International Journal of Research in Advent Technology, Vol.3, No.9, September 2015
E-ISSN: 2321-9637
Available online at www.ijrat.org

CPU Mining Malware: Complete Analysis of Real World Virus

Rohitkumar Gautam[1,3], Sanjeev Kumar[2,4], Jhilik Bhattacharya[15]
[1] *Computer Science and Engineering Department, Thapar University, Patiala,Punjab*
[2]*Cyber Security Technology Division, CDAC, Mohali, Punjab*
{[3] rohitkumargautam@live.com, [4]sanjeev@cdac.in [5] jhilik@thapar.edu }

Abstract- Malwares are one of the crucial security threats. Malwares are needed to be analyzed in order to fix them. Malware analysis provides the information required to respond to an intrusion. Malware analysis not only helps to understand the working of malware but also helps in devising solution (signature) for a virus or malware. Aim of malware analysis is to exactly determine what suspected executable can do, how to detect it in first place, and how to measure and contain damage. This research paper presents a complete malware analysis of a very recent virus which has hit Facebook in year 2014. This virus replicates by sending a message to every victim's Facebook friend which contain a zip file with a text message "lol". Once virus get download and execute, it earns bitcoins by mining victim machine's CPU.

Index Terms- Malware, Static Analysis, Dynamic Analysis, Virus, Bitcoins, Analysis Environment.

1. INTRODUCTION

Software that causes damage to users, computer, or network can be termed as malware. Virus, Trojan horse, worm, rootkit, scareware, spyware all are example of different types of malware [1]. Malware do come in variety but the technique used to analyze them is same and applies to all. Malware analysis helps in developing signature of a malware. Signature can be host-base or network based. Host based signatures are used to detect malicious piece of software on victim machine. Whereas network based signatures are used to detect malicious code by monitoring the network traffic. The two fundamental approach of malware analysis are:

- Static malware analysis
- Dynamic malware analysis

1.1 Static Malware Analysis

Static analysis [2,3,4,5,6,7] of malware includes examining the executable without running it .In static analysis important steps are to find out referenced ASCII or UNICODE strings in executable and read instruction by using a disassembler.

Following static analysis tools can be used during static malware analysis:

- Md5deep
- Virustotal.com
- Strings utility
- PEiD
- Dependency Walker
- PEView

1.2 *Dynamic Malware Analysis*

Dynamic analysis [2,3,4,5,6,7] of malware is performed after running it in controlled environment. Results obtained from dynamic analysis are much more reliable as compared with static as those can be mere guesses. Tools used during dynamic malware analysis are:

- Ollydbg
- Sandbox
- Process viewer
- Regshot
- Fakenet
- ApateDNS
- InetSim
- Virtual Box

During malware analysis especially in dynamic analysis we require a controlled environment in order to contain the virus in a box. The next section discusses in detail how this controlled environment is set up for malware analysis.

2. MALWARE ANALYSIS ENVIRONMENT

Malware analysis environment [5] consists of two hosts: the malware analysis Windows virtual machine and Linux virtual machine running INetSim. Linux machine is listening on various ports through the use of INetSim. The DNS server of windows is configured to loop back address (127.0.0.1). ApateDNS is used to redirect all requests to Linux virtual machine. All request or domain will resolve to Linux machine IP address which will make it easy to

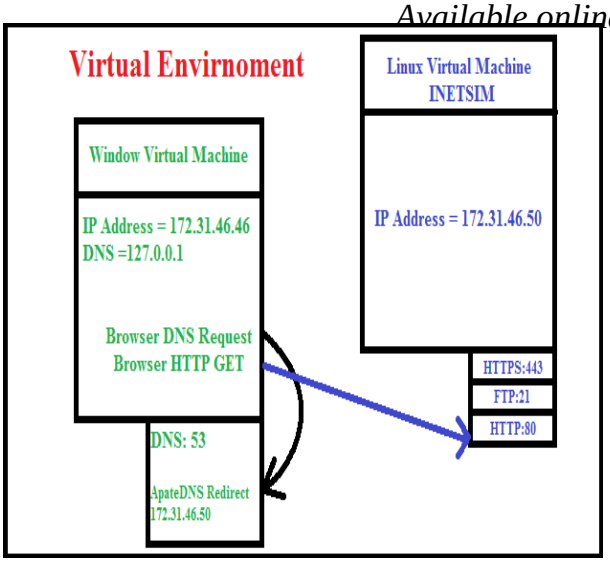

Fig. 1. Malware Analysis Environment

analyze the network behavior of infected machine with the help of INetSim.\

3. COMPLETE ANALYSIS OF LOL ZIP VIRUS

As discussed in previous section now we will employ traditional malware analysis technique in order to dissect this new virus. Figure 2 show LOL zip virus message. Initially we have following information about this malware:

- This message is sent from victim's computer without his knowledge to all his Facebook

Fig. 2. Facebook Message containing Malware

friends.

- Victims also reported sluggish performance of system after the infection.

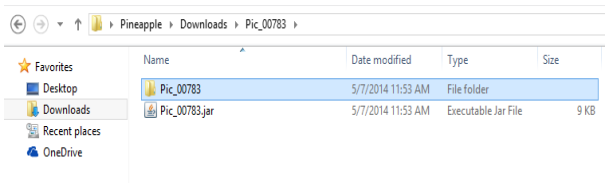

Fig. 3. JAR file inside Zipped folder

We will safely download this file on our virtual

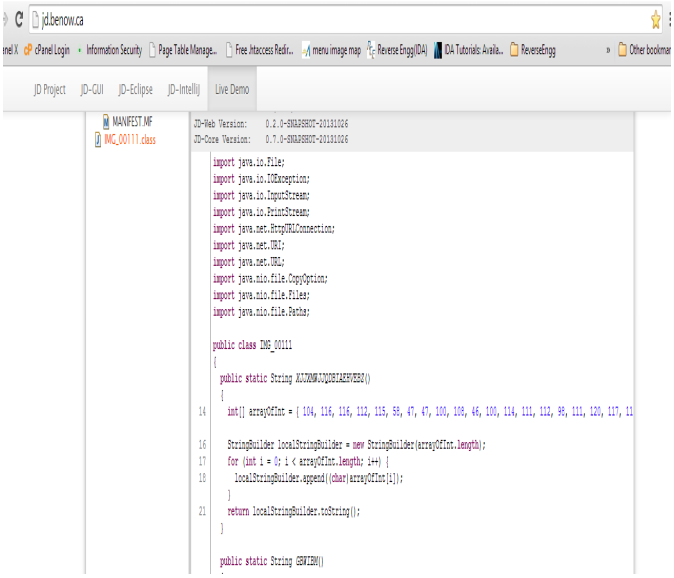

Fig. 4. Decompiled Jar File

controlled environment to carry the further analysis. After unzipping it we found a JAR (Java Archived File) in it.

Using online available java decompiler [8] we have converted the byte code (JAR file) to source code file.

3.1 *Decompilation and comprehension of JAR File*

Source of JAR file is obfuscated so it is needed to decompile such a file. We will use simple print instruction to find out the meaning of all the functions. After analyzing each function we have collected following information. These operations are performed by JAR file in order to download and install the actual malware.

- Jar file creates a directory in C drive if not present, namely C:\\TEMP\\
- Jar file created file name 081C854F.dat as hidden alternate stream of TEMP folder.
- Jar file then opened the http connection to location https://dl.dropboxusercontent.com/s/+string1="f6pb6ya5e5vc0q5/d.dat?dl=1@@@21urb4zg9n2on4s/d.dat?dl=1@@@8h46y6oodr4dxxf/d.dat?dl=1@@@8opd2hw50b974mx/d.dat?dl=1@@@tix3692duv4yts9/d.dat?dl=1"
- "@@@" is splitter that means fg6oxlrzau4egd5/d.dat?dl=1 constitutes one link. Jar will be downloading something from one of these links. Jar file will try all the links in order to download the malicious DLL.

- Jar file then downloads a file named d.dat. Then the jar file copies the content of d.dat (Main malicious file) to 081C854F.dat
- Final step is to run d.dat or 081C854F.dat. Lastly the Jar file installs the malicious file as a system level service using following command:
  regsvr32 /s C:\\temp: 081C854F.dat (/s stands for silent install.)

```
59127 e|,R
59128 USERPROFILE
59129 Control Panel\Desktop\
59130 WallPaper
59131 Bliss
59132 \Desktop\*
59133 <unknown>
59134 <unknown>
59135 Checking PID: %d
59136 ntdll.dll
59137 ZwUnmapViewOfSection
59138 TEMP
59139 %s\%s
59140 enfodthisfile!
59141 %s\%s
59142 enfodthisfile!
59143 enfodthisfile!
59144 %s%d
59145 enfodthisfile!
59146 %s%d
59147 %s\%s
59148 C:\Temp:rnd
59149 %08X
59150 %s%08X
59151 NUMBER_OF_PROCESSORS
59152 explorer -o stratum+tcp://%s:%d -O %s:%s -t %d -R 1
59153 explorer -o stratum+tcp://%s:%d -O %s:%s -t %d -R 1
59154 explorer -o stratum+tcp://%s:%d -O %s:%s -t 1 -R 1
59155 explorer -o stratum+tcp://%s:%d -O %s:%s -t %d -R 1
59156 explorer -o stratum+tcp://%s:%d -O %s:%s -t 1 -R 1
59157 explorer -o stratum+tcp://%s:%d -O %s:%s -t %d -R 1
```

Fig. 5. Strings.exe Output

3.2 *Static Analysis*

Next we will use strings [10] utility to dump out all the ASCII and UNICODE present in d.dat (DLL file) which we downloaded from the URL found above.

We found temp directory using strings utility inside the malicious DLL d.dat. This temp directory contains the Alternate Data Stream which is denoted by :rnd. So the Temp directory confirms the previous deduction. Also many commands were found which were referring to the explorer.exe.

If we go to TEMP and do "dir /r" then we can see the hidden alternate data stream in infected machine. Tools like alternate data stream viewer can be used to find out the contents of these alternate data streams.

Alternate data stream of TEMP folder contains 5 files as shown in Figure 6.

- d.dat (DLL file)
- PID1
- PID2
- SRV
- Rnd.dat/List file

If rnd.dat/list file is opened after extracting alternate data stream of TEMP folder then we will find profile id of all Facebook friends separated by comma inside it:

```
C:\temp>dir /r
 Volume in drive C has no label.
 Volume Serial Number is C6F3-879A

 Directory of C:\temp

05/12/2014  09:20 PM    <DIR>          .
                         3,975,680 .:081C854F.dat:$DATA
                                 4 .:pid1:$DATA
                                 4 .:pid2:$DATA
                                 4 .:rnd.dat:$DATA
                                22 .:srv:$DATA
05/12/2014  09:20 PM    <DIR>          ..
               0 File(s)              0 bytes
               2 Dir(s)  22,211,162,112 bytes free
```

Fig. 6. NTFS Alternate Streams found in infected windows8

1540406190,1088325702,100003798814520,100001180842834,1577102434,1826467191,100002007766718,100000384349191,100000274840242,100001620936327,100000011479860,100006282366359,100000110848326,100000454653025,100001517571086,837101841,100006592025690

These Facebook profile IDs are used by the virus to further replicate to other users via Facebook. This virus replicates itself from victim account to the accounts of people in victim's profile by sending a message containing copy of virus.

If SRV file is opened then a link to a file containing a message from the creator of this virus is found inside it: http://ideone.com/plain/74R13B

```
59072 libcurl-4.dll
59073 pthreadGC2.dll
59074 zlib1.dll
59075 minerd.exe
59076 C:\Temp:pid1
59077 C:\Temp:pid2
59078 1HExTe48BHniHC7kmUDciLLjepgzna1wpZ
59079 d=16
59080 useast.wafflepool.com
59081 uswest.wafflepool.com
59082 eu.wafflepool.com
59083 sea.wafflepool.com
59084 explorer.exe
```

Fig. 7. Strings.exe Output

The message says virus aim is CPU mining.

In order to confirm that the virus actually performs CPU mining we will dig further.

DLL (d.dat) will run from a hidden alternate data stream. So the malicious DLL is hidden on TEMP folder alternate data stream.

By carefully going through the strings we can conclude

```
58763 Usage: minerd [OPTIONS]
58764 Options:
58765  -a, --algo=ALGO       specify the algorithm to use
58766                          scrypt    scrypt(1024, 1, 1) (default)
58767                          sha256d   SHA-256d
58768  -o, --url=URL         URL of mining server (default: http://127.0.0.1:9332/)
58769  -O, --userpass=U:P    username:password pair for mining server
58770  -u, --user=USERNAME   username for mining server
58771  -p, --pass=PASSWORD   password for mining server
58772      --cert=FILE       certificate for mining server using SSL
58773  -x, --proxy=[PROTOCOL://]HOST[:PORT]  connect through a proxy
58774  -t, --threads=N       number of miner threads (default: number of processors)
58775  -r, --retries=N       number of times to retry if a network call fails
58776                          (default: retry indefinitely)
58777  -R, --retry-pause=N   time to pause between retries, in seconds (default: 30)
58778  -T, --timeout=N       network timeout, in seconds (default: 270)
58779  -s, --scantime=N      upper bound on time spent scanning current work when
58780                          long polling is unavailable, in seconds (default: 5)
58781      --no-longpoll     disable X-Long-Polling support
58782      --no-stratum      disable X-Stratum support
58783  -q, --quiet           disable per-thread hashmeter output
58784  -D, --debug           enable debug output
58785  -P, --protocol-dump   verbose dump of protocol-level activities
58786      --benchmark       run in offline benchmark mode
58787  -c, --config=FILE     load a JSON-format configuration file
58788  -V, --version         display version information and exit
58789  -h, --help            display this help text and exit
58790 MhP?
```

Fig. 8. Minerd.exe Operation List

- Virus injects minerd mining utility in explorer.exe memory.
- Virus is registered as a service in system using a DLL file stored in alternate data stream of C:/TEMP folder.
- Attacker have used curl module to automate LOL zip message to Facebook friends of the victim.

Quickstart guide

- Download a compatible miner:
  - cgminer 3.7.2 (for radeon gpu)
  - cudaminer (for nvidia gpu)
  - cpuminer (for cpu mining)
  - BFGMiner (gpu alternative)
- Run your miners with these parameters:
  - **Host:** Pick the closest to you!
    - **US East:** stratum+tcp://useast.wafflepool.com:3333
    - **US West:** stratum+tcp://uswest.wafflepool.com:3333
    - **Europe:** stratum+tcp://eu.wafflepool.com:3333
    - **Asia:** stratum+tcp://sea.wafflepool.com:3333
  - **Username:** Your Bitcoin Address [invalid mining addresses are considered a donation to the pool]
  - **Password:** Any Password is acceptable
  - **Mode:** --scrypt
- Example: ./cgminer -o stratum+tcp://useast.wafflepool.com:3333 -u your_bitcoin_address -p x --scrypt
- More mining setup help can be found here

Fig. 9. Wafflepool Website [9]

3.3 Dynamic Analysis

We will now look at InetSim data to confirm the hypothesis made after doing static analysis:
InetSim log shows and confirms the web address that we found as ASCII string in Malicious DLL. Figure 7 below shows the site (wafflepool.com) and confirms that explorer.exe is injected with minerd utility code.

Thus we can conclude form Figures 9 and 5 that explorer.exe has been injected with malicious code. Explorer does not take any command normally. Therefore the commands given to explorer in Figure 5 were able to run because of the injected minerd utility. The injection was done by LOL file in the main memory.

4. CONCLUSION

We concluded the following operations which are being performed by this malware sample.

- Virus injects minerd mining utility in explorer.exe memory. This executable is used for mining of bitcoins. These BitCoins can directly be redeemed for money in digital world. BitCoins is a peer to peer internet payment system in which the bitcoins serve as internet money.
- Virus is registered as a service in system using a DLL file stored in alternate data stream of C:/TEMP folder. This will ensure that the malware gets system privilege and runs automatically every time victim logs on in Facebook.
- Attacker have used curl module to automate LOL zip message to Facebook friends of the victim. This ensures that as soon as a victim opens the LOL jar file, all the Facebook friends of the victim receive a copy of LOL zip file as well.

This confirms that aim of attacker is to earn BITCOINS by CPU mining which also fits with our initial information i.e. degraded CPU performance in victim's machine.
Both the static and dynamic analysis techniques were able to uncover the details of the malware sample. Both the methods were used in conjugation with each other to confirm the findings.

REFERENCES

[1] Computer Economics, 2007 Malware Report: The Economic Impact of Viruses, Spyware, Adware, Botnets and Other Malicious Code, Retrieved 2007, November 23 from http://www.computereconomics.com/article.cfm?id=1225

[2] Eldad Eilam, (2005). Reversing: Secrets of Reverse Engineering.: WileyPublishing.

[3] Ed Skoudis,& Lenny Zeltser (2003). Malware: Fighting Malicious Code. Upper Saddle River,NJ: Prentice Hall PTR.
[4] Skoudis, E., & Liston, T. (2003). Malware: Fighting Malicious Code. Prentice Hall.
[5] Zeltser, L. (2006, November 11). Virtual Machine Detection in Malware via Commercial Tools. Retrieved January 18, 2007, from SANS Internet Storm Center Website: http://isc.sans.org/diary.html?storyid=1871&rss
[6] Michael Sikorski and Andrew Honig,(2012).Practical Malware Analysis, No starch press
[7] Dennis Distler,Malware Analysis: An Introduction http://www.sans.org/reading_room/whitepapers/malicious/malware-analysis-introduction_2103
[8] Java Decompiler http://jd.benow.ca/
[9] A profit based altcoin mining pool, www.wafflepool.com
[10] Mark Russinovich, Strings.exe, Strings v2.52, http://technet.microsoft.com/en-in/sysinternals/bb897439.aspx