YOROI®

TINEXTA GROUP

ON THE FOOTSTEPS OF HIVE RANSOMWARE

22/07/2022

DEFENCE BELONGS TO HUMANS

Date	Activity	Author
22/07/2022	Report delivery	Luigi Martire, Carmelo Ragusa

Table of Content

Introduction

Hive ransomware is one of the most active financially motivated threat actors of this period, adopting the current Double Extorsion model. They started their malicious activities in June of the past year, and just in a year of activity they collected a big number of victims, demonstrating the capability to hit even critical infrastructures.

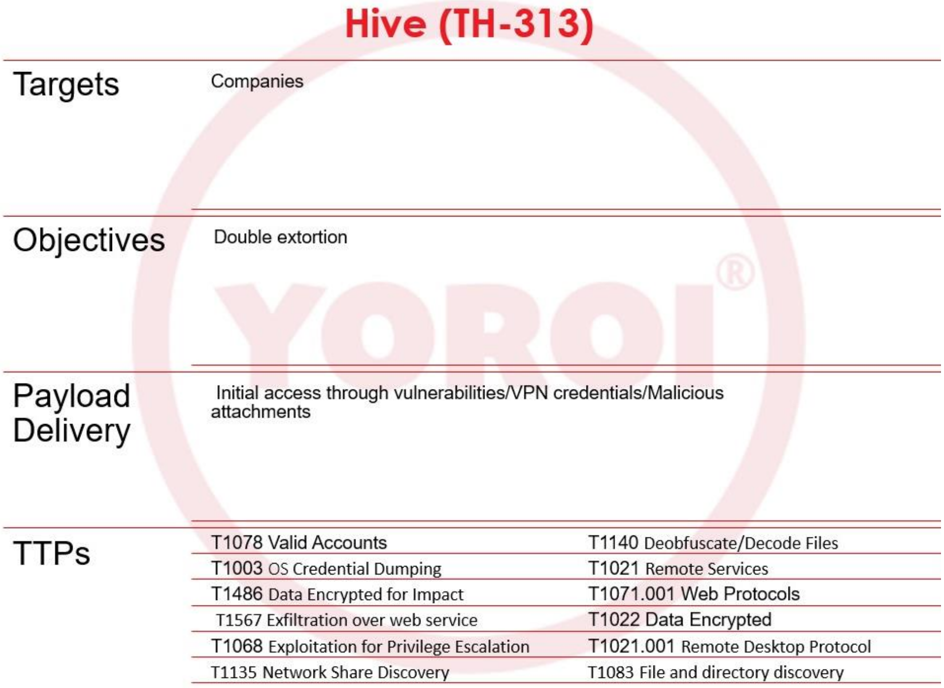
The criminal group distinguished from other ones also for attacking healthcare organization during the 2021 when we had to face off the Covid-19 pandemic. It was emblematic that one of the first victims was the Memorial Health System in August 2021.

For these reasons, Yoroi's Malware ZLab decided to keep track of this infamous threat actor and observe any modification of its modus operandi, in order to provide a guideline focusing on the evolution of the locker sample of the cyber gang.

About Hive

Hive (TH-313) is a Ransomware group firstly spotted in June 2021 and it gathered a big popularity inside the cybersecurity community because it was able to attack a large variety of sectors, starting from healthcare facilities and arriving to critical infrastructures, passing through manufacturers during just a year of activity.

In addition, the group was able to refine its toolkit and then its TTPs with a surprising speed: the business model is the Double-Extorsion and Ransomware-as-a-Service, with a self-made ransomware payload.



Hive (TH-313)

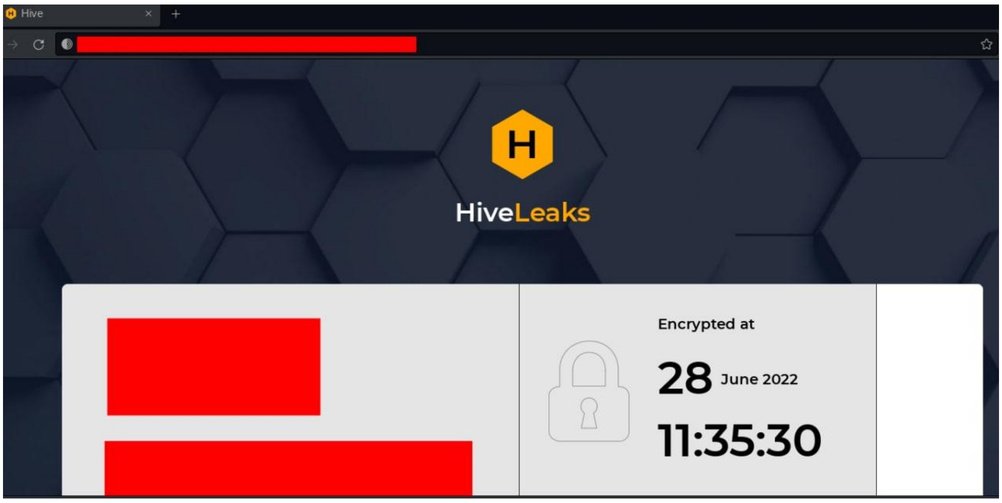
Targets	Companies	
Objectives	Double extortion	
Payload Delivery	Initial access through vulnerabilities/VPN credentials/Malicious attachments	
TTPs	T1078 Valid Accounts	T1140 Deobfuscate/Decode Files
	T1003 OS Credential Dumping	T1021 Remote Services
	T1486 Data Encrypted for Impact	T1071.001 Web Protocols
	T1567 Exfiltration over web service	T1022 Data Encrypted
	T1068 Exploitation for Privilege Escalation	T1021.001 Remote Desktop Protocol
	T1135 Network Share Discovery	T1083 File and directory discovery

Figure 1: Hive (TH-313)

So, in this report we have decided to focus our attention on the ransomware payload evolution, providing a timeline of the development of Hive Ransomware Payloads.

Timeline of the development of Hive Ransomware

Inside the criminal group, there is surely a high-profile development team, with deep knowledge of programming in both newer and older programming languages. The first versions of the encryptor payload are written in Golang, then, starting from the v5 version, the dev team of Hive switched into Rust.



[redacted]

[redacted]

[redacted]

Encrypted at

28 June 2022

11:35:30

Figure 2: Leak site

In the following timeline, we provide a quick overview of the evolution of the malware and how the cyber gang adopted an incremental development process on its TTPs:

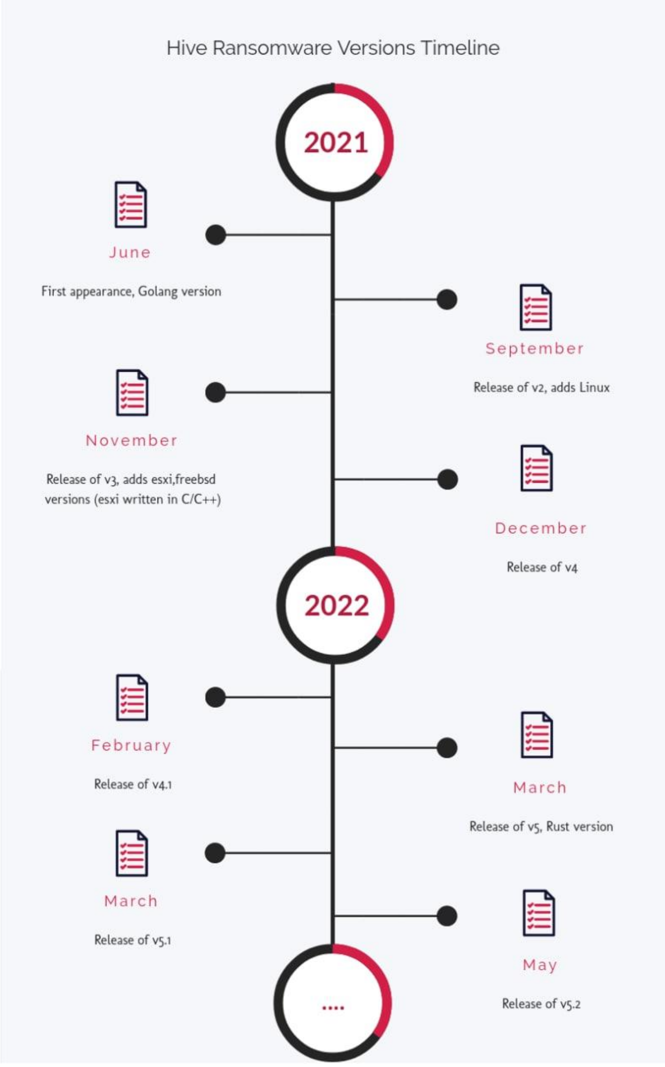
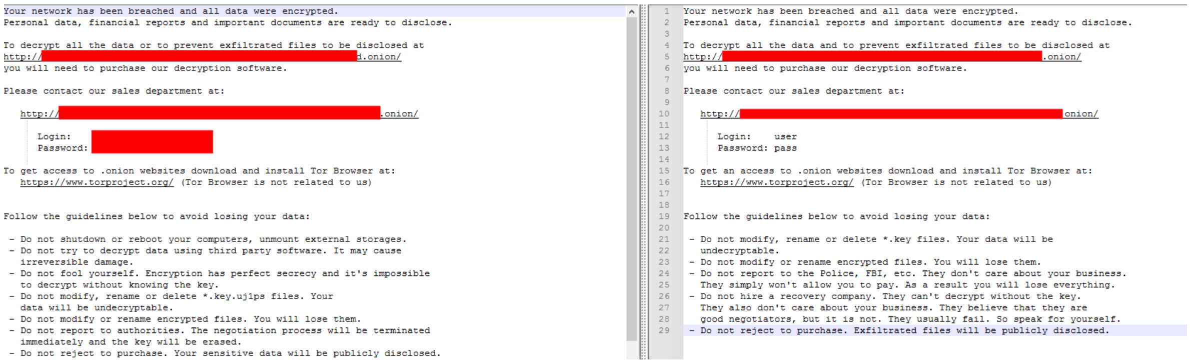

Hive Ransomware Versions Timeline
2021
June
First appearance, Golang version
September
Release of v2, adds Linux
November
Release of v3, adds esxi,freebsd versions (esxi written in C/C++)
December
Release of v4
2022
February
Release of v4.1
March
Release of v5, Rust version
March
Release of v5.1
May
Release of v5.2
....

Figure 3: Hive Timeline

In the same way, even the Ransom Note changed during the evolution: first, the credentials were hardcoded inside the sample, but now the operators pass them as a parameter when the locker process is launched. Below a comparison between an earlier version and a later one:



```
Your network has been breached and all data were encrypted.
Personal data, financial reports and important documents are ready to disclose.

To decrypt all the data or to prevent exfiltrated files to be disclosed at
http://[redacted]d.onion/
you will need to purchase our decryption software.

Please contact our sales department at:

   http://[redacted].onion/

      Login:
      Password: [redacted]

To get access to .onion websites download and install Tor Browser at:
   https://www.torproject.org/ (Tor Browser is not related to us)


Follow the guidelines below to avoid losing your data:

 - Do not shutdown or reboot your computers, unmount external storages.
 - Do not try to decrypt data using third party software. It may cause
   irreversible damage.
 - Do not fool yourself. Encryption has perfect secrecy and it's impossible
   to decrypt without knowing the key.
 - Do not modify, rename or delete *.key.uj1ps files. Your
   data will be undecryptable.
 - Do not modify or rename encrypted files. You will lose them.
 - Do not report to authorities. The negotiation process will be terminated
   immediately and the key will be erased.
 - Do not reject to purchase. Your sensitive data will be publicly disclosed.
```

```
Your network has been breached and all data were encrypted.
Personal data, financial reports and important documents are ready to disclose.

To decrypt all the data and to prevent exfiltrated files to be disclosed at
http://[redacted].onion/
you will need to purchase our decryption software.

Please contact our sales department at:

   http://[redacted]onion/

      Login:    user
      Password: pass

To get an access to .onion websites download and install Tor Browser at:
   https://www.torproject.org/ (Tor Browser is not related to us)


Follow the guidelines below to avoid losing your data:

 - Do not modify, rename or delete *.key files. Your data will be
   undecryptable.
 - Do not modify or rename encrypted files. You will lose them.
 - Do not report to the Police, FBI, etc. They don't care about your business.
   They simply won't allow you to pay. As a result you will lose everything.
 - Do not hire a recovery company. They can't decrypt without the key.
   They also don't care about your business. They believe that they are
   good negotiators, but it is not. They usually fail. So speak for yourself.
 - Do not reject to purchase. Exfiltrated files will be publicly disclosed.
```

Figure 4: Ransom note comparison

Victimology

During its activity, Hive Group hit a large number of victims and during that period some of them paid the ransom, after that the victims were removed from the "walk of shame". We tracked a total of 130 victims listed on their leak site, the affected companies belong to different sectors and nations. However, we have evidence that occasionally some victims of the group, despite being attacked by the threat actor, are never reported onto the site.

Moreover, the group does not exclude hospitals, companies that provide medical equipment and non-profit organizations. An example is the attack on the "Memorial Health System" in August 2021 or more recently on the "Partnership HealthPlan of California", a non-profit organization.

The following graph shows the total progress of the victims so far, indicating that the group is consolidating its role as one of the principal threats in the panorama.

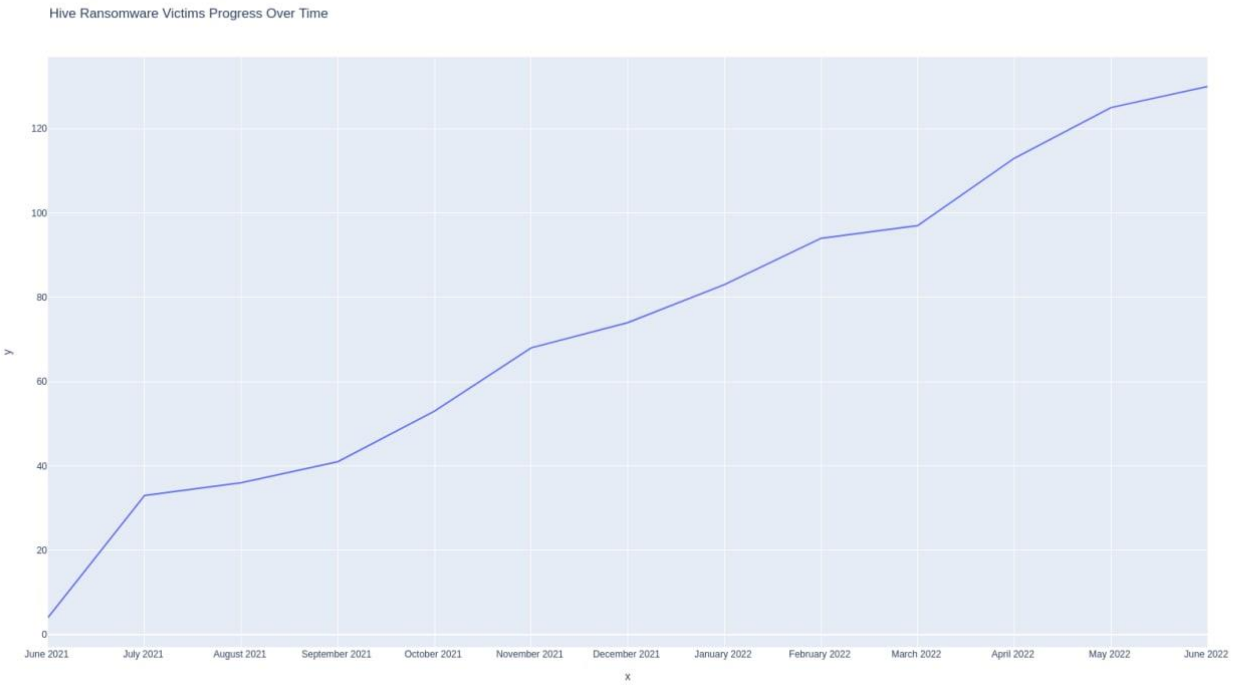


Figure 5: Hive Ransomware Victims Progress over time

Another view of the same information is represented in the following graph, where the focus is pointed to highlight the month in which most victims were published on their leak site, which turns out to be July 2021, shortly after the group started. So, it means that the ransomware operators gathered a consistent number of victims during the startup phase, in order to create a solid placement inside the threat landscape. After that phase, the gang continued to threaten with huge aggression.

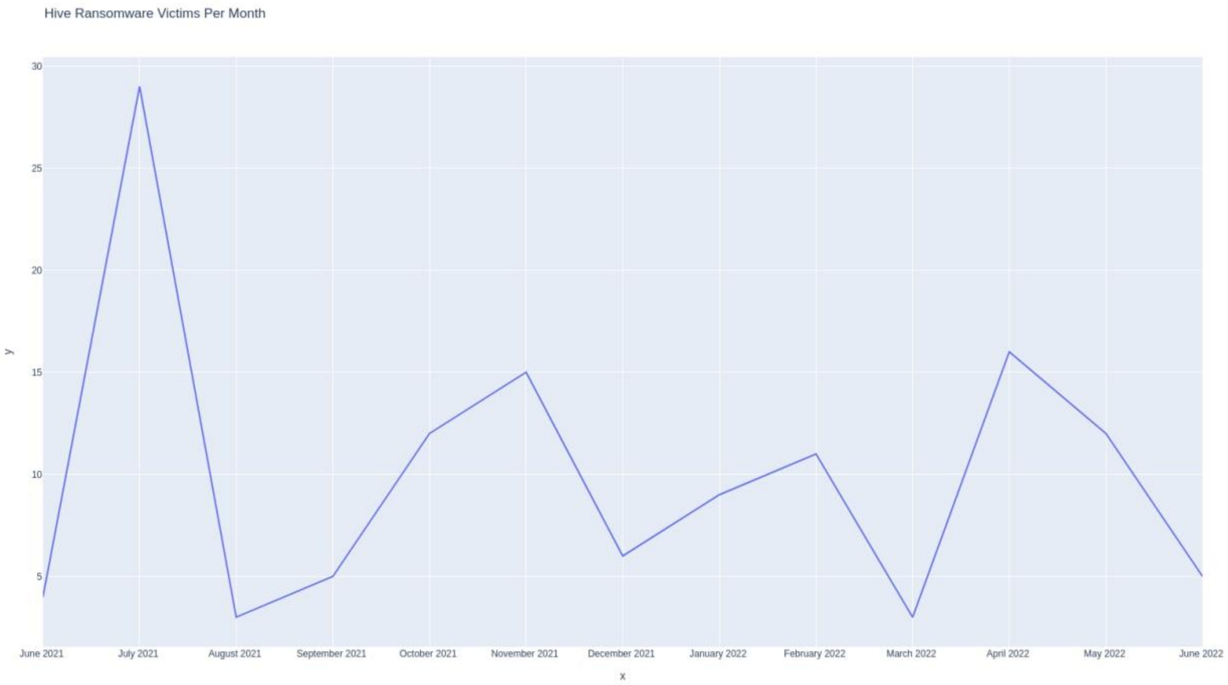


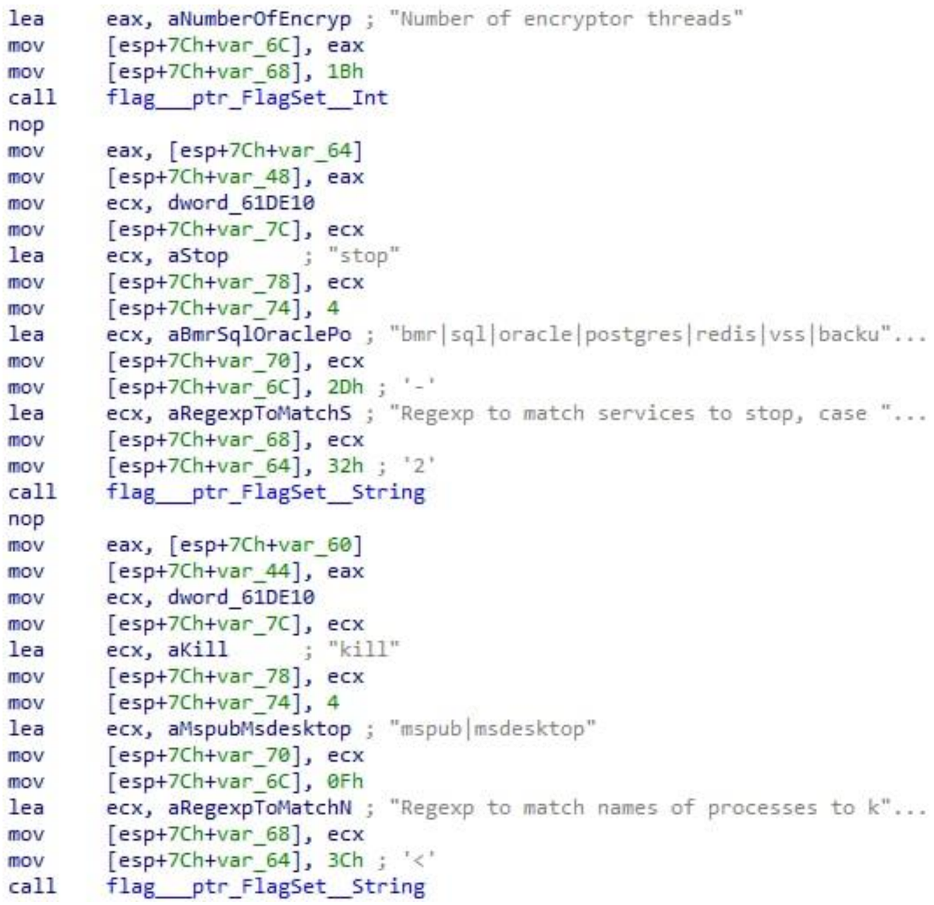
Figure 6: Hive Victims Per Month

Hive v1

Hash	88f7544a29a2ceb175a135d9fa221cbfd3e8c71f32dd6b09399717f85ea9afd1
Threat	Ransomware
Brief Description	Hive Ransomware v1
SSDEEP	12288:CinNFNkY/yU97ppM4NSBG81Np2C9H4S3iDjlLtc4wCIlTIQaOI6NrwacVYV+4MsT:CinN3n/y67jM4v4kCSPDjlLtbwt8IQLH

Table 1: Hive v1

The first version, written in Golang, was a sophisticated encryptor program, but, due to the newness of the malicious activity, there is no track of obfuscation, and the strings can be easily seen, the following figure shows some of the available parameters:



```
lea     eax, aNumberOfEncryp ; "Number of encryptor threads"
mov     [esp+7Ch+var_6C], eax
mov     [esp+7Ch+var_68], 1Bh
call    flag___ptr_FlagSet__Int
nop
mov     eax, [esp+7Ch+var_64]
mov     [esp+7Ch+var_48], eax
mov     ecx, dword_61DE10
mov     [esp+7Ch+var_7C], ecx
lea     ecx, aStop      ; "stop"
mov     [esp+7Ch+var_78], ecx
mov     [esp+7Ch+var_74], 4
lea     ecx, aBmrSqlOraclePo ; "bmr|sql|oracle|postgres|redis|vss|backu"...
mov     [esp+7Ch+var_70], ecx
mov     [esp+7Ch+var_6C], 2Dh ; '-'
lea     ecx, aRegexpToMatchS ; "Regexp to match services to stop, case "...
mov     [esp+7Ch+var_68], ecx
mov     [esp+7Ch+var_64], 32h ; '2'
call    flag___ptr_FlagSet__String
nop
mov     eax, [esp+7Ch+var_60]
mov     [esp+7Ch+var_44], eax
mov     ecx, dword_61DE10
mov     [esp+7Ch+var_7C], ecx
lea     ecx, aKill      ; "kill"
mov     [esp+7Ch+var_78], ecx
mov     [esp+7Ch+var_74], 4
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_68], ecx
mov     [esp+7Ch+var_64], 3Ch ; '<'
call    flag___ptr_FlagSet__String
```

Figure 7: Available parameters

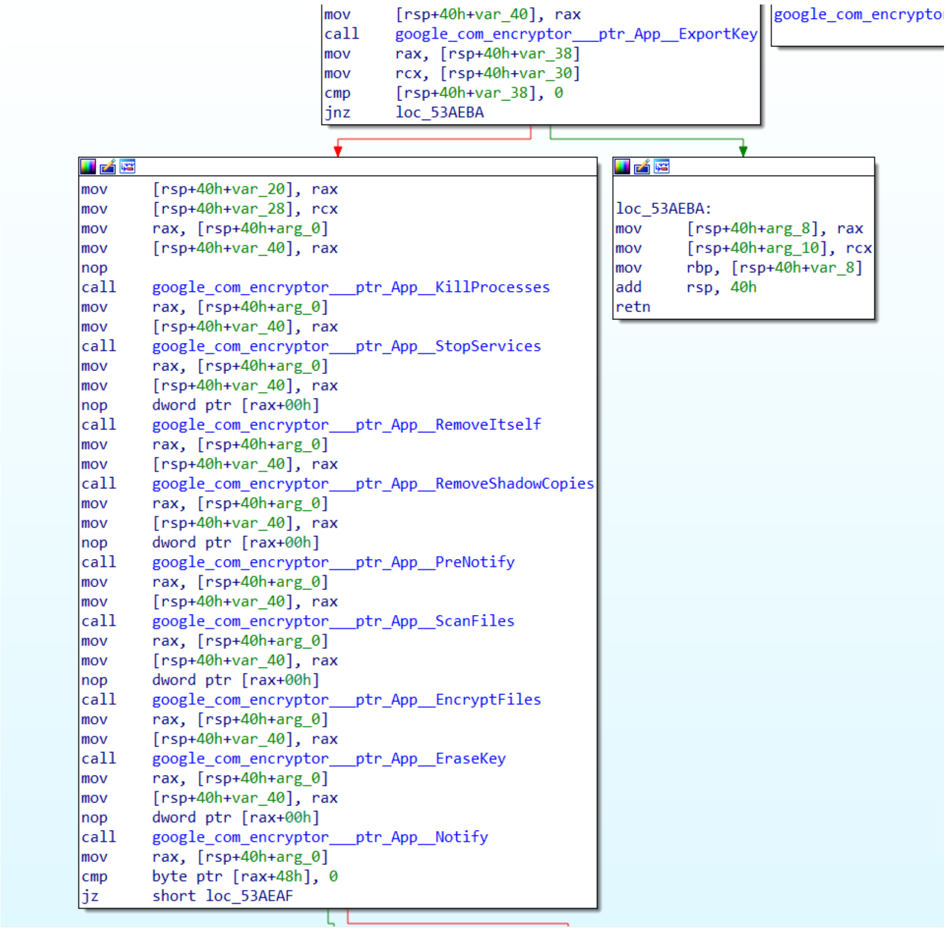
The initial effort of the gang was to make a product quite customizable according to the infection and the encryption process to perform. In this way, the malware writers provided a series of parameters to launch an ad-hoc infection profile.

The following table describes all the available parameters found in this version:

Parameter	Description
-kill	Regex, names of the processes to kill. Default values: “mspub|msdesktop”
-no-clean	Skip clean disk space stage
-skip	Regex, names of the files to skip. Default values: “\\.lnk”
-skip-before	Skip files before a specific date. Defaut value: “03.09.2016”
-stop	Regex, names of the services to stop. Default values: "bmr|sql|oracle|postgres|redis|vss|backup|sstp"
-t	Number of threads

Table 2: Hive v1 Parameters

Once the parameters are parsed, creating the desired infection profile, the control flow passes to the core malicious operations.



```
mov     [rsp+40h+var_40], rax
call    google_com_encryptor___ptr_App__ExportKey
mov     rax, [rsp+40h+var_38]
mov     rcx, [rsp+40h+var_30]
cmp     [rsp+40h+var_38], 0
jnz     loc_53AEBA
```

```
google_com_encrypto
```

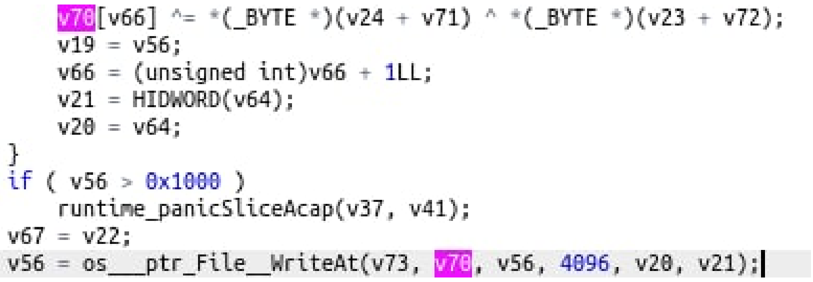
```
mov     [rsp+40h+var_20], rax
mov     [rsp+40h+var_28], rcx
mov     rax, [rsp+40h+arg_0]
mov     [rsp+40h+var_40], rax
nop
call    google_com_encryptor___ptr_App__KillProcesses
mov     rax, [rsp+40h+arg_0]
mov     [rsp+40h+var_40], rax
call    google_com_encryptor___ptr_App__StopServices
mov     rax, [rsp+40h+arg_0]
mov     [rsp+40h+var_40], rax
nop     dword ptr [rax+00h]
call    google_com_encryptor___ptr_App__RemoveItself
mov     rax, [rsp+40h+arg_0]
mov     [rsp+40h+var_40], rax
call    google_com_encryptor___ptr_App__RemoveShadowCopies
mov     rax, [rsp+40h+arg_0]
mov     [rsp+40h+var_40], rax
nop     dword ptr [rax+00h]
call    google_com_encryptor___ptr_App__PreNotify
mov     rax, [rsp+40h+arg_0]
mov     [rsp+40h+var_40], rax
call    google_com_encryptor___ptr_App__ScanFiles
mov     rax, [rsp+40h+arg_0]
mov     [rsp+40h+var_40], rax
nop     dword ptr [rax+00h]
call    google_com_encryptor___ptr_App__EncryptFiles
mov     rax, [rsp+40h+arg_0]
mov     [rsp+40h+var_40], rax
call    google_com_encryptor___ptr_App__EraseKey
mov     rax, [rsp+40h+arg_0]
mov     [rsp+40h+var_40], rax
nop     dword ptr [rax+00h]
call    google_com_encryptor___ptr_App__Notify
mov     rax, [rsp+40h+arg_0]
cmp     byte ptr [rax+48h], 0
jz      short loc_53AEAF
```

```
loc_53AEBA:
mov     [rsp+40h+arg_8], rax
mov     [rsp+40h+arg_10], rcx
mov     rbp, [rsp+40h+var_8]
add     rsp, 40h
retn
```

Figure 8: Hive core function

The locker sample proceeds to export the key, to kill the processes and services specified and to remove the shadow copies then it iterates the directories and starts encrypting the files.

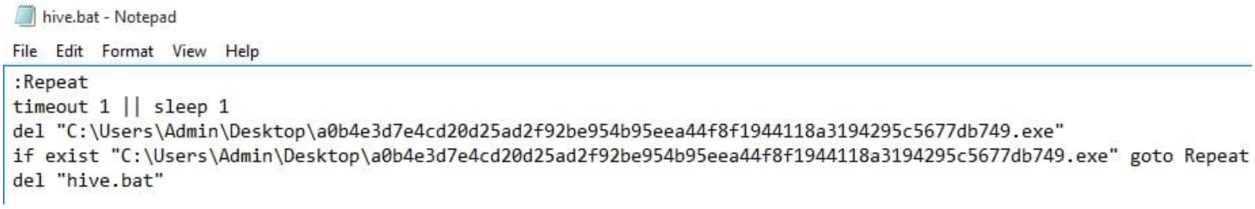
The core of the encryption scheme of Hive ransomware is a union of XOR+RSA algorithms. In the figure below we can see the XOR related routine:



```
    v70[v66] ^= *(_BYTE *)(v24 + v71) ^ *(_BYTE *)(v23 + v72);
    v19 = v56;
    v66 = (unsigned int)v66 + 1LL;
    v21 = HIDWORD(v64);
    v20 = v64;
}
if ( v56 > 0x1000 )
    runtime_panicSliceAcap(v37, v41);
v67 = v22;
v56 = os___ptr_File__WriteAt(v73, v70, v56, 4096, v20, v21);
```

Figure 9: Usage of XOR algorithm

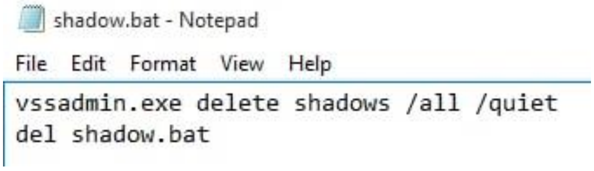
Then, in this first version, it uses ".hive" as extension to the encrypted files, later it is used a unique ID instead. Moreover, the **RemoveItself routine** drops "hive.bat" to remove itself. But, since the second version of the malware calls the related function after the encryption is complete:



```
:Repeat
timeout 1 || sleep 1
del "C:\Users\Admin\Desktop\a0b4e3d7e4cd20d25ad2f92be954b95eea44f8f1944118a3194295c5677db749.exe"
if exist "C:\Users\Admin\Desktop\a0b4e3d7e4cd20d25ad2f92be954b95eea44f8f1944118a3194295c5677db749.exe" goto Repeat
del "hive.bat"
```

Figure 10: hive.bat

RemoveShadowCopies drops "shadow.bat" to remove the shadow copies, from the second version will directly execute the command instead of dropping a .bat:



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

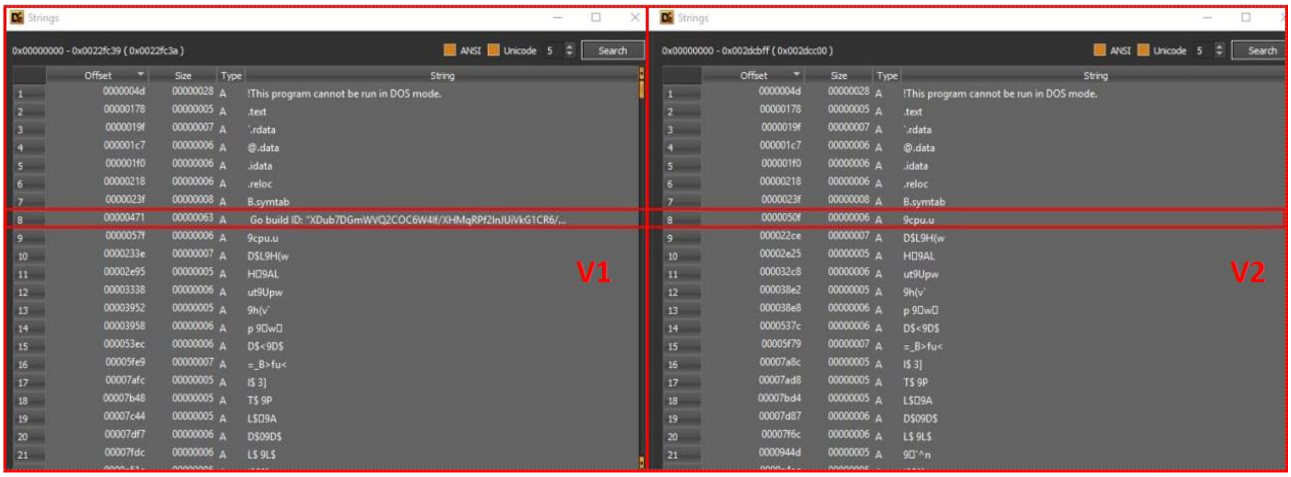
Figure 11: shadow.bat

Hive v2

Hash	25bfec0c3c81ab55cf85a57367c14cc6803a03e2e9b4afd72e7bbca9420fe7c5
Threat	Ransomware
Brief Description	Hive Ransomware v2
SSDEEP	12288:Sw41dVZvThPCsM18GLHe7wlDdkPAQEtxr0fflvRmhEBWtdUJiAUtP/T/kAfMvgV:dod1HDmlDdkZ4YXPpaTTXMw

Table 3: Hive v2

With the second version of Hive, the malware writers started to complicate the code in order to make the analysis more difficult for the analyst. The initial step is to obfuscate the “Go Build ID” header present in all golang- written binaries.

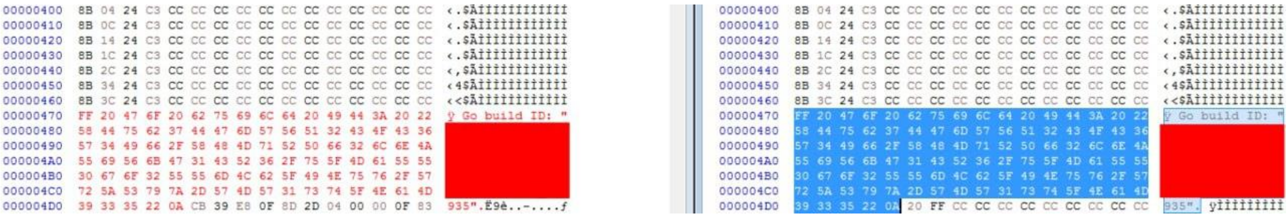


```
0x00000000 - 0x0022fc39 ( 0x0022fc3a )

     Offset    Size      Type  String
1    0000004d  00000028  A     !This program cannot be run in DOS mode.
2    00000178  00000005  A     .text
3    0000019f  00000007  A     `.rdata
4    000001c7  00000006  A     @.data
5    000001f0  00000006  A     .idata
6    00000218  00000006  A     .reloc
7    0000023f  00000008  A     B.symtab
8    00000471  00000063  A      Go build ID: "XDub7DGmWVQ2COC6W4If/XHMqRPf2InJUiVkG1CR6/...
9    0000057f  00000006  A     9cpu.u
10   0000233e  00000007  A     D$L9H(w
11   00002e95  00000005  A     H□9AL
12   00003338  00000006  A     ut9Upw
13   00003952  00000005  A     9h(v`
14   00003958  00000006  A     p 9□w□
15   000053ec  00000006  A     D$<9D$
16   00005fe9  00000007  A     =_B>fu<
17   00007afc  00000005  A     I$ 3]
18   00007b48  00000005  A     T$ 9P
19   00007c44  00000005  A     L$□9A
20   00007df7  00000006  A     D$09D$
21   00007fdc  00000006  A     L$ 9L$
```

```
0x00000000 - 0x002dcbff ( 0x002dcc00 )

     Offset    Size      Type  String
1    0000004d  00000028  A     !This program cannot be run in DOS mode.
2    00000178  00000005  A     .text
3    0000019f  00000007  A     `.rdata
4    000001c7  00000006  A     @.data
5    000001f0  00000006  A     .idata
6    00000218  00000006  A     .reloc
7    0000023f  00000008  A     B.symtab
8    0000050f  00000006  A     9cpu.u
9    000022ce  00000007  A     D$L9H(w
10   00002e25  00000005  A     H□9AL
11   000032c8  00000006  A     ut9Upw
12   000038e2  00000005  A     9h(v`
13   000038e8  00000006  A     p 9□w□
14   0000537c  00000006  A     D$<9D$
15   00005f79  00000007  A     =_B>fu<
16   00007a8c  00000005  A     I$ 3]
17   00007ad8  00000005  A     T$ 9P
18   00007bd4  00000005  A     L$□9A
19   00007d87  00000006  A     D$09D$
20   00007f6c  00000006  A     L$ 9L$
21   0000944d  00000005  A     9□`^n
```

Figure 12: Strings comparison

The simple trick causes that, when opening a disassembler, like IDA, the analyst can immediately see Golang not being recognized. However, a simple fix provides the overwriting of the build-id with a legit one.

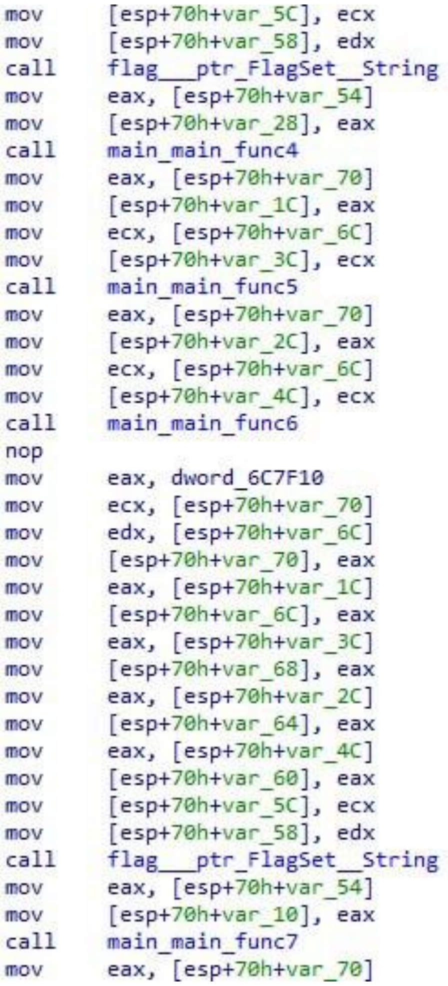


```
00000400  8B 04 24 C3 CC CC CC CC CC CC CC CC CC CC CC CC  ‹.$ÃÌÌÌÌÌÌÌÌÌÌÌÌ
00000410  8B 0C 24 C3 CC CC CC CC CC CC CC CC CC CC CC CC  ‹.$ÃÌÌÌÌÌÌÌÌÌÌÌÌ
00000420  8B 14 24 C3 CC CC CC CC CC CC CC CC CC CC CC CC  ‹.$ÃÌÌÌÌÌÌÌÌÌÌÌÌ
00000430  8B 1C 24 C3 CC CC CC CC CC CC CC CC CC CC CC CC  ‹.$ÃÌÌÌÌÌÌÌÌÌÌÌÌ
00000440  8B 2C 24 C3 CC CC CC CC CC CC CC CC CC CC CC CC  ‹,$ÃÌÌÌÌÌÌÌÌÌÌÌÌ
00000450  8B 34 24 C3 CC CC CC CC CC CC CC CC CC CC CC CC  ‹4$ÃÌÌÌÌÌÌÌÌÌÌÌÌ
00000460  8B 3C 24 C3 CC CC CC CC CC CC CC CC CC CC CC CC  ‹<$ÃÌÌÌÌÌÌÌÌÌÌÌÌ
00000470  FF 20 47 6F 20 62 75 69 6C 64 20 49 44 3A 20 22  ÿ Go build ID: "
00000480  58 44 75 62 37 44 47 6D 57 56 51 32 43 4F 43 36  [redacted]
00000490  57 34 49 66 2F 58 48 4D 71 52 50 66 32 6C 6E 4A
000004A0  55 69 56 6B 47 31 43 52 36 2F 75 5F 4D 61 55 55
000004B0  30 67 6F 32 55 55 6D 4C 62 5F 49 4E 75 76 2F 57
000004C0  72 5A 53 79 7A 2D 57 4D 57 31 73 74 5F 4E 61 4D
000004D0  39 33 35 22 0A CB 39 E8 0F 8D 2D 04 00 00 0F 83  935".Ë9è..-....ƒ
```

```
00000400  8B 04 24 C3 CC CC CC CC CC CC CC CC CC CC CC CC  ‹.$ÃÌÌÌÌÌÌÌÌÌÌÌÌ
00000410  8B 0C 24 C3 CC CC CC CC CC CC CC CC CC CC CC CC  ‹.$ÃÌÌÌÌÌÌÌÌÌÌÌÌ
00000420  8B 14 24 C3 CC CC CC CC CC CC CC CC CC CC CC CC  ‹.$ÃÌÌÌÌÌÌÌÌÌÌÌÌ
00000430  8B 1C 24 C3 CC CC CC CC CC CC CC CC CC CC CC CC  ‹.$ÃÌÌÌÌÌÌÌÌÌÌÌÌ
00000440  8B 2C 24 C3 CC CC CC CC CC CC CC CC CC CC CC CC  ‹,$ÃÌÌÌÌÌÌÌÌÌÌÌÌ
00000450  8B 34 24 C3 CC CC CC CC CC CC CC CC CC CC CC CC  ‹4$ÃÌÌÌÌÌÌÌÌÌÌÌÌ
00000460  8B 3C 24 C3 CC CC CC CC CC CC CC CC CC CC CC CC  ‹<$ÃÌÌÌÌÌÌÌÌÌÌÌÌ
00000470  FF 20 47 6F 20 62 75 69 6C 64 20 49 44 3A 20 22  ÿ Go build ID: "
00000480  58 44 75 62 37 44 47 6D 57 56 51 32 43 4F 43 36  [redacted]
00000490  57 34 49 66 2F 58 48 4D 71 52 50 66 32 6C 6E 4A
000004A0  55 69 56 6B 47 31 43 52 36 2F 75 5F 4D 61 55 55
000004B0  30 67 6F 32 55 55 6D 4C 62 5F 49 4E 75 76 2F 57
000004C0  72 5A 53 79 7A 2D 57 4D 57 31 73 74 5F 4E 61 4D
000004D0  39 33 35 22 0A 20 FF CC CC CC CC CC CC CC CC CC  935". ÿÌÌÌÌÌÌÌÌÌ
```

Figure 13: “Go Build ID” patch

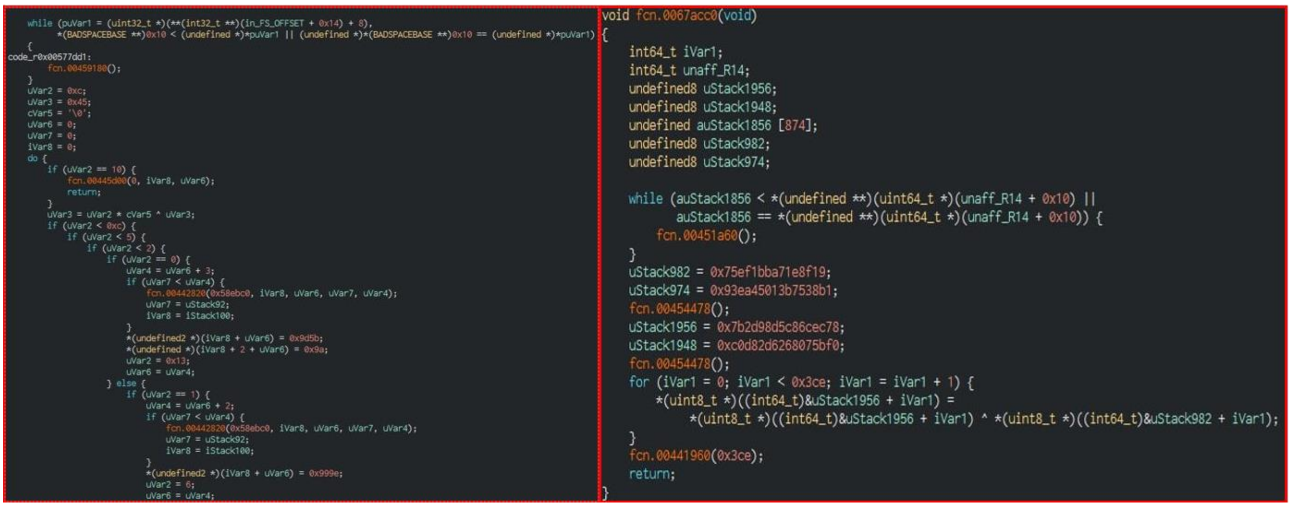
In addition, now the strings are obfuscated, and the names of the functions present inside the main are not visible in cleartext:



```
mov     [esp+70h+var_5C], ecx
mov     [esp+70h+var_58], edx
call    flag___ptr_FlagSet__String
mov     eax, [esp+70h+var_54]
mov     [esp+70h+var_28], eax
call    main_main_func4
mov     eax, [esp+70h+var_70]
mov     [esp+70h+var_1C], eax
mov     ecx, [esp+70h+var_6C]
mov     [esp+70h+var_3C], ecx
call    main_main_func5
mov     eax, [esp+70h+var_70]
mov     [esp+70h+var_2C], eax
mov     ecx, [esp+70h+var_6C]
mov     [esp+70h+var_4C], ecx
call    main_main_func6
nop
mov     eax, dword_6C7F10
mov     ecx, [esp+70h+var_70]
mov     edx, [esp+70h+var_6C]
mov     [esp+70h+var_70], eax
mov     eax, [esp+70h+var_1C]
mov     [esp+70h+var_6C], eax
mov     eax, [esp+70h+var_3C]
mov     [esp+70h+var_68], eax
mov     eax, [esp+70h+var_2C]
mov     [esp+70h+var_64], eax
mov     eax, [esp+70h+var_4C]
mov     [esp+70h+var_60], eax
mov     [esp+70h+var_5C], ecx
mov     [esp+70h+var_58], edx
call    flag___ptr_FlagSet__String
mov     eax, [esp+70h+var_54]
mov     [esp+70h+var_10], eax
call    main_main_func7
mov     eax, [esp+70h+var_70]
```

Figure 14: Obfuscated parameters

In the following screen two different routines for the strings obfuscation is provided:



```
    while (puVar1 = (uint32_t *)(**(int32_t **)(in_FS_OFFSET + 0x14) + 8),
          *(BADSPACEBASE **)0x10 < (undefined *)*puVar1 || (undefined *)*(BADSPACEBASE **)0x10 == (undefined *)*puVar1)
    {
code_r0x00577dd1:
        fcn.00459180();
    }
    uVar2 = 0xc;
    uVar3 = 0x45;
    cVar5 = '\0';
    uVar6 = 0;
    uVar7 = 0;
    iVar8 = 0;
    do {
        if (uVar2 == 10) {
            fcn.00445d00(0, iVar8, uVar6);
            return;
        }
        uVar3 = uVar2 * cVar5 ^ uVar3;
        if (uVar2 < 0xc) {
            if (uVar2 < 5) {
                if (uVar2 < 2) {
                    if (uVar2 == 0) {
                        uVar4 = uVar6 + 3;
                        if (uVar7 < uVar4) {
                            fcn.00442820(0x58ebc0, iVar8, uVar6, uVar7, uVar4);
                            uVar7 = uStack92;
                            iVar8 = iStack100;
                        }
                        *(undefined2 *)(iVar8 + uVar6) = 0x9d5b;
                        *(undefined *)(iVar8 + 2 + uVar6) = 0x9a;
                        uVar2 = 0x13;
                        uVar6 = uVar4;
                    } else {
                        if (uVar2 == 1) {
                            uVar4 = uVar6 + 2;
                            if (uVar7 < uVar4) {
                                fcn.00442820(0x58ebc0, iVar8, uVar6, uVar7, uVar4);
                                uVar7 = uStack92;
                                iVar8 = iStack100;
                            }
                            *(undefined2 *)(iVar8 + uVar6) = 0x999e;
                            uVar2 = 6;
                            uVar6 = uVar4;
```

```
void fcn.0067acc0(void)
{
    int64_t iVar1;
    int64_t unaff_R14;
    undefined8 uStack1956;
    undefined8 uStack1948;
    undefined auStack1856 [874];
    undefined8 uStack982;
    undefined8 uStack974;

    while (auStack1856 < *(undefined **)(uint64_t *)(unaff_R14 + 0x10) ||
          auStack1856 == *(undefined **)(uint64_t *)(unaff_R14 + 0x10)) {
        fcn.00451a60();
    }
    uStack982 = 0x75ef1bba71e8f19;
    uStack974 = 0x93ea45013b7538b1;
    fcn.00454478();
    uStack1956 = 0x7b2d98d5c86cec78;
    uStack1948 = 0xc0d82d6268075bf0;
    fcn.00454478();
    for (iVar1 = 0; iVar1 < 0x3ce; iVar1 = iVar1 + 1) {
        *(uint8_t *)((int64_t)&uStack1956 + iVar1) =
             *(uint8_t *)((int64_t)&uStack1956 + iVar1) ^ *(uint8_t *)((int64_t)&uStack982 + iVar1);
    }
    fcn.00441960(0x3ce);
    return;
}
```

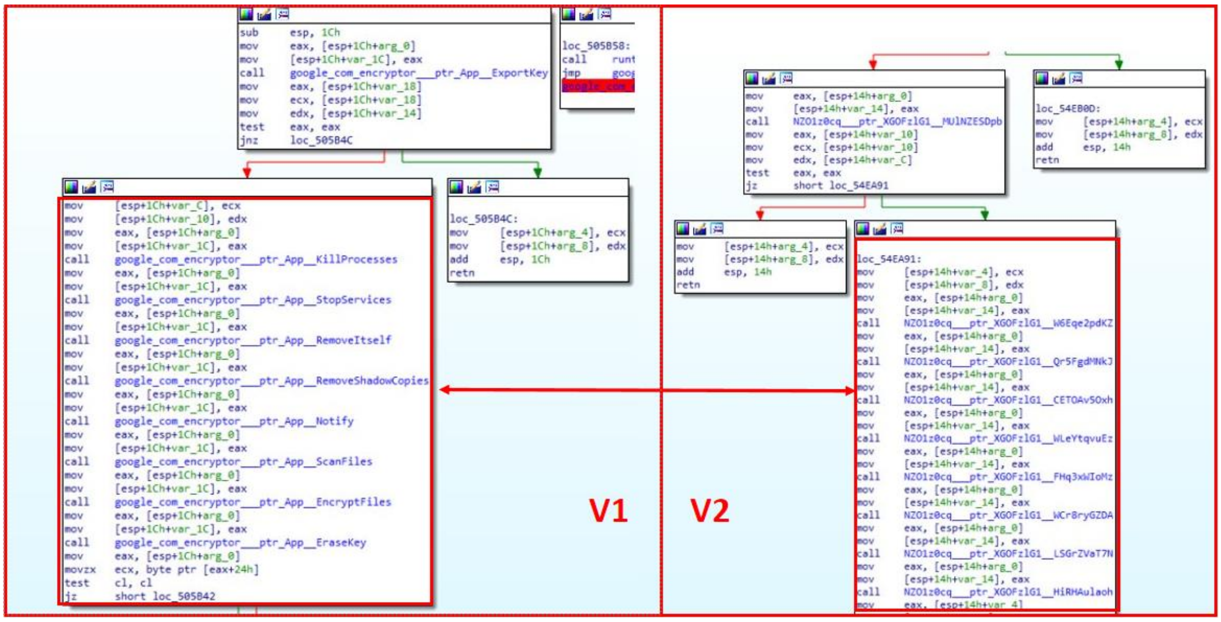
Figure 15: Strings decryption routines

The help command has also changed, it has more default values, the "-t" and "-skip" parameters have been removed, "-grant" has been added and "-no-clean" renamed to "-no-wipe"

Parameter	Description
-grant	Grant permissions to all files
-kill	Regex, names of the processes to kill. Default values: "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 of free space
-stop	Regex, names of the services to stop. Default values: "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|kavfsslp|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|SMTPSvc|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"

Table 4: Hive v2 parameters

The string obfuscation process does not impact the structure of the main function, following a comparison of these two versions.



```
sub     esp, 1Ch
mov     eax, [esp+1Ch+arg_0]
mov     [esp+1Ch+var_1C], eax
call    google_com_encryptor___ptr_App__ExportKey
mov     eax, [esp+1Ch+var_18]
mov     ecx, [esp+1Ch+var_18]
mov     edx, [esp+1Ch+var_14]
test    eax, eax
jnz     loc_505B4C
```

```
loc_505B58:
call    runt
jmp     goog
```

```
mov     [esp+1Ch+var_C], ecx
mov     [esp+1Ch+var_10], edx
mov     eax, [esp+1Ch+arg_0]
mov     [esp+1Ch+var_1C], eax
call    google_com_encryptor___ptr_App__KillProcesses
mov     eax, [esp+1Ch+arg_0]
mov     [esp+1Ch+var_1C], eax
call    google_com_encryptor___ptr_App__StopServices
mov     eax, [esp+1Ch+arg_0]
mov     [esp+1Ch+var_1C], eax
call    google_com_encryptor___ptr_App__RemoveItself
mov     eax, [esp+1Ch+arg_0]
mov     [esp+1Ch+var_1C], eax
call    google_com_encryptor___ptr_App__RemoveShadowCopies
mov     eax, [esp+1Ch+arg_0]
mov     [esp+1Ch+var_1C], eax
call    google_com_encryptor___ptr_App__Notify
mov     eax, [esp+1Ch+arg_0]
mov     [esp+1Ch+var_1C], eax
call    google_com_encryptor___ptr_App__ScanFiles
mov     eax, [esp+1Ch+arg_0]
mov     [esp+1Ch+var_1C], eax
call    google_com_encryptor___ptr_App__EncryptFiles
mov     eax, [esp+1Ch+arg_0]
mov     [esp+1Ch+var_1C], eax
call    google_com_encryptor___ptr_App__EraseKey
mov     eax, [esp+1Ch+arg_0]
movzx   ecx, byte ptr [eax+24h]
test    cl, cl
jz      short loc_505B42
```

```
loc_505B4C:
mov     [esp+1Ch+arg_4], ecx
mov     [esp+1Ch+arg_8], edx
add     esp, 1Ch
retn
```

```
mov     eax, [esp+14h+arg_0]
mov     [esp+14h+var_14], eax
call    NZO1z0cq___ptr_XGOFzlG1__MUlNZESDpb
mov     eax, [esp+14h+var_10]
mov     ecx, [esp+14h+var_10]
mov     edx, [esp+14h+var_C]
test    eax, eax
jz      short loc_54EA91
```

```
loc_54EB0D:
mov     [esp+14h+arg_4], ecx
mov     [esp+14h+arg_8], edx
add     esp, 14h
retn
```

```
mov     [esp+14h+arg_4], ecx
mov     [esp+14h+arg_8], edx
add     esp, 14h
retn
```

```
loc_54EA91:
mov     [esp+14h+var_4], ecx
mov     [esp+14h+var_8], edx
mov     eax, [esp+14h+arg_0]
mov     [esp+14h+var_14], eax
call    NZO1z0cq___ptr_XGOFzlG1__W6Eqe2pdKZ
mov     eax, [esp+14h+arg_0]
mov     [esp+14h+var_14], eax
call    NZO1z0cq___ptr_XGOFzlG1__Qr5FgdMNkJ
mov     eax, [esp+14h+arg_0]
mov     [esp+14h+var_14], eax
call    NZO1z0cq___ptr_XGOFzlG1__CETOAv5Oxh
mov     eax, [esp+14h+arg_0]
mov     [esp+14h+var_14], eax
call    NZO1z0cq___ptr_XGOFzlG1__WLeYtqvuEz
mov     eax, [esp+14h+arg_0]
mov     [esp+14h+var_14], eax
call    NZO1z0cq___ptr_XGOFzlG1__FHq3xWIoMz
mov     eax, [esp+14h+arg_0]
mov     [esp+14h+var_14], eax
call    NZO1z0cq___ptr_XGOFzlG1__WCr8ryGZDA
mov     eax, [esp+14h+arg_0]
mov     [esp+14h+var_14], eax
call    NZO1z0cq___ptr_XGOFzlG1__LSGrZVaT7N
mov     eax, [esp+14h+arg_0]
mov     [esp+14h+var_14], eax
call    NZO1z0cq___ptr_XGOFzlG1__HiRHAulaoh
mov     eax, [esp+14h+var_4]
```

Figure 16: Functions name comparison

Hive v3

Hash	8a461e66ae8a53ffe98d1e2e1dc52d015c11d67bd9ed09eb4be2124efd73ccd5
Threat	Ransomware
Brief Description	Hive Ransomware v3
SSDEEP	49152:gWVNVvSGbjmrb/T6vO90dL3BmAFd4A64nsfJuhQ8jmp4S3C5CEg+eNgiQJfOqAD:gWYQjPhQCmppnMfO

Table 5: Hive v3

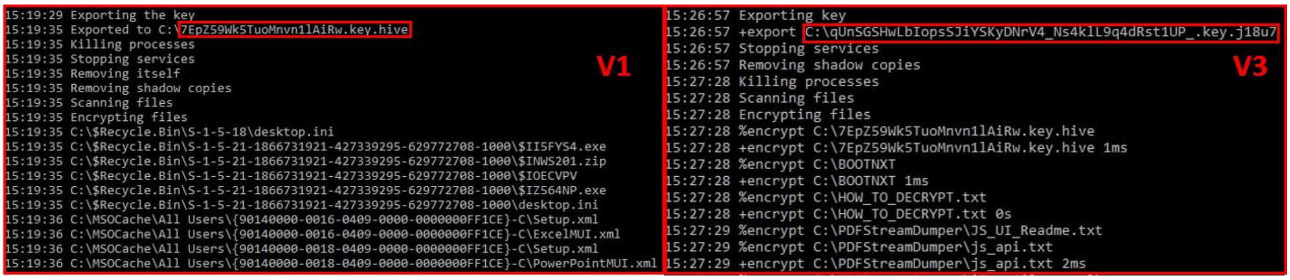
In this version the "-skip" parameter has been restored and, in another sample (3858e95bcf18c692f8321e3f8380c39684edb90bb622f37911144950602cea21), we found a new parameter named "scan":

Parameter	Description
-scan	Scan local network for shares

Table 6: Hive v3 Additional parameters

Comparing the logs from the v1 we can spot the following differences:

- The key name is longer and it has a random extension
- It shows the time elapsed for the encryption of each file



```
15:19:29 Exporting the key
15:19:35 Exported to C:\7EpZ59Wk5TuoMnvn1lAiRw.key.hive
15:19:35 Killing processes
15:19:35 Stopping services
15:19:35 Removing itself
15:19:35 Removing shadow copies
15:19:35 Scanning files
15:19:35 Encrypting files
15:19:35 C:\$Recycle.Bin\S-1-5-18\desktop.ini
15:19:35 C:\$Recycle.Bin\S-1-5-21-1866731921-427339295-629772708-1000\$II5FYS4.exe
15:19:35 C:\$Recycle.Bin\S-1-5-21-1866731921-427339295-629772708-1000\$INWS201.zip
15:19:35 C:\$Recycle.Bin\S-1-5-21-1866731921-427339295-629772708-1000\$IOECVPV
15:19:35 C:\$Recycle.Bin\S-1-5-21-1866731921-427339295-629772708-1000\$IZ564NP.exe
15:19:35 C:\$Recycle.Bin\S-1-5-21-1866731921-427339295-629772708-1000\desktop.ini
15:19:36 C:\MSOCache\All Users\{90140000-0016-0409-0000-0000000FF1CE}-C\Setup.xml
15:19:36 C:\MSOCache\All Users\{90140000-0016-0409-0000-0000000FF1CE}-C\ExcelMUI.xml
15:19:36 C:\MSOCache\All Users\{90140000-0018-0409-0000-0000000FF1CE}-C\Setup.xml
15:19:36 C:\MSOCache\All Users\{90140000-0018-0409-0000-0000000FF1CE}-C\PowerPointMUI.xml
```

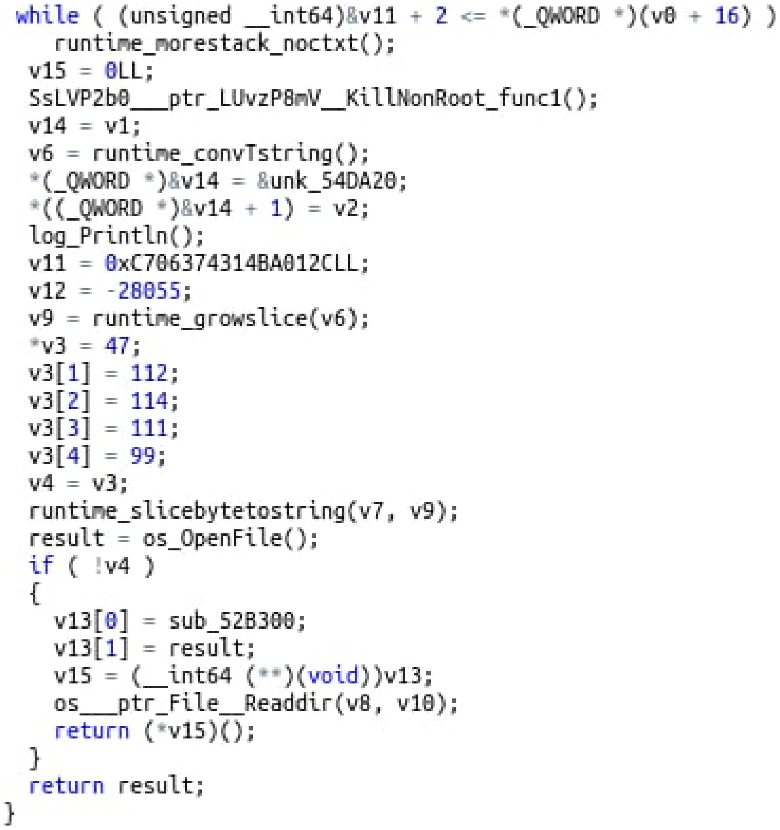
```
15:26:57 Exporting key
15:26:57 +export C:\qUnSGSHwLbIopsSJiYSKyDNrV4_Ns4klL9q4dRst1UP_.key.j18u7
15:26:57 Stopping services
15:26:57 Removing shadow copies
15:27:28 Killing processes
15:27:28 Scanning files
15:27:28 Encrypting files
15:27:28 %encrypt C:\7EpZ59Wk5TuoMnvn1lAiRw.key.hive
15:27:28 +encrypt C:\7EpZ59Wk5TuoMnvn1lAiRw.key.hive 1ms
15:27:28 %encrypt C:\BOOTNXT
15:27:28 +encrypt C:\BOOTNXT 1ms
15:27:28 %encrypt C:\HOW_TO_DECRYPT.txt
15:27:28 +encrypt C:\HOW_TO_DECRYPT.txt 0s
15:27:29 %encrypt C:\PDFStreamDumper\JS_UI_Readme.txt
15:27:29 %encrypt C:\PDFStreamDumper\js_api.txt
15:27:29 +encrypt C:\PDFStreamDumper\js_api.txt 2ms
```

Figure 17: Comparison of logs

Linux/FreeBSD version

The third version of the development of Hive ransomware saw the porting of the codebase for other operating systems, such as Linux/FreeBSD and ESXi.

The Linux (12389b8af28307fd09fe080fd89802b4e616ed4c961f464f95fdb4b3f0aaf185) and FreeBSD (bdf3d5f4f1b7c90dfc526340e917da9e188f04238e772049b2a97b4f88f711e3) versions are almost identical to the Windows one, despite the obvious OS differences. One of those differences is the following function "KillNonRoot" aimed at killing all non-root processes:



```
 while ( (unsigned __int64)&v11 + 2 <= *(_QWORD *)(v0 + 16) )
    runtime_morestack_noctxt();
  v15 = 0LL;
  SsLVP2b0___ptr_LUvzP8mV__KillNonRoot_func1();
  v14 = v1;
  v6 = runtime_convTstring();
  *(_QWORD *)&v14 = &unk_54DA20;
  *((_QWORD *)&v14 + 1) = v2;
  log_Println();
  v11 = 0xC706374314BA012CLL;
  v12 = -28055;
  v9 = runtime_growslice(v6);
  *v3 = 47;
  v3[1] = 112;
  v3[2] = 114;
  v3[3] = 111;
  v3[4] = 99;
  v4 = v3;
  runtime_slicebytetostring(v7, v9);
  result = os_OpenFile();
  if ( !v4 )
  {
    v13[0] = sub_52B300;
    v13[1] = result;
    v15 = (__int64 (**)(void))v13;
    os___ptr_File__Readdir(v8, v10);
    return (*v15)();
  }
  return result;
}
```

Figure 18: "KillNonRoot" Function

Hive v3 ESXI

Hash	822d89e7917d41a90f5f65bee75cad31fe13995e43f47ea9ea536862884efc25
Threat	Ransomware
Brief Description	Hive Ransomware v3
SSDEEP	3072:3Zp7gZzdfvjRCMj1Yk36ioyJ1zgjIlOhXYopNL+V7o0xvvkB/37Nt7xhew8A2Mz:P7gDj8S1Hlx14+opNClvk977ew8A2M

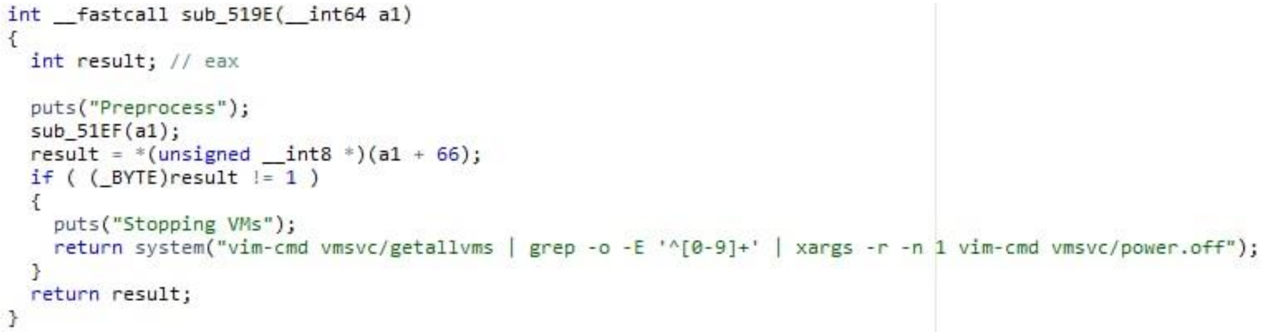
Table 7: Hive v3 ESXI

In this case, the malware is written in C/C++, in order to have a better compatibility with the target operating system, the strings are not obfuscated, and we have found some new parameters:

Parameter	Description
-no-stop	Don't stop virtual machines
-low-cpu	Single thread encryption

Table 8: Hive v3 ESXI Parameters

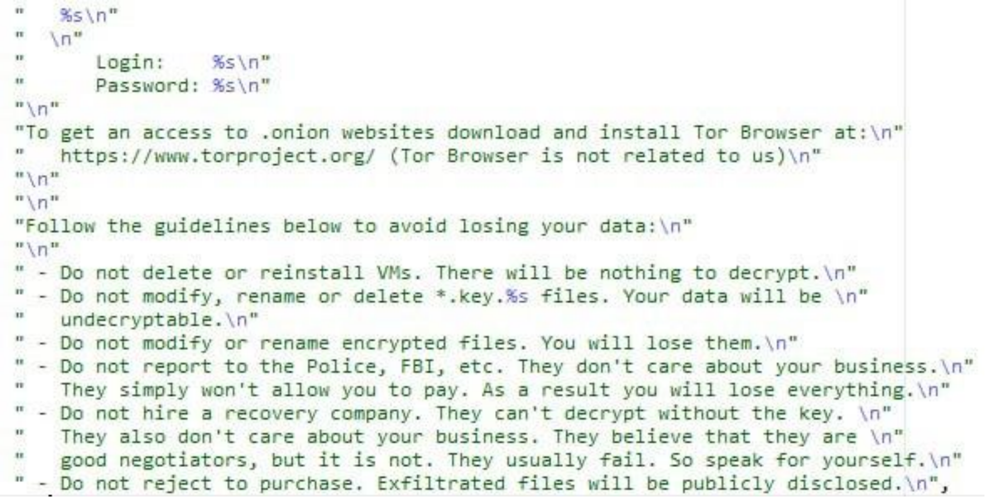
After the routine of exporting the keys already seen in the previous paragraphs, the sample stops all the running virtual machines in order to encrypt them without problems:



```
int __fastcall sub_519E(__int64 a1)
{
  int result; // eax

  puts("Preprocess");
  sub_51EF(a1);
  result = *(unsigned __int8 *)(a1 + 66);
  if ( (_BYTE)result != 1 )
  {
    puts("Stopping VMs");
    return system("vim-cmd vmsvc/getallvms | grep -o -E '^[0-9]+' | xargs -r -n 1 vim-cmd vmsvc/power.off");
  }
  return result;
}
```

Figure 19: "Stopping virtual machines

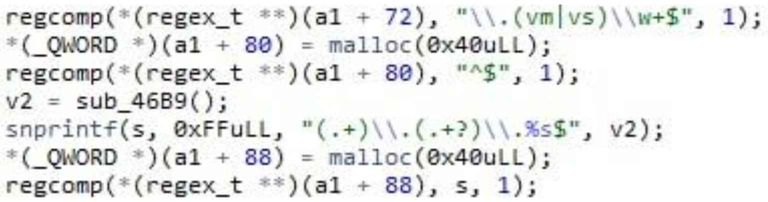
The ransom note contains also a reference to not delete or reinstall the virtual machines:



```
"   %s\n"
"  \n"
"       Login:    %s\n"
"       Password: %s\n"
"\n"
"To get an access to .onion websites download and install Tor Browser at:\n"
"   https://www.torproject.org/ (Tor Browser is not related to us)\n"
"\n"
"\n"
"Follow the guidelines below to avoid losing your data:\n"
"\n"
" - Do not delete or reinstall VMs. There will be nothing to decrypt.\n"
" - Do not modify, rename or delete *.key.%s files. Your data will be \n"
"   undecryptable.\n"
" - Do not modify or rename encrypted files. You will lose them.\n"
" - Do not report to the Police, FBI, etc. They don't care about your business.\n"
"   They simply won't allow you to pay. As a result you will lose everything.\n"
" - Do not hire a recovery company. They can't decrypt without the key. \n"
"   They also don't care about your business. They believe that they are \n"
"   good negotiators, but it is not. They usually fail. So speak for yourself.\n"
" - Do not reject to purchase. Exfiltrated files will be publicly disclosed.\n",
```

Figure 20: Ransom note

As said, the objective of this version is to encrypt the virtual machines hosted on the ESXi server, so, the malware goes to find the virtual machines deployed on the server, by using a custom regex aimed at finding the words “vm” or “vs”.



```
regcomp(*(regex_t **)(a1 + 72), "\\.(vm|vs)\\w+$", 1);
*(_QWORD *)(a1 + 80) = malloc(0x40uLL);
regcomp(*(regex_t **)(a1 + 80), "^$", 1);
v2 = sub_46B9();
snprintf(s, 0xFFuLL, "(.+)\\.(.+?)\\.%s$", v2);
*(_QWORD *)(a1 + 88) = malloc(0x40uLL);
regcomp(*(regex_t **)(a1 + 88), s, 1);
```

Figure 21: Regex for ESXI version

Hive v4

Hash	33aceb3dc0681a56226d4cfce32eee7a431e66f5c746a4d6dc7506a72b317277
Threat	Ransomware
Brief Description	Hive Ransomware v4
SSDEEP	49152:e2NiZPNNirb/T2vO90dL3BmAFd4A64nsfJk0NuXCdmTQb0/6VCrrPrsbg11VgWA:e2ANB04yIa0hsirubO

Table 9: Hive v4

The fourth version of Hive locker is an effort to obfuscate also the code. We haven't noticed new features or upgrades except for a more serious obfuscation of the code and changes in the details of the key generation and encryption.

In detail, this version adopts the control flow flattering obfuscation technique, which is largely adopted by many attackers, thanks to its actual effectiveness. Below an example of that technique:

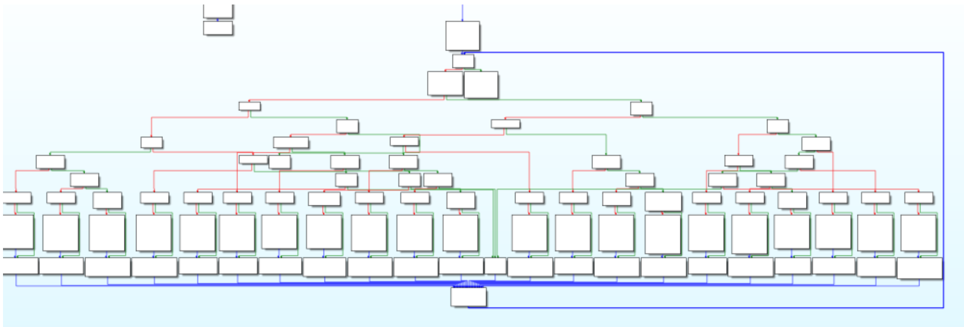


Figure 22: Control flattening obfuscation

Hive v5

The fifth version of hive represents a sort of revolution inside the entire codebase. In this version, the major differences include the changing of the base programming language and the refinement encryption algorithm.

Hash	b6b1ea26464c92c3d25956815c301caf6fa0da9723a2ef847e2bb9cd11563d8b
Threat	Ransomware
Brief Description	Hive Ransomware v5.2
SSDEEP	12288:BLF6OtM1z8JLbA689tSfvTvFSYIzp4yzhrWbttQfaa4Gxjzgdlo/AhwN/eh9z/E:BLF6gb0xqx9z/EO3BxhR

Table 10: Hive v5

Hive is now written in Rust and for this reason the difficulty has increased, along with a complex encryption scheme makes the analysis harder even for experienced analysts.

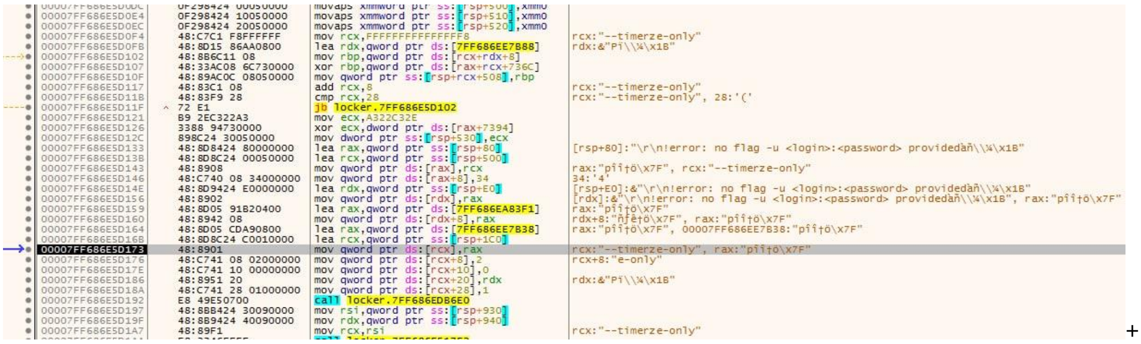
The refinement of the encryption process considers the passing from "***XOR+RSA***" of the previous versions, arriving to "***ECDH+Curve25519+XChaCha20-Poly1305***"

For this version we found the following parameters:

Parameter	Description
-no-local	Don't encrypt local files
-no-mounted	Don't encrypt on mounted network volumes
-no-discovery	Don't discover network volumes
-local-only	Encrypt only local files
-network-only	Encrypt only network volumes
-explicit-only	Encrypt specified folders
-min-size	Minimum file size
-timerze-only	N/A
-da	N/A

Table 11: Hive v5 parameters

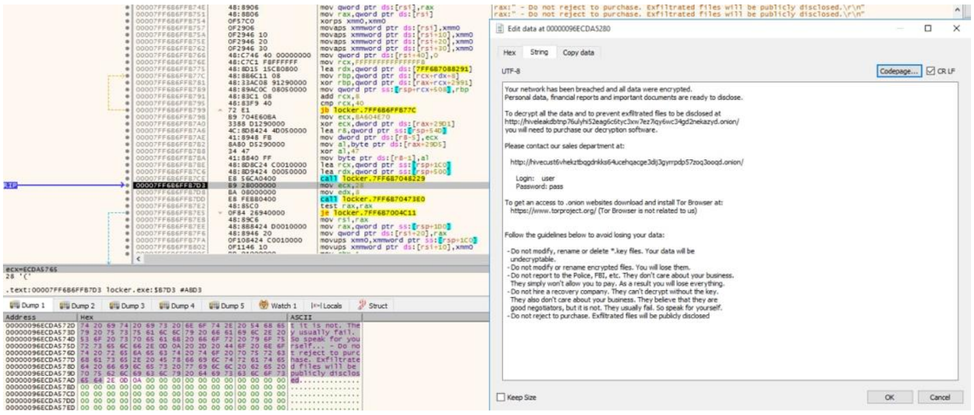
Once executed, the sample checks for the parameter "-u", which should contain the "*username:password*" used as credentials for the victim and written in the ransom note, if this unique parameter is missing, the program exits.



```
00007FF686E5D0DC  0F298424 00050000     movaps xmmword ptr ss:[rsp+500],xmm0
00007FF686E5D0E4  0F298424 10050000     movaps xmmword ptr ss:[rsp+510],xmm0
00007FF686E5D0EC  0F298424 20050000     movaps xmmword ptr ss:[rsp+520],xmm0
00007FF686E5D0F4  48:C7C1 F8FFFFFF      mov rcx,FFFFFFFFFFFFFFF8                  rcx:"--timerze-only"
00007FF686E5D0FB  48:8D15 86AA0800      lea rdx,qword ptr ds:[7FF686EE7B88]       rdx:&"Pï\\¼\x1B"
00007FF686E5D102  48:8B6C11 08          mov rbp,qword ptr ds:[rcx+rdx+8]
00007FF686E5D107  48:33AC08 6C730000    xor rbp,qword ptr ds:[rax+rcx+736C]
00007FF686E5D10F  48:89AC0C 08050000    mov qword ptr ss:[rsp+rcx+508],rbp
00007FF686E5D117  48:83C1 08            add rcx,8                                 rcx:"--timerze-only"
00007FF686E5D11B  48:83F9 28            cmp rcx,28                                rcx:"--timerze-only", 28:'('
00007FF686E5D11F  72 E1                 jb locker.7FF686E5D102
00007FF686E5D121  B9 2EC322A3           mov ecx,A322C32E
00007FF686E5D126  3388 94730000         xor ecx,dword ptr ds:[rax+7394]
00007FF686E5D12C  898C24 30050000       mov dword ptr ss:[rsp+530],ecx
00007FF686E5D133  48:8D8424 80000000    lea rax,qword ptr ss:[rsp+80]             [rsp+80]:"\r\n!error: no flag -u <login>:<password> providedàñ\\¼\x1B"
00007FF686E5D13B  48:8D8C24 00050000    lea rcx,qword ptr ss:[rsp+500]
00007FF686E5D143  48:8908               mov qword ptr ds:[rax],rcx                rax:"pîî†ö\x7F", rcx:"--timerze-only"
00007FF686E5D146  48:C740 08 34000000   mov qword ptr ds:[rax+8],34               34:'4'
00007FF686E5D14E  48:8D9424 E0000000    lea rdx,qword ptr ss:[rsp+E0]             [rsp+E0]:&"\r\n!error: no flag -u <login>:<password> providedàñ\\¼\x1B"
00007FF686E5D156  48:8902               mov qword ptr ds:[rdx],rax                [rdx]:&"\r\n!error: no flag -u <login>:<password> providedàñ\\¼\x1B", rax:"pîî†ö\x7F"
00007FF686E5D159  48:8D05 91B20400      lea rax,qword ptr ds:[7FF686EA83F1]       rax:"pîî†ö\x7F"
00007FF686E5D160  48:8942 08            mov qword ptr ds:[rdx+8],rax              rdx+8:"ñƒê†ö\x7F", rax:"pîî†ö\x7F"
00007FF686E5D164  48:8D05 CDA90800      lea rax,qword ptr ds:[7FF686EE7B38]       rax:"pîî†ö\x7F", 00007FF686EE7B38:"pîî†ö\x7F"
00007FF686E5D16B  48:8D8C24 C0010000    lea rcx,qword ptr ss:[rsp+1C0]
00007FF686E5D173  48:8901               mov qword ptr ds:[rcx],rax                rcx:"--timerze-only", rax:"pîî†ö\x7F"
00007FF686E5D176  48:C741 08 02000000   mov qword ptr ds:[rcx+8],2                rcx+8:"e-only"
00007FF686E5D17E  48:C741 10 00000000   mov qword ptr ds:[rcx+10],0
00007FF686E5D186  48:8951 20            mov qword ptr ds:[rcx+20],rdx             rdx:&"Pï\\¼\x1B"
00007FF686E5D18A  48:C741 28 01000000   mov qword ptr ds:[rcx+28],1
00007FF686E5D192  E8 49E50700           call locker.7FF686EDB6E0
00007FF686E5D197  48:8BB424 30090000    mov rsi,qword ptr ss:[rsp+930]
00007FF686E5D19F  48:8B9424 40090000    mov rdx,qword ptr ss:[rsp+940]
00007FF686E5D1A7  48:89F1               mov rcx,rsi                               rcx:"--timerze-only"
```

Figure 23: -u parameter

Even the routine to decrypt the ransom note changed. In this case, the protection of the ransom note relies on a XOR key.



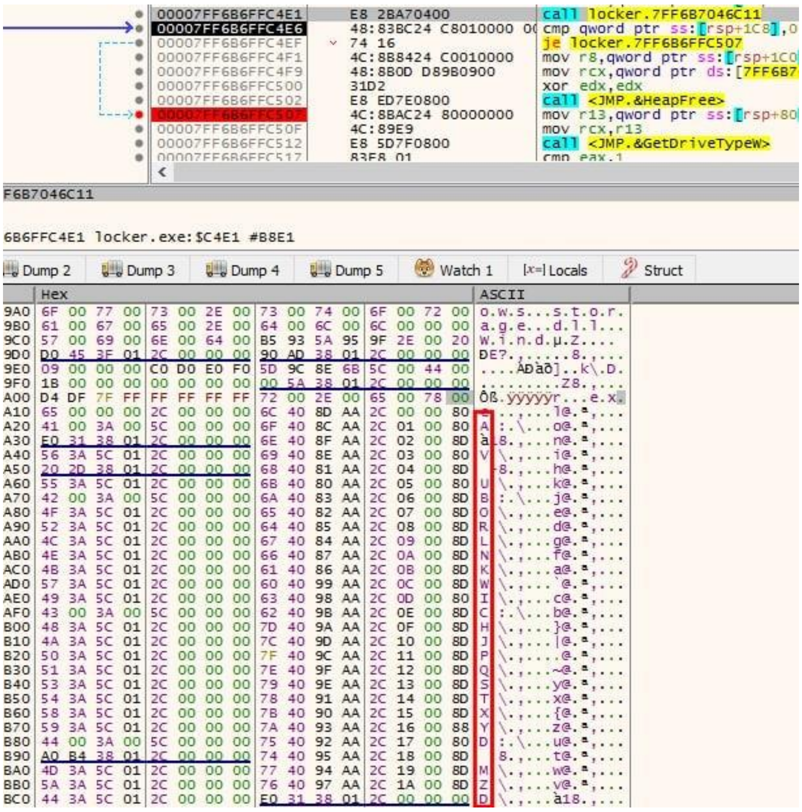
```
00007FF686FFB74E  48:8906               mov qword ptr ds:[rsi],rax            rax:" - Do not reject to purchase. Exfiltrated files will be publicly disclosed.\r\n"
00007FF686FFB751  48:8B06               mov rax,qword ptr ds:[rsi]            rax:" - Do not reject to purchase. Exfiltrated files will be publicly disclosed.\r\n"
00007FF686FFB754  0F57C0                xorps xmm0,xmm0
00007FF686FFB757  0F2906                movaps xmmword ptr ds:[rsi],xmm0
00007FF686FFB75A  0F2946 10             movaps xmmword ptr ds:[rsi+10],xmm0
00007FF686FFB75E  0F2946 20             movaps xmmword ptr ds:[rsi+20],xmm0
00007FF686FFB762  0F2946 30             movaps xmmword ptr ds:[rsi+30],xmm0
00007FF686FFB766  48:C746 40 00000000   mov qword ptr ds:[rsi+40],0
00007FF686FFB76E  48:C7C1 F8FFFFFF      mov rcx,FFFFFFFFFFFFFFF8
00007FF686FFB775  48:8D15 15CB0800      lea rdx,qword ptr ds:[7FF687088291]
00007FF686FFB77C  48:8B6C11 08          mov rbp,qword ptr ds:[rcx+rdx+8]
00007FF686FFB781  48:33AC08 91290000    xor rbp,qword ptr ds:[rax+rcx+2991]
00007FF686FFB789  48:89AC0C 08050000    mov qword ptr ss:[rsp+rcx+508],rbp
00007FF686FFB791  48:83C1 08            add rcx,8
00007FF686FFB795  48:83F9 40            cmp rcx,40
00007FF686FFB799  72 E1                 jb locker.7FF686FFB77C
00007FF686FFB79B  B9 704E60BA           mov ecx,BA604E70
00007FF686FFB7A0  3388 D1290000         xor ecx,dword ptr ds:[rax+29D1]
00007FF686FFB7A6  4C:8D8424 4D050000    lea r8,qword ptr ss:[rsp+54D]
00007FF686FFB7AE  41:8948 FB            mov dword ptr ds:[r8-5],ecx
00007FF686FFB7B2  8A80 D5290000         mov al,byte ptr ds:[rax+29D5]
00007FF686FFB7B8  34 47                 xor al,47
00007FF686FFB7BA  41:8840 FF            mov byte ptr ds:[r8-1],al
00007FF686FFB7BE  48:8D8C24 C0010000    lea rcx,qword ptr ss:[rsp+1C0]
00007FF686FFB7C6  48:8D9424 00050000    lea rdx,qword ptr ss:[rsp+500]
00007FF686FFB7CE  E8 56CA0400           call locker.7FF687048229
00007FF686FFB7D3  B9 28000000           mov ecx,28
00007FF686FFB7D8  BA 08000000           mov edx,8
00007FF686FFB7DD  E8 FEBB0400           call locker.7FF6870473E0
00007FF686FFB7E2  48:85C0               test rax,rax
00007FF686FFB7E5  0F84 26940000         je locker.7FF687004C11
00007FF686FFB7EB  48:89C6               mov rsi,rax
00007FF686FFB7EE  48:8B8424 D0010000    mov rax,qword ptr ss:[rsp+1D0]
00007FF686FFB7F6  48:8946 20            mov qword ptr ds:[rsi+20],rax
00007FF686FFB7FA  0F108424 C0010000     movups xmm0,xmmword ptr ss:[rsp+1C0]
00007FF686FFB802  0F1146 10             movups xmmword ptr ds:[rsi+10],xmm0

ecx=ECDA5765
28 '('

.text:00007FF686FFB7D3 locker.exe:$B7D3 #A8D3
```

```
Address           Hex                                              ASCII
00000096ECDA572D  74 20 69 74 20 69 73 20 6E 6F 74 2E 20 54 68 65  t it is not. The
00000096ECDA573D  79 20 75 73 75 61 6C 6C 79 20 66 61 69 6C 2E 20  y usually fail. 
00000096ECDA574D  53 6F 20 73 70 65 61 6B 20 66 6F 72 20 79 6F 75  So speak for you
00000096ECDA575D  72 73 65 6C 66 2E 0D 0A 20 2D 20 44 6F 20 6E 6F  rself... - Do no
00000096ECDA576D  74 20 72 65 6A 65 63 74 20 74 6F 20 70 75 72 63  t reject to purc
00000096ECDA577D  68 61 73 65 2E 20 45 78 66 69 6C 74 72 61 74 65  hase. Exfiltrate
00000096ECDA578D  64 20 66 69 6C 65 73 20 77 69 6C 6C 20 62 65 20  d files will be 
00000096ECDA579D  70 75 62 6C 69 63 6C 79 20 64 69 73 63 6C 6F 73  publicly disclos
00000096ECDA57AD  65 64 2E 0D 0A 00 00 00 00 00 00 00 00 00 00 00  ed..............
00000096ECDA57BD  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000096ECDA57CD  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000096ECDA57DD  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000096ECDA57ED  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
```

```
Your network has been breached and all data were encrypted.
Personal data, financial reports and important documents are ready to disclose.

To decrypt all the data and to prevent exfiltrated files to be disclosed at
http://hiveleakdbtnp76ulyhi52eag6c6tyc3xw7ez7iqy6wc34gd2nekazyd.onion/
you will need to purchase our decryption software.

Please contact our sales department at:

  http://hivecust6vhekztbqgdnkks64ucehqacge3dij3gyrrpdp57zoq3ooqd.onion/

    Login:    user
    Password: pass

To get an access to .onion websites download and install Tor Browser at:
  https://www.torproject.org/ (Tor Browser is not related to us)


Follow the guidelines below to avoid losing your data:

 - Do not modify, rename or delete *.key files. Your data will be
   undecryptable.
 - Do not modify or rename encrypted files. You will lose them.
 - Do not report to the Police, FBI, etc. They don't care about your business.
   They simply won't allow you to pay. As a result you will lose everything.
 - Do not hire a recovery company. They can't decrypt without the key.
   They also don't care about your business. They believe that they are
   good negotiators, but it is not. They usually fail. So speak for yourself.
 - Do not reject to purchase. Exfiltrated files will be publicly disclosed
```

Figure 24: Decrypted Ransom Note

Another update is the expansion on the other drives. The sample generates an array of drive labels and uses **GetDriveTypeW** to check if the path is invalid:

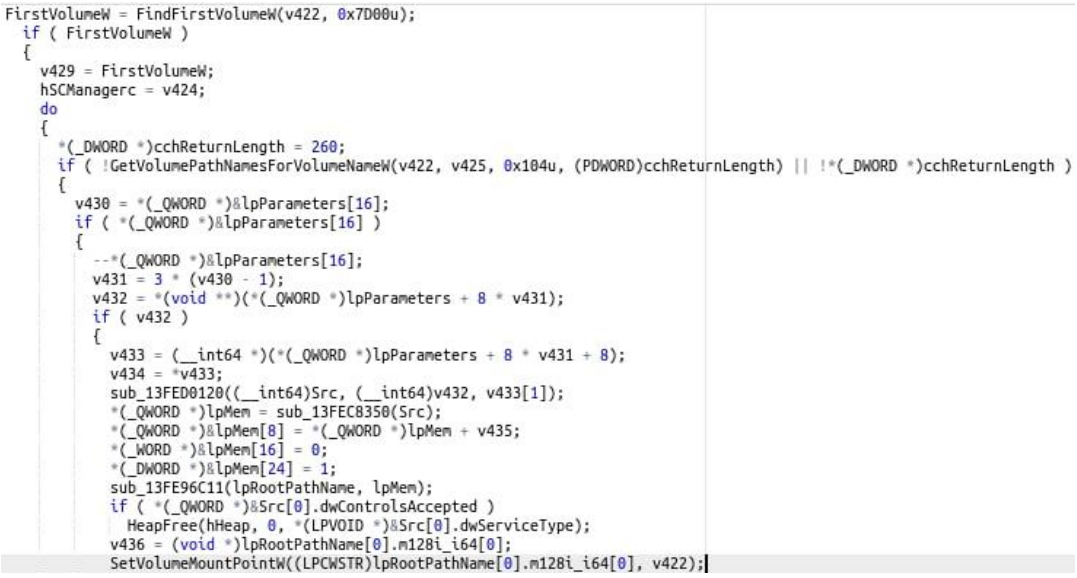


```
00007FF6B6FFC4E1   E8 2BA70400              call locker.7FF6B7046C11
00007FF6B6FFC4E6   48:83BC24 C8010000 00    cmp qword ptr ss:[rsp+1C8],0
00007FF6B6FFC4EF   74 16                    je locker.7FF6B6FFC507
00007FF6B6FFC4F1   4C:8B8424 C0010000       mov r8,qword ptr ss:[rsp+1C0
00007FF6B6FFC4F9   48:8B0D D89B0900         mov rcx,qword ptr ds:[7FF6B7
00007FF6B6FFC500   31D2                     xor edx,edx
00007FF6B6FFC502   E8 ED7E0800              call <JMP.&HeapFree>
00007FF6B6FFC507   4C:8BAC24 80000000       mov r13,qword ptr ss:[rsp+80
00007FF6B6FFC50F   4C:89E9                  mov rcx,r13
00007FF6B6FFC512   E8 5D7F0800              call <JMP.&GetDriveTypeW>
00007FF6B6FFC517   83F8 01                  cmp eax,1

F6B7046C11

6B6FFC4E1 locker.exe:$C4E1 #B8E1
```

```
    Hex                                              ASCII
9A0 6F 00 77 00 73 00 2E 00 73 00 74 00 6F 00 72 00  o.w.s...s.t.o.r.
9B0 61 00 67 00 65 00 2E 00 64 00 6C 00 6C 00 00 00  a.g.e...d.l.l...
9C0 57 00 69 00 6E 00 64 00 B5 93 5A 95 9F 2E 00 20  W.i.n.d.µ.Z....
9D0 D0 45 3F 01 2C 00 00 00 90 AD 38 01 2C 00 00 00  ÐE?.,.....8.,...
9E0 09 00 00 00 C0 D0 E0 F0 5D 9C 8E 6B 5C 00 44 00  ....ÀÐàð]..k\.D.
9F0 1B 00 00 00 00 00 00 00 00 5A 38 01 2C 00 00 00  .........Z8.,...
A00 D4 DF 7F FF FF FF FF FF 72 00 2E 00 65 00 78 00  Ôß.ÿÿÿÿÿr...e.x.
A10 65 00 00 00 2C 00 00 00 6C 40 8D AA 2C 00 00 80  e...,...l@.ª,...
A20 41 00 3A 00 5C 00 00 00 6F 40 8C AA 2C 01 00 80  A.:.\...o@.ª,...
A30 E0 31 38 01 2C 00 00 00 6E 40 8F AA 2C 02 00 8D  à18.,...n@.ª,...
A40 56 3A 5C 01 2C 00 00 00 69 40 8E AA 2C 03 00 80  V:\.,...i@.ª,...
A50 20 2D 38 01 2C 00 00 00 68 40 81 AA 2C 04 00 8D   -8.,...h@.ª,...
A60 55 3A 5C 01 2C 00 00 00 6B 40 80 AA 2C 05 00 80  U:\.,...k@.ª,...
A70 42 00 3A 00 5C 00 00 00 6A 40 83 AA 2C 06 00 8D  B.:.\...j@.ª,...
A80 4F 3A 5C 01 2C 00 00 00 65 40 82 AA 2C 07 00 8D  O:\.,...e@.ª,...
A90 52 3A 5C 01 2C 00 00 00 64 40 85 AA 2C 08 00 8D  R:\.,...d@.ª,...
AA0 4C 3A 5C 01 2C 00 00 00 67 40 84 AA 2C 09 00 8D  L:\.,...g@.ª,...
AB0 4E 3A 5C 01 2C 00 00 00 66 40 87 AA 2C 0A 00 8D  N:\.,...f@.ª,...
AC0 4B 3A 5C 01 2C 00 00 00 61 40 86 AA 2C 0B 00 8D  K:\.,...a@.ª,...
AD0 57 3A 5C 01 2C 00 00 00 60 40 99 AA 2C 0C 00 8D  W:\.,...`@.ª,...
AE0 49 3A 5C 01 2C 00 00 00 63 40 98 AA 2C 0D 00 80  I:\.,...c@.ª,...
AF0 43 00 3A 00 5C 00 00 00 62 40 9B AA 2C 0E 00 8D  C.:.\...b@.ª,...
B00 48 3A 5C 01 2C 00 00 00 7D 40 9A AA 2C 0F 00 8D  H:\.,...}@.ª,...
B10 4A 3A 5C 01 2C 00 00 00 7C 40 9D AA 2C 10 00 8D  J:\.,...|@.ª,...
B20 50 3A 5C 01 2C 00 00 00 7F 40 9C AA 2C 11 00 8D  P:\.,....@.ª,...
B30 51 3A 5C 01 2C 00 00 00 7E 40 9F AA 2C 12 00 8D  Q:\.,...~@.ª,...
B40 53 3A 5C 01 2C 00 00 00 79 40 9E AA 2C 13 00 8D  S:\.,...y@.ª,...
B50 54 3A 5C 01 2C 00 00 00 78 40 91 AA 2C 14 00 8D  T:\.,...x@.ª,...
B60 58 3A 5C 01 2C 00 00 00 7B 40 90 AA 2C 15 00 8D  X:\.,...{@.ª,...
B70 59 3A 5C 01 2C 00 00 00 7A 40 93 AA 2C 16 00 88  Y:\.,...z@.ª,...
B80 44 00 3A 00 5C 00 00 00 75 40 92 AA 2C 17 00 80  D.:.\...u@.ª,...
B90 A0 B4 38 01 2C 00 00 00 74 40 95 AA 2C 18 00 8D   ´8.,...t@.ª,...
BA0 4D 3A 5C 01 2C 00 00 00 77 40 94 AA 2C 19 00 8D  M:\.,...w@.ª,...
BB0 5A 3A 5C 01 2C 00 00 00 76 40 97 AA 2C 1A 00 8D  Z:\.,...v@.ª,...
BC0 44 3A 5C 01 2C 00 00 00 E0 31 38 01 2C 00 00 00  D:\.,...à18.,...
```

Figure 25: Finding of logical drives

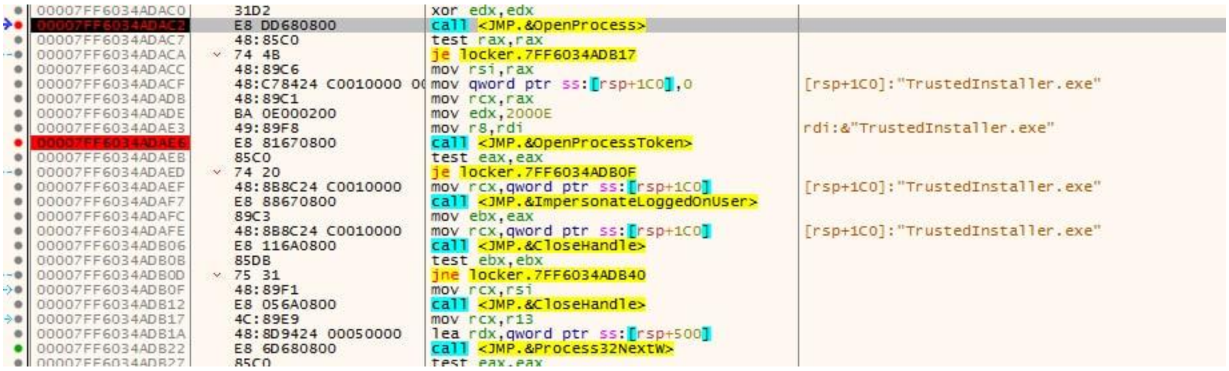
Once the attached volumes are found, it calls **FindFirstVolumeW** and **SetVolumeMountPointW** to mount eventual unmounted volumes:



```
FirstVolumeW = FindFirstVolumeW(v422, 0x7D00u);
  if ( FirstVolumeW )
  {
    v429 = FirstVolumeW;
    hSCManagerc = v424;
    do
    {
      *(_DWORD *)cchReturnLength = 260;
      if ( !GetVolumePathNamesForVolumeNameW(v422, v425, 0x104u, (PDWORD)cchReturnLength) || !*(_DWORD *)cchReturnLength )
      {
        v430 = *(_QWORD *)&lpParameters[16];
        if ( *(_QWORD *)&lpParameters[16] )
        {
          --*(_QWORD *)&lpParameters[16];
          v431 = 3 * (v430 - 1);
          v432 = *(void **)(*(_QWORD *)lpParameters + 8 * v431);
          if ( v432 )
          {
            v433 = (__int64 *)(*(_QWORD *)lpParameters + 8 * v431 + 8);
            v434 = *v433;
            sub_13FED0120((__int64)Src, (__int64)v432, v433[1]);
            *(_QWORD *)lpMem = sub_13FEC8350(Src);
            *(_QWORD *)&lpMem[8] = *(_QWORD *)lpMem + v435;
            *(_WORD *)&lpMem[16] = 0;
            *(_DWORD *)&lpMem[24] = 1;
            sub_13FE96C11(lpRootPathName, lpMem);
            if ( *(_QWORD *)&Src[0].dwControlsAccepted )
              HeapFree(hHeap, 0, *(LPVOID *)&Src[0].dwServiceType);
            v436 = (void *)lpRootPathName[0].m128i_i64[0];
            SetVolumeMountPointW((LPCWSTR)lpRootPathName[0].m128i_i64[0], v422);
```

Figure 26: Mounting available volumes

After that, the operation of privilege escalation is performed though abusing the “TrustedInstaller” service to recover its access token. In this way, the malware is able to read and write files with the same privileges of TrustedInstaller Group.



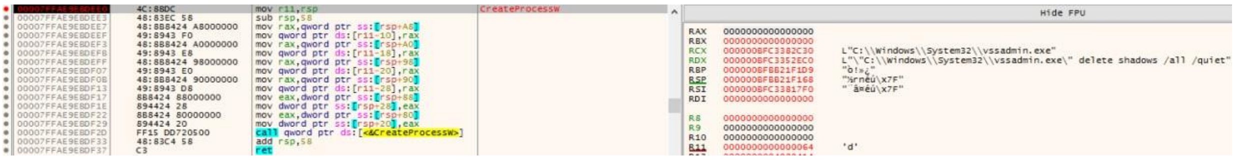
```
00007FF6034ADAC0    31D2                  xor edx,edx
00007FF6034ADAC2    E8 DD680800           call <JMP.&OpenProcess>
00007FF6034ADAC7    48:85C0               test rax,rax
00007FF6034ADACA    74 4B                 je locker.7FF6034ADB17
00007FF6034ADACC    48:89C6               mov rsi,rax
00007FF6034ADACF    48:C78424 C0010000 0  mov qword ptr ss:[rsp+1C0],0       [rsp+1C0]:"TrustedInstaller.exe"
00007FF6034ADADB    48:89C1               mov rcx,rax
00007FF6034ADADE    BA 0E000200           mov edx,2000E
00007FF6034ADAE3    49:89F8               mov r8,rdi                         rdi:&"TrustedInstaller.exe"
00007FF6034ADAE6    E8 81670800           call <JMP.&OpenProcessToken>
00007FF6034ADAEB    85C0                  test eax,eax
00007FF6034ADAED    74 20                 je locker.7FF6034ADB0F
00007FF6034ADAEF    48:8B8C24 C0010000    mov rcx,qword ptr ss:[rsp+1C0]     [rsp+1C0]:"TrustedInstaller.exe"
00007FF6034ADAF7    E8 88670800           call <JMP.&ImpersonateLoggedOnUser>
00007FF6034ADAFC    89C3                  mov ebx,eax
00007FF6034ADAFE    48:8B8C24 C0010000    mov rcx,qword ptr ss:[rsp+1C0]     [rsp+1C0]:"TrustedInstaller.exe"
00007FF6034ADB06    E8 116A0800           call <JMP.&CloseHandle>
00007FF6034ADB0B    85DB                  test ebx,ebx
00007FF6034ADB0D    75 31                 jne locker.7FF6034ADB40
00007FF6034ADB0F    48:89F1               mov rcx,rsi
00007FF6034ADB12    E8 056A0800           call <JMP.&CloseHandle>
00007FF6034ADB17    4C:89E9               mov rcx,r13
00007FF6034ADB1A    48:8D9424 00050000    lea rdx,qword ptr ss:[rsp+500]
00007FF6034ADB22    E8 6D680800           call <JMP.&Process32NextW>
00007FF6034ADB27    85C0                  test eax,eax
```

Figure 27: Retrieving TrustedInstaller access token

Moreover, in the previous versions, we saw bat files and other escamotages for the erasing of the backup mechanisms provided by the Microsoft Environment.

In the fifth version analyzed, there are the following tricks:

- vssadmin.exe delete shadows /all /quiet

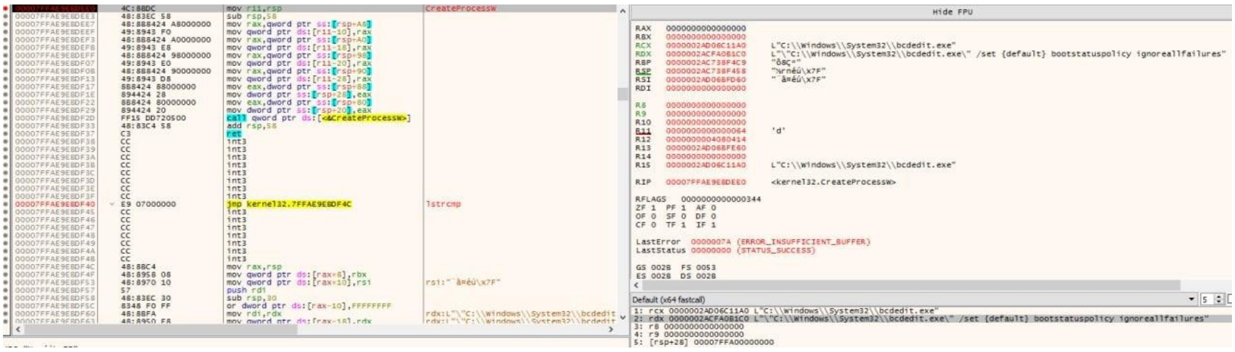


```
00007FFAE9EBDEE0    4C:8BDC               mov r11,rsp                         CreateProcessW
00007FFAE9EBDEE3    48:83EC 58            sub rsp,58
00007FFAE9EBDEE7    48:8B8424 A8000000    mov rax,qword ptr ss:[rsp+A8]
00007FFAE9EBDEEF    49:8943 F0            mov qword ptr ds:[r11-10],rax
00007FFAE9EBDEF3    48:8B8424 A0000000    mov rax,qword ptr ss:[rsp+A0]
00007FFAE9EBDEFB    49:8943 E8            mov qword ptr ds:[r11-18],rax
00007FFAE9EBDEFF    48:8B8424 98000000    mov rax,qword ptr ss:[rsp+98]
00007FFAE9EBDF07    49:8943 E0            mov qword ptr ds:[r11-20],rax
00007FFAE9EBDF0B    48:8B8424 90000000    mov rax,qword ptr ss:[rsp+90]
00007FFAE9EBDF13    49:8943 D8            mov qword ptr ds:[r11-28],rax
00007FFAE9EBDF17    8B8424 88000000       mov eax,dword ptr ss:[rsp+88]
00007FFAE9EBDF1E    894424 28             mov dword ptr ss:[rsp+28],eax
00007FFAE9EBDF22    8B8424 80000000       mov eax,dword ptr ss:[rsp+80]
00007FFAE9EBDF29    894424 20             mov dword ptr ss:[rsp+20],eax
00007FFAE9EBDF2D    FF15 DD720500         call qword ptr ds:[<&CreateProcessW>]
00007FFAE9EBDF33    48:83C4 58            add rsp,58
00007FFAE9EBDF37    C3                    ret
```

```
Hide FPU

RAX   0000000000000000
RBX   0000000000000000
RCX   000000BFC3382C30   L"C:\\Windows\\System32\\vssadmin.exe"
RDX   000000BFC3352EC0   L"\"C:\\Windows\\System32\\vssadmin.exe\" delete shadows /all /quiet"
RBP   000000BFBB21F1D9   "ò!»¿"
RSP   000000BFBB21F168   "½rnéû\x7F"
RSI   000000BFC33817F0   "¨â¤éû\x7F"
RDI   0000000000000000

R8    0000000000000000
R9    0000000000000000
R10   0000000000000000
R11   0000000000000064   'd'
```

Figure 28: vssadmin

- bcedit /set {default} bootstatuspolicy ignoreallfailures

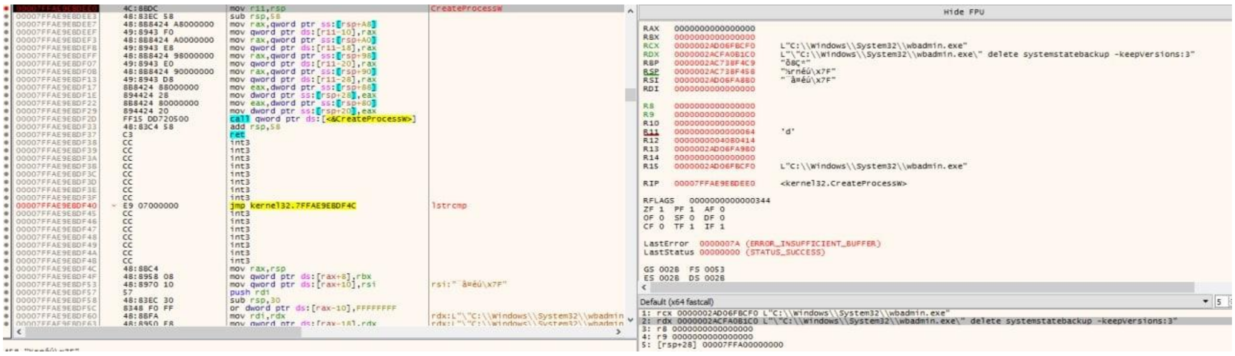


```
00007FFAE9EBDEE0    4C:8BDC               mov r11,rsp                         CreateProcessW
00007FFAE9EBDEE3    48:83EC 58            sub rsp,58
00007FFAE9EBDEE7    48:8B8424 A8000000    mov rax,qword ptr ss:[rsp+A8]
00007FFAE9EBDEEF    49:8943 F0            mov qword ptr ds:[r11-10],rax
00007FFAE9EBDEF3    48:8B8424 A0000000    mov rax,qword ptr ss:[rsp+A0]
00007FFAE9EBDEFB    49:8943 E8            mov qword ptr ds:[r11-18],rax
00007FFAE9EBDEFF    48:8B8424 98000000    mov rax,qword ptr ss:[rsp+98]
00007FFAE9EBDF07    49:8943 E0            mov qword ptr ds:[r11-20],rax
00007FFAE9EBDF0B    48:8B8424 90000000    mov rax,qword ptr ss:[rsp+90]
00007FFAE9EBDF13    49:8943 D8            mov qword ptr ds:[r11-28],rax
00007FFAE9EBDF17    8B8424 88000000       mov eax,dword ptr ss:[rsp+88]
00007FFAE9EBDF1E    894424 28             mov dword ptr ss:[rsp+28],eax
00007FFAE9EBDF22    8B8424 80000000       mov eax,dword ptr ss:[rsp+80]
00007FFAE9EBDF29    894424 20             mov dword ptr ss:[rsp+20],eax
00007FFAE9EBDF2D    FF15 DD720500         call qword ptr ds:[<&CreateProcessW>]
00007FFAE9EBDF33    48:83C4 58            add rsp,58
00007FFAE9EBDF37    C3                    ret
00007FFAE9EBDF38    CC                    int3
00007FFAE9EBDF39    CC                    int3
00007FFAE9EBDF3A    CC                    int3
00007FFAE9EBDF3B    CC                    int3
00007FFAE9EBDF3C    CC                    int3
00007FFAE9EBDF3D    CC                    int3
00007FFAE9EBDF3E    CC                    int3
00007FFAE9EBDF3F    CC                    int3
00007FFAE9EBDF40    E9 07000000           jmp kernel32.7FFAE9EBDF4C           lstrcmp
00007FFAE9EBDF45    CC                    int3
00007FFAE9EBDF46    CC                    int3
00007FFAE9EBDF47    CC                    int3
00007FFAE9EBDF48    CC                    int3
00007FFAE9EBDF49    CC                    int3
00007FFAE9EBDF4A    CC                    int3
00007FFAE9EBDF4B    CC                    int3
00007FFAE9EBDF4C    48:8BC4               mov rax,rsp
00007FFAE9EBDF4F    48:8958 08            mov qword ptr ds:[rax+8],rbx
00007FFAE9EBDF53    48:8970 10            mov qword ptr ds:[rax+10],rsi       rsi:"¨â¤éû\x7F"
00007FFAE9EBDF57    57                    push rdi
00007FFAE9EBDF58    48:83EC 30            sub rsp,30
00007FFAE9EBDF5C    8348 F0 FF            or dword ptr ds:[rax-10],FFFFFFFF
00007FFAE9EBDF60    48:8BFA               mov rdi,rdx                         rdx:L"\"C:\\Windows\\System32\\bcdedit
00007FFAE9EBDF63    48:8950 E8            mov qword ptr ds:[rax-18],rdx       rdx:L"\"C:\\Windows\\System32\\bcdedit
```

```
Hide FPU

RAX   0000000000000000
RBX   0000000000000000
RCX   0000002AD06C11A0   L"C:\\Windows\\System32\\bcdedit.exe"
RDX   0000002ACFA0B1C0   L"\"C:\\Windows\\System32\\bcdedit.exe\" /set {default} bootstatuspolicy ignoreallfailures"
RBP   0000002AC738F4C9   "ô8Ç°"
RSP   0000002AC738F458   "½rnéû\x7F"
RSI   0000002AD06BFD60   "¨â¤éû\x7F"
RDI   0000000000000000

R8    0000000000000000
R9    0000000000000000
R10   0000000000000000
R11   0000000000000064   'd'
R12   0000000004080414
R13   0000002AD06BFE60
R14   0000000000000000
R15   0000002AD06C11A0   L"C:\\Windows\\System32\\bcdedit.exe"

RIP   00007FFAE9EBDEE0   <kernel32.CreateProcessW>

RFLAGS   0000000000000344
ZF 1  PF 1  AF 0
OF 0  SF 0  DF 0
CF 0  TF 1  IF 1

LastError  0000007A (ERROR_INSUFFICIENT_BUFFER)
LastStatus 00000000 (STATUS_SUCCESS)

GS 002B  FS 0053
ES 002B  DS 002B

Default (x64 fastcall)   5
1: rcx 0000002AD06C11A0 L"C:\\Windows\\System32\\bcdedit.exe"
2: rdx 0000002ACFA0B1C0 L"\"C:\\Windows\\System32\\bcdedit.exe\" /set {default} bootstatuspolicy ignoreallfailures"
3: r8 0000000000000000
4: r9 0000000000000000
5: [rsp+28] 00007FFA00000000
```

Figure 29: bcedit

- wbadmin delete systemstatebackup –keepversions:3



```
00007FFAE9E8DEE0   4C:8BDC              mov r11,rsp                              CreateProcessW
00007FFAE9E8DEE3   48:83EC 58           sub rsp,58
00007FFAE9E8DEE7   48:8B8424 A8000000   mov rax,qword ptr ss:[rsp+A8]
00007FFAE9E8DEEF   49:8943 F0           mov qword ptr ds:[r11-10],rax
00007FFAE9E8DEF3   48:8B8424 A0000000   mov rax,qword ptr ss:[rsp+A0]
00007FFAE9E8DEFB   49:8943 E8           mov qword ptr ds:[r11-18],rax
00007FFAE9E8DEFF   48:8B8424 98000000   mov rax,qword ptr ss:[rsp+98]
00007FFAE9E8DF07   49:8943 E0           mov qword ptr ds:[r11-20],rax
00007FFAE9E8DF0B   48:8B8424 90000000   mov rax,qword ptr ss:[rsp+90]
00007FFAE9E8DF13   49:8943 D8           mov qword ptr ds:[r11-28],rax
00007FFAE9E8DF17   8B8424 88000000      mov eax,dword ptr ss:[rsp+88]
00007FFAE9E8DF1E   894424 28            mov dword ptr ss:[rsp+28],eax
00007FFAE9E8DF22   8B8424 80000000      mov eax,dword ptr ss:[rsp+80]
00007FFAE9E8DF29   894424 20            mov dword ptr ss:[rsp+20],eax
00007FFAE9E8DF2D   FF15 DD720500        call qword ptr ds:[<&CreateProcessW>]
00007FFAE9E8DF33   48:83C4 58           add rsp,58
00007FFAE9E8DF37   C3                   ret
00007FFAE9E8DF38   CC                   int3
00007FFAE9E8DF39   CC                   int3
00007FFAE9E8DF3A   CC                   int3
00007FFAE9E8DF3B   CC                   int3
00007FFAE9E8DF3C   CC                   int3
00007FFAE9E8DF3D   CC                   int3
00007FFAE9E8DF3E   CC                   int3
00007FFAE9E8DF3F   CC                   int3
00007FFAE9E8DF40 ˅ E9 07000000          jmp kernel32.7FFAE9E8DF4C                lstrcmp
00007FFAE9E8DF45   CC                   int3
00007FFAE9E8DF46   CC                   int3
00007FFAE9E8DF47   CC                   int3
00007FFAE9E8DF48   CC                   int3
00007FFAE9E8DF49   CC                   int3
00007FFAE9E8DF4A   CC                   int3
00007FFAE9E8DF4B   CC                   int3
00007FFAE9E8DF4C   48:8BC4              mov rax,rsp
00007FFAE9E8DF4F   48:8958 08           mov qword ptr ds:[rax+8],rbx
00007FFAE9E8DF53   48:8970 10           mov qword ptr ds:[rax+10],rsi            rsi:"¨â¤éû\x7F"
00007FFAE9E8DF57   57                   push rdi
00007FFAE9E8DF58   48:83EC 30           sub rsp,30
00007FFAE9E8DF5C   8348 F0 FF           or dword ptr ds:[rax-10],FFFFFFFF
00007FFAE9E8DF60   48:8BFA              mov rdi,rdx                              rdx:L"\"C:\\Windows\\System32\\wbadmin
00007FFAE9E8DF63   48:8950 E8           mov qword ptr ds:[rax-18],rdx            rdx:L"\"C:\\Windows\\System32\\wbadmin
```

```
RAX     0000000000000000
RBX     0000000000000000
RCX     0000002AD06FBCF0     L"C:\\Windows\\System32\\wbadmin.exe"
RDX     0000002ACFA0B1C0     L"\"C:\\Windows\\System32\\wbadmin.exe\" delete systemstatebackup -keepVersions:3"
RBP     0000002AC738F4C9     "õBÇ±"
RSP     0000002AC738F458     "½rnéû\x7F"
RSI     0000002AD06FA8B0     "¨â¤éû\x7F"
RDI     0000000000000000

R8      0000000000000000
R9      0000000000000000
R10     0000000000000000
R11     0000000000000064     'd'
R12     0000000004080414
R13     0000002AD06FA980
R14     0000000000000000
R15     0000002AD06FBCF0     L"C:\\Windows\\System32\\wbadmin.exe"

RIP     00007FFAE9E8DEE0     <kernel32.CreateProcessW>

RFLAGS    0000000000000344
ZF 1  PF 1  AF 0
OF 0  SF 0  DF 0
CF 0  TF 1  IF 1

LastError  0000007A (ERROR_INSUFFICIENT_BUFFER)
LastStatus 00000000 (STATUS_SUCCESS)

GS 002B  FS 0053
ES 002B  DS 002B
```

```
1: rcx 0000002AD06FBCF0 L"C:\\Windows\\System32\\wbadmin.exe"
2: rdx 0000002ACFA0B1C0 L"\"C:\\Windows\\System32\\wbadmin.exe\" delete systemstatebackup -keepVersions:3"
3: r8 0000000000000000
4: r9 0000000000000000
5: [rsp+28] 00007FFA00000000
```

Figure 30: wbadmin

Conclusion

Hive threat actor is one of the most sophisticated active threats. It does not care about the target, the only objective is to maximize the illicit profits, even by causing the interruption of critical services. The continuous development of the ransomware payload should not be underestimated, and in the same way organizations must upgrade their cyber protections.

We at Yoroi ZLab believe that collaboration and sharing more information possible about attackers is the right way to pursue to defend these entities. We know that having to deal with these threats is challenging, so we are pointing to create the best expertise needed to handle such incidents whether they happen.

In conclusion, we need to create a solid and reliable strategy to defend our customers. we encourage our customers to make assessments and awareness campaigns for their employees. The goal of the Defence Center of Yoroi is to guarantee the best protection in every phase of the attack, starting from the continuous monitoring arriving to the Incident Response engagements.

Appendix

Indicators of Compromise

Hive v1

- 88f7544a29a2ceb175a135d9fa221cbfd3e8c71f32dd6b09399717f85ea9afd1 (Sample)
- d158f9d53e7c37eadd3b5cc1b82d095f61484e47eda2c36d9d35f31c0b4d3ff8 (shadow.bat)

Hive v2:

- 25bfec0c3c81ab55cf85a57367c14cc6803a03e2e9b4afd72e7bbca9420fe7c5

Hive v3

- 8a461e66ae8a53ffe98d1e2e1dc52d015c11d67bd9ed09eb4be2124efd73ccd5

Hive v3 Linux/FreeBSD

- 12389b8af28307fd09fe080fd89802b4e616ed4c961f464f95fdb4b3f0aaf185 (Linux)
- Bdf3d5f4f1b7c90dfc526340e917da9e188f04238e772049b2a97b4f88f711e3 (FreeBSD)

Hive v3 ESXI

- 822d89e7917d41a90f5f65bee75cad31fe13995e43f47ea9ea536862884efc25

Hive v4

- 33aceb3dc0681a56226d4cfce32eee7a431e66f5c746a4d6dc7506a72b317277

Hive v5.2

- b6b1ea26464c92c3d25956815c301caf6fa0da9723a2ef847e2bb9cd11563d8b

Yara Rules

```
rule hive_v1_32_win
{
  strings:
    $1 =
{648b0d140000008b89000000003b61080f86e401000083ec40e8?2f?feff8b04248b4c240485c90f8556010000b941000000
31d231db8d2d?4??6300eb0341d1e883f95a0f8f29010000a90100000074ed895c2434896c243c894c2424895424308944 24
288d44242c}
  condition:
    $1 and uint16(0) == 0x5A4D
}
```

```
rule hive_v1_64_win
{
  strings:
    $1 = { 65 4? 8b 0c ?5 28 00 00 00 4? 8b 89 00 00 00 00 4? 3b 61 10 0f 86 ?? ?? ?? ?? 4? 83 ec 40 4? 89 6c ?4 38 4? 8d 6c ?4
38 4? 8b 44 ?4 48 4? 89 04 ?4 e8 ?? ?? ?? ?? 4? 8b 44 ?4 08 4? 8b 4c ?4 10 4? 83 7c ?4 08 00 0f 85 ?? ?? ?? ?? 4? 89 44 ?4 20 4?
89 4c ?4 18 4? 8b 44 ?4 48 4? 89 04 ?4 90 e8 ?? ?? ?? ?? 4? 8b 44 ?4 48 4? 89 04 ?4 e8 ?? ?? ?? ?? 4? 8b 44 ?4 48 4? 89 04 ?4 0f
1f 40 00 e8 ?? ?? ?? ?? 4? 8b 44 ?4 48 4? 89 04 ?4 e8 ?? ?? ?? ?? 4? 8b 44 ?4 48 4? 89 04 ?4 0f 1f 40 00 e8 ?? ?? ?? ?? 4? 8b 44 ?4
48 4? 89 04 ?4 e8 ?? ?? ?? ?? 4? 8b 44 ?4 48 4? 89 04 ?4 0f 1f 40 00 e8 ?? ?? ?? ?? 4? 8b 44 ?4 48 4? 89 04 ?4 e8 ?? ?? ?? ?? 4? 8b
44 ?4 48 4? 89 04 ?4 0f 1f 40 00 e8 ?? ?? ?? ?? 4? 8b 44 ?4 48 80 78 48 00 74 ?? 90 0f 57 c0 0f 11 44 ?4 28 4? 8d 05 ?? ?? ?? ??
4? 89 ?? ?4 28 4? 8d 05 ?? ?? ?? ?? 4? 89 ?? ?4 30 4? 8d 44 ?4 28 4? 89 04 ?4 4? c7 44 ?4 08 01 00 00 00 4? c7 44 ?4 10 01 00
00 00 e8 ?? ?? ?? ?? 4? 8b 44 ?4 20 4? 89 44 ?4 50 4? 8b 44 ?4 18 4? 89 44 ?4 58 4? 8b 6c ?4 38 4? 83 c4 40 c3 4? 89 04 ?4
e8 ?? ?? ?? ?? eb ?? 4? 89 44 ?4 50 4? 89 4c ?4 58 4? 8b 6c ?4 38 4? 83 c4 40 c3  }

  condition:
    $1 and uint16(0) == 0x5A4D
}
```

```
rule hive_v2_v3_32_win
{
  strings:
//prenotify routine
    $1 = { 64 8b 0d 14 00 00 00 8b 89 00 00 00 00 3b 61 08 0f 86 ?? ?? ?? ?? 83 ec ?? c7 44 ?4 04 ?? ?? ?? ?? c7 04 ?4 ?? ?? ?? ??
e8 ?? ?? ?? ?? e8 ?? ?? ?? ?? 8b 04 ?4 8b 4c ?4 04 c7 44 ?4 4c 00 00 00 00 c7 44 ?4 50 00 00 00 00 89 04 ?4 89 4c ?4 04
e8 ?? ?? ?? ?? 8b 44 ?4 08 8d 0d ?? ?? ?? ?? 89 4c ?4 4c 89 44 ?4 50 8d 44 ?4 4c 89 04 ?4 c7 44 ?4 04 01 00 00 00 c7 44 ?4 08
01 00 00 00 e8 ?? ?? ?? ?? 8b 44 ?4 68 89 04 ?4 e8 ?? ?? ?? ?? 8b 44 ?4 04 89 44 ?4 48 8b 4c ?4 08 89 4c ?4 38 31 d2 eb ??  }
  condition:
    $1 and uint16(0) == 0x5A4D
}
```

```
rule hive_v2_64_win
{
  strings:
      $1 =
{654?8b0c?5280000004?8b8900000000004?8d44????4?3b41100f86????????4?81ec????????4?89ac??????????4?8dac??????????4?
b8????????????????4?8904?4e8????????e8????????4?8b04?44?8b4c?4080f57c00f1184??????????4?8904?44?894c?408e8????????
4?8b44?4104?8d0d????????4?898c??????????4?8984??????????4?8d84??????????4?8904?44?c744?408010000004?c744?410010
00000e8????????4?8b84??????????4?8904?40f1f440000e8????????4?8b44?4084?8b4c?4104?85c97e??4?89?c}
  condition:
    $1 and uint16(0) == 0x5A4D
}
```

```
rule hive_v3_v4_64_win
{
  strings:
    $1 = {4? 3b 66 10 0f 86 ?? ?? ?? ?? 4? 83 ec 30 4? 89 6c ?4 28 4? 8d 6c ?4 28 4? 89 44 ?4 20 0f 1f 00 e8 ?? ?? ?? ?? 4? 85 c0 0f
85 ?? ?? ?? ?? 4? 8b 44 ?4 20 e8 ?? ?? ?? ?? 4? 85 c0 74 ?? 4? 8b 6c ?4 28 4? 83 c4 30 c3 4? 89 44 ?4 10 4? 89 5c ?4 18 4? 8b
44 ?4 20 e8 ?? ?? ?? ?? 4? 8b 44 ?4 20 e8 ?? ?? ?? ?? 4? 8b 44 ?4 20 e8 ?? ?? ?? ?? 4? 89 c3 4? 8b 44 ?4 20 e8 ?? ?? ?? ?? 4? 8b
44 ?4 20 e8 ?? ?? ?? ?? 4? 8b 44 ?4 20 e8 ?? ?? ?? ?? 4? 8b 44 ?4 20 e8 ?? ?? ?? ?? 4? 8b 44 ?4 20 90 e8 ?? ?? ?? ?? 4? 8b 44 ?4 10
4? 8b 5c ?4 18 4? 8b 6c ?4 28 4? 83 c4 30 c3 4? 8b 6c ?4 28 4? 83 c4 30 c3}
  condition:
    $1 and uint16(0) == 0x5A4D
}
```

```
rule hive_v5_32_win
{
  strings:
    $1 =
{5589e553575681ec440400008b75108b7d0c89d3894dc88d85b0fbffff68000400006a0050e8????????83c40c0fbec3b9abaa
aaaa8b0485c0b949008945e889f0f7e1d1ea8d045229c683f60389f0f7e131c9d1ea8d04528d570229c68b45148955cc8975e
48b00}
  condition:
    $1 and uint16(0) == 0x5A4D
}
```

```
rule hive_v5_64_win
{
  strings:
    $1 =
{415741564155415456575553 4881ec880400004c89cd448844243789d648894c2450488b9c24f0040000488bbc24f804000
0488d8c248800000041b80004000031d2e8??????00480fbec6488d0d??????004c8b24c148b9abaaaaaaaaaaaaaa4889d848f
7e148d1ea488d04524989de4929c64983f6034c89f048f7e148d1ea488d04524929c6488b07}
  condition:
    $1 and uint16(0) == 0x5A4D
}
```

```
rule hive_v3_esxi
{
  strings:
    $s1 = "+ prenotify %s"
    $s2 = "Stopping VMs"
    $s3 = "(.+)\\.(.+?)\\.%s$"
    $s4 = "\\.(vm|vs)\\w+$"

    $c = {f3 0f 1e fa 55 4? 89 e5 4? 83 ec 20 4? 89 7? ?? 4? 8b 4? ?? 4? 89 c7 e8 ?? ?? ?? ?? 89 4? ?? 83 7? ?? 00 74 ?? 8b 4? ??
eb ?? 4? 8b 4? ?? 4? 89 c7 e8 ?? ?? ?? ?? 89 4? ?? 83 7? ?? 00 74 ?? 4? 8d 3d ?? ?? ?? ?? e8 ?? ?? ?? ?? 8b 4? ?? eb ?? 4? 8b 4? ?? 4?
89 c7 e8 ?? ?? ?? ?? 4? 8b 4? ?? 4? 89 c7 e8 ?? ?? ?? ?? 4? 8b 4? ?? 4? 89 c7 e8 ?? ?? ?? ?? 4? 8b 4? ?? 4? 89 c7 e8 ?? ?? ?? ?? 4? 8b
4? ?? 4? 89 c7 e8 ?? ?? ?? ?? b8 00 00 00 00 c9 c3}

  condition:
    (all of ($s*) or $c) and uint32(0) == 0x464C457F
}
```

Yoroi S.r.l.
www.yoroi.company - info@yoroi.company

Piazza Sallustio, 9
00187 – Roma (RM)
+39 (051) 0301005