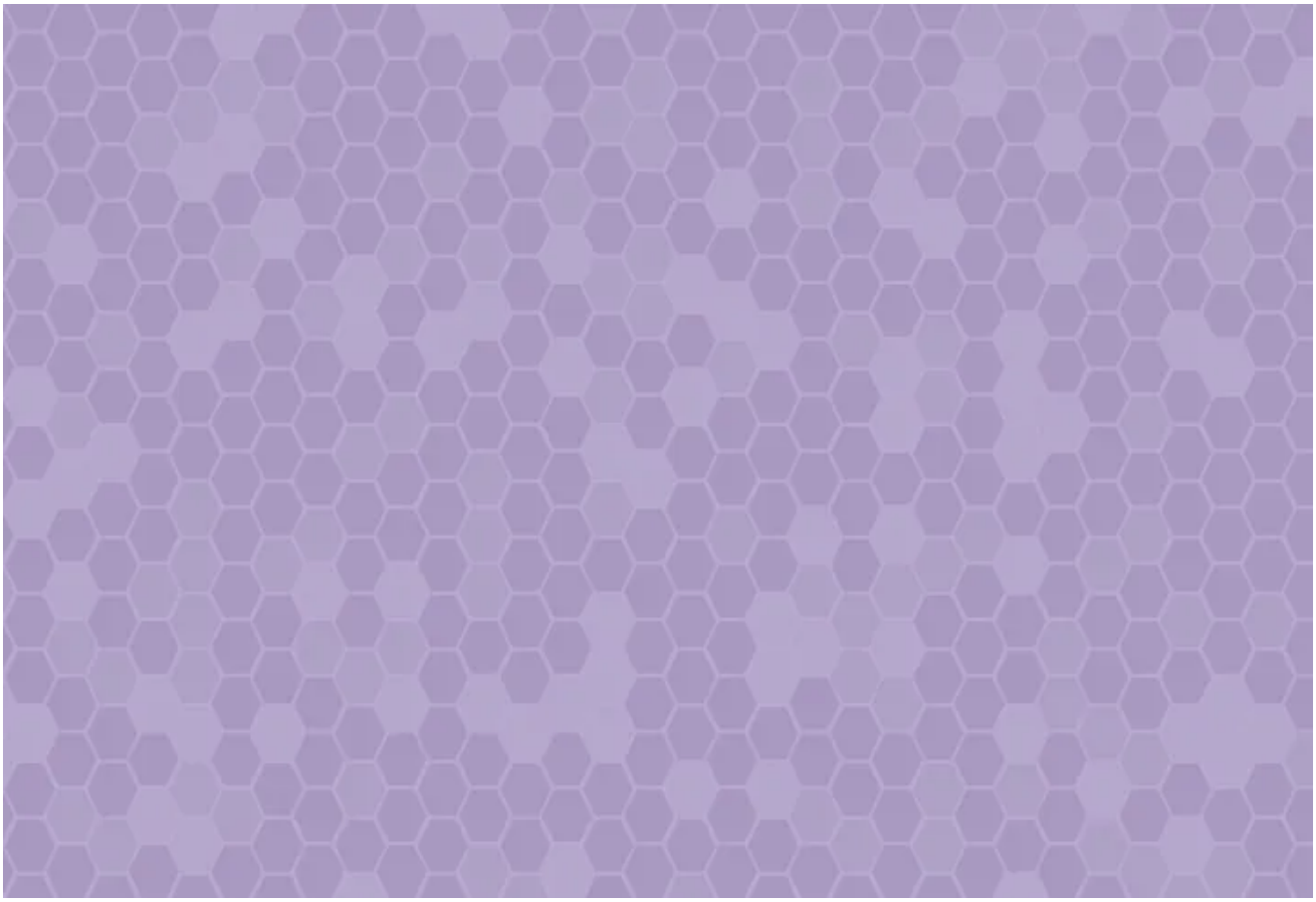
How to Decrypt the Files Encrypted by the Hive Ransomware

lifars.com/2022/02/how-to-decrypt-the-files-encrypted-by-the-hive-ransomware/

February 24, 2022



South Korean researchers published an academic paper that presents a method to decrypt the files encrypted by the Hive Ransomware [1]. This article highlights the main findings of the paper and our comments regarding the feasibility of creating a universal decryptor for Hive.

Hive ransomware appeared around June 2021 and infected multiple companies. Two important victims of Hive are the Altus Group [2] and Memorial Health System [3].

The ransomware generates 10MB of random data using the rand function in the Go language, which is called the Master Key. For each file that will be encrypted, Hive extracts 1MB and 1KB of data from specific offsets and uses them as a keystream. These offsets are stored in the encrypted file names and can be used to extract the keystream used for encryption.

Hive encrypts a file using the XOR operator with a random keystream. The Master Key is encrypted using a hard-coded RSA-2048 public key and stored in the “C:\” folder or “C:\Users\<User>\AppData\Local\VirtualStore” depending on the user’s privileges.

The ransomware creates a ransom note called HOW_TO_DECRYPT.txt in every targeted directory. Whether it’s running with admin privileges, Hive also encrypts the files located in the “C:\Users\Program Files(x86)”, “C:\Users\Program Files”, and “C:\Users\ProgramData” directories.

The malware generates two random numbers R1 and R2, which are 8 bytes long using the rand function of the Math package. It defines the following values:

- Keystream1 offset (SP1) = R1 % 0x900000
- Keystream2 offset (SP2) = R2 % 0x9FFC00
- Keystream1 = Master Key [SP1:SP1+0x100000], 0x100000 bytes = 1MB
- Keystream2 = Master Key [SP2:SP2+0x400], 0x400 bytes = 1KB

The above selection is also shown in figure 1.

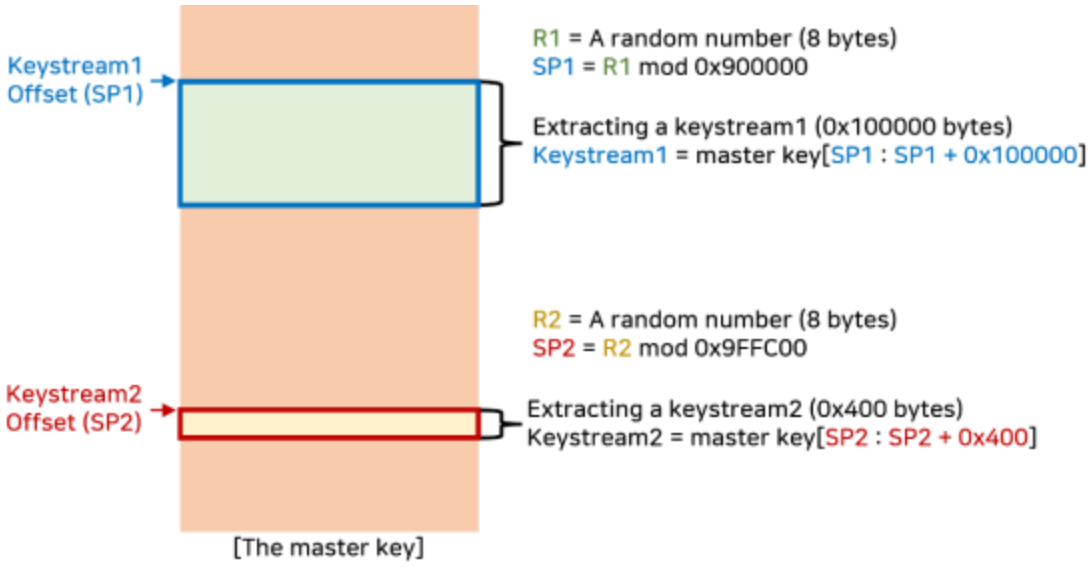


Figure 1

The files are encrypted using the XOR operator:

> Encrypted Data [i] = Data [i] XOR Keystream1 [i % 0x100000] XOR Keystream2 [i % 0x400]

which is equivalent to:

- EKS [i] = Keystream1 [i] XOR Keystream2 [i % 0x400], i = 0,1, 2, …, 0xFFFFF
- Encrypted Data = Data [offset] XOR EKS [offset % 0x100000]

All files are partially encrypted: 0x1000 encrypted bytes (4KB) alternatively followed by the non-encrypted data block (NBS bytes computed based on a file size). A visual representation of an encrypted file is displayed in figure 2.

Figure 2

A weakness of the above process is the fact that some keystreams will be partially reused when encrypting a lot of files. At least one of the following two conditions is required in order to decrypt files:

- Some original files corresponding to the encrypted files are available
- A lot of files with known signatures (.pdf, .xlsx, .hwp) have to be encrypted

An example provided by researchers regarding obtaining the original files represents the content of the Program Files, Program Files (x86), and ProgramData directories that are encrypted by the Hive ransomware. Some of these files are software files (Java, Python, Microsoft Office) that aren't related to the OS, which can be downloaded from the Internet. The keystream (EKS) can be computed by XOR-ing the original file with the encrypted content.

The academic paper presents the pseudocode of 3 algorithms: "Calculation of the non-encrypted data block size", "Calculation of the start offsets of Keystream1 and Keystream2", and "Hive ransomware master key recovery".

The researchers performed multiple experiments and concluded that it's better to mix various file sizes for recovering the Master Key:

Average file size	Number of files	Master key recovery rate (%)
1KB	9	95.85
2–127KB	24	95.85
128–1,023KB	19	95.85
1–10MB	19	95.85

10–100MB	20	95.85

Using the Master Key that was partially recovered, we could decrypt most of the encrypted files, as highlighted in figure 3:

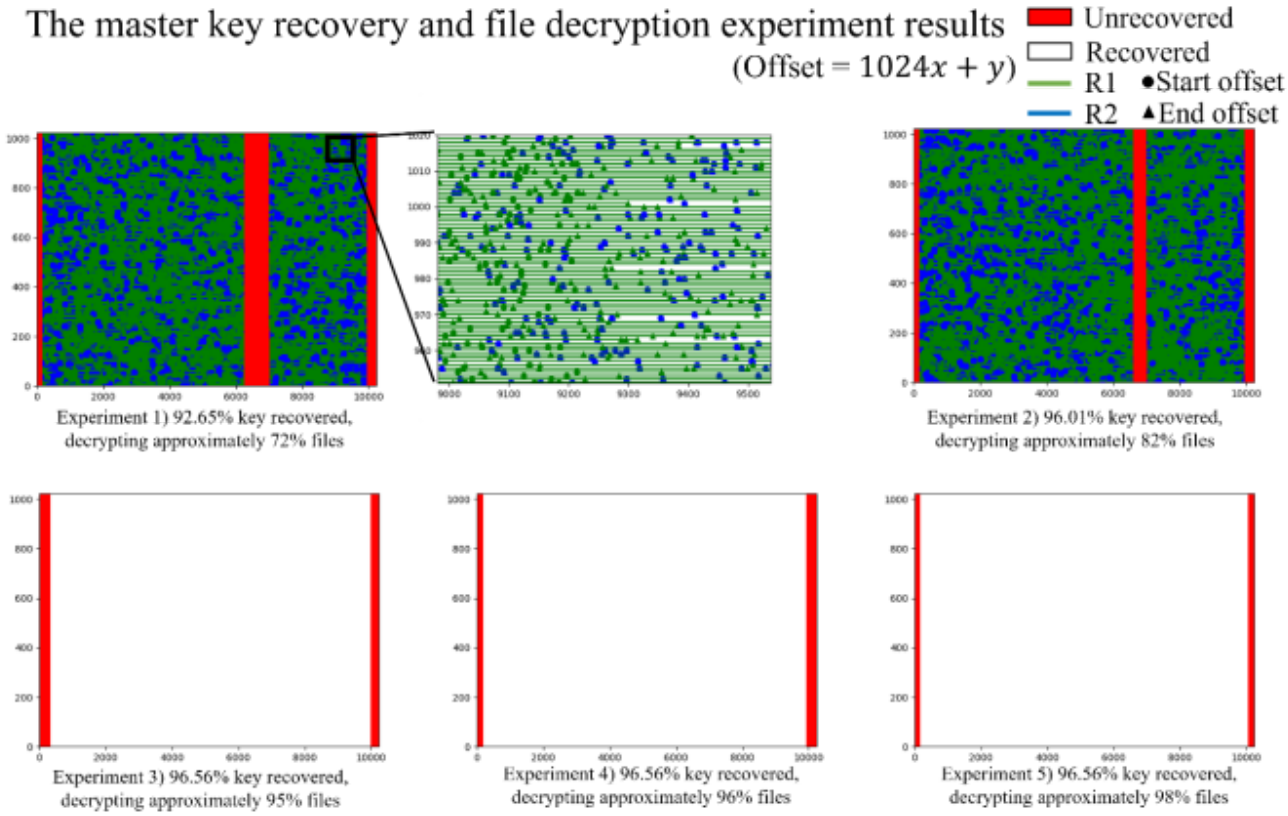


Figure 3

Our company has been engaged in a lot of ransomware engagements in the past and has analyzed most of the ransomware families that are still active in the wild. We believe that even if there is a theoretical chance to create a universal decryptor for the Hive ransomware, it's way easier to investigate an individual infection and decide whether the environment meets at least one of the necessary conditions to decrypt the files: at least tens of the unencrypted files should be available, or a lot of files with known signatures should have been encrypted (there is no number specified in the paper, this might be even infeasible).

References:

[1] https://arxiv.org/pdf/2202.08477.pdf

[2] https://cybernews.com/news/new-ransomware-group-hive-leaks-altus-group-sample-files/

[3] https://www.bleepingcomputer.com/news/security/hive-ransomware-attacks-memorial-health-system-steals-patient-data/