ISSN 2277-3061

Analysis and Detection of Bot performing Keylogging Activities

Miss.Takale Jyoti D.
Department of Computer Science & Technology
Shivaji University, Kolhapur, Maharashtra, India.
jyotitakale@gmail.com

ABSTRACT

The focus on computer security has increased due to the ubiquitous use of Internet. Botnets are one of the biggest cyber threats. Botnet is a malware controlled by a Botmaster using Command and Control (C&C). Botnet is expanded with infecting fresh computers through social networking sites like facebook, twitter, etc. ZeuS is famous type of botnet for financial gain. It targets bank websites for stealing user's credentials like password, credit card information,etc. In this paper, an application framework is designed for analysis and detection of ZeuS bot residing on host victim's machine. The detection phase is based on analysis of bot's infection stategy means in what way it affects the victims's pc. All the related files are wiped out from the system in removal phase. The communication between command and control server and the victim machine is analysed in a virtual environment.

Indexing terms/Keywords

Botnet; Bots; Botmaster; Command and control server; Zeus.

ISSN 2277-3061

INTRODUCTION

The remarkable growth of the Internet technologies over the past few years changes the lifestyle of most people. The Internet applications are mistreated by perpetrators and hackers for committing different kinds of crimes. The extensive use of Internet motivates the malicious activities which took place over the past several years. Formerly malicious programs have been classified as viruses, worms or Trojan horses based on their behaviors. The social networking sites such as Facebook, Bebo, and Twitter being used by hackers to spread malware easier and faster than before [5]. A Botnet is a group / network of infected machines that are controlled by a Botmaster using command and control (C&C) mechanism [2][3]. Botnet targets data stealing, combating cyber attacks such as DDoS and hacking into bank accounts to get a financial gain. ZeuS is one of the latest Windows based botnet with latest techniques being used by botnets of current age. The majority of zeus infections are unnoticed by antivirus products.Most of the products are unable to detect bot's presence as it hides itself and appear as a system file on the victim's machine.The bot could be detected only prior to the executable execution.The large number of zeus infections occur on machines which have an installed an up to date antivirus product[14].

LITERATURE SURVEY

A. Evolution of Botnets

This section describes the history of Botnets [6]. Timeline for common Botnets from 1999 up to 2011 is shown in Figure 1. The first GMBot was developed in late 1980s. Its objective was to emulate a live person IRC sessions. GMBot was not a malicious bot. In 1999 first malicious IRC based bots Sub7 and Pretty Park were emerged. From time to time, the objectives of Botnets developed from corrupting or stealing computers data to financial gain or a way to make a huge amount of fortune. Zeus bot in 2006 sold for several thousand dollars. Initially Botnets utilised IRC and then further developed to use more sophisticated protocols such as HTTP, ICMP and SSL for C&C of a compromised network [1]. In mid-2011, code for bots Zeus was leaked and the secrets behind development of these Bots were exposed to anyone who wants to create their own Botnet.

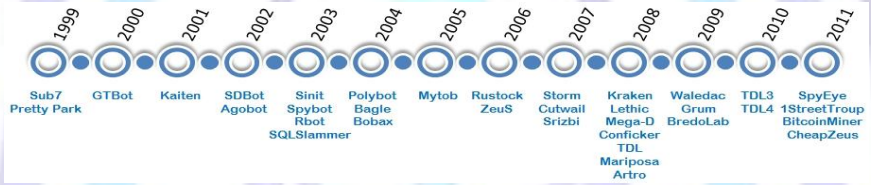

Figure 1: History of Botnet

B. Botnets

The Command & Control(C&C) structure is the way a Botmaster communicates with the slaves or Bots. When a bot is installed on the victim machine, it communicates and synchronizes with the C&C server. The Steps for synchronization of a bot with a C&C server is shown in Figure 2. When a bot is synchronized with its C&C server, it registers it in its database and bot becomes ready to listen to C&C server for commands and scripts to perform operations as requested by Botmaster on the C&C server [7].

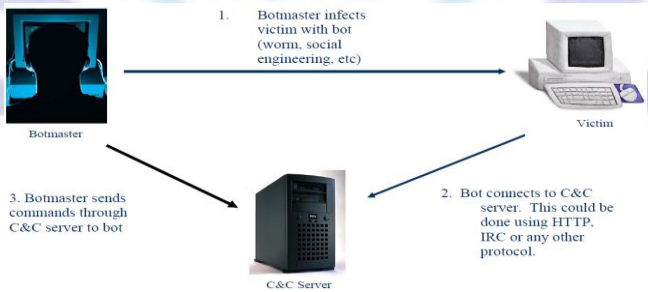

Figure 2: Botnet Life Cycle

ZEUS BOTNET OVERVIEW

Zeus bot originated in Russia and was first identified in 2007, when it was used to steal information from the U. S. Department of Transportation. It got spread in March 2009 and currently it is belonging to the top active botnets. In 2009, security company Prevx discovered that ZeuS compromised 74,000 FTP accounts on popular websites including Bank of America, Monster.com, Oracle, Cisco, Amazon and NASA [8]. The main purpose of Zeus bot is financial gain stealing online credentials such as email, online banking, and other online passwords. ZeuS is known by several names like

WSNPoem, PRG, and others. Zeus has ability to steal the certificate store with private keys, to steal cookies of the specified URL and to hijack banking sessions and inject custom data into returned HTML.

The new computers are infected to expand a botnet. Key logging is a technique used by a Botnet to store the keystrokes while typed and sent to the hacker [12]. Some banks use a technique called virtual keyboard that is on-screen keyboard. ZeuS botnet counter this attack by capturing the screenshots [4]. Normally, general keylogger captures every keystroke, but Zeus can be configured to grab the form information from targeted bank web sites and online stores, etc. So, the botmaster's sorting time to go through large quantity of useless keylogged data is saved. All the keylogged information is stored in an SQL database and a query tool is used to access it.

The Zeus comes in the form of Crimeware Toolkit [13] for creating botnets.The toolkit is a set of programs which is designed to setup a botnet over a high-scaled networked infrastructure. The toolkit consists of following components:

a) Control panel: It consists of a set of PHP scripts. The botnet is monitored through this control panel and stolen information is stored into MySQL database.

b) Configuration files: They are used for customizing the botnet parameters. It contains two files: the configuration file config.txt which lists the basic information and other, webinjects.txt identifying the targeted websites and defines the content injection rules.

c) Builder program: It generates the encrypted version of configuration file called config.bin and the malware binary file bt.exe.

Network Communications

The communication between the bot and the command and control server is done using the HTTP protocol(Figure 3). The data is encoded using RC4 and the key specified by the botmaster. Initially, the bot sends a GET request to the botmaster to retrieve the configuration file.The server replies with the configuration file. Whenever the bot needs to send information to the command and control server, it sends a POST request to the url_server URL specified in the dynamic configuration file. In response, the server sends a HTTP/200 with an OK code. The server may also send additional data to be executed by the bot, such as script commands created by the bot master.

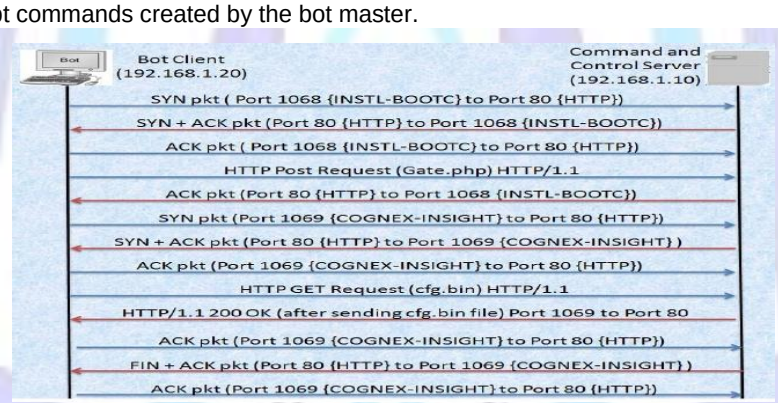

Figure 3: Communication Pattern of Zeus Botnet

The analysis of ZeuS bot is performed in three layers i.e. binary, application and communication layer[15].

1. Binary layer analysis: The static binary analysis of bot executable itself is done before execution. The PE Explorer tool is used along with REC studio to perform reverse engineering of the bot executable including headers info.

2. Application layer analysis: This type of analysis is performed using Procmon to analyse file system, registry and processes during ZeuS bot configuration binary building using "ZeuS Configuration Builder" tool, ZeuS bot executable binary building using "ZeuS builder" tool, ZeuS bot removal using "ZeuS Configuration Builder" bot removal feature and Bot executable "bot.exe" execution.

3. Communication layer analysis: The communication between ZeuS bot and the control panel (CP) on the ZeuS C&C server is analysed using Wireshark by capturing the packets sent/received by ZeuS bot from ZeuS C&C server. These packets are analysed in terms of port numbers and bytes pattern to create snort rules to implement host based intrusion detection system (HIDS) on the "Bot victim" machine.

Reverse Engineering

The reverse engineering steps are presented to deobfuscate the malware[13]. Also, configuration information is extracted from the binary and its structure is presented. A method is described to extract the encryption key which is used to decrypt communications and decrypt the local storage of stolen information and additional configuration information. The reverse engineering analysis is performed on the Zeus crimeware toolkit components. The two binaries are analysed: the Zeus builder and Zeus bot executable. *IDAPro*, the disassembler is used for analyzing the binaries. *IDAPro* interprets a string of bytes as mnemonics of machine instructions.

ISSN 2277-3061

1. The Zeus Builder Program Analysis

Zeus builder is one of the components of the crimeware toolkit. It configures a Zeus botnet and generate the bot executables. Using *PEiD* it comes to know that the builder is packed using *UPX* and must be unpacked using *UPX*. After this, the *PaiMei is used for* reverse engineering framework, to see which functions within the Zeus builder are invoked by a specific action within the GUI.

2. Zeus Bot Binary Analysis

The bot binary file contains four segments: a *text/code* segment, an *imports* segment, a *sources* segment, and a *data* segment. The analysis is done at the malware entry point (EP) residing in the *text/code* segment.

De-obfuscation Process

The *IDAPro* debugger is able to debug the malware and step through the instructions to analyze and understand the logic of the de-obfuscation routines(Figure 4). Each routine revealed some information which is used by the other routines until all obfuscation layers are removed.

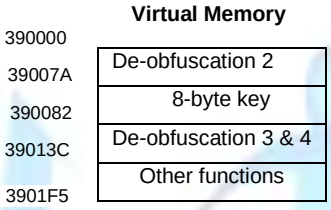

Figure 4: De-obfuscated Code in Virtual Memory

The Zeus bot employs two additional layers of obfuscation. These two layers are de-obfuscated during the installation procedure. The logic behind these two routines is described in Algorithm 1 and Algorithm 2.

```
Algorithm 1: DECRYPT_STRING(enc_string)
seed = 0xBA;
String new_string = new String(enc_string.length());
for i = 0 to enc_string.length()
do
    new_string[i] = (enc_string[i] + seed) %256;
    seed = (seed + 2);
return (new_string)
```

```
Algorithm 2: DECRYPT_URL(enc_url)
String new_url = new String(enc_url.length());
for i = 0 to enc_url.length()
do
    if (i%2 == 0)
    then
    new_url[i] = (enc_url[i] + 0xF6 - i * 2) %256;
    else
    new_url[i] = (enc_url[i] + 0x7 + i * 2)%256;
return (new_url)
```

Automatic Key Extraction

The substitution table from recovered static configuration can be used to decrypt all the communications of the Zeus botnet. It can be used to decrypt the dynamic configuration file as well as the stolen information. Extract the 256-byte substitution table located at memory address 0x41602A. This key will allow for decrypting the botnet's communication traffic and all the encrypted files.

DETECTION METHOD

A. Bot Detection System

ZeuS botnet experimental setup and analysis is performed in a virtual environment (Figure 5) having two machines i.e. "Bot victim and "C&C server" that are isolated from host machine running VMware and the Internet [11].

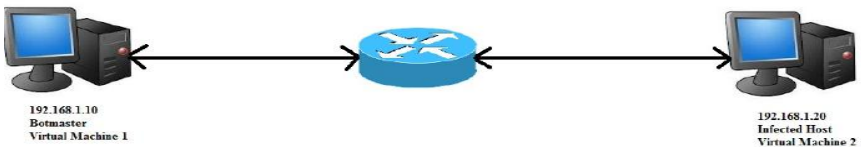

Figure 5: Diagram of Experimental Virtual Network

Command & Control Server:

- Install the ZeuS Command and Control Panel using web server like XAMPP .
- Build bot executable,bt.exe using zeus builder program.
- Send the bot executable towards the other machine which is to be get infected.

Victim Machine:

- Install the bot executable on victim's machine to infect it with bot.
- Visit a bank website and enter user credentials at the website.
- The proposed system is designed and implemented at the infected host.

- The designed system checks for possible infection of bot and
- Removes the bot performing keylogging activities from the victim's machine successfully. Bot Removal could be verified by accessing the bank website and giving credentials details and check whether they are being logged at the botmaster.
- Analyze the Zeus bot at different layers like application layer & communication layer using tools like ProcMonitor & Wireshark respectively[9][10]**.**

The proposed detection system works as per the following functions as shown in the Figure 6.

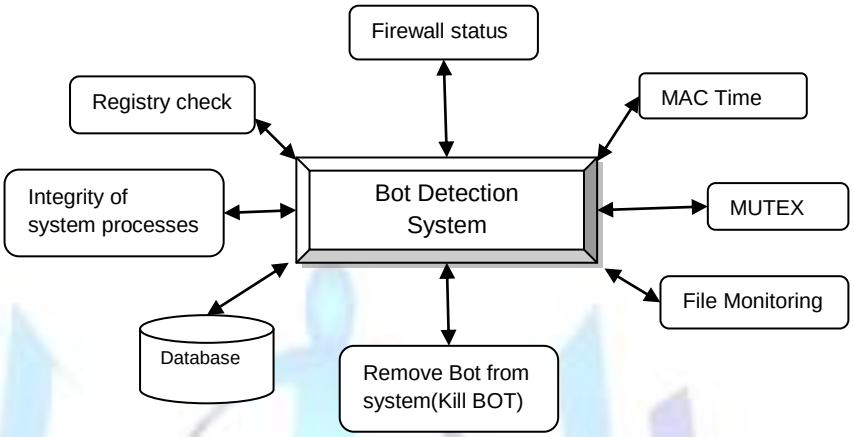

Figure 6: Bot Detection System

1. Firewall Status

The module should check the firewall status from registry settings. As the zeus bot gets installed on victim's machine, first of all it disables the firewall. This is the indication to know about something undesired activity is going on. The designed system would detect status of firewall whether it has been on/off.

2. Enable Firewall

If firewall has been switched off, the system enables the user to switch on by checking the registry settings. If the firewall is not disabled, the user should be notified that firewall is already enabled.

3. Check Bot Infection

The possible bot infection should be checked at the administrator and for all the users having account on that machine. The bot is able to infect multiple users on the same victim's machine.

- *Analyze Windows/System32 path:* The zeus bot creates certain hidden files(sdra64.exe) and folders(lowsec) in windows/system32 path. All the hidden files and folders from windows/system32 are analyzed in terms of certain aspects to determine suspicious activity.
- **Process injections:** The bot injects it's code into certain processes. The initial hash value of important key system processes like winlogon.exe,svchoost.exe which are to be get infected by the bot executable is calculated using Seure Hash Algoritham (SHA) and stored on the machine. This process is done on clean installation of WindowsXP. The newly calculated hash value of system processes is compared with the old one. If both differs, then there is possible rootkit infection in the victim's machine.
- **Check Registry Entries:** Zeus makes registry changes using one of two subkeys to ensure that the sdra64.exe dropper is executed upon startup.

I. If the logged-in account at the time of infection has administrative privileges:

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\WindowsNT\CurrentVersion\Winlogon\"Userinit"="% System%\userinit.exe,%System%\sdra64.exe"

II. If the account has limited privileges:

HKEY_CURRENT_USER\SOFTWARE \Microsoft \Windows \CurrentVersion \Run\"userinit" ="%UserProfile %\ Application Data\sdra64.exe"

The userinit registry key's value is checked to determine whether it is appended with additional path or not.The module should check these two subkey's entries.

- **MAC:** The bot duplicates the Modification Access, and creation times (MAC times) information from Ntdll.dll library, and applies them to the sdra64.exe. This is carried out to hide itself and make sdra64.exe behave as a system file created when the operating system was first installed. The MAC information of hidden files in windows/system32 is checked to find out a suspicious file.

- **Check Mutexes:** The communication between various injected components of processes is done with mutexes and pipes, maliciously named _AVIRA_x, where x is a number (eg: x=2109 in winlogon.exe, x=2108 in svchost.exe). Each Zeus infection creates a mutex with the name _AVIRA_2109, so Zeus can be detected by attempting to open the mutex _AVIRA_2109.

The weights are identified using Analytic Hierachy Process(AHP) technique and assigned to each possible infection level and the threshold is set. If the resulting outcome crosses the threshold value, then suspicious bot's activity is detected successfully.

4. Bot Removal

This module should remove the bot infection from all the users who have been infected by the botnet. The running processes maintaining registry entries are unhooked. The registry key is corrected to ensure that malware is not loaded at each reboot. The function should remove the files and folders created by the bot executable inside the infected computer. This module should first call the "Check Infection" module and if an infection is found should perform its task.

5. Monitoring Module

FileSystem Watcher which is in continuous monitoring state, monitors the active directories where the bot is likely to create its files and folders on a successful execution. On suspicion it should check for particular files and folders and if found should generate alert messages to the user in a particular format.

CONCLUSION

Botnets pose a significant threat to the Internet community. The paper aims at discussing the concept of botnets, botnet topologies and the attacks being carried out by botnets. Botnet detection through host-based and network-based detection was also described. A famous banking bot, known as Zeus Botnet is discussed with all aspects. It's working mechanism, infection method and how to use to target Indian banking websites is described. We also presented an approach to detect bot infection by going through possible infection levels. Zeus bot is analysed at certain layers namely application & communication. An analysis is performed on the Botnet's infection procedure and a mechanism is proposed based on the findings.

ACKNOWLEDGMENTS

For sound and safe building a strong foundation is prime requirement.Similarly for any project to be a great success a good guidance is required. I would like to express my deep gratitude towards my guide Prof. P.C. Bhaskar, Coordinator & Head of Department, Department of Electronics,Department of Technology, Shivaji University, Kolhapur for their valuable guidance and constant motivation.

REFERENCES

[1] Trend Micro (Nov. 2006), Taxonomy of Botnet Threats: Trend Micro

[2] Microsoft (2012), Botnets Today - HTTP Botnets, Retrieved 08 May, 2012 from http://www.microsoft.com/security/sir/story/default.aspx#!botnetsection_http

[3] Microsoft (2012), Botnets today - Other Protocols, Retrieved 26 Mar, 2012 from http://www.microsoft.com/security/sir/story/default.aspx#!botnetsection_other

[4] Shah, C., (Sep 2010), Zeus Crimeware Toolkit, Retrieved 14 Jan, 2012 from http://blogs.mcafee.com/mcafee-labs/zeus-crimeware-toolkit

[5] Nazario, J. (Aug. 2009), Twitter-based Botnet Command Channel, Retrieved 14 Feb, 2012 from http://ddos.arbornetworks.com/2009/08/twitter-based-botnet-command-channel/

[6] Schiller, C. Binkley, J., Harley, D., Evron, G., Bradley, T., Willems, C., Cross, M. (Eds.), (2007), Botnets- The Killer Web App, Massachusetts: Syngress.

[7] Kamluk, V. (May. 2008), The botnet business, Retrieved 22 Jun, 2012 from http://www.securelist.com/en/analysis/204792003/The_botnet_business

[8] Raju, P., (Jul. 2010), Zeus/Zbot Trojan Attacks Credit Cards of Banks, Retrieved 19 Apr, 2012 from http://techpp.com/2010/07/15/zeuszbot-trojan-attacks-credit-cards-of-banks/

[9] Russinovich, M., and Cogswell, B., (Jun. 2012), Process Monitor v3.03, Retrieved 02 Jan, 2012 from http://technet.microsoft.com/en-us/sysinternals/bb896645.aspx

[10] Kurose, F., and Ross, K., (2007), Wireshark Lab: Getting Started, retrieved 09 Aug, 2012 from

http://www.eng.tau.ac.il/~netlab/resources/booklet/Wireshark_INTRO.pdf

[11] Zeltser, L., (May. 2007), Using VMware for malware analysis, Retrieved 09 Jul, 2012 from

http://searchsecurity.techtarget.com/tip/Using-VMware-for-malware-analysis

[12] Spamlaws (2012), Avoiding Keystroke Loggers, Retrieved 08 Aug, 2012 from

http://www.spamlaws.com/keystroke-loggers.html

[13] Binsalleeh, H., Ormerod, T., Boukhtouta, A., Sinha, P., Youssef, A., Debbabi, M., Wang, L. (Eds.), (Aug. 2010), On the Analysis of the Zeus Botnet Crimeware Toolkit, Quebec: IEEE

[14] Zeus and Antivirus

[15] Shahzad Waheed, (2012), Implementation and evaluation of a botnet analysis and detection methods in a virtual environment, http://researchrepository.napier.ac.uk/5667/1/Waheed.pdf

Author's Photo