Water Basilisk Uses New HCrypt Variant to Flood Victims with RAT Payloads

Published: 2021-09-20 · Archived: 2026-04-05 13:12:59 UTC

In this blog entry we look into a fileless campaign that used a new HCrypt variant to distribute numerous remote access trojans (RATs) in victim systems. This new variant also uses an updated obfuscation mechanism which we detail.

By: Aliakbar Zahravi, William Gamazo Sanchez Sep 20, 2021 Read time: 5 min (1415 words)

We encountered a fileless campaign that used a new HCrypt variant to distribute numerous remote access trojans (RATs) in victim systems. This new variant uses a newer obfuscation mechanism compared to what has been observed in past reports. It reached the peak of activity in the middle of August 2021.

HCrypt is a crypter and multistage generator that is considered difficult to detect. It is identifiedopen on a new tab as a crypter-as-a-service, paid for by threat actors to load a RAT (or in this case RATs) of their choosing. The campaign also showed new obfuscation techniques and attack vectors, different from those that were observed in the past.

Overview of the Water Basilisk campaign

In this campaign, which we have labelled Water Basilisk, the attacker mostly used publicly available file hosting services such as “archive.org”, “transfer.sh”, and "discord.com", to host the malware while hacked WordPress websites were used to host phishing kits.

The malicious file is hidden as an ISO that is distributed through a phishing email or website. This file contains an obfuscated VBScript stager responsible for downloading and executing the next stage of the VBScript content onto the infected system memory.

The final stage is an obfuscated PowerShell script that contains the payloads and is responsible for deobfuscating and injecting them into the assigned process. In some cases, the final stage PowerShell script contained up to seven various RATs. These are typically NjRat, BitRat, Nanocore RAT, QuasarRat, LimeRat, and Warzone.

HCrypt version 7.8

In a nutshell, Water Basilisk’s attack chain is a combination of the VBScript and PowerShell commands. HCrypt creates various obfuscated VBScripts and PowerShell to deliver or inject the final payload into a given process in a victim system. The latest version of this crypter is 7.8, based on what we have seen in its builder and website.

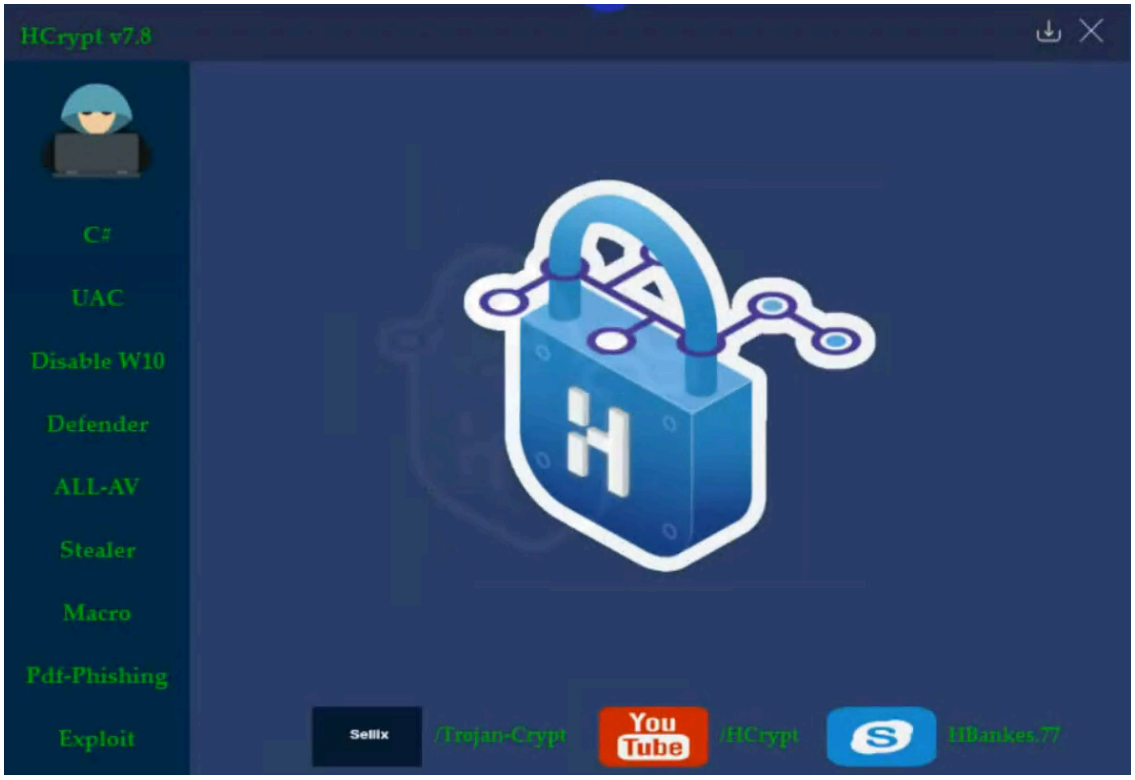

Figure 1. The HCrypt v7.8 builder

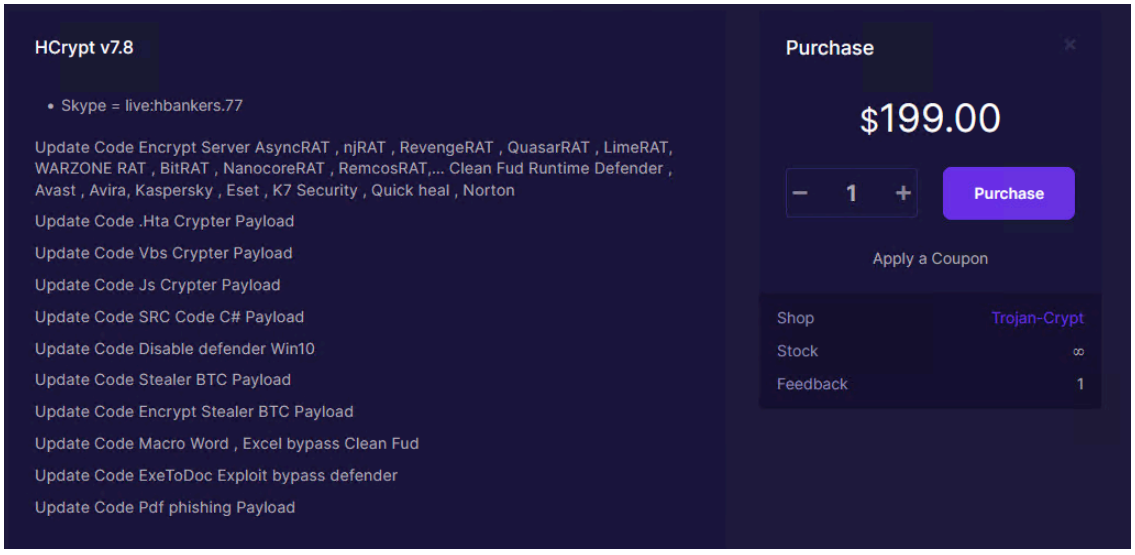

Figure 2. HCrypt v7.8 updates that also list RAT variants and the purchase price

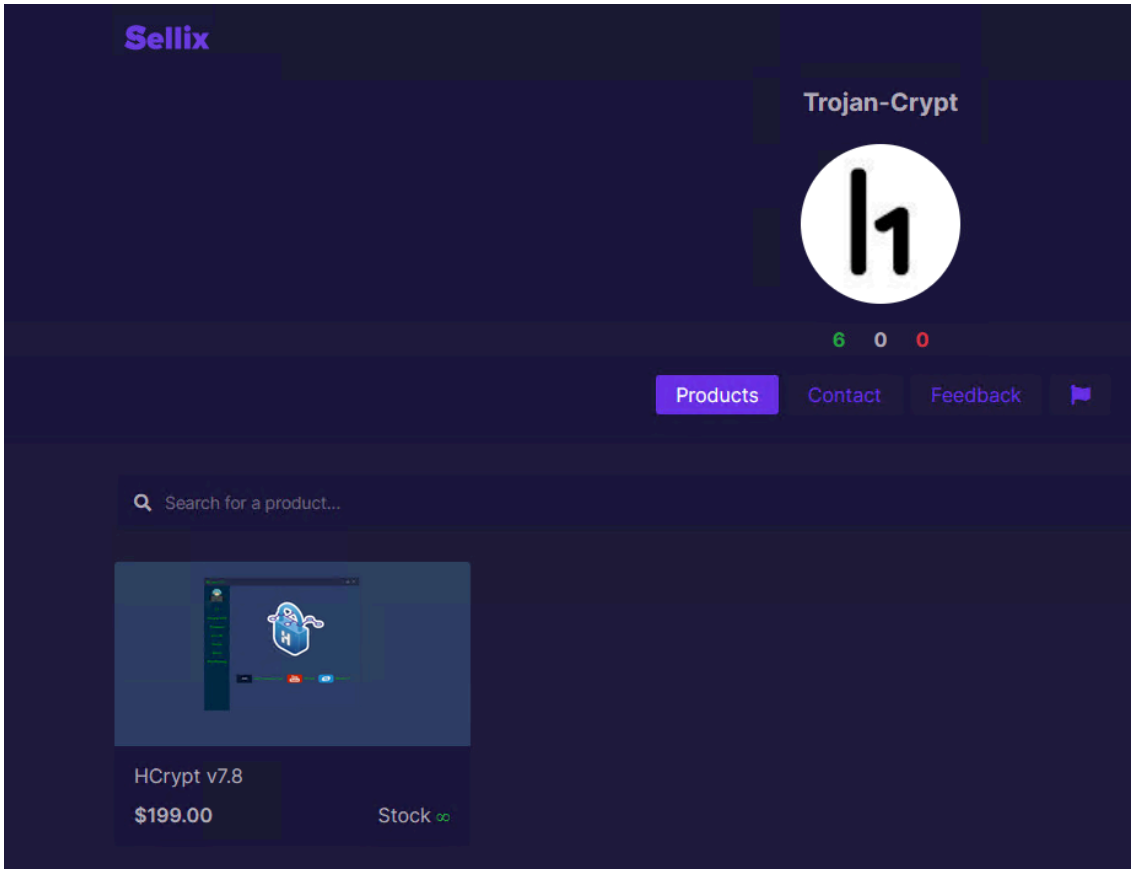

Figure 3. HCrypt v7.8 on Sellix

As can be seen in Figures 1 to 3, HCrypt 7.8 is being sold for US$199. Figure 2 also lists, as part of an update, the various RATs that can be loaded using this variant that we mentioned earlier.

Attack analysis

This section discusses how this version works. Figure 4 summarizes Water Basilisk. The infection chain goes as follows:

- A phishing email or website tricks a user into downloading and executing the malicious ISO file that contains the initial VBScript stager
- The initial VBScript downloads and executes the next stage VBScript content via a PowerShell command in memory
- The downloaded VBScript would be responsible for achieving persistence on the victim system and downloads and executes the final stage via a PowerShell command in memory
- The final stage PowerShell is responsible for deobfuscating and injecting the payload (RATs) into the given process

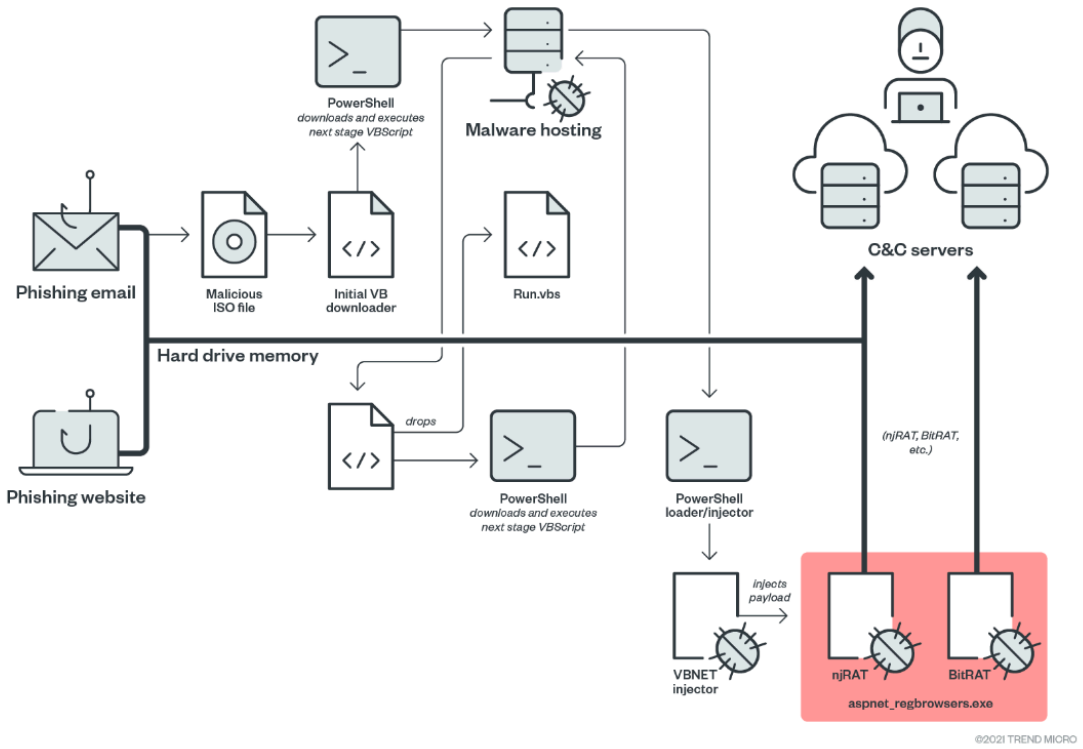

Figure 4. An overview of the attack

This campaign uses two different attack vectors: phishing websites and emails. Both have the same infection chain, which we have already described. The attack begins with the malicious ISO image file.

We can assume two reasons why this attack uses ISO files. One is how ISO images tend to have larger file sizes, making it so that email gateway scanners would not be able to scan ISO file attachments properly. Another is how opening an ISO file in new operating systems is as simple as double-clicking the file, due to native IOS mounting tools. This improves the chances of a victim opening the file and infecting their system.

As we have also mentioned, and as seen in Figure 4, an interesting aspect of this attack is how HCrypt developers host stager scripts were hosted from public file hosting services such as Transfer.sh and Internet Archive (archive.org). Once the ISO file is opened the needed scripts are downloaded from this hosting archive. Figure 5 is an example of the archive.org account used to host scripts.

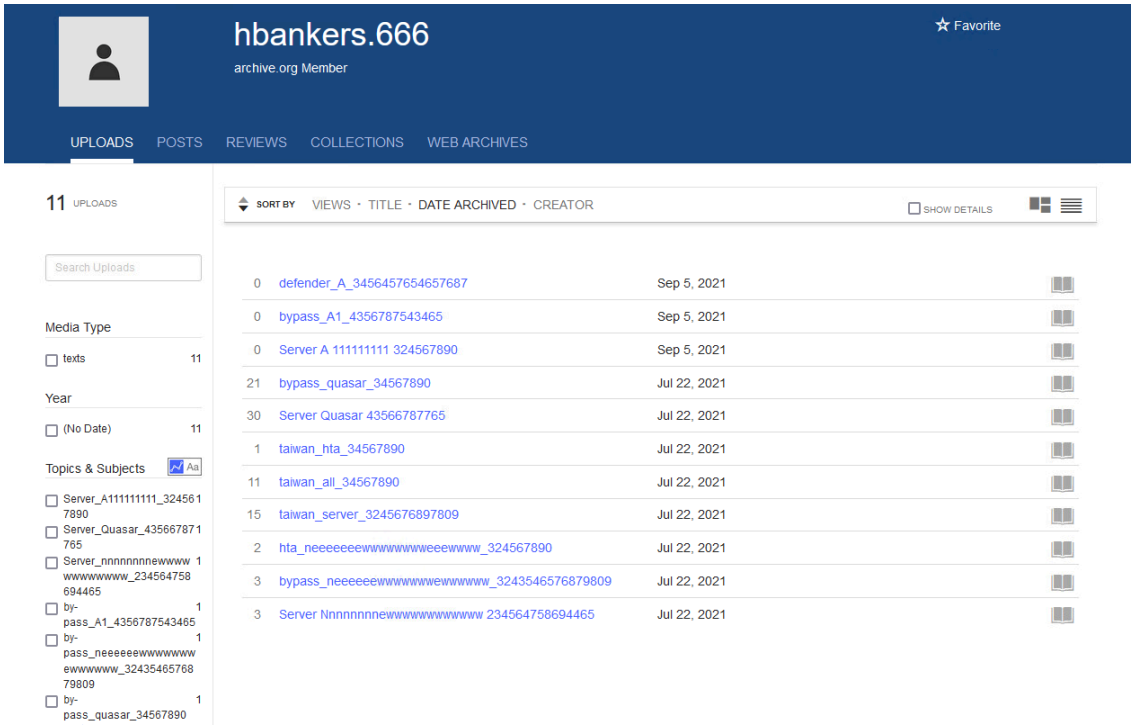

Figure 5. The archive.org account hosting the loader's scripts

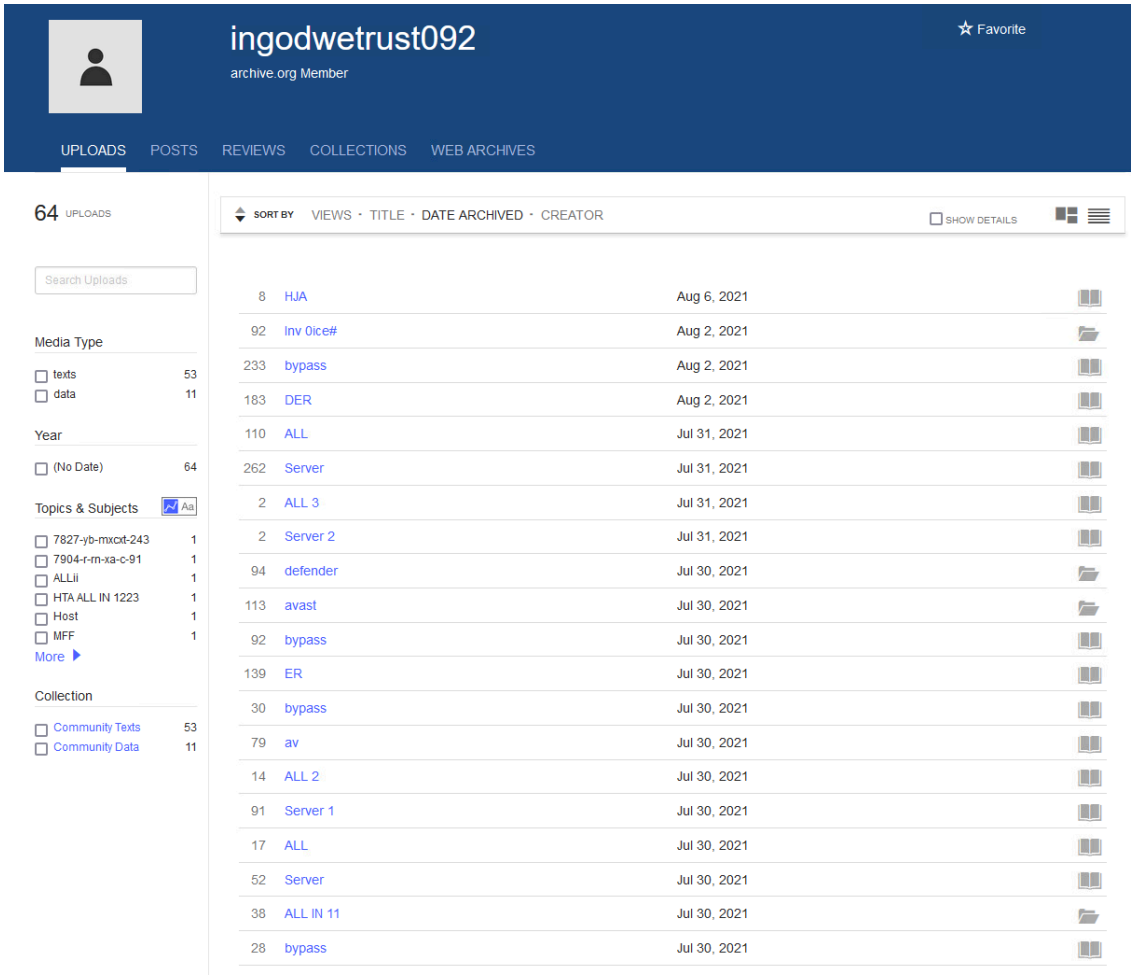

Figure 6. The archive.org account hosting the loader's scripts

Figure 7 shows an example of the hacked WordPress website that hosts a phishing kit that downloads the "Spectrum Bill.iso" file. Figure 8 shows the malicious content added by the attacker in the said website.

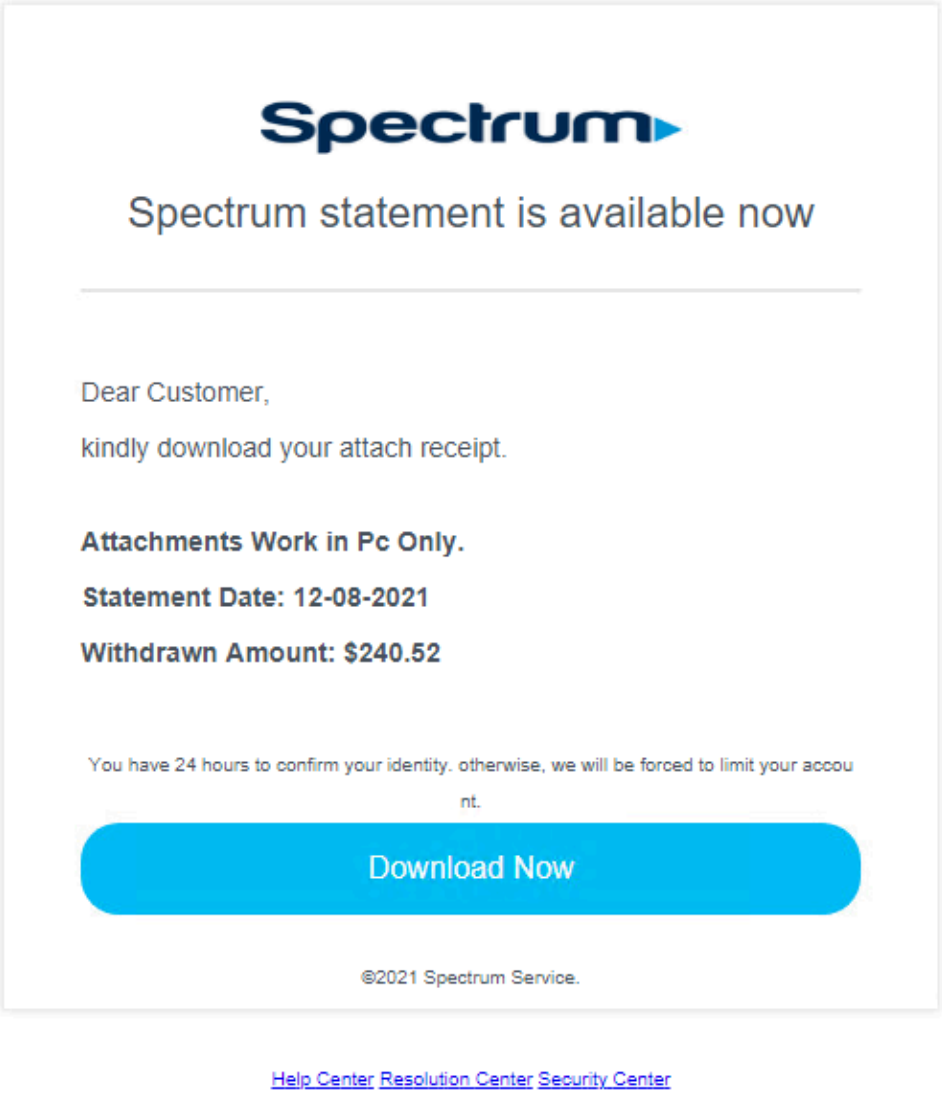

Figure 7. The phishing website used in this campaign

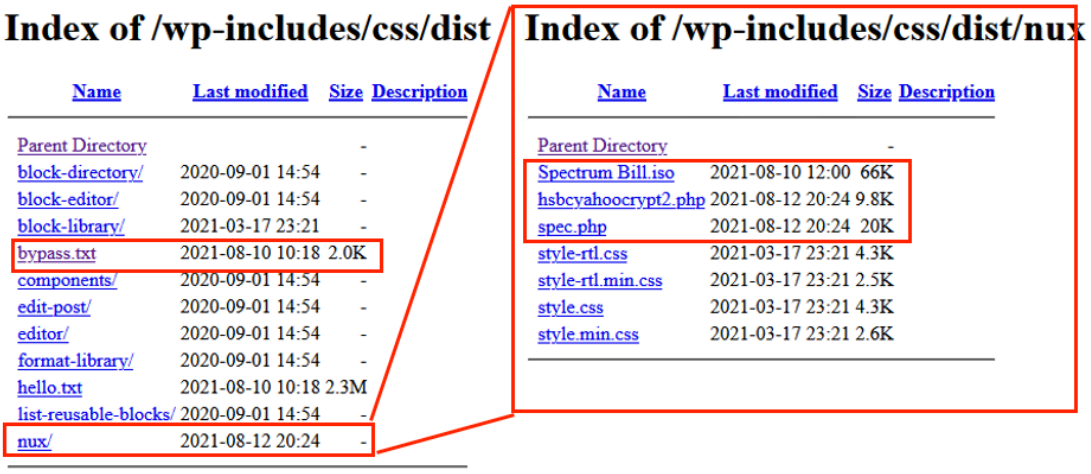

Figure 8. Malicious content uploaded by the attacker

The “Spectrum Bill.iso” file contains an HCrypt obfuscated VBScript stager that is responsible for downloading and executing the next stage via a PowerShell command. We note here that, with the exception of this second stage for persistence, all scripts, PowerShell, and binaries are fileless and execute in memory.

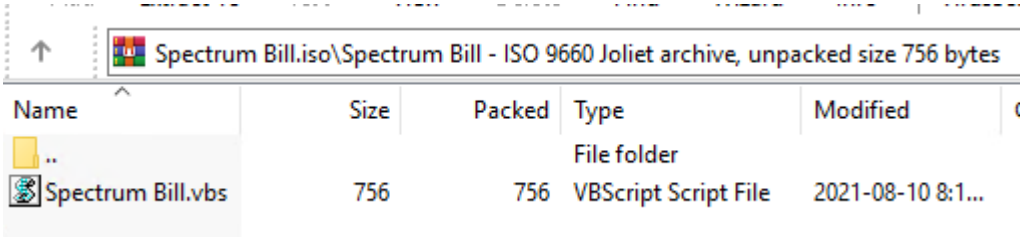

Figure 9. “Spectrum Bill.iso” content

Figure 10. "Spectrum Bill.vbs" content and cleanup code

The downloaded content in memory, “bx25.txt,” is another obfuscated HCrypt VBScript. As mentioned, this code is for achieving persistence and is the only one not executed in memory. It achieves persistence by creating the file C:\Users\Public\Run\Run.vbs, adding it to the Startup path, and downloading and executing the final stage in memory.

Each time an infected computer starts, the malware downloads the latest payload(s) from the given URL. The attacker can therefore change the final payload(s) and its command and control (C&C) server easily, reducing their fingerprints on an infected system.

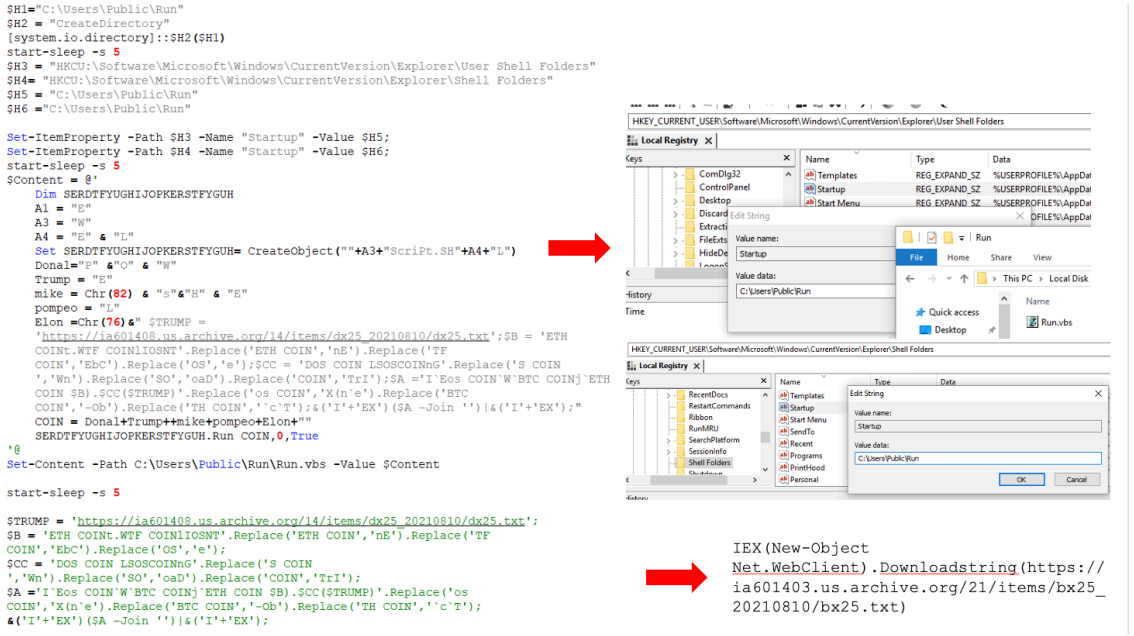

Figure 11. The cleaned code of bx.25, the second VBScript stage for persistenc

Run.vbs (“dx25.txt”) is the final stage PowerShell that contains the final payload(s). This executes on an infected system memory and its responsible for deobfuscating, loading, and injecting payload(s) into the given hardcoded legitimate process. In some cases, the malware loads up to seven RATs on an infected system. The snippet in Figure 12 demonstrates this behaviour of the malware.

Figure 12. The code of the file dx25.txt, the PowerShell loader

Among the loaded binaries is a DLL injector called “VBNET,” which reflectively loads a .NET PE payload in a selected .NET legitimate process. In Figure 12, $HH1 is a VBNET PE injector DLL and $HH5 contains a PowerShell command to pass a final malware payload ($HH3) into the given process, which is “aspnet_regbrowsers.exe.”

To automate the final payload extraction we developed a Python script to deobfuscated and extract the payloads from the final PowerShell stage which simply accept a directory where an obfuscated PowerShell script are stored and output directory where the extracted payload will be stored. The Python script can be viewed here.

Bitcoin and Ethereum Hijacker

We were also able to observe Bitcoin/Ethereum address hijacker binaries among the loaded RATs in an infected system. These binaries search the victim's clipboard content for Bitcoin and Ethereum addresses using regex, then replaces them with the attacker's own address. Figure 13 shows where the binary can be generated in the HCrypt interface.

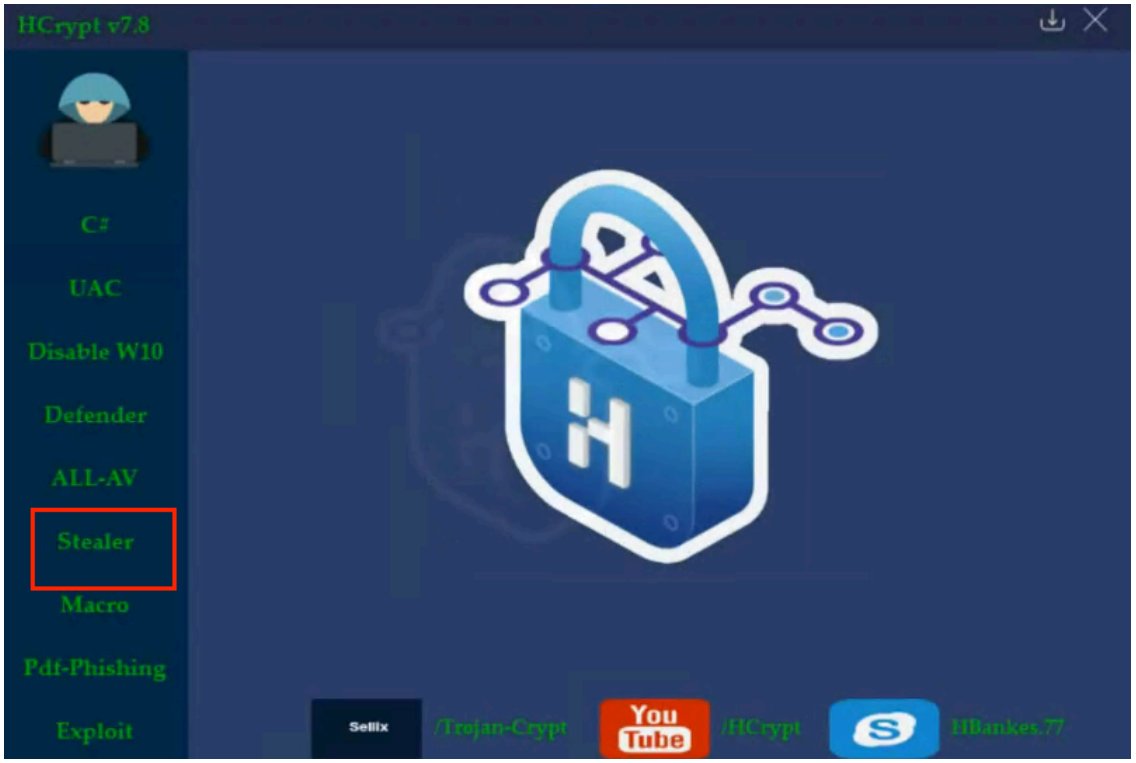

Figure 13. HCrypt builder interface showing where to start generating the hijacker binaries

By default, the HCrypt stealer builder shows built-in Ethereum and Bitcoin addresses, likely belonging to the malware's author.

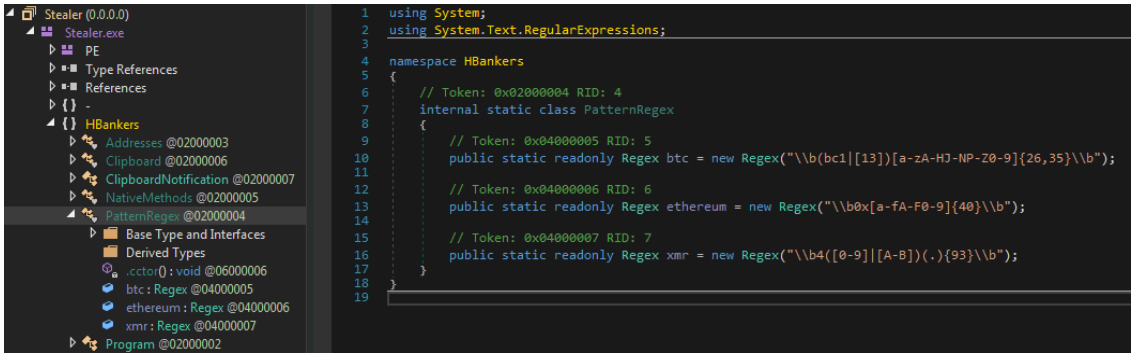

Figure 14. Built-in Ethereum and Bitcoin addresses, potentially belonging to the author(s), seen here as "HBankers"

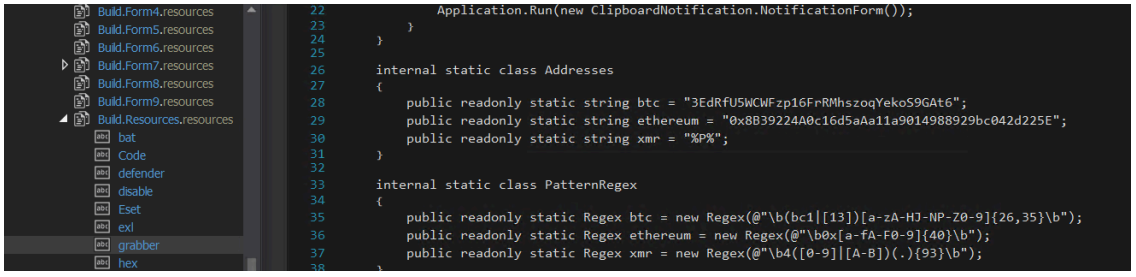

Figure 15. Using regex to search for Bitcoin and Ethereum addresses in the victim's clipboard content

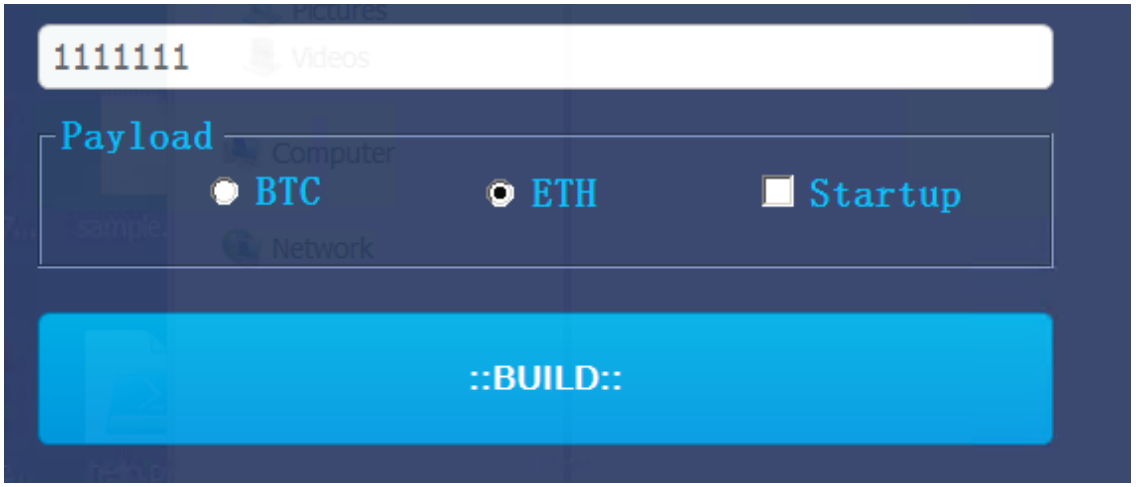

Figure 16. The HCrypt builder where the user (attacker) can only choose either Bitcoin or Ethereum

The stealer builder will only accept one option, either Bitcoin or Ethereum, from a user. As shown in the example in Figure 16, in such a scenario the crypto address hijacker will replace the victim's Ethereum address with "1111111," generate the payload, and replace the bitcoin address with the HCrypt builder author's (HBankers) address. Overall, this shows the HCrypt's developers' attempt to also make a profit from attacks that use this loader.

Conclusion

This case shows how cybercriminals can take an advantage of crypter tools, such as HCrypt, to dynamically distribute malware. HCrypt also shows signs of undergoing active development. It would be best to anticipate newer versions to cover more RAT variants and an updated obfuscation algorithm to reduce the chances of detection.

Organizations should also remain vigilant against phishing tactics that remain a staple in cyberattacks. Users should be wary of opening ISO files, especially from suspicious sources, as threat actors have used image files in their campaigns before. They are too easy to open and can bypass email gateway scanners, giving users less chances to consider whether the file is malicious.

Organizations can also consider security solutions that provide a multilayered defense systemproducts that helps in detecting, scanning, and blocking malicious URLs.

The indicators of compromise (IOCs) can be found here.

Tags

Source: https://www.trendmicro.com/en_us/research/21/i/Water-Basilisk-Uses-New-HCrypt-Variant-to-Flood-Victims-with-RAT-Payloads.html