

A Study on the Analysis of Netbot and Design of Detection Framework

Kyoung Soo Han and Eul Gyu Im

Department of Electronics and Computer Engineering
Hanyang University, Seoul, Korea
{1hanasun, imeg}@hanyang.ac.kr

Abstract. Recently, the cyber-attacks using botnets are increasing. Attackers can execute the DDoS attack more easily by using tools, such as Netbot which is a kind of botnet, even if they do not have expert knowledge. Besides, Netbot includes functions that enable attackers to control and monitor compromised systems remotely, as well as to launch DDoS attacks. Therefore, it can be led to secondary damages because attackers illegally get the private information of users and data stored in the computer. Actually, many web-sites such as game item trading sites, internet portals and internet banking web-sites in Korea experienced DDoS attacks since 2007. In this paper, in order to detect the Netbot in an early stage and to reduce the damages, we constructed an environment for the Netbot analysis. In addition, we analyzed the changes of files, registries and traffics as well as malicious behavioral patterns of Netbot in zombie computers. We also proposed a framework to detect Netbot agents with these analyzed results.

Keywords: Netbot, Botnet, Netbot Analysis, Netbot Detection Framework

1 Introduction

Recently, cyber-attacks using attacking tools are steadily increasing on the Internet. Many attackers use botnets for cyber-attacks. Botnet is a kind of network and it consist of malicious codes called bot. Attackers compromise other user's computer with illegal intention to turn the computers into zombies. Thousands to tens of thousands of infected zombies can be connected through a network and remotely controlled by attackers[1][2][4]. One of botnets, Netbot is a HTTP-based botnet used for DDoS attack. It is a malicious program that not only infects computers like worms, but also controls systems while exchanging commands with them. Major functions of Netbot include DDoS attack and backdoor functions such as remote control. The infected computers can be abused for malicious behaviors such as illegally get the private information of users and data stored in the computers, attacking of specific servers and web-sites. Actually, many web-sites such as game item trading sites, internet portals and internet banking web-sites in Korea experienced DDoS attacks since 2007, which caused troubles such as disabled or delayed access[13]. Moreover, in these days, even if attackers do not have expert

knowledge they can launch the DDoS attack more easily by using tools such as Netbot, and cyber crimes such as claiming money using DDoS attacks are increasing.

In this paper, we discussed the functions of Netbot, system changes by Netbot, and the analysis results of attack traffic, and also described the design of detection framework.

2 Related Work

2.1 HTTP-based Centralized Botnets

Botnets are classified into IRC botnets, HTTP botnets, and P2P botnets depending on the protocol used for the delivery of attacker's control commands. They can also be classified into centralized botnets and distributed botnets depending on their structure. IRC botnets and HTTP botnets belong to centralized botnets whereas P2P botnets belong to distributed botnets[1]. Recent HTTP-based centralized botnets that are often used for DDoS attacks include Netbot, Black Energy, Fybot, and Prorat.

One characteristic of HTTP-based botnets are that they make detecting and blocking the attacks difficult using Web services that are used frequently on the Internet as a medium of command delivery, because it is difficult to block the port number-80 which is used for Web page transmission and has a high volume of normal traffic[1].

Figure 1 illustrates the structure of HTTP-based centralized botnets. Attackers can make cyber-attacks and spread bots to vulnerable PCs and servers by sending attack and control commands to zombies via a C&C (Command and Control) server.

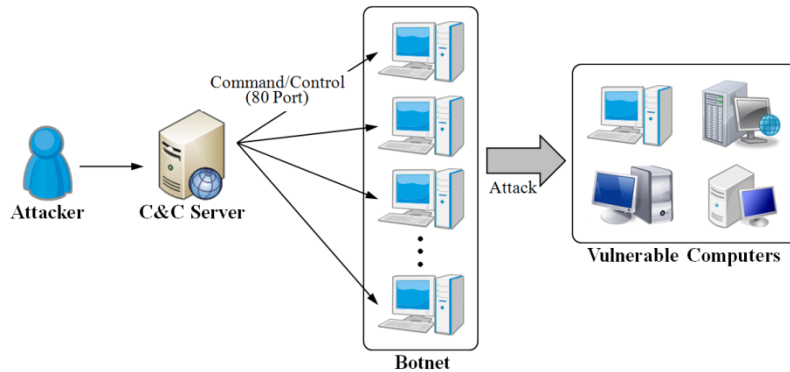


Fig. 1. Structure of HTTP-based botnets

2.2 Infection Routes of Botnets

User's computers can be infected with bots through various routes: By clicking a executable file attached in SPAM mail, by accessing a Web page that contains malicious codes, through the instant messenger, by downloading attached files of messages posted on a blog or community board. As shown in Figure 2, a survey conducted by S21sec[14] found that the greatest infection route was through browsers (65%), followed by mails (13%), vulnerability of operating systems (11%), file downloads (9%), and others (2%).

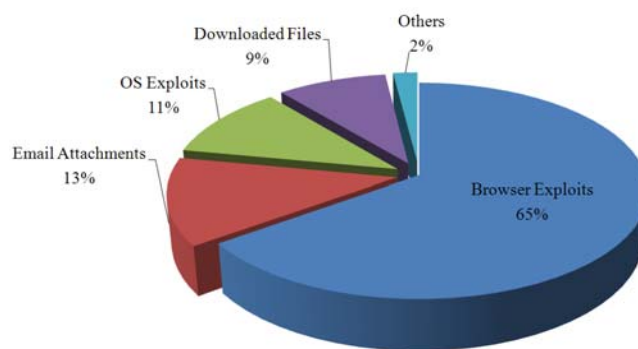


Fig. 2. Infection routes of bots

3 Analysis of Netbot's Characteristics

In this paper, we actually generated agent files with Netbot Attacker version 5.1 and infected computers with them in accordance with the operation process of Netbot so as to analyze the changes of files and registries, service changes, traffic, etc.

3.1 Operations of Netbot

Figure 3 illustrates the operation process of Netbot. First, an attacker generates an agent file which infects user computers using the Netbot Attacker tool, and attaches the agent file to messages posted on blog or community sites, or sends SPAM mails through a social engineering method. When users click and run the agent file, their computers are infected and become zombies, and a network of many zombies is organized. Each zombie downloads the IP address of C&C server from a relay site and accesses the C&C server of the attacker to let their existence known. The attacker remotely controls these infected zombies or make attacks using them.

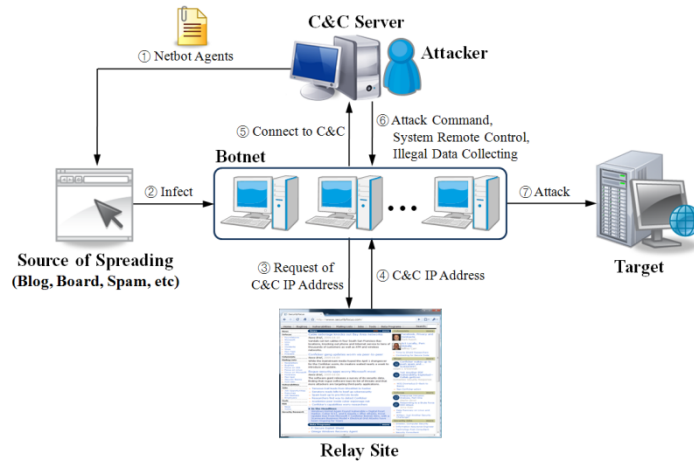


Fig. 3. Operation process of Netbot

3.2 Functions of Netbot

The functions of Netbot Attacker used by the attacker can be divided into remote control functions listed in Table 1 and DDoS attack functions listed in Table 2.

Table 1. Remote control functions of Netbot

Remote Control Menu	Description
Windows Explorer	Explore file(s) and folder(s)
Screen Monitor	Monitoring user's computer screen
Task Manager	Explore and terminate current process(es)
Command Line Interface	Execute cmd.exe
System Shutdown / Log-off	Shutdown and log-off the user's computer

Table 2. DDoS functions of Netbot

Attack Mode	Type of Attacks
Common Attack	SYN Flood, ICMP Flood, UDP Flood, UDP Small Size, TCP Flood, TCP Multi-Connect
Web Attack	No Cache Get Flood, CC Attack, HTTP GET Nothing
Special Attack	CQ Game Attack, Route Attack, Smart Auto Attack
Combine Attack	SYN+UDP Flood, ICMP+TCP Flood, UDP+TCP Flood
For Korean	Fin_Wait1 Attack, Fin_Wait2 Attack, Established Attack

Using the remote control menu shown in Table 1, the attacker can monitor the user's computers and acquire data stored in user's computers. The DDoS attacks shown in Table 2 can be largely divided into packet flooding attack and Web load attack. Packet flooding attack sends a large number of packets such as TCP, UDP, and

ICMP packets to the target in Common Attack which is an attack mode included in Netbot, thus causing network overload[5][11]. Web load attack corresponds to Web Attack mode included in Netbot, which sends a large number of requests to the target web-server. Web load attack brings about troubles in the web-server such as disabled or delayed access to web-pages by increasing the occupancy of the memory and CPU to over 90% [6][11]. The Web Attack mode of Netbot offers such functions as No Cache Get Flood, CC Attack, and HTTP GET Nothing. No Cache Get Flood or HTTP GET Nothing is a kind of HTTP Get Flood attack.

3.3 File Changes

We generated an agent file by Netbot Attacker and ran it on the analysis environment like honeypot and virtual machine. Filemon and Process Monitor[15], which are tools for checking file changes were used to observe file changes, and the results are summarized in Table 3.

Table 3. File changes by Netbot

Sequence	Description
0	Execute the agent file
1	Generate the ' <i>1955984_res.tmp</i> ' in C:\Documents and Settings\User\Local Settings\Temp
2	Rename ' <i>1955984_res.tmp</i> ' to ' <i>NetNtEx.dll</i> '
3	Move ' <i>NetNtEx.dll</i> ' to C:\WINDOWS\system32
4	Modify ' <i>beep.sys</i> ' in C:\WINDOWS\system32\drivers
5	Automatically delete the agent file
6	Modify ' <i>SysEvent.Evt</i> ' in C:\WINDOWS\system32\config

When we run an agent file of Netbot, temporary file such as 1955984_res.tmp is generated and the first seven numbers are random. This temporary file is renamed to NetNtEx.dll and moved to the system32 folder of Windows. Furthermore, the beep.sys file is modified, which is a rootkit that modifies the SSDT table so as to obstruct with monitoring using analysis tools. Once the NetNtEx.dll file is generated and the beep.sys file is modified, the original agent file is automatically deleted, and the SysEvent.Evt file is modified to delete event logs for these file changes.

3.4 Registry Changes

Two options are available when generating an agent file using Netbot Attacker. The first option is to change the normal service "BITS (Background Intelligent Transfer Service)" in zombies, and the second option is to generate a new "svchost.exe" and register it to services. The choice of option during the generation of an agent file affects the registry changes and service registrations in infected computers. RegMon and Process Monitor[15], the tools for checking the registry changes, were used to

observe the important registry changes for each option and the results are shown in Table 4 and Table 5.

Table 4. Registry changes by Netbot - BITS

Sequence	Description
1	Modify the value of registry key HKLM\SYSTEM\CurrentControlSet\Services\BITS\Parameters Name: <i>ServiceDll</i> , Data: <i>C:\WINDOWS\system32\NetNtEx.dll</i>
2	Generate the registry key HKLM\SYSTEM\CurrentControlSet\Enum\Root\LEGACY_BITS\0000\Control
3	Modify the value of registry key HKLM\SYSTEM\CurrentControlSet\Enum\Root\LEGACY_BITS\0000\Control Name: <i>ActiveService</i> , Data: <i>BITS</i> HKLM\SYSTEM\CurrentControlSet\Enum\Root\LEGACY_BITS\0000\Control Name: <i>Start</i> , Data: <i>2</i>

In sequence number 1 in Table 4, the “ServiceDll” registry value of the path records the location of the NetNtEx.dll file. The registry keys generated and modified in sequence number 2 and 3 are used to control the functions of Netbot by dynamic services.

Table 5. Registry changes by Netbot - SVCHOST

Sequence	Description
1	Generate the registry key HKLM\SYSTEM\CurrentControlSet\Services\MediaCenter\Parameters
2	Modify the value of registry key HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Svchost Name: <i>krnlsrcv</i> , Data: <i>0x4e006500740042006f0074000000</i> HKLM\SYSTEM\CurrentControlSet\Services\MediaCenter\Parameters Name: <i>ServiceDll</i> , Data: <i>C:\WINDOWS\system32\NetNtEx.dll</i>

The registry changes in Table 5 modify the “krnlsrcv” value of Svchost so that it can be registered to services, and the ServiceDll registry key records the location of the NetNtEx.dll file. Furthermore, to automatically start when Windows is boot, the service is registered.

No.	Time	Source	Destination	Protocol	Info
96	177.790363	166.	192.168.107.128	HTTP	Continuation or non-HTTP traffic
▣ Frame 96 (166 bytes on wire, 166 bytes captured) ▣ Transmission Control Protocol, Src Port: http (80), Dst Port: mxrxlogin (1035), Seq: 33, Ack: 185, Len: 112 ▣ Hypertext Transfer Protocol ▣ Data (112 bytes) Data: 08000000687474703a2f2f7777772e7461726765742e636f...					
0000	00	0c 29 f8 36 75 00 50 56 f8 cc d4 08 00 45 00			...)6u.P V....E.
0010	00	98 17 91 00 00 80 06 bf 42 a6 68 90 fb c0 a8		B.H...
0020	6b 80 00 50 04 0b 47 3e 2e 27 3b 7e c7 89 50 18				k..P..G..:~.P.
0030	fa f0 eb 6f 00 00 08 00 00 00 68 74 74 70 3a 2f				...o... .http:
0040	2f 77 77 77 2e 74 61 72 67 65 74 2e 63 6f 6d 2f				/www.tar get.com/
0050	f3 68 6f 77 2e 61 73 70 2f 69 64 3d 31 22 32 00				show.asp ?id=123
0060	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00			
0070	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00			
0080	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00			
0090	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00			
00a0	00 00 0a 00 00 00			

Fig. 6. Sending attack command to zombie

No.	Time	Source	Destination	Protocol	Info
97	177.792434	192.168.107.128	166.	TCP	ams > http [SYN] Seq=0 win=65535 Len=0 MSS=1460
98	177.792850	192.168.107.128	166.	TCP	mtqp > http [SYN] Seq=0 win=65535 Len=0 MSS=1460
99	177.793290	192.168.107.128	166.	TCP	sbj > http [SYN] Seq=0 win=65535 Len=0 MSS=1460
100	177.793764	192.168.107.128	166.	TCP	netarx > http [SYN] Seq=0 win=65535 Len=0 MSS=1460
101	177.795702	192.168.107.128	166.	TCP	danf-ak2 > http [SYN] Seq=0 win=65535 Len=0 MSS=1460
102	177.795961	192.168.107.128	166.	TCP	afrog > http [SYN] Seq=0 win=65535 Len=0 MSS=1460
103	177.796215	192.168.107.128	166.	TCP	boinc-cltent > http [SYN] Seq=0 win=65535 Len=0 MSS=1460
104	177.796483	192.168.107.128	166.	TCP	dcutlity > http [SYN] Seq=0 win=65535 Len=0 MSS=1460
105	177.796734	192.168.107.128	166.	TCP	fpitp > http [SYN] Seq=0 win=65535 Len=0 MSS=1460
106	177.796968	192.168.107.128	166.	TCP	wfremotertm > http [SYN] Seq=0 win=65535 Len=0 MSS=1460
107	177.797243	192.168.107.128	166.	TCP	neod1 > http [SYN] Seq=0 win=65535 Len=0 MSS=1460
108	177.797861	192.168.107.128	166.	TCP	neod2 > http [SYN] Seq=0 win=65535 Len=0 MSS=1460
109	177.798329	192.168.107.128	166.	TCP	td-postman > http [SYN] Seq=0 win=65535 Len=0 MSS=1460
110	177.798812	192.168.107.128	166.	TCP	cma > http [SYN] Seq=0 win=65535 Len=0 MSS=1460
111	177.799242	192.168.107.128	166.	TCP	optima-vnet > http [SYN] Seq=0 win=65535 Len=0 MSS=1460
112	177.799641	192.168.107.128	166.	TCP	ddt > http [SYN] Seq=0 win=65535 Len=0 MSS=1460
113	177.800240	192.168.107.128	166.	TCP	remote-as > http [SYN] Seq=0 win=65535 Len=0 MSS=1460
114	177.800874	192.168.107.128	166.	TCP	brvread > http [SYN] Seq=0 win=65535 Len=0 MSS=1460
115	177.801274	192.168.107.128	166.	TCP	ansyslmd > http [SYN] Seq=0 win=65535 Len=0 MSS=1460
116	177.801521	192.168.107.128	166.	TCP	vfo > http [SYN] Seq=0 win=65535 Len=0 MSS=1460

Fig. 7. Sending attack packets by zombie

However, as time passed by, the range of port numbers used for attack changed. The attack process is consisted of many cycles and each cycle has 3 subcalls. In one cycle, 20 port numbers are used to send attack packets. The first 20 port number set is from 1037 to 1056, and the next set is from 1057 to 1076, and this pattern was repeated. Table 6 shows this pattern.

Table 6. Pattern of used port number

Cycle	Call	Range of port numbers
1	1	1037~1056
	2	
	3	
2	1	1057~1076
	2	
	3	
3	1	1077~1096
	2	
	3	
:	:	:

4 Design of Detection Framework

In this paper, we proposed the design of a Netbot agent detection framework on the basis of the analysis results from Netbot. Note that features and detection framework of Netbot which is described in this paper should not be propagated to the outside because attackers can use it as a reference to avoid detection system. In the feature extraction phase of Netbot, registry change monitoring, file change monitoring, domain accessing monitoring, and code analyzing were performed through the environment for characteristic analysis. This process was detailed to derive the malicious behaviors and code characteristics that can be used in the Netbot agent detection framework. The derived characteristics were stored in the DB which becomes a tool for comparative analysis with the characteristics of target malicious codes.

After the derived characteristics of Netbot were stored in the DB, and the derived characteristics were combined and the detection process was applied to detect the Netbot agent. In other words, when a file suspected to be an agent of Netbot appear as an input, dynamic behavior-based analysis and static code-based analysis are performed and the results are compared with the DB. As a result, the characteristics of malicious behavior and codes and the classifications of malicious codes are reported. Furthermore, they can be feedback to the analysis process and stored in the DB. Figure 8 illustrates this process, and Figure 9 shows the detection framework that has been materialized on the basis of Netbot Attacker version 5.1.

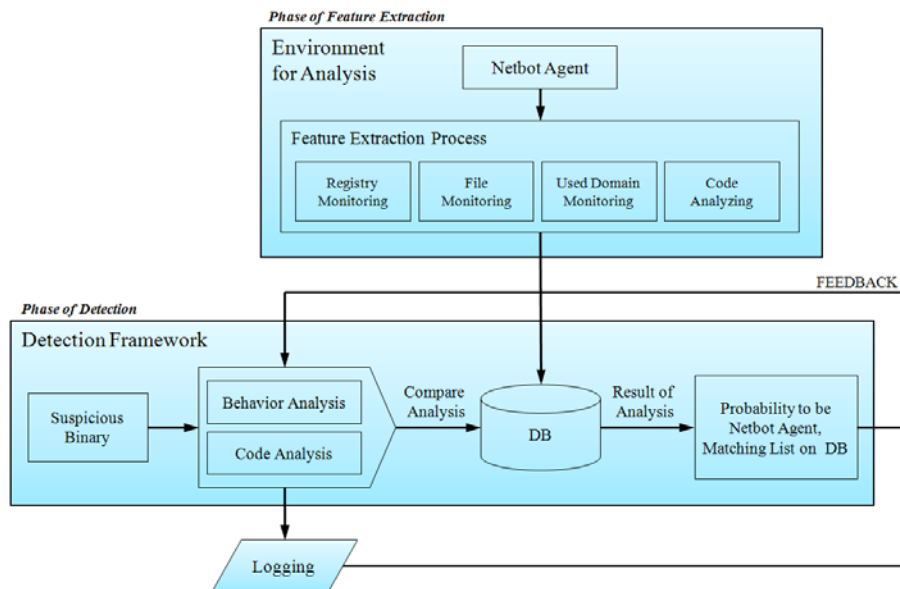


Fig. 8. Conceptual diagram for Netbot agent detection

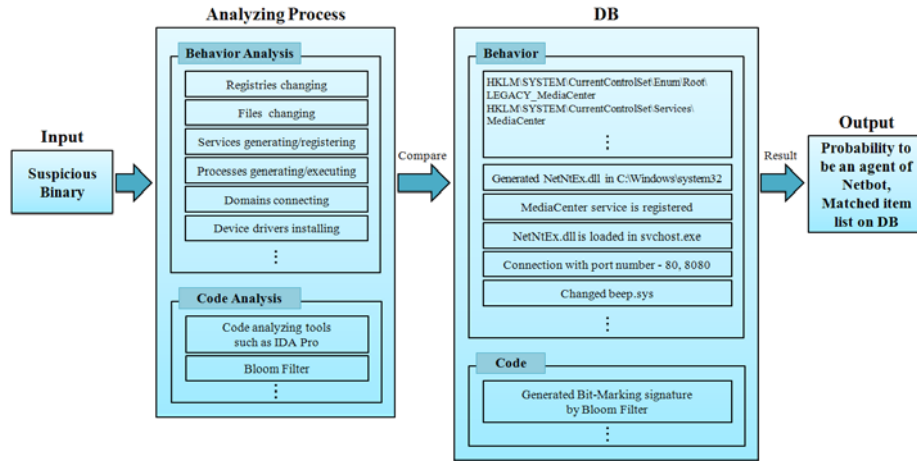


Fig. 9. Detection framework for Netbot agent

In the analyzing process, a dynamic behavior-based analysis and a static code-based analysis are performed for the binary codes suspected to be an agent of Netbot. The factors of behavior-based analysis include registries and files changing services generating and registering, processes generating and executing, domain connecting, and device driver installing (rootkit). The code-based analysis uses the code analysis tools such as IDA Pro[12] and Ollydbg[16], and divides the malicious codes into functional unit modules. Furthermore, as shown in Figure 10, Bit-Marking is performed for the divided functional modules and Basic Block[7] through Bloom Filter[9][10] to derive the similarity of codes.

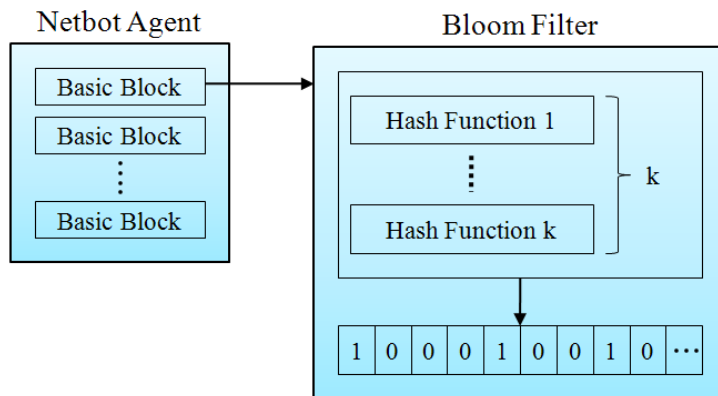


Fig. 10. Bloom Filtering for Basic Block

Codes can be divided into smaller units such as Basic Block[7]. Basic Block is a set of continuous commands that can be divided at the minimum. That is, Basic Block

has one branch point and one entry point. Thus, an attack that performs specific functions can include the same basic block. For example, to execute a DDoS attack, specific data will be copied to one buffer, and new simple loop statements will be used for continuous packet transmission. In addition, the transferred data will be repeatedly sent dozens to hundreds of times with one memory copy, and this can be defined as one characteristic of the DDoS attack function. The space efficiency and comparative speed are improved by removing false negatives using the Bloom Filter[9][10].

Next, the characteristics derived from analysis results are comparatively analyzed against the characteristics of malicious codes stored in the DB. The stored characteristics about behaviors and the stored characteristics about codes are compared. Particularly the characteristics about codes will be compared with the Bit-Marking signature that was generated and recorded through Bloom Filter. Furthermore, the similarity of codes is calculated for such factors as the characteristics of used libraries, APIs, and functions through the code analysis tool.

By putting together the characteristics derived from the results of these analysis and comparisons with the DB, the probability to be an agent of Netbot and the list of items matching with the DB can be reported. For the numerical value of the probability, it must be determined by assigning a different weight value to each matching item.

5 Conclusion

These days, Internet has been developed a lot and dependency on the Internet has also been increased, but as a side effect, cyber-attacks using the tools called botnets are also increasing. Since 2007, the construction of botnets through Netbot and DDoS attacks using them against the servers and web-sites of specific companies have been increasing by attackers. Therefore, in this paper, we constructed an analysis environment on the basis of the operation process of Netbot, and actually generated and ran agents of Netbot Attacker version 5.1 so as to analyze the malicious behaviors of Netbot as well as the file changes, registry changes, and traffic in zombies. Through the results of this analysis, we proposed the design of a framework that can be used to detect the Netbot agents.

In the future, we plan to materialize and implement this framework to detect agents generated by updated versions of Netbot Attacker. To that end, the characteristics of each version of Netbot Attacker must be generalized, various factors must be applied to the analyzing process, and the characteristics must be stored in the DB. Comparative analysis techniques between the DB and the analysis process are another important subject of our research.

Acknowledgement

This work was funded by the Korea Meteorological Administration Research and Development Program under Grant CATER 2009-3213.

References

1. Yong-Hee Jeon, "Introduction and Analysis of Botnet Techniques", Proceedings of the Korea Institutes of Information Security and Cryptology, Vol. 18 No. 3, pp. 101-108, June 2008.
2. Kyoung-Soo Han, Eul-Gyu Im, "A Study on the Traffic Analysis of P2P Botnet in Honeynet Environment", Proceedings of the 12th Conference on Next Generation Communication Software (NCS 2008), pp. 10-13, December 2008.
3. Han-Woo Lee et al., "DNS-based Botnet Detection System", Proceedings of the Korea Information Processing Society Conference, Vol. 13 No. 2, November 2006.
4. David Barroso, "Botnets - The Silent Threat", ENISA Position Paper No. 3, November 2007.
5. L. Garber, "Denial-of-Service Attacks rip the Internet", IEEE Computer pp. 12-17, April 2000.
6. P. Reinecke et al., "Experimental Analysis of the Correlation of HTTP GET invocations", Proceedings of the Formal Methods and Stochastic Models for Performance Evaluation, Third European Performance Engineering Workshop, LNCS Vol. 4054, pp. 226-237, Springer Verlag, 2006.
7. Marius Gheorghescu, "An Automated Virus Classification System", Proceedings of 16th Virus Bulletin International Conference (VB 2005), pp. 294-300, 2005.
8. Ramneek Puri, "Bots & Botnet: An Overview", GSEC Practical Assignment Version 1.4b, SANS Institute, August 2003.
9. Broder A et al., "Network applications of bloom filters: A survey", Proceedings of the 40th Conference on Communication, Control, and Computing, University of Illinois: Urbana-Champaign, IL, 2002.
10. Dharmapurikar, S et al., "Deep Packet Inspection Using Parallel Bloom Filters", Proceedings of 11th Symp. High Performance Interconnects (HOTI'03), pp. 44-51, Stanford, California, 2003.
11. Kyung Chul Choi, "Web, Hacking & Defence", Freerec, 2008.
12. Chris Eagle, "The IDA Pro Book", San Francisco: No Starch Press, 2008.
13. Security-News, <http://www.boannews.com/>
14. S21sec, <http://www.s21sec.com/>
15. Microsoft Technet, <http://technet.microsoft.com/>
16. O. Yuschuk. Ollydbg. <http://www.ollydbg.de/>