Botnet Detection

Ji ung Choi

UAHuntsville
The University of Alabama
in Huntsville,
Huntsville, AL 35899 USA

Department of Electrical
and Computer Engineering

Advisor: Dr. Yoo

7/31/2012

1. Introduction

Recently, there are many cases that malware bots damage our own computers. Skilled botmasters are using elaborating tricks and hiding skills regarding to botnet activities. For this research, I have found bots that control infected systems, through SSL communication. These bots are generally communicating servers with encrypted SSL communication and it makes very hard to monitor bots' activities with packets. The communication port that I have found was not using the IRC port. The bot I was working on was one of the Phatbots. Analyzing the botnet's Command and Control (C&C) server, I found its IP addresses. Then I analyzed the system and the files that were infected. I had encrypted communication to control the infected system. I installed a Rootkit and unreal Internet Relay Chat daemon (IRCd). When an infected system tried to communicate with C&C server, the communication was encrypted to avoid a catch-up. I used an Internet Explorer to connect to the server with TCP/436 port. To confirm SSL communication, I used https protocol. The botnet C&C server was listening TCP/436 server. Then, I have used IRC client to connect the botnet C&C server. The C&C server was communicating with 168 bit SSL. I accessed a specific channel that the botmaster already set up and performed additional commands ordered by the botmaster. After the infected system was connected to the botnet's C&C server, it communicated with a specific channel to install spyware, adware, and more. To follow the commands that botmaster had ordered, I downloaded those wares to install in the infected system. Some specific bots infect others without connecting C&C server. However, most of them have to connect to C&C server to communicate with the botmaster. After monitoring the botnet's C&C server for several days, the botmaster keeps trying to infect others as follows.

Channel	Users	Topic
#!!PHAT!^#	422	[$$$$]

Generally, botmasters command their orders to infected systems with channel topic. Above statement shows how the botmasters are using to attack some TCPs. A set-up file contained port numbers and admin passwords for infected system to use. The botnet C&C server was set by a server password not to be detected easily. Other than the port TCP/436, the server was using other ports too. I have used unreal IRCd for botnet C&C program. The botmaster have changed source code to be controlled only by itself. Also, it had blocked some specific IPs or domains to protect its own botnet C&C server. The botmaster seems to know how to avoid some networks or how to attack the other networks as the following a few statements show. These lines kill IRC connection on some specific domains or IPs.

```
M: 41.2.142.*:akilled:*
M:*.xxxx.com:akilled:*
M:*.gov:akilled:*
```

2. Method

Before my analysis, the number of infected systems by the bots through C&C server was over 300. I found some IPs to extract bot binary files. However, after I closed the server, over 1000 systems were infected. Most of worms, viruses, and bots are compressed, before they are spread, to reduce their sizes not to be detected or be analyzed easily. Some are even unable to decompress (e.g., Morphine). I have tried to vaccinate but no vaccines found malwares. The infected system's OS was Windows XP and the file was located on system32 folder. It has changed the system's registry and set on start-up menu. Then, how do bots follow commands after connected through C&C server? Some specific channel topics have scan.startall and then the system follows the command. Then, they start to have scanning attacks on some specific ports to spread bots using a weaker point that Windows OS has. Followings are few botnet codes for scanning.

```
MOV EAX.DWORD PTR DS:[XXX]
LEA ECX.DWORD PTR DS:[EAX+4]
```

```
CALL [xxxx], XXXXXXX
PUSH ESI
PUSH EBX
, , ,
```

Nowadays, bots are adding some more codes to be disassembled by detectors. If bots detect anti-debugging tools like Softice or OllyDbg [4][5], they terminate their programs right away. The following line is one of the codes to do that.

```
MOVX EAX, BYTE PTR[EAX+0X2]
```

Likewise, source codes for Phatbot keep updating their codes whenever they found some weaker points [5]. Also, some bots are calculating TCP/IP checksum in assembler to gain speed [1][2]. Botmasters insert SSL methods to encrypt their commands not to be monitored easily by network monitoring. The following lines are few codes for encrypting.

```
PUSH EAX
CALL xxxx
PUSH xxxy
PUSH 0001. . .
ADD ESP, xx . . . MOV EAX, 1 . . .    RETN, NOP, NOP, NOP …
```

By network monitoring, I have tried to capture a communication which was encrypted with SSL. However, I could not make decoding due to the encryption. To capture those communications, I need to study a sniffing method for future works [1]. Currently, most bots are not activated on VMWare because botmasters know that many honeypot methods are using VMWare to capture them and not to be analyzed. The following code is an example to avoid VMWare [4].

```
Int IsVMWarePresent() {
int version = Version();
if(version| return ture; else return false;}
```

3. Results

To detect bots and botnet C&C server, I need an intrusion detection tool or software to monitor network packets. Table 1 shows the ports that Bots used for the C&C server.

Port numbers				
3267	4141	5662	6666	7000
8029	8249	9136	9998	31031

Table 1. Normal ports that Botnet C&C server usually used

Nowadays, instead of those ports listed in Table 1, botmasters are using randomly selected ports to avoid conflicts with detectors [3]. The method that I used was somewhat successful and I could move forward. To detect botnet C&C server, I need to know the ports for which botnet C&C servers use frequently. Even though there are many unorganized ports, I need to get more information about suspicious ports. Bots are more evolving and they will try more enough not to be found. This research was not fully done for every botnet activities but I will keep trying to work on the most recent important threats.

4. Conclusion

There is a saying “You can always win when you know your enemy and you.” This research will be the first actual movement to advance my educational and career objective. Moreover, the knowledge I obtained from this research could produce my future security works. In order to get rid of those malicious botnets which control networks to harm our systems, we need to develop knowledge about them first. Even though I encountered the shortage of botnet resources, I could get some knowledge of the methods for enumerating bot-infected hosts and accurate assessment using entropy-based measurement and research about the servers that the botnet uses to command and to control. We can see botmasters are very intelligent to avoid our detection mechanisms. Therefore, we need security consciousness for everybody. There are already many botnet take-down methods out there. The IRC-based C&C is well spread and the most usages of hackers; however, hackers are also evolving to avoid their weaknesses by moving their bases to HTTP or P2P [2]. Also, as the mobile market grows so fast, we now need to focus on not only one specific but also every electric device. As we can see the malwares or bots in spam could harm mobile phones, the detection mechanism for spam over mobile phones could also be the problems in near future. The mechanism that I mentioned and researched here has not fully implemented, but I will keep making research and find out what is going on for the botnet world. The thing is that the nuclear weapon in the world is not a good thing because it kills people. However, if we develop our botnet nuclear bombs to get rid of those botnets, it will be the most successive invention, and it will cause people to live peaceful Internet lives. Now, let us find mitigation and take-down methods against advanced botnet architecture in the computer world studying botnet C&C, every electric device connected over network and others that have been evolved.

Acknowledgments

This research was funded by the RCEU program with funds provided the Presidents/Provosts office, funds provided by the Vice President for Research, funds provided by the Chemistry Department through their patent account, and external funding from the Alabama Space Grant Consortium.

References

[1] Gu, Guofei, “BotSniffer: Detecting Botnet Command and Control Channels in Network Trafic,” Georgia Institue of Technology, Atlanta, GA 30332.

[2] Zeidanloo, Hossein Rouhani, “Botnet Command and Control Mechanisms,” UTM International Campus, UTM

[3] Cho, Chia Yuan, “Inference and Analysis of Formal Models of Botnet Command and Control Protocols,” University of California, Berkeley

[4] http://old.honeynet.org/papers/bots/botnet-code.html

[5] http://pen-testing.sans.org/resources/papers/gcih/eradicating-masses-1-phatbot-106262