Hive Attacks | Analysis of the Human-Operated Ransomware Targeting Healthcare

labs.sentinelone.com/hive-attacks-analysis-of-the-human-operated-ransomware-targeting-healthcare/

Jim Walter

By Jim Walter & Juan Andres Guerrero-Saade

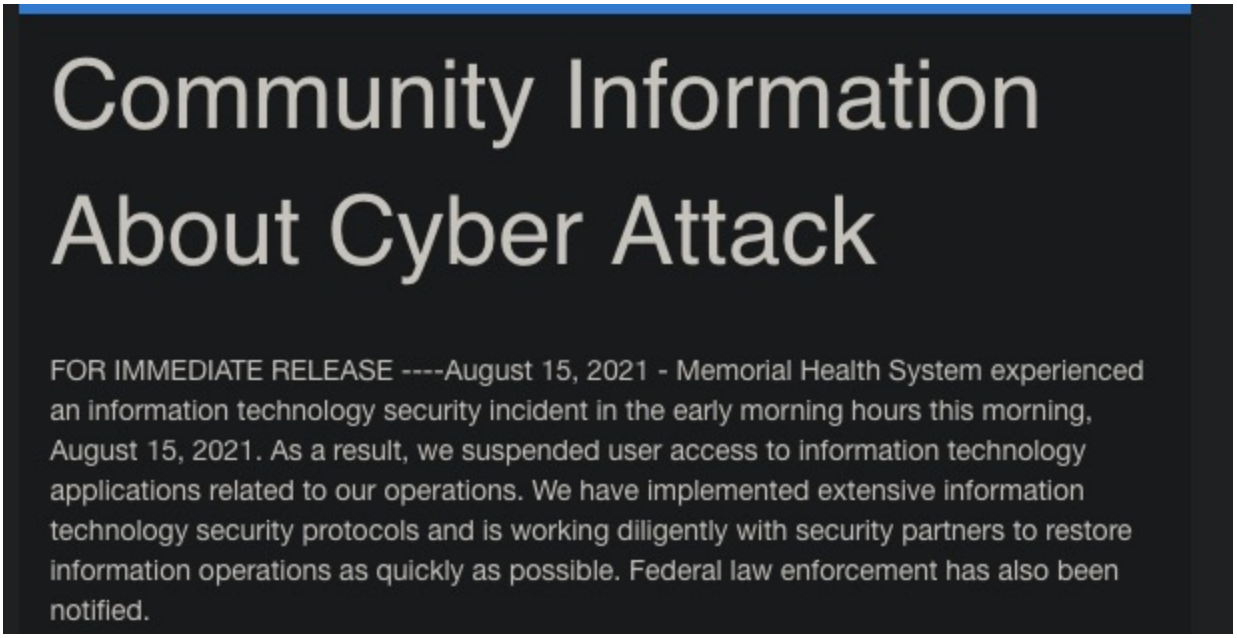
Executive Summary

- Hive is a double-extortion ransomware group that first appeared in June 2021.
- The group is notable in its undiscerning choice of targets, having no limits when it comes to healthcare providers and hospitals, as evidenced in a recent attack on Memorial Health System hospitals in Ohio.
- Hive ransomware is written in Go to take advantage of the language's concurrency features to encrypt files faster.
- This report offers an overview of Hive TTPs as well as a reverse engineering deep dive into the ransomware payloads.
- Hive remains active with as many as 30 victim companies listed on its Hive Leaks onion site at the time of writing.

Background

While many active ransomware groups have committed to forgoing attacks on medical targets in deference to the current global situation, Hive is not one of them. On August 15, 2021, news broke of a Hive campaign against Memorial Health System, an Ohio healthcare provider. As a result, the hospital was forced to advise some patients to seek treatment at separate facilities.

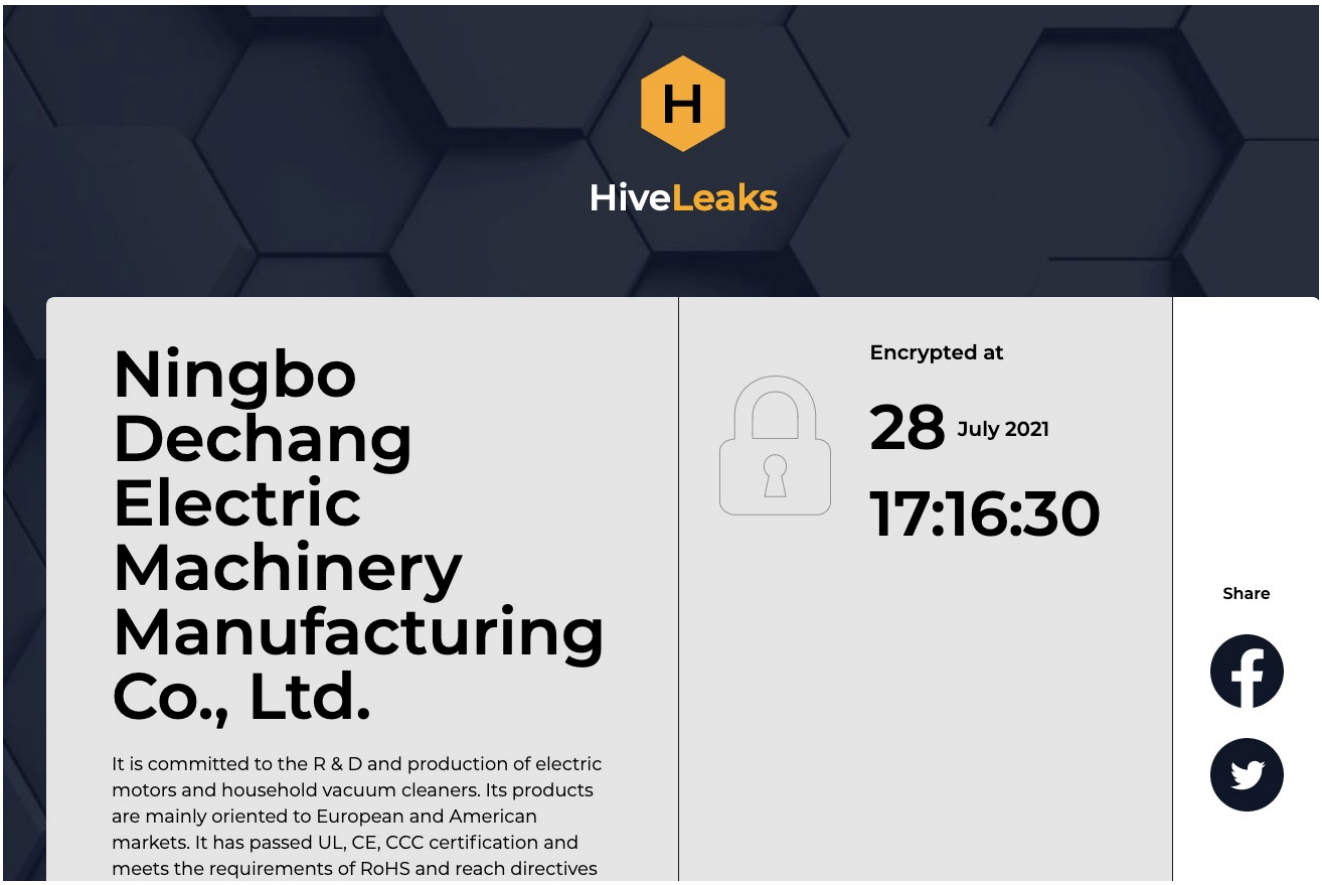
While some ransomware attacks hitting public health and critical infrastructure targets can be the result of a shotgun approach to targetting – mass phishing campaigns that execute malware blindly on victim devices without awareness of the victim environment – that is not the case with Hive. This is a human-operated ransomware attack designed to take input from the command line, indicating the attackers are both aware of the environment and tailoring their attacks for maximum impact.



Memorial Health Systems open statement on ransomware attack

Who is Hive?

Hive or “HiveLeaks” is a relatively new ransomware outfit that made its appearance on the scene in late June, 2021. Hive is yet another double extortion group, making their money off of a two-pronged attack: exfiltrating sensitive data before locking up the victims’ systems. This allows them to pressure the victim into paying greater sums than a conventional ransomware attack as they also face the threat of a mass leak of sensitive data. Hive’s schemes have proven successful so far as multiple leaks are currently posted on their victim blog. As of the time of writing, there are 30 companies currently named on the HiveLeaks site.



HiveLeaks site showing the timer before releasing victim files

We can't put the toothpaste back in the tube for Memorial Health Systems, but we can at least contribute a breakdown of the Hive operators' preferred techniques and a deep dive into their ransomware toolkit to help other potential victims.

Technical Analysis

Initial acces can vary. Cobalt Strike implants are most often the tool of choice. They are delivered via phishing or emails in order to establish initial access. These beacons maintain persistence and allow the operators to expand their reach within the compromised environment. They are also used to launch the Hive payloads.

Recent campaigns opt for the use of ConnectWise. ConnectWise is a legitimate commercial remote administration tool that has been abused by multiple ransomware operators in recent years. This allows for persistence and management of their malware in environments where Cobalt Strike hasn't been successful.

Once inside, attackers will attempt to dump credentials by way of `consvcs.dll` (MinDump) though `rundll32.exe`:

```
Windowssystem32cmd.exe /C rundll32.exe
WindowsSystem32comsvcs.dll MinDump 752 lsass.dmp full
```

Additionally, `WDigest` may be manipulated to allow for the caching of cleartext credential data:

```
Windowssystem32cmd.exe /C reg add
HKLMSYSTEMCurrentControlSetControlSecurityProvidersWDigest /v
UseLogonCredential /t REG_DWORD /d 1 && gpupdate /force
```

Additional tools like ADRecon may be used to further understand and traverse the compromised Active Directory (AD) environment. ADRecon is an open-source tool designed to do just that– to map, traverse and enumerate an AD environment.

The Hive Payload

While the tools, techniques, and procedures mentioned above are fairly standard for ransomware groups these days, Hive utilizes their own closed-source ransomware. The payloads are written in Go and packed with UPX. After unpacking, the ransomware itself is over 2MB in size owing to the way Go packages statically-link all dependencies to create a reliably portable executable.

The developers are taking advantage of some of the native benefits of Go, particularly the ability to implement easy and reliable concurrency. On the other hand, Go is known for enabling easy cross-compilation across different operating systems but the manner in which Hive implements its functionality makes it Windows-specific, at this time.

The ransomware is designed to take input from the command line, indicating that it's meant to be run directly by an operator or a script containing the desired parameters. The available flags are as follows.

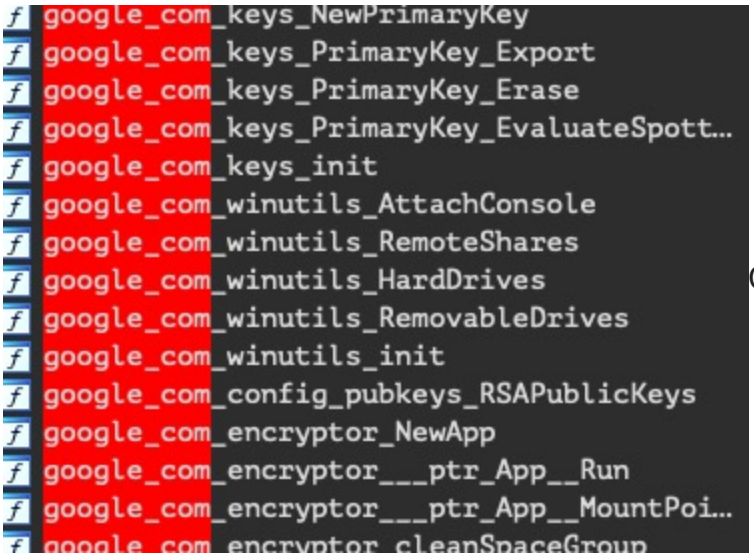
Flag	Type	Functionality
t	Int	Number of threads to run in parallel
stop	String	Regex for services to stop
kill	String	Regex for process to kill (case insensitive)
skip	String	Regex for filenames to skip (case insensitive)
no-cleanpollDesc	Bool	Skip clean disk space stage

Flags used by Hive Ransomware

These flags are largely self-explanatory with the exception of the final option, *no-cleanpollDesc*. This refers to a final phase in the ransomware's functionality that looks for a file named `swap.tmp` in all logical drives and deletes it before the ransomware exits. The

developers refer to this as ‘cleaning space’. At this time we don’t know what this file does, whether it’s a component generated during their operations, a native Windows file, or perhaps a reference to incomplete cross-platform functionality intended for future builds.

Go malware is usually considered difficult to reverse engineer, primarily due to the wealth of tangentially-related imported code baked into every executable. It’s important to isolate the code contributed by the malware developers. In this case, Hive devs contributed four packages orchestrated by the `main()` function: ***encryptor***, ***keys***, ***winutils***, and ***config***.



Custom packages under ‘google.com’ parent directory

Cursory examination might miss these as they’re housed under a parent package named ***google.com***, perhaps to give the appearance that these are standard packages.

The main function parses the flags provided by the operator and before initializing the ransomware functionality under `encryptor.NewApp()`. First it generates and exports the encryption keys and generates the ransom note. It directs the victim to a password-protected Onion domain:

```
http://hivecust6vhekztbqgdnkks64ucehqacge3dij3gyrrpdp57zoq3ooqd[.]onion/
```

It also warns the victim of the impending disclosure of their stolen data at the Hive Leaks site:

```
http://hiveleakdbtnp76ulyhi52eag6c6tyc<redacted>.onion/
```

The main functionally is housed under `encryptor.(*App).Run()`, which does the following:

1. `App.ExportKeys()` wraps around standard go ***crypto*** functions, which it uses to generate RSA keys. A key file is exported.
2. `MountPoints()` enumerates different types of drives and appends them to a slice (a dynamically-sized array in Go). This includes native logical drives, removable drives, and remote shares.

3. Based on the *kill* flag, the malware proceeds to kill processes matching the regex provided. If no custom value is provided, the following default is used:

```
"bmr|sql|oracle|postgres|redis|vss|backup|sstp"
```

4. Based on the *stop* flag, the malware connects to the Windows service control manager and proceeds to stop services matching the regex provided.
5. The malware creates a batch file to self-delete with the filename `hive.bat`, removing its own components from the disk via a new process.

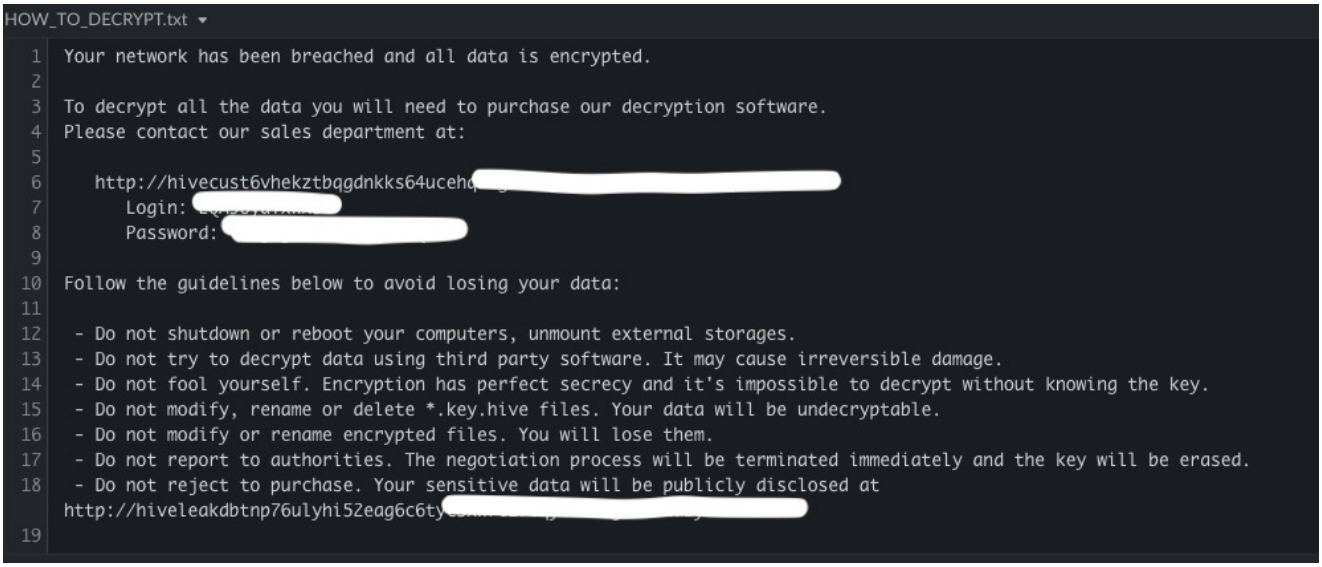
```
timeout 1 || sleep 1
del "C:Usersadmin1Desktophmod4.exe"
if exist "C:Usersadmin1Desktophmod4.exe" goto Repeat
del "hive.bat"
```

6. It creates a batch file to delete shadow copies under the filename `shadow.bat` and executes it as a separate process.

```
vssadmin.exe delete shadows /all /quiet
del shadow.bat
```

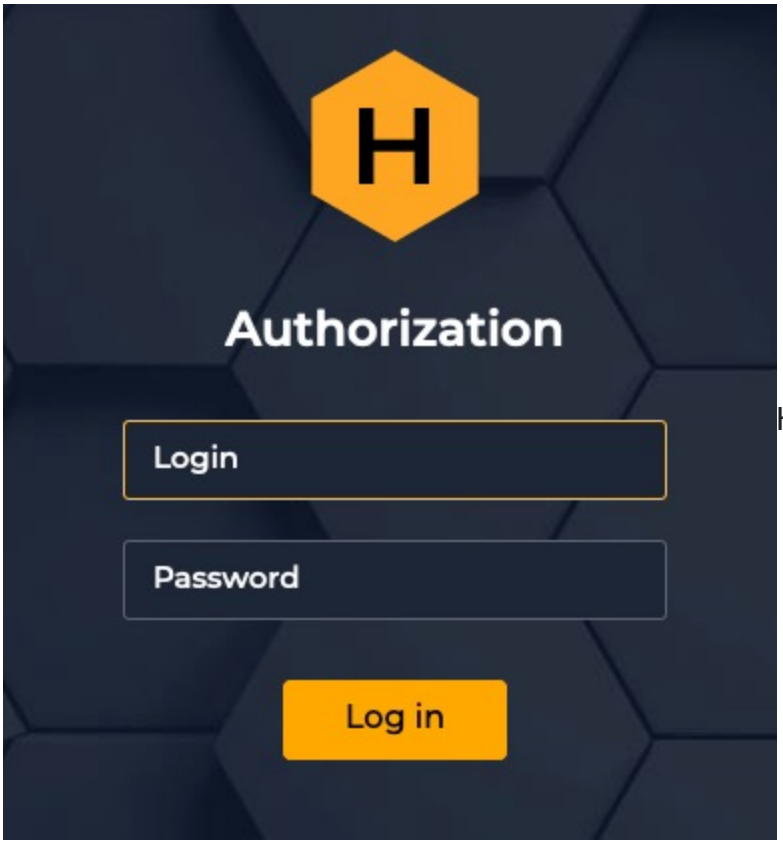
7. In order to take advantage of Go's concurrency features, the Hive devs run a `Notify()` function that is meant to watch the `WaitGroup` that keeps track of the parallel threads. As long as there are threads pending, this function will keep the program running.
8. Now onto the real business of ransomware. `ScanFiles()` will populate a list of absolute filepaths fed into a channel (a queue of sorts). `EncryptFiles()` will then spawn threads that each take a file from that queue and encrypt it. This concurrency feature is the main advantage of writing this ransomware in Go and allows for much faster file encryption.
9. Finally, the devs make sure to erase the encryption key from memory.

Ransom notes are deposited into each folder containing encrypted files (skipping the `C:windows`) directory.



The ‘HOW_TO_DECRYPT.TXT’ ransom note

The ransom note instructs victims to visit the Hive portal via TOR and login with their assigned unique ID to continue the payment process.



Hive Victim Portal

Each infection campaign is assigned unique credentials available in the ransom note. This portal leads the victim to the standard ransomware ‘support’ area where they can upload freebie test files, communicate with their attackers, and receive their decryptor should they choose to pay (which, in an ideal world, they shouldn’t).

Watch Video At:

https://youtu.be/MEi0QWuyLtw

Conclusion

As these attacks continue to escalate and become more egregious, the need for true attack 'prevention' is all the more critical. While well-maintained and tested backup strategies are a must, they are not enough in these double-extortion cases.

Once executed, most modern ransomware will go after backup and storage volumes in fairly smart ways. Many have even evolved to target specific NAS devices and platforms. Some groups will bypass the encryption phase altogether and opt for pilfering data to openly extort victims with. While the latter scenario may seem preferable due to a lack of disruption, the reputational damage, potential liability, and threat to business viability remains. Hence our emphasis on prevention.

We urge all defenders to explore and embrace modern endpoint protection technologies that go beyond static checks, basic signatures, and other outdated components. Contextual awareness and automated behavioral classification are among the most powerful weapons defenders should avail themselves of.

Indicators of Compromise

FILE HASHES

SHA1
67f0c8d81aefcfc5943b31d695972194ac15e9f2
edba1b73ddd0e32784ae21844c940d7850531b82

2877b32518445c09418849eb8fb913ed73d7b8fb
cd8e4372620930876c71ba0a24e2b0e17dcd87c9
eaa2e1e2cb6c7b6ec405ffdf204999853ebbd54a
0f9484948fdd1b05bad387b14b27dc702c2c09ed
e3e8e28a70cdfa2164ece51ff377879a5151abdf
9d336b8911c8ffd7cc809e31d5b53796bb0cc7bb
1cc80ad88a022c429f8285d871f48529c6484734
3b40dbdc418d2d5de5f552a054a32bfbac18c5cc
2f3273e5b6739b844fe33f7310476afb971956dd
7777771aec887896be773c32200515a50e08112a
5dbe3713b309e6ecc208e2a6c038aeb1762340d4
480db5652124d4dd199bc8e775539684a19f1f24
Dc0ae41192272fda884a1a2589fe31d604d75af2

Hive.bat
C9471adc8db180a7da5a56966b156b440483856f

Shadow.bat
4714f1e6bb75a80a8faf69434726d176b70d7bd8

SHA256
a0b4e3d7e4cd20d25ad2f92be954b95eea44f8f1944118a3194295c5677db749
50ad0e6e9dc72d10579c20bb436f09eeaa7bfdbcb5747a2590af667823e85609
5ae51e30817c0d08d03f120539aedc31d094b080eb70c0691bbfbaa4ec265ef3
77a398c870ad4904d06d455c9249e7864ac92dda877e288e5718b3c8d9fc6618
e1a7ddbf735d5c1cb9097d7614840c00e5c4d5107fa687c0ab2a2ec8948ef84e
ed614cba30f26f90815c28e189340843fab0fe7ebe71bb9b4a3cb7c78ff8e3d2
c5fe23c626413a18cba8fb4ea93df81529c85f470577fb9c2336d6b692689d9d
88f7544a29a2ceb175a135d9fa221cbfd3e8c71f32dd6b09399717f85ea9afd1
2f7d37c22e6199d1496f307c676223dda999c136ece4f2748975169b4a48afe5
fdbc66ebe7af710e15946e1541e2e81ddfd62aa3b35339288a9a244fb56a74cf
1e21c8e27a97de1796ca47a9613477cf7aec335a783469c5ca3a09d4f07db0ff
bf7bc94506eb72daec1d310ba038d9c3b115f145594fd271b80fbe911a8f3964
c04509c1b80c129a7486119436c9ada5b0505358e97c1508b2cfb5c2a177ed11
612e5ffd09ca30ca9488d802594efb5d41c360f7a439df4ae09b14bce45575ec
0df750bf15895d410c3f6ce45279ab0329c5c723af38b99aadc9a60bcdc9a71d
5954558d43884da2c7902ddf89c0cf7cd5bf162d6feefe5ce7d15b16767a27e5

Hive.bat
93852dbd3a977cf2662b0c4db26b627736ba51c0df627eb36b41fdbde093c3c3

Shadow.bat
D158f9d53e7c37eadd3b5cc1b82d095f61484e47eda2c36d9d35f31c0b4d3ff8

COMMUNICATIONS

Cobalt Beacon: 176.123.8.228

MITRE ATT&CK

T1574.001 – Hijack Execution Flow: DLL Search Order Hijacking
TA0005 – Defense Evasion
TA0004 – Privilege Escalation
T1486 – Data Encrypted for Impact
T1027.002 – Obfuscated Files or Information: Software Packing
T1003.001 – OS Credential Dumping: LSASS Memory
T1007 – System Service Discovery
T1059 – Command and Scripting Interpreter
T1059.001 – Command and Scripting Interpreter: PowerShell
T1059.003 – Command and Scripting Interpreter: Windows Command Shell
T1490 – Inhibit System Recovery