Letter_Y50.js Analysis

Sample obtained from app.any.run

SHA1: 514e6350dde291e84912754efd3e5da8ed0ae213

Analysis

After getting the letter_Y50.js file I began by bringing the file into Sublime-Text. The code was hidden within comments. The file was full of long text of nonsense sentences. Each line had a double back slash. The real code had quadruple back slashes. I began by using regular expressions and looking for all lines that begin with quad back slashes and removed them. Next I ran a regular expressions search and selected \\.* to select the entire line of nonsense sentences. Removed them all and lastly removed all the empty lines ^\n. This brought all the code up together nicely. The code itself was not obfuscated.

I began looking at what the script was doing. First thing is creates and activeXObject, winmgmts:\\\\.\\root\\cimv2 variable. It then begins to go through an enumerate what drives are present on the system. It was trying to find a valid drive letter that is not currently being used. Once it does this it maps an external URL and specifies port 80 and a \share location. It waits 30 seconds before checking the connection. If it’s successful, it runs an install command pointing to an msi file stored in the newly created mapped drive.

The last section of the script is calling a new activeXObject, Scripting.FileSystemObject. There it opens a script file and reads each line. Storing it into the variable ‘a’. I could not see where this scripting file is defined or what it’s referring to. I connected to the Tor network and browsed to the share location. I wanted to see if there was another file stored in the share directory. There was not.

The next step was to look at the MSI file. I ran strings and reviewed what was in it. I could see image files and two Windows PE files. Ran strings to see what was visible. Hopefully it had pdbs embedded with it. It did. When I exported the strings the output was all bunched up. Spent some time moving new strings to new lines to get a better understanding of what this was. As far as I could tell this was just an installer. Within it I could several places where svg files were in. I could see references to other image files made in Photoshop. Two Windows PE files could be seen, based on the “This program cannot run in DOS mode”. Lastly there was a cab file within.

I needed an easy way to get these files within the msi file. I grabbed msitools. First I ran msiinfo suminfo on the msi file. I got the following:

```
Subject: Impv studio
Author: Beatss Inc
Keywords: Installer, MSI, Database
Comments: This installer database contains the logic and data required to install Impv studio.
Template: ;1033
Last author:
Revision number (UUID): {32CC3CB8-8734-4CAA-8F29-C1CE70440B8D}
Last printed: Fri Dec 11 04:47:44 2009
Created: Fri Dec 11 04:47:44 2009
Last saved: Fri Sep 18 08:06:51 2020
Version: 200 (c8)
Source: 10 (a)
```

Application: Impv studio
Security: 0 (0)

Next I ran msidump and got all the files from within. Knowing this had to be malicious I decided to have VirusTotal look at each file. All were cleared except for the cab file. This was flagged as IceID. Ran 7zip on the cab file to pull out what was in it. Inside was a file called map.dll. Scanned the dll file and it was flagged as IceID.

The malware hid itself within a real installer, though it was old software from 2009.

Indicators

The biggest indicator this file is malicious is the fact it hides its code within a sea of commented lines. The lines contained gibberish text.

The script looks over used drive letters and makes a new share folder pointing to its share directory on its sketchy website. Then automatically grabs and installs an msi containing IceID.

Interesting Strings

- !This program cannot be run in DOS mode. >> Found this part way down
    - .text
    - .rdata
    - @.data
    - .rsrc
    - @.reloc
    - GetProcessId
    - C:\JobRelease\win\Release\custact\x86\viewer.pdb >> PDB file for this malware?
    - .rdata
    - .rdata$T
    - .rdata$r
    - .rdata$sxdata
    - .rdata$voltmd
    - .rdata$zzzdbg
    - Sleep
    - ShellExecuteExW
    - ShellExecuteW
    - SHELL32.dll
    - /(c) 2006 thawte, Inc. - For authorized use only1
    - http://t2.symcb.com
    - !http://t1.symcb.com/ThawtePCA.crl0
    - https://www.thawte.com/cps0/
    - !https://www.thawte.com/repository0W
    - msi.dll
    - NETAPI32.dll

- SHELL32.dll
- WS2_32.dll
- SHLWAPI.dll
- IPHLPAPI.DLL
- KERNEL32.dll
- USER32.dll
- GDI32.dll
- COMDLG32.dll
- LogonUserW
- ole32.dll
- OLEAUT32.dll
- This installer database contains the logic and data required to install Impv studio.

URLs Found

178.23.190.199@80\\share

Hashes

Letter_y50.js - 514e6350dde291e84912754efd3e5da8ed0ae213 1/60

gsm.msi - c437c27f55f7b34dd7bd964ec48b0df6ffee985f 4/60 (identified as IceID)

extracted exe - f6c31c9cd832ae2aebcd88e7b2fa6803ae93fc83 0/71

extracted dll - 67340739f51e1134ae8f0ffc5ae9dd710e8e3a08 0/69

disk1.cab 7fdb3a6c776f0e775b41be60d457fdde9f102c19 - 3/56 (IceID)

map.dll 80803c4fcb49bcc6bcd4035d455f8c095d607424 7/71 (IceID)