Malware Analysis — Cobalt Strike

By 0xMrMagnezi

Published: 2024-02-29 · Archived: 2026-04-05 13:54:07 UTC

Cobalt Strike is a versatile tool for Red Team operations and penetration testing. However, threat actors also use it for malicious activities like establishing covert communication, conducting post-exploitation tasks, moving laterally across networks, crafting and delivering weaponized payloads, and executing social engineering attacks.

Press enter or click to view image in full size

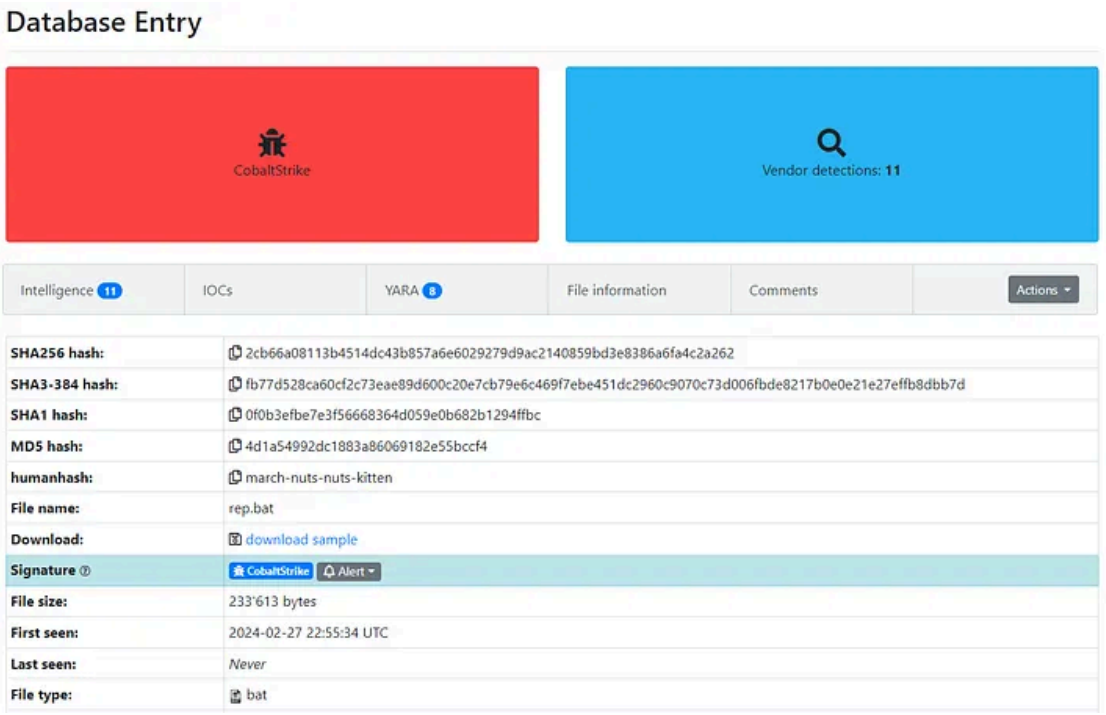


Figure 1: Malware Bazaar sample

After downloading and extracting the zip file, two sections were observed in the BAT file. The first section contained two chunks of Base64-encoded code, as shown in Figure 2. It was suspected that these might be a Base64-encoded file, but decoding them did not yield any results. The second section appeared to utilize a simple replace obfuscation, as shown in Figure 3.

Press enter or click to view image in full size

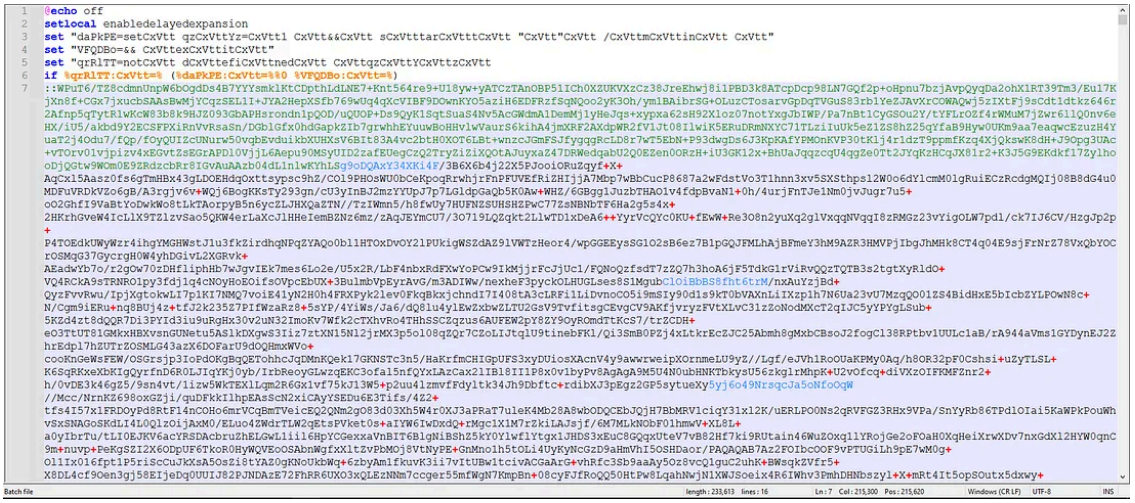


Figure 2: First section of the original BAT file

Press enter or click to view image in full size

Figure 3: Marking what looked like replacing “CxVtt” with NULL

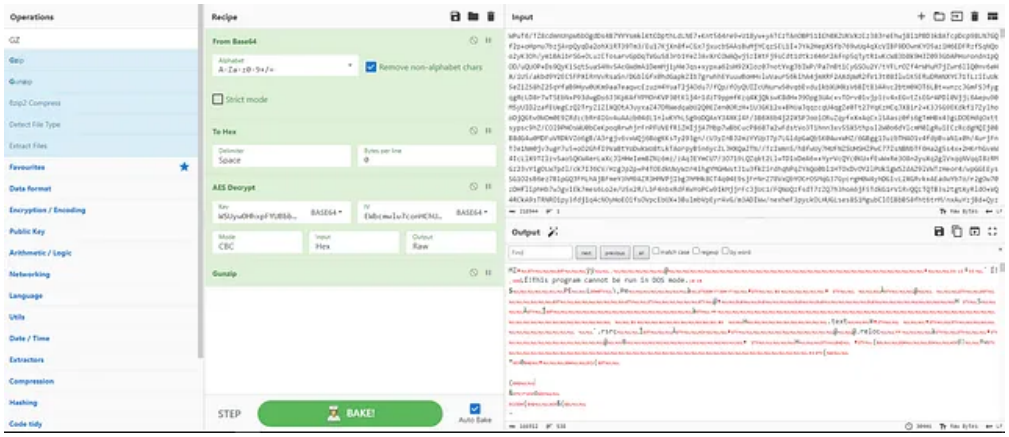
The second section contained additional instances of the replace function. Consequently, the cleaning process was continued using it. This part of the code seemed to decrypt something using AES encryption and decompress it using Gzip, as illustrated in Figure 4.

Press enter or click to view image in full size

Figure 4: After Cleaning of the 2nd part

It was decided to attempt using CyberChef to decrypt what was suspected to be another dropped file. Subsequently, it was confirmed that the suspicion was correct, and the dropped files were successfully extracted, as illustrated in Figure 5.

Press enter or click to view image in full size



Press enter or click to view image in full size

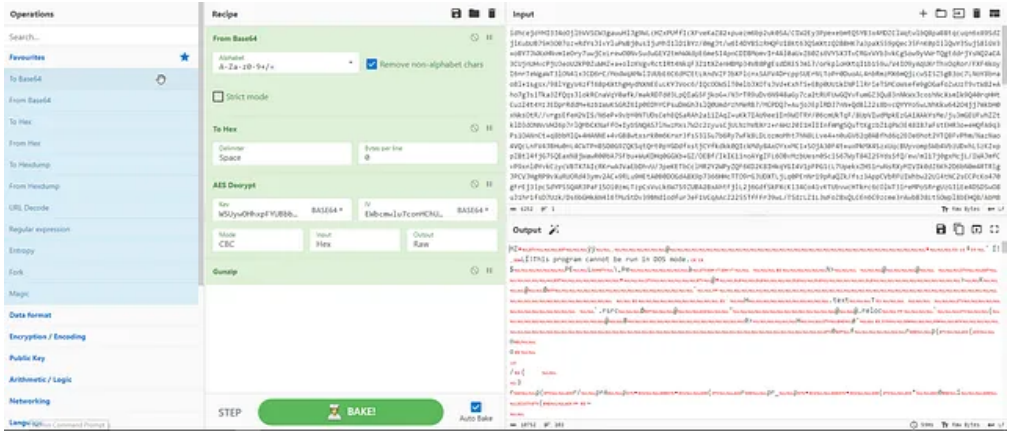


Figure 5: Using the AES Key+IV and Gunzip on the Base64 code , which resulted an EXE file

The extracted files were both written in .NET, allowing for debugging in DNSPY. During debugging, additional memory manipulation was observed, leading to the suspicion that there might be another hidden file.

Press enter or click to view image in full size

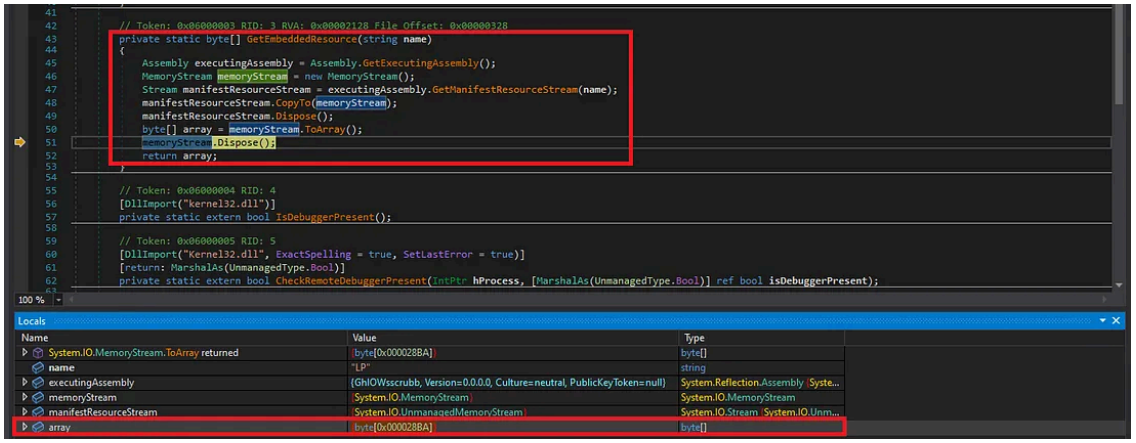


Figure 6: Embedding Resource

Observing the array in memory confirmed my suspicion. It was observed that the memory started with the header “1F 8B,” indicating a Gzip file as shown in Figure 7.

Press enter or click to view image in full size

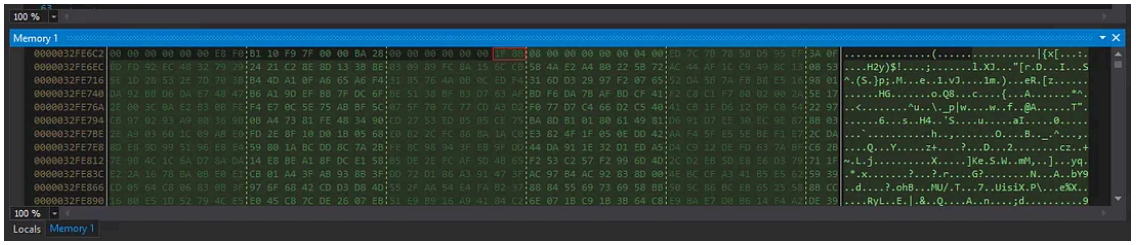


Figure 7: Finding the embedded file inside the memory

For the last time I used CyberChef to Decompress as shown in Figure 8.

Press enter or click to view image in full size

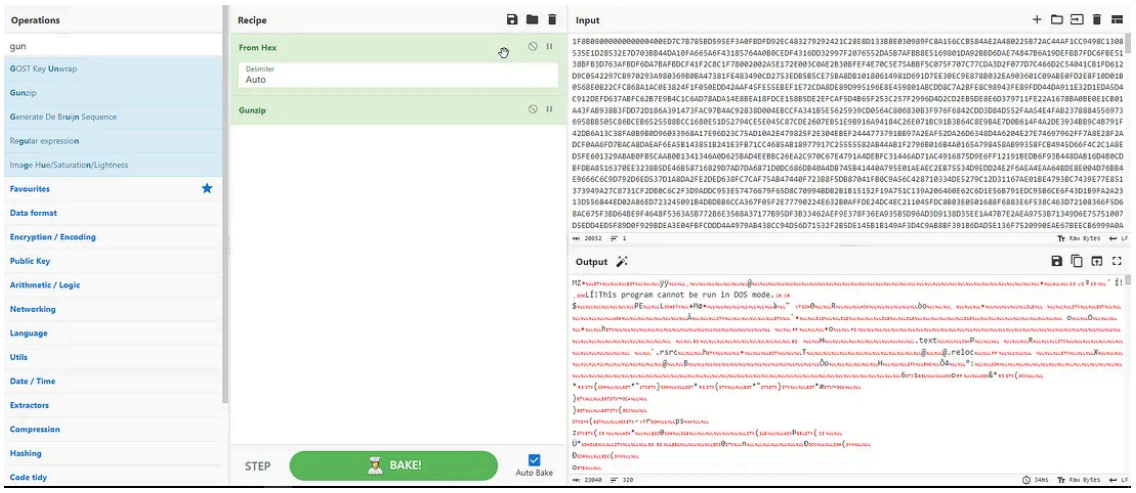


Figure 8: Extracting embedded EXE file

At that point, a decision was made to execute the file and gather some artifacts and IOCs. As shown below, an EXE embedded in the running process was successfully dumped.

Press enter or click to view image in full size

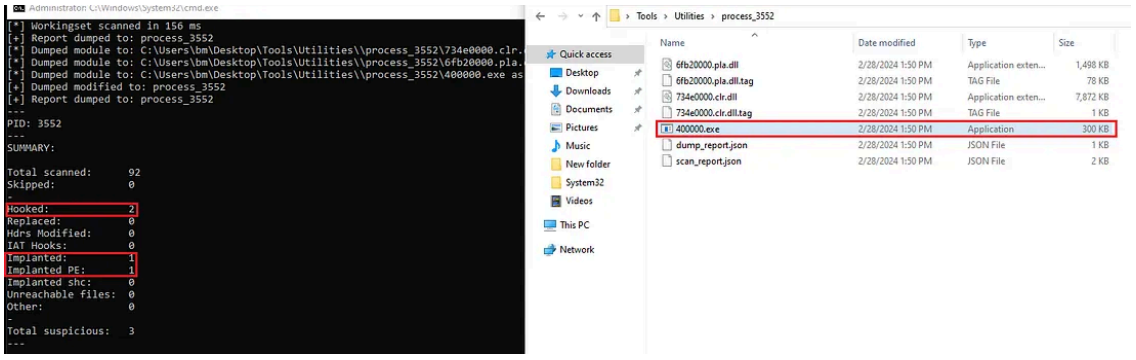


Figure 9: Dumping EXE from running process

The network traffic was encrypted, so the combination of inetsim + Fiddler was used to follow the requests without the risk of being infected.

Inetsim is an open-source tool for simulating internet services like HTTP, DNS, and FTP. It helps observe malware behavior without connecting to real servers, reducing infection risks.

Fiddler is a web debugging proxy tool that logs and modifies HTTP/HTTPS traffic. It decrypts HTTPS traffic with a root certificate.

Using inetsim with Fiddler allows to safely intercept and analyze encrypted network traffic, gaining insights into malware communication and payloads without risking infection.

Press enter or click to view image in full size

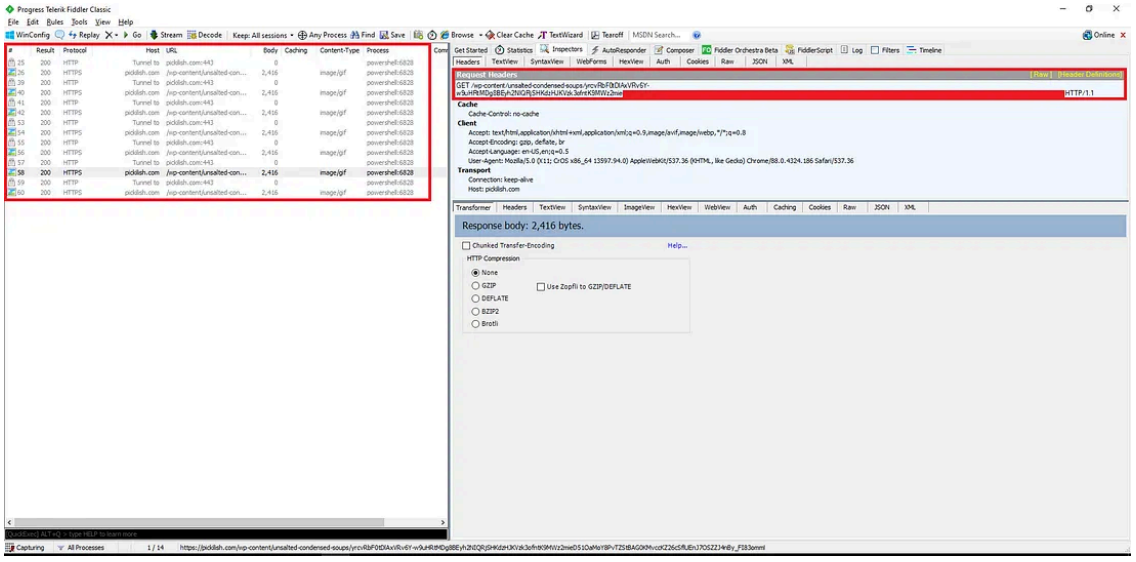


Figure 10: Analyzing network traffic using the combination of Fiddler + inetsim

IOCs:

- origin.bat — 4d1a54992dc1883a86069182e55bccf4
- out1.exe — c58f43348436a19ca37a676b477a137f
- out2.exe — 8d8fe14374cb94fe10070d9591fea3bb
- 4000.exe — 30d2256f99c9dc5e6846838f655fae34
- pickilish[.]com

In conclusion, a sample linked to Cobalt Strike was dissected. The process involved decoding Base64-encoded sections and unraveling obfuscation. AES encryption and Gzip compression were used to conceal and deploy malicious payloads. Tools like CyberChef and DNSPY were instrumental in extracting and examining the dropped files. Further investigation uncovered hidden files embedded in memory confirming the sophisticated nature of the malware. This comprehensive analysis demonstrates the complexity and stealth of threats associated with Cobalt

Strike, emphasizing the importance of robust cybersecurity measures to detect and mitigate such threats effectively.
