Lab 1

Kevin Turkington

I. Binary Analysis

A. *evil.exe*

During my analysis with McAfee's FileInsight program I found the Malware delete or create a collection of scheduled tasks to run every 30 minutes through out the day that run the file: *c:/ntldrs/svchest.exe*. upon further research I found the *svchest.exe* is a windows process that hosts multiple windows processes, which could mean the malware is attempting to stop firewalls or anti-virus scanners from running regularly. Additionally a couple different files are being stored within the ntldrs directory which may be dependencies for the malware to function or a seperate piece of malware altogether. Indicating the sample provided *evil.exe* is probably a downloader, intended for the delivery and installation of the actual payload. upon further inspection of the malicious software I found a lot of vba specfic commands indicating the malware is written in visual basic.

Files that are being stored in the boot directory:

- ntldrs/Isinter.gif
- ntldrs/system.yf
- ntldrs/svchest.exe
- ntldrs/funbots.bat

B. *tongji2.exe*

During FileInsight of this file, I found a couple Delphi specfic commands and keywords in a string search. Which could mean the *tongji2.exe* is the main payload of the malware. However the only thing that I could descipher from the keywords was that the program was attempting to spawn seperate threads. Which could mean machines are being for computational capabilitys as a whole.

II. Run time Analysis

After setting up the recommnded tools from the lab instructions I found during run time the malware immediately created a couple different GET/POST requests to an outside server specfically to the url *timeless888.com*. Then *evil.exe* proceeded to create a collection of files. while inspecting the windows task scheduler, it seemed to somewhat reflect what was examined in the binary file from *evil.exe*, creating a 30 minute task running *svchest.exe*. Once the exact hour came and the tasks were run the malware created a GET request to timeless888.com for a page called *tong.htm*, however while conducting a system wide search for *tong.htm* I could not find it. Leading to the conclusion that the website is stored in memory and will be served back to the victum in one way or another.

III. RESEARCH

After examining article from piazaa posted cas donoghue[1], I found out the malware was a backdoor Trojan with backdoor capabilities with the addition of serving up a fake webpage (*tong.htm*) used for stealing social security numbers from Korean citizens. The payload of the virus was deployed in several stages starting from a simple visual basic script (*evil.exe*), then downloading the dummy website, and finally installing a backdoor (*tongji2.exe*).

REFERENCES

[1] Analysis of chinese attack against korean banks. https://blog.avast.com/2013/03/19/analysis-of-chinese-attack-against-korean-banks/. (Accessed on 01/16/2018).