The ‘STOP’ Ransomware Variant

angle.ankura.com/post/102het9/the-stop-ransomware-variant

Vishal Thakur

In recent weeks, we have observed a spike in infections involving the STOP ransomware variant. STOP is also known as DJVU by other vendors in the industry. In this article, we’ve looked at the latest version circulating in the wild. We will look at some of the main characteristics of this malware variant, along with detections that can be used to prevent infection and IOCs that we were able to extract during analysis.

The STOP ransomware has been around for some time, dating back to 2019. The latest version has been found to be distributed broadly in the past few weeks. Like the ones in the past, this variant is a portable executable that uses a public key to encrypt data on the victim’s machine and drops a ransom note in folder directories as it goes through the entire file system encrypting files using the Salsa20 encryption algorithm. The threat actors behind STOP have gone for a flat rate of USD $980 to provide the decryption keys to victims and

have also offered a ‘discounted’ rate of USD $490 if the victims contact them within 72 hours of the attack occurring. This tactic is consistent with what has been observed in the past for this ransomware group.

Based on the tactics and techniques used by the malware, the threat actors behind the variant are likely from the Russian region as the malware avoids encryption explicitly on systems geo-located in or near Russia.

Quick Snapshot:
Class: DOS
Type: PE32
Machine: X86-64
OS: Windows
Entry Point: 0009D410
MD5: a2f33095ef25b4d5b061eb53a7fe6548

Figure 1: Quick Snapshot of STOP Ransomware

Mitigation

This section provides information that can be used to prevent infection by the STOP ransomware. We have included detections, IOC list, and YARA Rules that can be used to defend against this threat.

YARA Rule

This YARA Rule can be used to detect STOP Ransomware. Download the entire ruleset here.

```
/*
author = "Vishal Thakur - malienist.medium.com"
date = "2021-12-20"
version = "1"
description = "Detects STOP Windows Ransomware"
info = "Generated from information extracted from the malware sample by manual analysis."
*/
rule stopRansomwareStatic
{
  strings:
  $header = { 21 54 68 69 73 20 70 72 6f 67 72 61 6d 20 63 61 6e 6e 6f 74 20 62 65 20 72 75 6e 20 69 6e 20 44 4f 53 20 6d 6f
  $block1 = { 43 3a 5c 6d 6f 7a 5c 76 69 64 61 6a 2e 70 64 62 }
  $block2 = { 39 2d 39 35 39 45 39 56 39 69 39 34 3a 3c 3a 43 3a 4f 3a }
  $block3 = { 32 25 32 2f 32 39 32 4a 32 53 32 5f 32 67 32 75 32 }
  $block4 = { 32 2a 33 2f 33 34 33 57 33 78 33 7d 33 }
  $block5 = { 3e 20 3e 37 3e 40 3e 48 3e 4f 3e 6c 3e }
    $str1 = { 66 3a 5c 64 64 5c 76 63 74 6f 6f 6c 73 5c 63 72 74 5f 62 6c 64 5c 73 65 6c 66 5f 78 38 36 5c 63 72 74 5c 73 72
    $str2 = { 73 66 74 62 75 66 2e 63 }
    $str3 = { 69 6f 69 6e 69 74 2e 63 }
    $str4 = { 73 74 64 65 6e 76 70 2e 63 }
    $str5 = { 78 38 36 5c 63 72 74 5c 73 72 63 5c 73 74 64 61 72 67 76 2e 63 }
    $str6 = { 63 5c 77 5f 65 6e 76 2e 63 }
    $str7 = { 66 5f 78 38 36 5c 63 72 74 5c 73 72 63 5c 6d 62 63 74 79 70 65 2e 63 }
    $str8 = { 48 61 74 61 7a 75 79 69 20 6a 75 62 6f 6b 20 79 69 62 2e 20 54 75 6d 61 6a 75 73 6f 20 6e 69 6e 69 74 6f 66 75
    $str9 = "%s%s%s%s%s%s%s%s%s%s"
    $str10 = { 78 38 36 5c 63 72 74 5c 73 72 63 5c 6d 62 63 74 79 70 65 2e 63 }

  condition:
     filesize < 1500KB and all of them
}
```

Figure 2: YARA Ruleset for STOP Ransomware

Detections

The following figure has the information that can be used to create detections for this malware. Download the entire list here.

The following strings are from the unpacked malware, and these can be found in memory during and after the malware has been fully executed. This information can be used to create detections for EDR tools that can access and read memory and take actions based on detection rules applied.

```
C:\moz\vidaj.pdb
"--Admin"
" IsNotAutoStart"
" IsNotTask"
"e:\doc\my work (c++)\_git\
"input != nullptr && output != nullptr"
"C:\SystemID\PersonalID.txt"
http://tzgl.org/fhsgtsspen6/get.php
manager@mailtemp.ch
helprestoremanager@airmail.cc
delself.bat
E:\Doc\My work (C++)\_Git\Encryption\Release\encrypt_win_api.pdb
e:\doc\my work (c++)\_git\encryption\encryptionwinapi\Salsa20.inl
C:\Build-OpenSSL-VC-32/ssl/private
https://api.2ip.ua/geo.json
```

Figure 3: Detections

IOC List

Download the entire list here.

```
02e36a484cb87c6c55122369fd726a44be6cbced7ca3b83a868d005852b52130
1562ac8d688d9bfbe272835e83bb8d772fa65fc41e55bf449fa7f5e0d4e1df96
a8ba55c38281587234f510217a07325490d4a25878271273b9592a8d59d9b543
b0d41e9b8c941d207a0958b92f57083dd9b9246958bd32e2e6e90c4ee0e12419
c22fbc68473199e473afd0468542434854bf5ab8f1fbd2932c044e0ce226b307
http://api.2ip.ua:443/
http://kotob.top/dl/build2.exe
http://tzgl.org/fhsgtsspen6/get.php
http://tzgl.org/files/1/build3.exe
https://api.2ip.ua/geo.json
api.2ip.ua
kotob.top
tzgl.org
1.248.122.240
104.18.30.182
104.18.31.182
110.14.121.125
116.121.62.237
14.51.96.70
175.126.109.15
180.69.193.102
183.100.39.157
187.156.124.76
```

Figure 4: IOC list

Execution

Once the STOP ransomware executes, it attempts to make a few network connections over the Internet for various purposes, such as; geo-checking, key retrieval, and further infection by downloading different malware. First, let's look at the start of the execution of this malware.

```
0049D410    51                  push ecx
0049D411    50                  push eax
0049D412    52                  push edx
0049D413    8D 0D 18 00 00 00   lea ecx,dword ptr ds:[18]
0049D419    64 8B 01            mov eax,dword ptr fs:[ecx]
0049D41C    01 C8               add eax,ecx
0049D41E    01 C8               add eax,ecx
0049D420    8B 00               mov eax,dword ptr ds:[eax]
0049D422    53                  push ebx
0049D423    8B 58 08            mov ebx,dword ptr ds:[eax+8]
0049D426    83 C0 0C            add eax,C
0049D429    8B 10               mov edx,dword ptr ds:[eax]
0049D42B    8D 0A               lea ecx,dword ptr ds:[edx]
0049D42D    83 C1 0C            add ecx,C
0049D430    8B 01               mov eax,dword ptr ds:[ecx]
0049D432    56                  push esi
0049D433    8B 48 18            mov ecx,dword ptr ds:[eax+18]
0049D436    83 F9 00            cmp ecx,0
0049D439    74 25               je stop.49D460
0049D43B    8B D0               mov edx,eax
0049D43D    83 C2 30            add edx,30
0049D440    8B 12               mov edx,dword ptr ds:[edx]
0049D442    8B 32               mov esi,dword ptr ds:[edx]
0049D444    81 E6 DF 00 DF 00   and esi,DF00DF
0049D44A    8B 52 0C            mov edx,dword ptr ds:[edx+C]
0049D44D    C1 E2 08            shl edx,8
0049D450    03 D6               add edx,esi
0049D452    81 C2 B5 CC BA CD   add edx,CDBACCB5
0049D458    85 D2               test edx,edx
0049D45A    0F 84 07 00 00 00   je stop.49D467
0049D460    8B 00               mov eax,dword ptr ds:[eax]
0049D462    E9 CC FF FF FF      jmp stop.49D433
0049D467    E8 FD 00 00 00      call stop.49D569
```

Figure 5: Malware Entry-point

Upon execution, the malware copies itself to the 'C:\Users\[username]\AppData\Local\[GUID]' directory on disk and tries to execute with escalated privileges, as shown in the figures below.

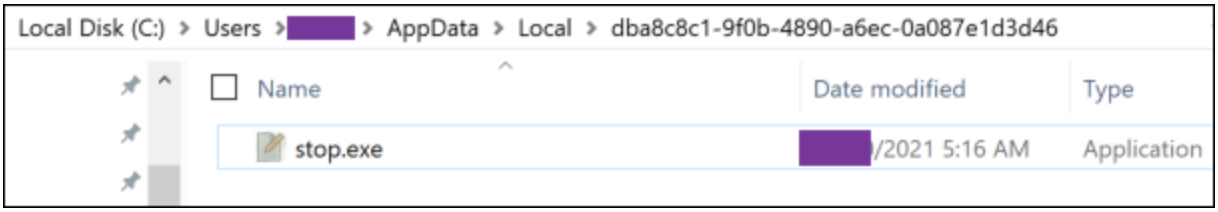

Figure 6: Malware copies itself to a different location

```
007EB788  028226F8  "\"C:[redacted]\stop.exe\" --Admin IsNotAutoStart IsNotTask"
```

Figure 7: Spawning new process with elevated privileges

The malware then attempts to connect over the Internet to "https://api.2ip.ua/geo.json" to verify the victim's geolocation. This link leads to a Russian site (screenshot below) that provides geolocation services based on public Internet IP addresses which the malware uses to ascertain the location of its victims. The malware has a hard-coded country codes list that is checked before it continues executing on the victim's system and will avoid encrypting victims within these countries.

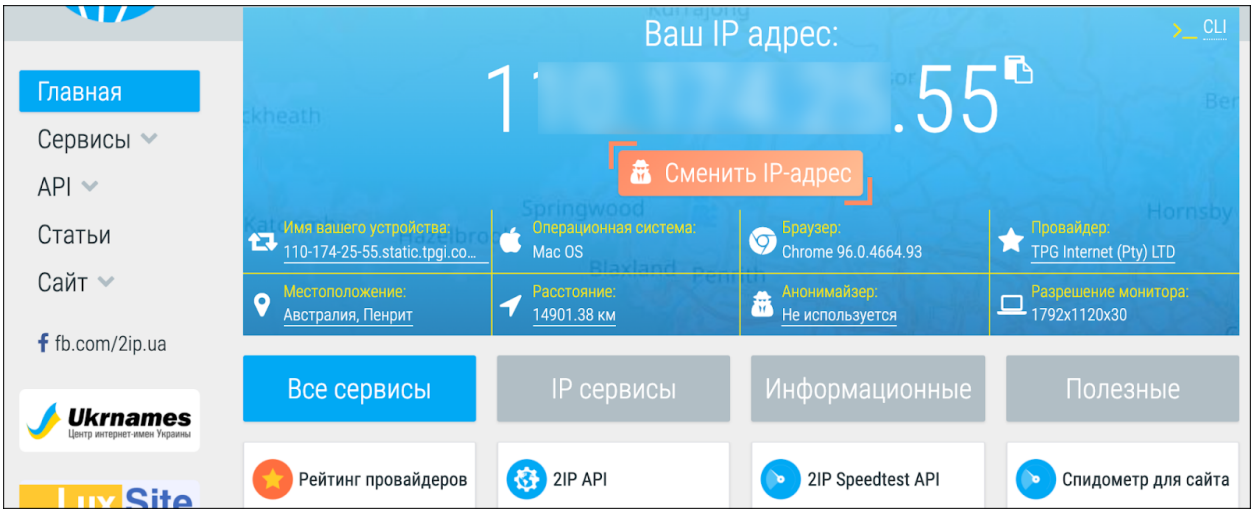

Figure 8: Geo-location service used by the malware

The site also offers an API-based service that the malware uses to determine the geolocation of the victim machines.

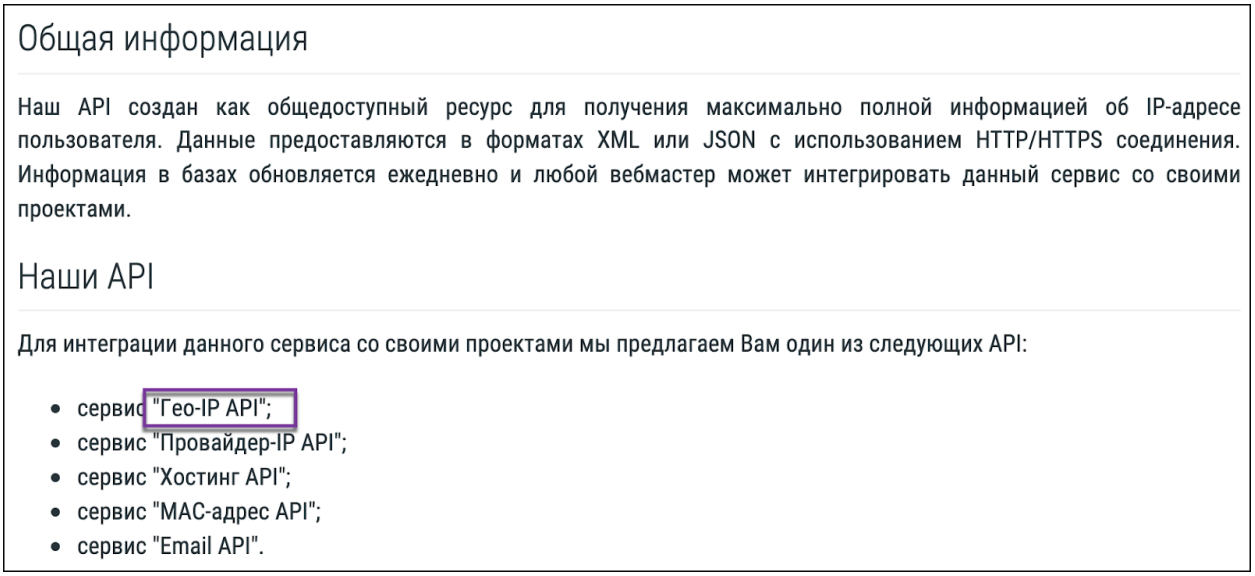

Figure 9: The specific API-based service the malware uses

The country code list can be seen in the figure below, showing the codes in memory during execution.

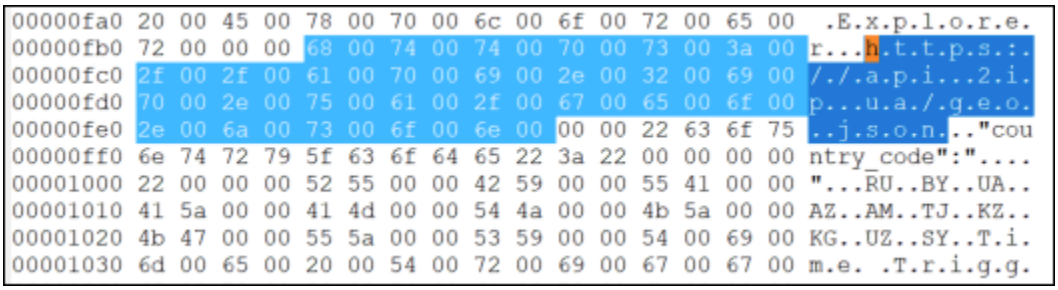

Figure 10: Country codes of locations this malware avoids

Next, the malware tries to connect to a command and control URI to get the public key for encryption. As we can see in the figure below, it sends a request to this URI with a PID created for the victim.

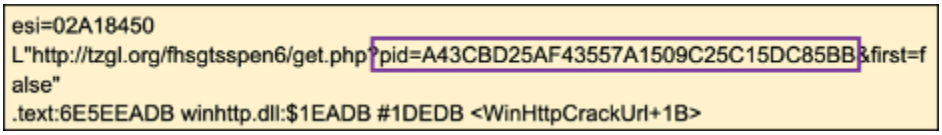

Figure 11: URI loaded into the Stack for processing

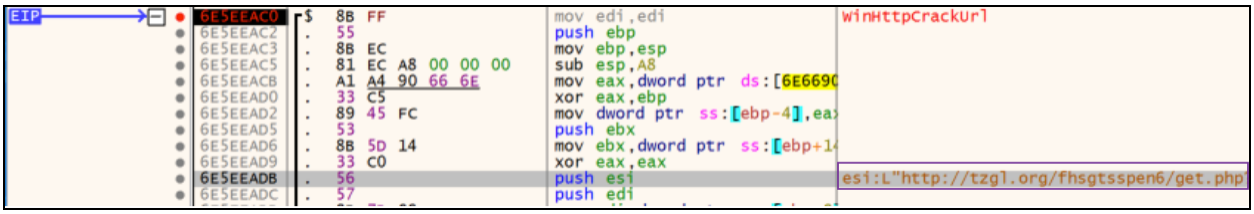

Figure 12: Connection to the C2 for public key

Once the request is successful, the malware uses the public key with the ID to encrypt the victim's data.

{"public_key":"-----BEGIN PUBLIC KEY-----\\nMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA9yBx9akK4qpsI+xBRN4i\\nC9qqbIJB5lyxPffC3XdKt8vcRFOfzJNYF7oyx6pwgJmJ79XDgLnmesbcz9mNL8+l\\nJY9ViOgd\/WAjC7gp\/rYTA4Wp9v5s6eGecRCcwSgr8ewPdjsbTyNXK3VzITC16LiB\\nFc0++0QREZOlbQsek7iq7B9TfnNicMLXklTBCk2\/DXqeqPANao6qouhRGntavGjx\\n9yZ\/41GwzBbS2MY9QwT2p5NZG1EppKc9YDh+KzVZnoLgO5JBYxDSiIQR9CktE78W\\nvxGneXMCSf0hMITKtxcZeafHoiLjvAefXFnmI3+EIXiJTEenkw+izXlRFge1C2MK\\niwIDAQAB\\n-----END PUBLIC KEY-----\\n","id":"Hsd92XfmqYxjBH2e4HbX2BE4AbcBjVw5Fu1wDp3t"}

Figure 13: Public Key for encryption served by the C2

The malware uses a standard encryption sequence, calling in the functions required to encrypt data from start to finish. The complete sequence can be seen in the figure below, in the order of called functions.

Encryption sequence:
.text:742A03B0 advapi32.dll:$203B0 #1F7B0 <CryptAcquireContextW>
.text:7429FB50 advapi32.dll:$1FB50 #1EF50 <CryptCreateHash>
.text:7429FC90 advapi32.dll:$1FC90 #1F090 <CryptHashData>
.text:7429FAB0 advapi32.dll:$1FAB0 #1EEB0 <CryptGetHashParam>
.text:742A0000 advapi32.dll:$20000 #1F400 <CryptDestroyHash>
.text:742A0740 advapi32.dll:$20740 #1FB40 <CryptReleaseContext>
.text:753DE250 kernel32.dll:$6E250 #5F250 <WriteFile>

Figure 14: Encryption Sequence of function calls

CSP – Cryptography Service Provider

The malware queries the Registry on the victim machine to set the CSP and CSP type. Note that type shown in the figure below is 'Type 001' which is the 'RSA Full' provider.

```
007EE8D8  029EBD90  "SOFTWARE\\Microsoft\\Cryptography\\Defaults\\Provider Types\\Type 001"
```

Figure 15: Malware query to Registry for the Type of CSP

The malware uses the Registry to set the provider type and subsequently the actual provider, which in this case happens to be RSA Full.

RegOpenKey

```
74BC9490    8B FF           mov edi,edi
74BC9492    55              push ebp
74BC9493    8B EC           mov ebp,esp
74BC9495    51              push ecx
74BC9496    6A 00           push 0
74BC9498    FF 75 18        push dword ptr ss:[ebp+18]
74BC949B    FF 75 14        push dword ptr ss:[ebp+14]
74BC949E    FF 75 10        push dword ptr ss:[ebp+10]
74BC94A1    FF 75 0C        push dword ptr ss:[ebp+C]
74BC94A4    FF 75 08        push dword ptr ss:[ebp+8]
74BC94A7    E8 14 00 00 00  call <kernelbase.RegOpenKeyExInternalA>
74BC94AC    59              pop ecx
74BC94AD    5D              pop ebp
74BC94AE    C2 14 00        ret 14
```

Figure 16: Registry functions used to determine the CSP

RegOpenKeyExA

Next, the malware queries the Registry to determine the actual CSP as can be seen in the figure below.

```
EBX   029D0B98   "SOFTWARE\\Microsoft\\Cryptography\\Defaults\\Provider\\Microsoft Strong Cryptographic Provider"
```

Figure 17: The absolute Registry path passing through the Registers

Name	Type	Data
(Default)	REG_SZ	(value not set)
Name	REG_SZ	Microsoft Strong Cryptographic Provider
TypeName	REG_SZ	RSA Full (Signature and Key Exchange)

Figure 18: The CSP highlighted in the Registry

Name	Type	Data
(Default)	REG_SZ	(value not set)
Image Path	REG_SZ	%SystemRoot%\system32\rsaenh.dll
SigInFile	REG_DWORD	0x00000000 (0)
Type	REG_DWORD	0x00000001 (1)

Figure 19: DLL image path to be called for the CSP

The malware uses the public key obtained from the command and control server to start the process of encryption on the victim's system.

```
7429FB50    8B FF           mov edi,edi
7429FB52    55              push ebp
7429FB53    8B EC           mov ebp,esp
7429FB55    5D              pop ebp
7429FB56    FF 25 44 10 2F 74   jmp dword ptr ds:[<&CryptCreateHash>]
7429FB5C    CC              int3
7429FB5D    CC              int3
7429FB5E    CC              int3
7429FB5F    CC              int3
7429FB60    8B FF           mov edi,edi
7429FB62    55              push ebp
7429FB63    8B EC           mov ebp,esp
7429FB65    83 EC 34        sub esp,34
7429FB68    A1 30 74 2E 74  mov eax,dword ptr ds:[742E7430]
7429FB6D    33 C5           xor eax,ebp
7429FB6F    89 45 FC        mov dword ptr ss:[ebp-4],eax
7429FB72    8B 4D 08        mov ecx,dword ptr ss:[ebp+8]
7429FB75    56              push esi
7429FB76    33 F6           xor esi,esi
7429FB78    89 75 CC        mov dword ptr ss:[ebp-34],esi
7429FB7B    85 C9           test ecx,ecx
7429FB7D    0F 85 6D 05 01 00   jne advapi32.742B00F0
```

Figure 20: Second function to be called in the Encryption Sequence

```
04FDF9FC  0040EB07  return to stop.0040EB07 from ???
04FDFA00  04FDFA48
04FDFA04  00000000
04FDFA08  00000000
04FDFA0C  00000001
04FDFA10  F0000000
04FDFA14  00540000  stop.00540000
04FDFA18  000005E4
04FDFA1C  00162588  "-----BEGIN PUBLIC KEY-----\\\\nMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA8
04FDFA20  000001D9
04FDFA24  000001D9
04FDFA28  04FDFA48
04FDFA2C  00420CAB  return to stop.00420CAB from ???
```

Figure 21: Public key loaded

Once the entire encryption sequence is completed for a directory, the final step is to write a ransom note to the directory with instructions on how to pay the ransom.

```
753DE250  ^ FF 25 C0 0E 3E 75    jmp dword ptr ds:[<&WriteFile>]
753DE256    CC                   int3
753DE257    CC                   int3
753DE258    CC                   int3
753DE259    CC                   int3
753DE25A    CC                   int3
753DE25B    CC                   int3
753DE25C    CC                   int3
753DE25D    CC                   int3
753DE25E    CC                   int3
753DE25F    CC                   int3
753DE260  ^ FF 25 C4 0E 3E 75    jmp dword ptr ds:[<&WriteFileEx>]
753DE266    CC                   int3
753DE267    CC                   int3
```

Figure 22: Ransom note 'write' initiated

The figure below shows the ransom note as strings being passed onto the Stack before it is written to the disk.

```
02882CF0  UNICODE "C:\_readme.txt"
028B32D0
02890DA8  ASCII "ATTENTION!♪◘♪◘Don't worry, you can return all your files!
0041475E  RETURN from stop.0042D8D0 to stop.0041475E
02994D38  UNICODE "C:\_readme.txt"
0291D338  UNICODE "C:\_readme.txt"
0000085D
```

Figure 23: Ransom note loaded into the Stack

Finally, the ransom note is written as a ‘txt’ file to the disk. This process is repeated for all directories in which the malware encrypts data. The figure below shows the newly created ransom note “_readme.txt”.

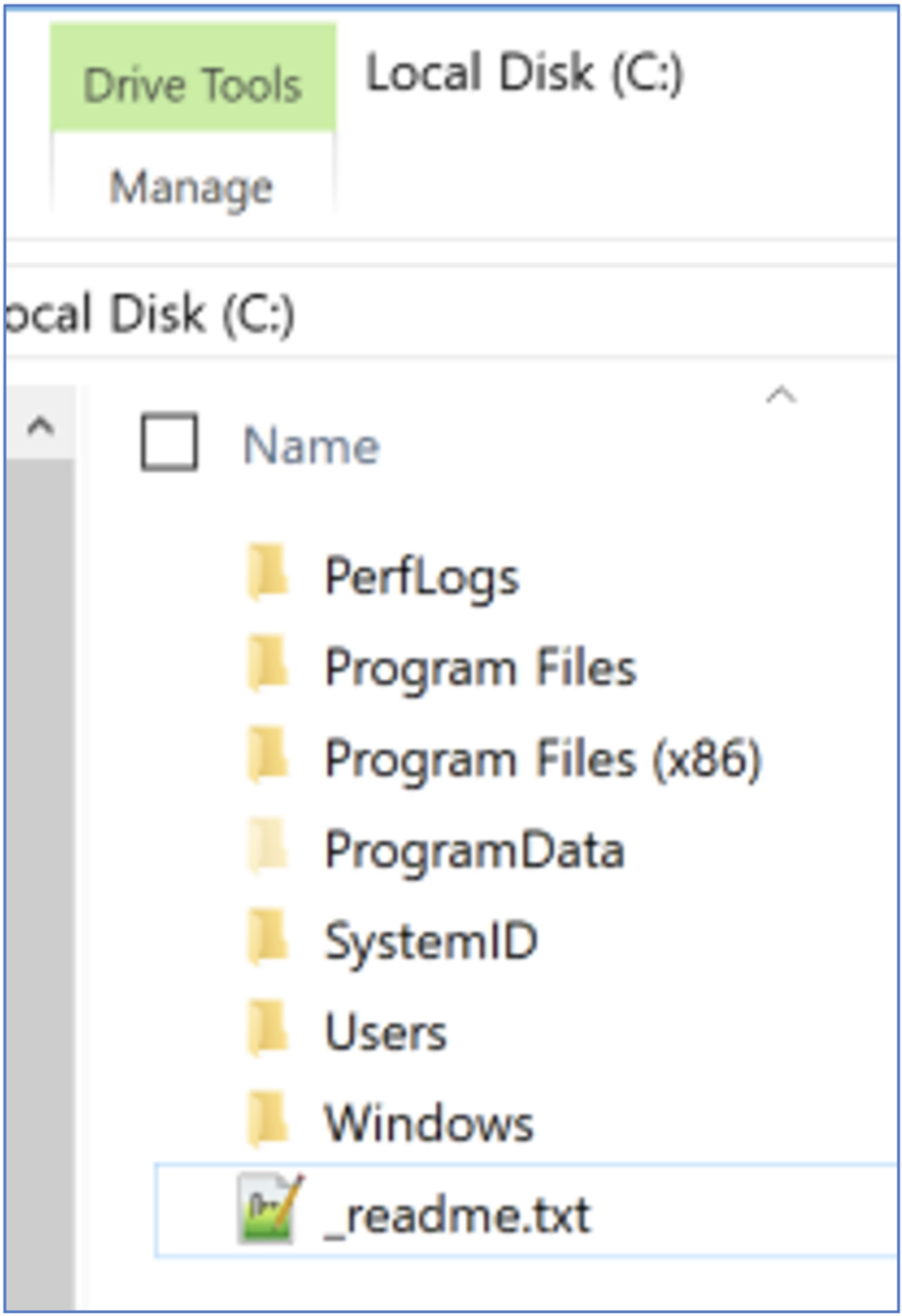
Drive Tools
Local Disk (C:)
Manage
ocal Disk (C:)
Name
PerfLogs
Program Files
Program Files (x86)
ProgramData
SystemID
Users
Windows
_readme.txt

Figure 24: Ransom note file written to the current directory

The ransom note has the instructions on how the victims can pay to get the decryption key and provides a unique ID that the victim needs to use to get the decryption key for their machine. There is also a link to a demo video showing how the decryption tool works. The note also provides a couple of email addresses for the victims to contact the ransomware group if needed.

```
ATTENTION!

Don't worry, you can return all your files!
All your files like pictures, databases, documents and other important are encrypted with strongest
encryption and unique key.
The only method of recovering files is to purchase decrypt tool and unique key for you.
This software will decrypt all your encrypted files.
What guarantees you have?
You can send one of your encrypted file from your PC and we decrypt it for free.
But we can decrypt only 1 file for free. File must not contain valuable information.
You can get and look video overview decrypt tool:
https://we.tl/t-NONb1QT9nD
Price of private key and decrypt software is $980.
Discount 50% available if you contact us first 72 hours, that's price for you is $490.
Please note that you'll never restore your data without payment.
Check your e-mail "Spam" or "Junk" folder if you don't get answer more than 6 hours.


To get this software you need write on our e-mail:
manager@mailtemp.ch

Reserve e-mail address to contact us:
helprestoremanager@airmail.cc

Your personal ID:
0361[redacted]wDp3t
```

Figure 25: Ransom note with instructions on next steps

This version of the STOP ransomware variant encrypts the file and replaces the file-extensions to ".shgv", as seen in the figure below.

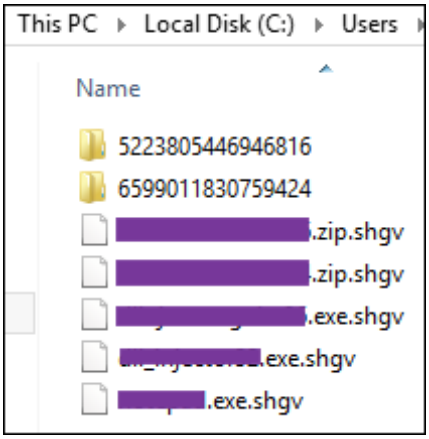

Figure 26: Files successfully encrypted

Downloader Module

Aside from performing common ransomware activities, this malware also tries to download and execute other malware:

```
http://tzgl.org/files/1/build3.exe$run
229b06ba702bdde53a3f4a89d9da20d47b972ddaf45b00997fa517014e4d5bec
```

Figure 27: Downloaded malware - Vidar Stealer

This downloaded PE is a variant of the Vidar malware family.

Vidar Stealer is malware designed to steal information, mainly distributed as spam mail or cracked versions of commercial software and keygen programs. When installed, data such as infected device information, account, and history recorded in the browser is collected and sent to a command and control server.

The group behind the development or distribution (or both) of STOP ransomware may be working with the group responsible for developing the Vidar malware.

Conclusion

STOP ransomware has been around for quite some time now. Early occurrences of infections by this ransomware can be traced back to 2019.

Compared to some other ransomware families, the execution standard is low and it's clear that this ransomware model is affiliation-leaning (working with other malware groups). We were able to link this malware to a different malware, the Vidar Stealer, which has been the case for quite some time.

The encryption is straightforward, with the threat actors not bothering to create their encryption algorithm or deploying any additional modules other than a downloader for a separate malware. The malware uses the Salsa20 algorithm for encryption. It is capable of both online and offline encryption.

This ransomware avoids infecting victims in and near Russia.

The ransomware seems to be targeted towards individuals or small businesses at best, as the asking price for the decryption key is not that high. They even offer an 'early bird' discount to top it all off.

Deep Analysis of Vidar Stealer - Sojun Ryu
YAYA ruleset for STOP Ransomware - Vishal Thakur
Detections list for STOP Ransomware - Vishal Thakur

IOC list of STOP Ransomware - Vishal Thakur