$\windows\temp\scythe.exe

SCYTHE used the SMB connection that was already established and created a new service called SCYTHEC to execute a new file it copied over at C:\windows\temp\scythe.exe. Clicking on the new device will result in the APT33 automation to run on the new device. Take a look at the whoami module that executed on the new device and note the payload is running with NT AUTHORITY\SYSTEM privileges; this is because Windows services run with SYSTEM privilege.

Defend against APT33

Detecting PowerShell when executed from powershell.exe should be a simple test case for defenders. Using sysmon may be the simplest method of detecting when a new process is created. Our friends at Blackhills Information Security have a great post on getting started with sysmon.

We will go into detected unmanaged powershell in another #ThreatThursday but in the meantime, our friends at Optiv have a two part post related to unmanaged powershell: Part 1 and Part 2.

Lets focus on detecting the lateral movement performed through the user interface of SCYTHE. It is important to understand what occurs in the background to defend against it. SCYTHE created a service called SCYTHEC with a command to execute C:\Windows\temp\scythe.exe To defend against this ensure you are monitoring when new services are created on systems, especially where there is no change record or ticket to install or modify a system.

Clean up

Make sure to clean up when complete. To delete the service: open a privileged cmd.exe, type “sc delete SCYTHEC” and press “Enter“. You should also delete C:\Windows\temp\scythe.exe

Conclusion

In this #ThreatThursday we learned about an Iranian threat known as APT33 or Elfin. We used the beta of ATT&CK Navigator with sub-techniques to gather Cyber Threat Intelligence and create a Threat Profile. We imported the adversary emulation plan from the community threats Github. We learned the difference between powershell.exe and unmanaged PowerShell. We moved laterally using the SCYTHE user interface and looked under the hood to understand how it executed the lateral movement. Lastly, we covered methods for detecting some of the new techniques introduced by APT33/Elfin on this #ThreatThursday. We hope you enjoyed it!

About SCYTHE

SCYTHE provides an advanced attack emulation platform for the enterprise and cybersecurity consulting market. The SCYTHE platform enables Red, Blue, and Purple teams to build and emulate real-world adversarial campaigns in a matter of minutes. Customers are in turn enabled to validate the risk posture and exposure of their business and employees and the performance of enterprise security teams and existing security solutions. Based in Arlington, VA, the company is privately held and is funded by Gula Tech Adventures, Paladin Capital, Evolution Equity, and private industry investors. For more information email info@scythe.io, visit https://scythe.io, or follow on Twitter @scythe_io.