Agent Tesla VBS analysis – Feb 10, 2024

Sha1-2cff9666dad3cf3afbfa379718f31081fb1ed57a

Downloaded from Abuse.ch

https://bazaar<dot>abuse<dot>ch/sample/f4691cad5e54ee1d239725e5d2ae270a3ab1e6ffec22bc5fe93d42756d78861e/

General Analysis

I began my analysis in Remnux.

Based on the report from Abuse.ch, the vbs script was delivered via malspam. Once run the script creates a winhttp request to paste.ee. I tried to curl the data from the specified location but could not get anything. I went directly to the specified URL in my Remnux VM. Once there I was shown a blob of text. I copied it into a new vbs script document in Sublime-text.

I could see lots of obfuscation. Gibberish ASCII characters, variable re-declared many times, adding more gibberish characters to the variable. Below this was a section where it called Replace and specified the variable previously mentioned. I saw it would take a few of the random characters and replace them by a single character. I replaced these in Sublime-text and got a the full line:

Figure 1: Part of string before replacement

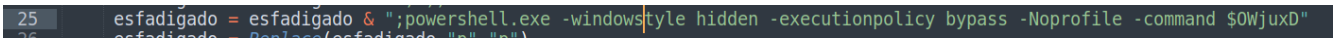

Figure 2: String after replacement

Continuing analysis on the second vbs file the variable defined at line 4 showed multiple segments of random characters and then the same set of variables being joined in random locations.

There is another replace where three variables are combined, they replace and become Z. I found all 64 instances of this and replaced it as Z. Removed all of the un-needed & and “ to combine the variable into one single string.

Figure 3: Obfuscated base64 blob

I took this base64 string and decoded it via the built in *base64 -d* command.

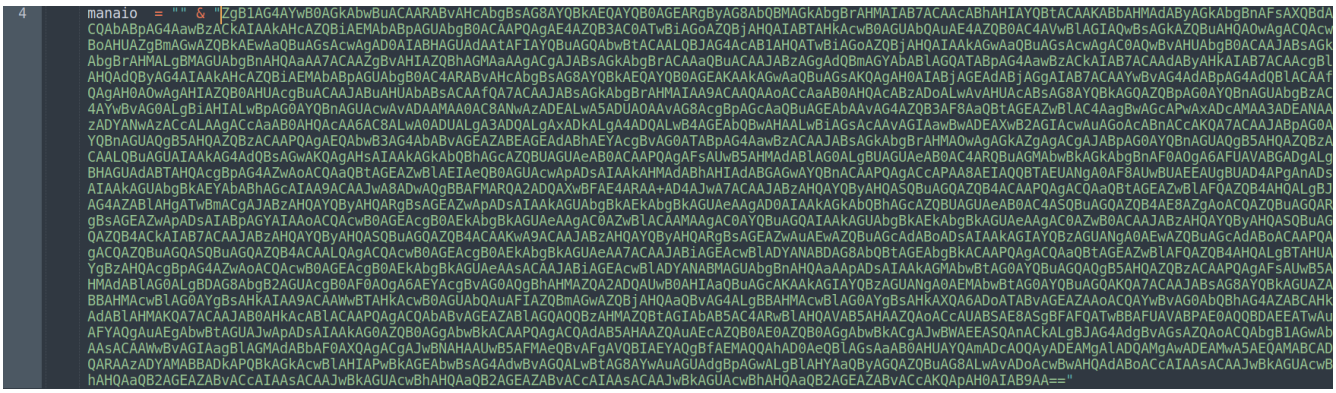

Figure 4: Base64 after deobfuscation

The second vbs script creates a wscript call with the first variable I analyzed. The wscript file runs PowerShell with execution policy set to bypass and no windows. The PowerShell command was hidden in the variable from line 4.

After reviewing both lines I see several urls. One is an image download site with a .br ending (Brazil). The second is an ip address, just http. The third is to a OneDrive location. I was able to download image files from he .br site and the IP address site. Compared the two files. They are identical. Based on the script it will try one site, if that fails it will try another.

Examining the jpg file you can see there is embedded base64 code within. I exported the strings from the file and removed all but the base64 data to decode. The character size of the base64 blob was 7,369,389. Next step is to decode this. I did this in terminal and redirected the output to a stage4 file.

Ran file on stage4 and saw it was a Windows PE32 DLL. Ran sha1sum and got:
7b0f1b9d14df489e4242fb8432337d31757eb6fb

Search within VirusTotal shows 38/70 hits and identified as likely a trojan injector.

I reviewed the dll in Detect It Easy. There were 4 sections within it. This showed the system was packed.

Indicators

Indicators program is malicious:

- File delivered via malspam
- Infected Office document
- Obfuscated code
  - Random characters
  - Character replacement
  - String broken up into part to make analysis harder
  - Some strings reversed, such as URLs. These are resolved right when needed in code
- Base64 encoded strings
- URL containing another script that will be autorun
  - Some URLs are also http and not https
  - One URL was just the IP address
- wscript calls to run PowerShell with calls for no window, and bypassing local system execution policy
- PowerShell command makes calls to the internet to gather new files
- Downloading image files with embedded dll file
- dll file is packed

URLs

List of URLs this malware attempts to contact from all stages analyzed:

- paste.ee/d/QkK2f (HTTP)
- 45.74.19.84/xampp/bkp/bkp1_vbs.jpg (HTTP)
- uploaddeimagens.com<.>br/images/004/731/958/original/new_image.jpg?1707143673 (HTTPS)
- onedrive.live.com/download?resid=9A063D4B0D931024%21297&authkey=!AC_BFHUXoySySpM (HTTPS)

SHA1 Hashes

- Initial VBS script – 2cff9666dad3cf3afbfa379718f31081fb1ed57a
  - VirusTotal score = 16/52 – Identified as Agent Tesla
- Second stage VBS script – f99cf72f174834efd7305ba48d5792b8486c18c6
  - VirusTotal score = 20/60
- JPG file with embedded DLL – 19351a79881daf08b3d28e7e895c6b8e3bbf20fe
  - VirusTotal score = 0/57-- Did detect the base64 encoded data
- DLL decoded from JPG – 7b0f1b9d14df489e4242fb8432337d31757eb6fb
  - VirusTotal score = 38/70

Summary

I was unable to identify what might be at the OneDrive location provided in the stage2 vbs file. It would not return anything when using *curl*, *wget* or via directly going to it in my Remnux VM. The other two URLs downloaded the same image file, an interesting picture of a Jupiter looking planet with stars.

Figure 5: Image with embedded dll

I did a basic Google Image Search and found the image on other sites:

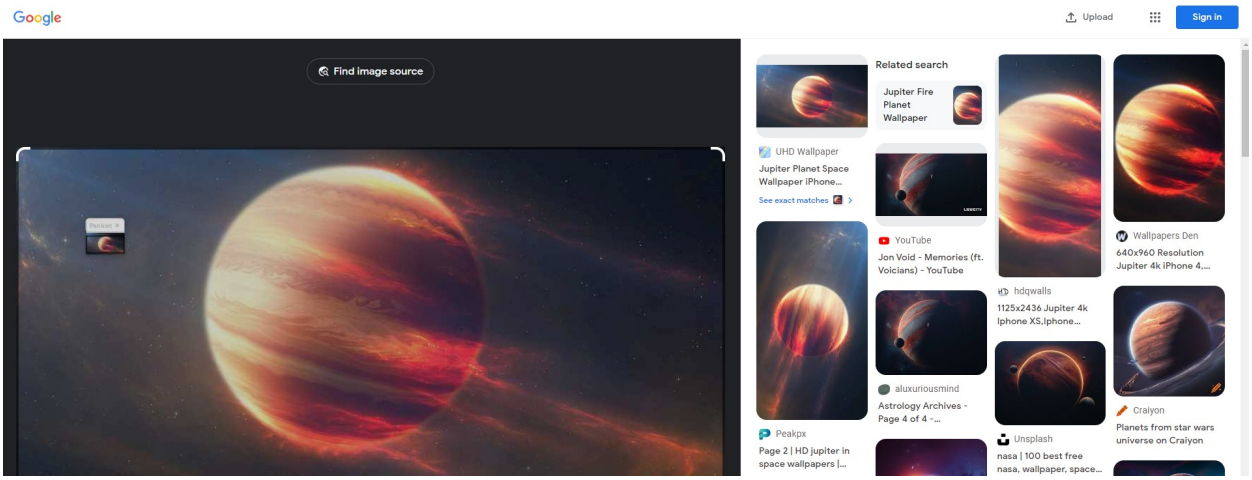

Figure 6: Google Image Search of Offending Image

Reviewed other analysis of Agent Tesla. This has been around since 2014. Exploits the equation editor in MS Office applications. Based on this, I have to assume the vba script I started analysis with was contained within a Word or Excel file. Other analysis show it does use stenography to deliver an exe or dll. In the case of my analysis the image file contained a dll.

To confirm this was Agent Tesla I reviewed some of the script code that was shown in Any.run. This showed identical code seen in the vbs files as well as identical PowerShell commands.

Based on what I've learned I have not yet obtained the final payload used in Agent Tesla. There appears to be one more stage where the payload is downloaded from an exe and two dlls, one which I was able to download.