International Journal of Hybrid Information Technology
Vol.6, No.6 (2013), pp.291-300
http://dx.doi.org/10.14257/ijhit.2013.6.6.26

Botnet Detection Based on Correlation of Malicious Behaviors

Chunyong Yin[1, 2], Mian Zou[1], Darius Iko[1] and Jin Wang[1]

[1]School of Computer & Software, Nanjing University of Information Science & Technology, Nanjing 210044, China

[2]Jiangsu Key Laboratory of Meteorological Observation and Information Processing, Nanjing University of Information Science & Technology, Nanjing 210044, China

Abstract

Botnet has become the most serious security threats on the current Internet infrastructure. Botnet is a group of compromised computers (Bots) which are remotely controlled by its originator (BotMaster) under a common Command and Control (C&C) infrastructure. Botnets can not only be implemented by using existing well known bot tools, but can also be constructed from scratch and developed in own way, which makes the botnet detection a challenging problem. Because the P2P (peer to peer) botnet is a distributed malicious software network, it is more difficult to detect this bot. In this paper, we proposed a new general Botnet detection correlation algorithm, which is based on the correlation of host behaviors and classification method for network behaviors. The experimental results show the proposed approach not only can successfully detect known botnet with a high detection rate but it can also detect some unknown malware.

Keywords: *botnet, botnet detection, network, behavior, host*

1. Introduction

"Bot" is derived from the word "robot" and is an automated process that interacts with other network services. Bots often automate tasks and provide information or services that would otherwise be conducted by a human being. A typical use of bots is to gather information (such as web crawlers), or interact automatically with instant messaging (IM), Internet Relay Chat (IRC), or other web interfaces. They may also be used to interact dynamically with websites.

Bots can be used for either good or malicious intent. A malicious bot is self-propagating malware designed to infect a host and connect back to a central server or servers that act as a command and control (C&C) center for an entire network of compromised devices, or "botnet." With a botnet, attackers can launch broad-based, "remote-control," flood-type attacks against their target(s). In addition to the worm-like ability to self-propagate, bots can include the ability to log keystrokes, gather passwords, capture and analyze packets, gather financial information, launch DoS attacks, relay spam, and open back doors on the infected host. Bots have all the advantages of worms, but are generally much more versatile in their infection vector, and are often modified within hours of publication of a new exploit. They have been known to exploit back doors opened by worms and viruses, which allows them to access networks that have good perimeter control. Bots rarely announce their presence with high scan rates, which damage network infrastructure; instead they infect networks in a way that escapes immediate notice.

ISSN: 1738-9968 IJHIT

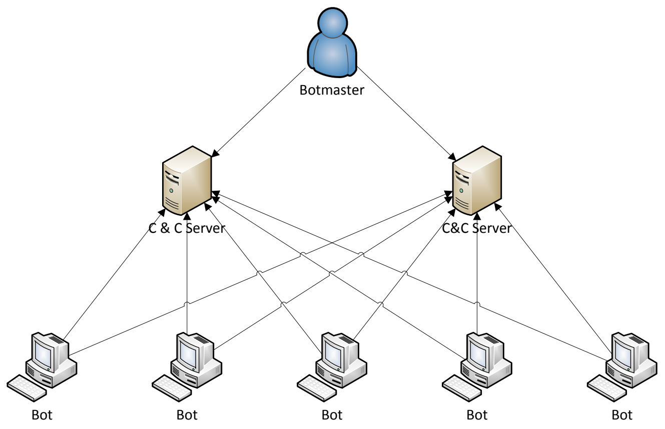


Figure 1. C&C Botnet

Figure 1 illustrates a network of five bots connected to two C&C server controlled by a Botmaster. Nowadays, Botnet is the most serious threat of advanced malware. Different from other Internet malware, Botnet has its own unique characteristic, namely its control communication network. Usually, a “Botnet” consists of a network of compromised computers controlled by a bot-master and has a large scale on the Internet. The disadvantage of C&C server (centralized server) is that it can be easily shut down or blocked by firewall once it has been aware by the victim. Therefore, botmaster design a new mechanism to the botnet system that it does not depend anymore on central server, but it depends on any computer of the system or P2P. Each computer can be act as client or server to any other computer in P2P network. P2P network is showed in the Figure 2.

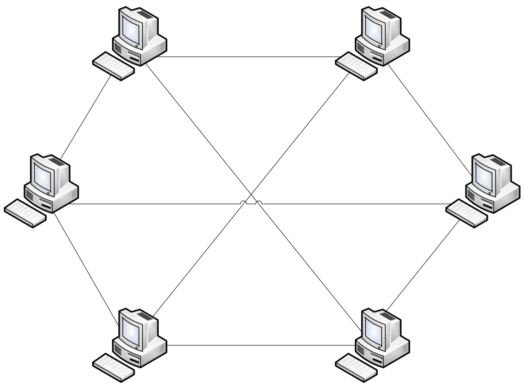


Figure 2. A P2P Network

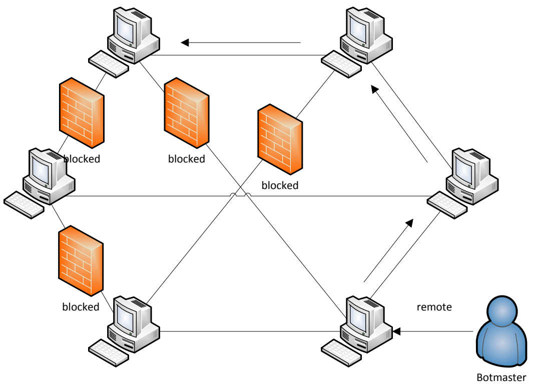


Figure 3. A command from Botmaster was sent through many computers

Detection on P2P Botnet is difficult as it has no central point (C&C server). Any host that connected to P2P Network can act as C&C server. Once the botmaster get a list of host connected to P2P network, he can control every host as he wish.

Although some computers are blocked by the firewall, but once a bot get connected to at least one bot in another computer, it can receive any command indirectly from the botmaster through another computer. This scenario is illustrated in the Figure 3.

2. System Vulnerability

System Vulnerability can turn even a highly-secure computer into security-compromise computer. No wonder most kinds of malware are likely to misuse this weakness to spread malware and to attack target.

System vulnerability is caused by the following points:

1. Complexity: Large, complex systems increase the probability of flaws and unintended access points.
2. Familiarity: Using common, well-known code, software, operating systems, and/or hardware increases the probability an attacker has or can find the knowledge and tools to exploit the flaw.
3. Connectivity: More physical connections, privileges, ports, protocols, and services and time each of those are accessible increase vulnerability.
4. Password management flaws: The computer user uses weak passwords that could be discovered by brute force. The computer user stores the password on the computer where a program can access it. Users re-use passwords between many programs and websites.
5. Fundamental operating system design flaws: The operating system designer chooses to enforce suboptimal policies on user/program management. For example operating systems with policies such as default permit grant every program and every user full access to the entire computer. This operating system flaw allows viruses and malware to execute commands on behalf of the administrator.
6. Internet Website Browsing: Some internet websites may contain harmful Spyware or Adware that can be installed automatically on the computer systems. After visiting those websites, the computer systems become infected and personal information will be collected and passed on to third party individuals.
7. Software bugs: The programmer leaves an exploitable bug in a software program. The software bug may allow an attacker to misuse an application.
8. Unchecked user input: The program assumes that all user input is safe. Programs that do not check user input can allow unintended direct execution of commands or SQL statements (known as Buffer overflows, SQL injection or other non-validated inputs).
9. Not learning from past mistakes: for example most vulnerabilities discovered in IPv4 protocol software were discovered in the new IPv6 implementations.

3. Bot Propagation

Bot infection usually occurs through trapped files, e-mail attachments or web pages. However, trend in bot technology includes a blending of Trojan-horse, virus, worm, and backdoor functionality.

Hacker uses spams to spread campaign message to dupe users. The messages are accompanied by a .zip file attachment. This archive file usually contains an executable with an icon resembles other trusted applications icon in order to trick the recipient. If the recipient opened the archive file and then launched the resulting executable, the machine will be hijacked by malware and added to the growing botnet.

Some kinds of bot usually infect other computers through the vulnerability of the services, just like the behavior of other forms of Malware. For example: Stack-based buffer overflow in certain Active Directory service functions of the Local Security Authority Subsystem Service (LSASS) in certain versions of Microsoft Windows allowing remote attackers to execute arbitrary code, can be helpful for a bot to transfer its code and execute it on another host. This scenario is illustrated in Figure 4.

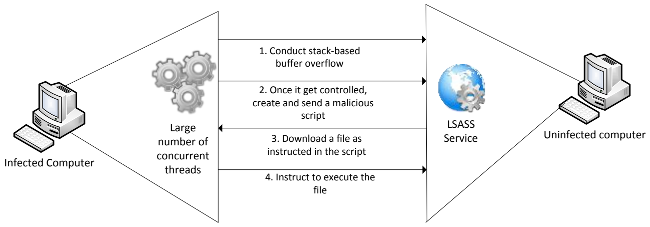

Infected Computer
Large number of concurrent threads
1. Conduct stack-based buffer overflow
2. Once it get controlled, create and send a malicious script
3. Download a file as instructed in the script
4. Instruct to execute the file
LSASS Service
Uninfected computer

Figure 4. Large number of threads on an infected computer spreading malware

Some kinds of bot also conduct a remote code execution by specially crafting a malicious RPC request in the Remote Procedure Call (RPC) service in computers running certain Windows Platform. Remote Procedure Call (RPC) is a protocol that a program can use to request a service from another program which is located on another computer in a network. More specifically, an infected source computer uses this malicious request to force a buffer overflow and execute shellcode on the target computer. On the source computer, the bot runs an HTTP server on a specific port, and the target shellcode connects back to this HTTP server to download a copy of itself, which it then installs to the system.

At some special cases, another kind of malware such as Worm can open a backdoor port to bypass normal authentication and allows remote access to the computer. Some bot also used this special backdoor port to propagate itself. Other methods are using the weak password-protected administrative shares (such as C$, ADMIN$, PRINT$) in Windows, WebDAV bug in IIS (Internet Information Service), Workstation service buffer overrun vulnerability, and so on.

4. Related Work

According to the different protocols, botnets can be classified into IRC-based, HTTP-based, P2P-based, and other-based. Meanwhile, botnets can also be divided into central mode and distribute mode based on their topologies. As for centralized architecture, researchers have designed some methods to detect and destroy such botnets [1, 2]. Compared to centralized botnets, distributed botnets are more difficult to be detected and taken down [3], because they do not have a central point. Besides, most of its existing studies are still in the analysis phase [3, 4]. This paper focus on P2P bot detection based on distributed architecture.

By now, current botnet detection research is mainly based on network behaviors [5, 6, 7, 8, 9, 10, 11, 12]. Few people have studied how to detect botnets based on host behaviors [13].

This paper proposes a botnet detection approach based on correlation between host behaviors and classification for network behaviors. Compared to existing botnet detection methods only based on single network behaviors or host behaviors, the proposed approach has a higher detection rate.

5. Host-based Detection

We know that interaction between the application and operating system is through the programming interface (API). Windows Operating System provides a lot API functions to let the application invoke. By invoking these interfaces, the application can request a service from the operating system, pass parameters, and receive the results of the operation. As the API calling are freely used by each running process, Operating System does not guarantee the security of the invocation by running process, especially the malware process.

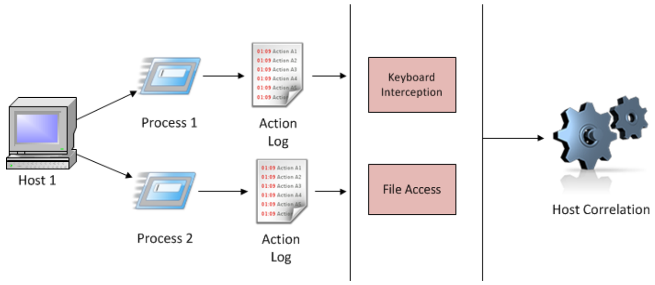


Figure 5. Large number of threads on an infected computer spreading malware

In order to deal with host behaviors of a bot program, we developed combined algorithm which correlates keyboard interception and file access of running processes, which is illustrated in Figure 5. We define a function called GetAsyncKeyState() as our keyboard interception function, and a function called WriteFile() as our file access function. Each process has multiple times of calling a function. We will group them into time groups.

Our algorithm consists of two steps: correlation step and decision step. The correlation step will employ Spearman's Rank Correlation (SRC) which is explained as follow:

Formula: Spearman's Rank Correlation

- Dataset 1 contains the number of function calls (GetAsyncKeyState) grouped by time group.
- Dataset 2 contains the number of function calls (WriteFile) grouped by time group.
- Rank the two data datasets by using the following rules:
  - The smallest number is ranked 1, the second smallest number is ranked 2 and so on.
  - Identical values (rank ties or value duplicates) are assigned a rank equal to the average of their positions in the ascending order of the values.
- Find the value difference in the ranks, named as d
- Square the difference, named as d2
- Calculate the coefficient Rs using the following equation:

$$Rs = 1 - \frac{6 \times \sum_{i=1}^{n} d_i^2}{n(n^2 - 1)}$$

And the decision step consists of algorithm which is shown as follow:

```
S1: Keyboard interception function call;
S2: File Access function call;
if Keyboard interception function is called (i.e. keylogging activity) then
   if SRC(S1,S2) >= HighThreshold then
      Strong detection;
   if SRC(S1,S2) > LowThreshold and SRC(S1,S2) < HighThreshold  then
      Normal detection;
   else
      Normal activity is considered;
end
```

6. Experiment for Host-based Detection

In the implementation step, we define the time groups be 15 second/group, the low threshold be 0.333 and the high threshold be 0.666. We will divide the implementation step into three experiments. In the first experiment, we will target RBot process as an example of malicious process. In the second and the third experiment, we will target mIRC client process and Internet Explorer process as two examples of benign process.

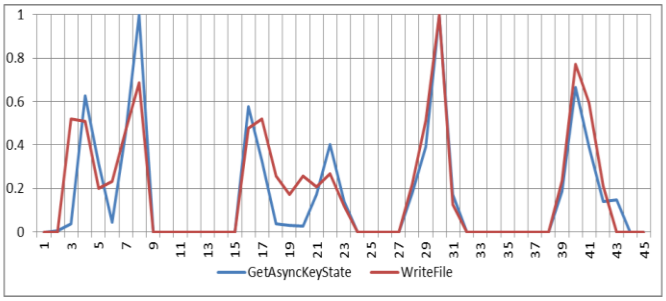


Figure 6. Correlation distribution for experiment 1

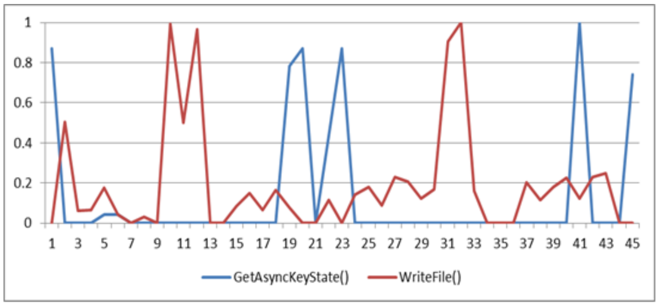


Figure 7. Correlation distribution for experiment 2

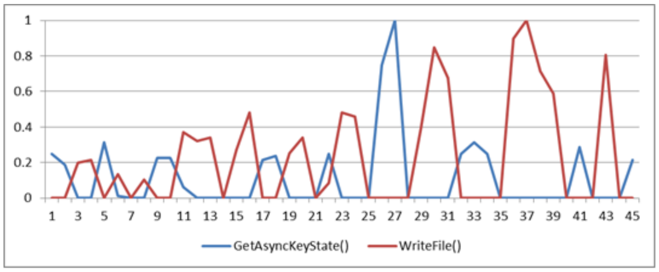


Figure 8. Correlation distribution for experiment 3

Figure 6 shows that in the first experiment, RBot process calls file access function at almost the same time as the keyboard interception function. Moreover, the SRC coefficient is 0.9260210803689064 (more than highthreshold (0.666) means highly suspicious).

Figure 7 shows that in the second experiment, mIRC process calls file access function at the different time as the keyboard interception function. Moreover, the SRC coefficient is 0.07384716732542818 (less than lowthreshold (0.333) then this activity is considered normal activity).

Figure 8 shows that in the second experiment, Internet Explorer process calls file access function at the different time as the keyboard interception function. Moreover, the SRC coefficient is -0.23577075098814237 (less than lowthreshold (0.333) then this activity is considered normal activity).

7. Network-based Detection

In this work, we will use ISOT Botnet Dataset 2010 as our target dataset, I extract 28 e-mail documents from the capture data set provided. 18 e-mail documents are real spam and 10 e-mail documents are ham (non-spam). The attributes for training set and testing set are defined as follow (each of them representing the frequency of word exists in an email):

Table 1. The attributes for training set and testing set

word_freq_ermx	word_freq_gladiator	word_freq_entremetrix
word_freq_0_089	word_freq_11_25	word_freq_martial
word_freq_arts	word_freq_biggest	word_freq_promoter
word_freq_dividend	word_freq_opportunities	word_freq_asset
word_freq_expansion	word_freq_tuesday	word_freq_hp
word_freq_hpl	word_freq_george	word_freq_lab
word_freq_labs	word_freq_telnet	word_freq_data
word_freq_technology	word_freq_parts	word_freq_pm
word_freq_direct	word_freq_cs	word_freq_meeting
word_freq_original	word_freq_project	word_freq_re
word_freq_edu	word_freq_table	word_freq_conference
word_freq_discount	word_freq_voucher	word_freq_coupon
word_freq_xxx	word_freq_save	word_freq_money
word_freq_sales		

First, we extracted each e-mail document to text file and each file attributes was extracted automatically by using program (codes are listing in appendix e-mail attributes extraction). Then, I collected all the extracted information and join them into a file that is compatible with Weka. This file would be used to make a classifier model by using Naïve Bayes classifier.

After the NumericToBinary filter is applied to the input, all the numeric frequency attributes are now converted to Booleans. Each e-mail is now represented by a 33 dimensional vector representing whether or not a particular word exists in an e-mail. We use ten-fold cross-validation which makes our results less prone to random variation.

The results shows that the accuracy is 89.3% and according to the confusion matrix, there are 3 instances with false positive/true negative, there are 15 instances classified as spam and 10 instances classified as non-spam. Then we used the test set that only contains 8 instances, and the results shows that the accuracy is 88.9% and according to the confusion matrix, there is only 1 instance with false positive/true negative, there are 5 instances classified as spam and 3 instances classified as non-spam.

Acknowledgements

This work was funded by Jiangsu Key Laboratory of Meteorological Observation and Information Processing (No.KDX1105) to Dr. Chunyong Yin and by the National Natural Science Foundation of China (No.61373134). It was also supported by the Industrial Strategic Technology Development Program (10041740) funded by the Ministry of Trade, Industry and Energy (MOTIE Korea), and by the Natural Science Foundation of Jiangsu Province (No. BK2012461). The authors wish to thank the reviewers for their valuable comments and suggestions which helped in improving the quality of the paper.

References

[1] C. Mazzariello, “Irc traffic analysis for botnet detection”, Information Assurance and Security 2008, ISIAS’08, Fourth International Conference on, Ieee, **(2008)**.

[2] B. McCarty, “Botnets: Big and bigger”, Security & Privacy, IEEE, vol. 1, no. 4, **(2003)**.

[3] M. Masud, J. Gao, L. Khan, J. Han and B. Thuraisingham, “Mining conceptdrifting data stream to detect peer to peer botnet traffic”, Univ. of Texas at Dallas Tech. Report# UTDCS-05-08, http://www. utdallas.edu/mmm058000/reports/UTDCS-05-08. pdf.

[4] G. Schaffer**,** “Worms and viruses and botnets, oh my! rational responses to emerging internet threats”, Security & Privacy, IEEE, vol. 4, no. 3, **(2006)**.

[5] J. Binkley and S. Singh, “An algorithm for anomaly-based botnet detection”, Proceedings of the 2nd conference on Steps to Reducing Unwanted Traffic on the Internet, **(2006)**.

[6] H. Choi, H. Lee, H. Lee and H. Kim, “Botnet detection by monitoring group activities in dns traffic”, Computer and Information Technology 2007, CIT 2007, 7th IEEE International Conference on, Ieee, **(2007)**.

[7] Heron, “Working the botnet: how dynamic dns is revitalising the zombie army”, Network Security, vol. 2007, no. 1, **(2007)**.

[8] W. Lu, G. Rammidi and A. Ghorbani, “Clustering botnet communication traffic based on n-gram feature selection”, Computer Communications, vol. 34, no. 3, **(2011)**.

[9] W. Lu, M. Tavallaee, G. Rammidi and A. Ghorbani, “Botcop: An online botnet traffic classifier”, 2009 Seventh Annual Communication Networks and Services Research Conference, IEEE, **(2009)**.

[10] L. Song, Z. Jin and G. Sun, “Modeling and analyzing of botnet interactions”, Physica A: Statistical Mechanics and its Applications, vol. 390, no. 2, **(2011)**.

[11] H. Tu, Z. Li and B. Liu, “Detecting botnets by analyzing dns traffic”, Intelligence and Security Informatics, **(2007)**.

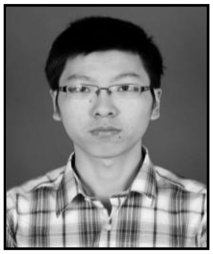
[12] H. Zeidanloo, F. Hosseinpour and P. Borazjani, “Botnet detection based on common network behaviors by utilizing artificial immune system (ais)”, Software Technology and Engineering (ICSTE), 2010 2nd International Conference on, vol. 1, IEEE, **(2010)**.

[13] Y. Al-Hammadi, U. Aickelin and J. Greensmith, “Dca for bot detection”, Evolutionary Computation 2008, CEC 2008, (IEEE World Congress on Computational Intelligence), IEEE Congress on, IEEE, **(2008)**.

Authors

Chunyong Yin

He is currently an associate Professor and Dean with the Nanjing University of Information Science & Technology, China. He received his Bachelor (SDUT, China, 1998), Master (GZU, China, 2005), PhD (GZU, 2008) and was Post-doctoral associate (University of New Brunswick, 2010).He has authored or coauthored more than twenty journal and conference papers. His current research interests include privacy preserving and network security.



Mian Zou

He received bachelor degree in computer science from Nanjing University of Information Science & Technology, Jiangsu, China in 2012 and now studying for his Master Degree in computer science and application in Nanjing University of Information Science & Technology. His research interesting includes computer networks, network security, and data encryption.

Darius Iko

He received bachelor degree in computer science from Bina Nusantara University, Jakarta, Indonesia in 2009 and now studying for his Master Degree in computer science and application in Nanjing University of Information Science & Technology. His research interesting includes computer networks, network security, and data encryption.

Jin Wang

He received the B.S. and M.S. degree from Nanjing University of Posts and Telecommunications, China in 2002 and 2005, respectively. He received Ph.D. degree from Kyung Hee University Korea in 2010. Now, he is a professor in the Computer and Software Institute, Nanjing University of Information Science and Technology. He has published more than 120 journal and conference papers. His research interests mainly include routing protocol and algorithm design, performance evaluation and optimization for wireless ad hoc and sensor networks. He is a member of the IEEE and ACM.