Unstructured Peer-to-Peer Botnet Simulation for Measuring Its Robustness

Sayed Muchallil[1], Subhasish Dutta Chowdhuri[2], Kiran Venkatesh[2], Abhiram Doddaballapur Venkatraman[2] and Candeep Singh[2]

[1]Jurusan Teknik Elektro, Fakultas Teknik Universitas Syiah Kuala,
[2]Department of Computer Science and Engineering
The University of Texas at Arlington
Email: sayed.muchallil@elektro.unsyiah.ac.id

***Abstract*— Malware attacks on the Internet have increased substantially in recent years for which botnets are a root cause. A "botnet" is a network of compromised computers controlled by an attacker known as the "botmaster". To be able to effectively detect and defend against botnets, it is very important to have a good understanding of their construction procedure and propagation methodology. In this work, we study the construction of an unstructured peer-to-peer botnet, its propagation methodology, diurnal properties and robustness. This simulation shows that the more frequently a node updates its buddy list, the lesser is the process overhead involved.**

Kata Kunci. ***botnet, peer-to-peer.***

I. Introduction

In the last decade, malware attacks on the Internet have increased substantially. Attacks such as phishing, e-mail spamming, keylogging, click fraud and Distributed Denial of Service (DDoS) are common on the Internet today for which botnets are the root cause [1-3]. A botnet is a network of compromised computers called bots controlled by a remote attacker called botmaster.

Most botnets have a Centralized Command and Control (C&C) architecture in which the bots directly connect to servers called C&C servers. These C&C servers receive commands from the botmaster and forward them to the bots in the network. However, these C&C servers are the primary weak points in the botnet architectures. The botmaster will lose control over the botnet if the C&C servers are shut down by the defenders.

Considering the weakness in the architecture of centralized botnets, some attackers use a peer-to-peer architecture for their botnets. This is done in order to avoid centralized C&C and make it difficult for the defenders to detect and shut down the botnet. A buddy list is a list of hosts that a host wants to keep track of. Each host then connects to other hosts that are in its buddy list. The attacker only has to send his commands to a few of these hosts and each host will propagate these commands to their buddies. Thus, by becoming one of the peers, the attacker can broadcast his commands over the entire network.

The goal of this work is to understand the creation of the botnet architecture, study the diurnal dynamics and robustness of the botnet as well as the process overhead involved in the generation and updating of buddy lists. We have also studied the propagation time for the attack commands to spread to all the bots in the botnet. Our work would help defenders develop effective detection and response systems for botnets.

We simulated an unstructured peer-to-peer botnet to study the aforesaid features with the aim of better understanding of how botnets work. Our work is organized as follows: Section 2 describes the related work, Section 3 presents the system model that we use in this simulation, Section 4 shows our experimental design, and Section 5 presents the results of our simulation whereas Section 6 and 7 discusses the conclusion and the future work.

II. Related Works

In recent years, botnets have become an active research topic. This section reviews the related works about botnets that we use as our reference in creating the simulation for this project.

Puri provided an overview of botnets in [4], which introduced botnets, some related terms as well as attacks that the botnet could perform.

In 2005, The Honeypot Project presented more details about botnet commands such as DDoS, bot spreading and downloading files from the Internet [5] using a botnet.

Wang et al. [6] proposed an advanced hybrid peer-to-peer botnet which, compared to current botnets is harder to shut down, monitored or hijacked. Vogt et al. [7] developed a simulation to show the attack effectiveness of many small botnets compared to one large botnet.

Steggink and Idziejczak [8] discussed three topologies used by botnets: centralized, decentralized and hybrid. They also analyzed the bot infection mechanisms and their behaviors in the test environment. Dagon et al. [9] discussed the diurnal properties in botnet activity using which they compared botnet propagation rates so as to prioritize responses

Chu et al. [10] discussed the population dynamics in peer-to-peer networks which showed that the availability of nodes was strongly influenced by the time of the day and most users tended to be available for short contiguous periods of time. Bhagwan et al. [11] provided an overview of the availability of peer-to-peer systems. Lua et al. [12] compared various structured and unstructured peer-to-peer networks.

Our work studies the construction, robustness and the diurnal dynamics of botnets as well as the underlying process overhead involved which would enable defenders to develop effective detection and response systems.

III. SYSTEM MODEL

In this section, we present the system model used in our simulation. We simulate an unstructured peer-to-peer network such as the one shown in Figure 1 in which the buddy list of each peer is constructed by a random process. We limit the neighbors of each peer to at most ten.

Our bot model is based on Phatbot [13] which uses a peer-to-peer network structure based on Gnutella to receive commands and send information. Since there is no server, the infected hosts have to find each other individually. This is accomplished by utilizing Gnutella cache servers – anyone can use the CGI scripts provided by these servers to register themselves as a Gnutella client [13]. However, in our simulation each bot will have a buddy list of its neighbors. The buddy list will be created using a random process to choose up to ten neighbors for each infected host.

In our simulation, after the initial botnet construction, an infected host will check its neighbors every 20, 15, 10 and 5 minutes to see whether they are alive or not. If five or fewer neighbors are alive, the infected host will replace its non-active neighbors with other hosts that are not listed in the buddy list and have been infected.

Each host in the peer-to-peer network will frequently leave and join the network. According to Chu et al. [10], the online population of a botnet is strongly affected by the time of the day. A small percentage of nodes are available for downloads at any instant: 31% of the time nodes were available for only about a 10-minute period before becoming unavailable again. The result of this experiment which we used for our simulation is shown in Figure 2.

IV. EXPERIMENTAL DESIGN

In order to study the botnet topology and its robustness through simulations, we first need to determine the simulation settings.

In our simulation, we assume that the botnet has a potential vulnerable population of 15,000 but stops growing after it reaches the size of 10,000. The botnet topology is created the first time the simulation runs. Subsequently, each infected host will check its neighbors every 20, 15, 10 and 5 minutes of simulation time respectively to determine whether they are alive or not. If five or fewer neighbors are alive, the infected host will replace its non-active neighbors with other infected hosts that are not in its buddy list.

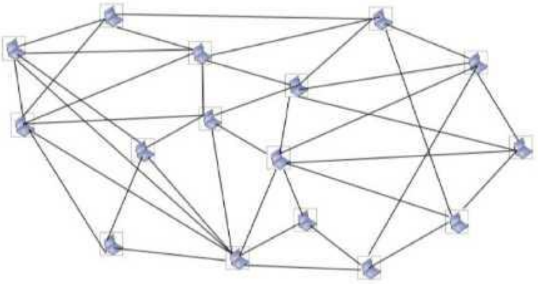

Figure1. Example of an unstructured peer- to-peer network

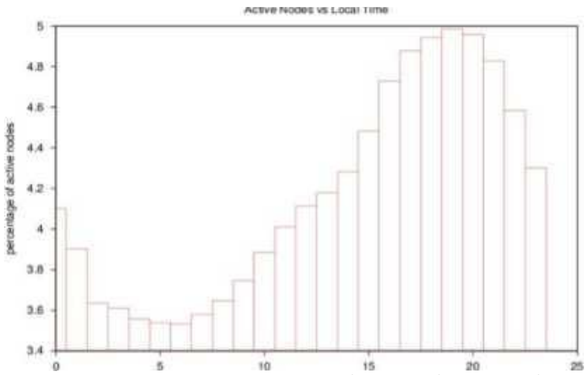

Figure 2. Node availability as a function of the hour of the day (Source: [10])

We did two types of simulation for our research. In the first experiment called Dynamic Population Simulation, we observed the botnet population behavior and the process overhead involved in the creation and updating of buddy lists. We ran this simulation ten times for 20, 15, 10 and 5 minutes of simulation time and took the median value of the results to arrive at the conclusions. In the second experiment called Attack Command Simulation we observed the time taken for the propagation of attack commands to all the live bots in the botnet.

A. Dynamic Population Simulation

In this experiment, we observe the dynamic process of the construction of the botnet topology and the process of addition of new neighbors by an infected host. In our simulation there are 10,000 infected hosts and 5,000 uninfected hosts. For the purpose of our simulation, we do not cover the primary infection procedure and assume that 10,000 hosts are already infected.

As soon as the simulation is run the topology is automatically constructed after which the infected hosts will check their neighbors every 20, 15, 10 and 5 minutes. If five or fewer neighbors are alive, it will try to add other infected hosts as its neighbors. For the purpose of our simulation, we limit each peer to have at most 10 neighbors. The diurnal dynamics of the botnet are simulated based on data presented in Chu et al. [10] which states that nodes are available only for about 10 minute periods before beco ming unavailable again. Moreover, at any instant a maximum of only 5% nodes are available.

This experiment also measures the process overhead involved during the construction of the botnet topology and updating of the buddy list by every node.

B. Attack Command Simulation

In this we observe the time required for the propagation of commands to all the live bots in the botnet. One of the current live, infected bot will be randomly chosen to initiate the command propagation process. The commands will be propagated to all the live and infected bots through the buddy list of each node.

V. RESULT

In this section, we present the results of our simulation.

Figure 3 depicts the diurnal dynamics of the botnet over a 24 hour time period. The results confirm the

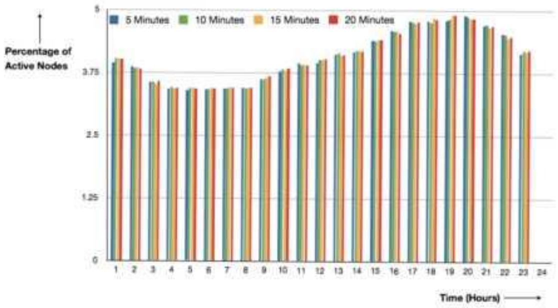

Figure 3. Percentage of Live Botnet Population over a 24 hour period

statistics presented in [10] which show that the diurnal properties of the botnet are strongly influenced by the time of the day and a maximum of only 5% nodes are active at any instant. It can be inferred from the graph that the maximum botnet population is during the second half of the day.

Since every node in a peer-to-peer botnet has to frequently update its buddy list, the process overhead involved is of prime concern. Figure 4 depicts the process overhead involved when each node in the botnet updates its buddy list every 20, 15, 10 and 5 minutes.

The results show that the more frequently a node updates its buddy list, the lesser is the process overhead involved. Our simulation results show that the process overhead involved is the minimum when every node in the botnet updates its buddy list every 5 minutes.

Figure 5 depicts the time needed for the propagation of commands to all the live bots in the botnet.

VI. Conclusion

In this work, we study the construction of an unstructured peer-to-peer botnet topology and its robustness with the help of simulations. From our work, we can conclude that the process overhead in a peer-to-peer botnet will be significantly lower if the bots in the botnet update their buddy lists frequently. We also observe that the diurnal dynamics of the botnet are strongly influenced by the time of the day. Therefore it can be inferred that the botnet is in its most vulnerable state when it is initially constructed and before its buddy list updating procedure is run for the first time.

VII. Future Work

Future work should study the topology building procedure and robustness of peer-to-peer botnets by increasing the total vulnerable population of the botnet, infected population of the botnet and the number of buddies of each host. The resiliency of botnets should also be studied by classifying the bots in a peer-to-peer botnet into servant bots and client bots [6] and then performing the buddy list updating procedure.

VIII. Acknowledgment

The authors would like to thank Dr. Matthew Wright for his invaluable cooperation and guidance.

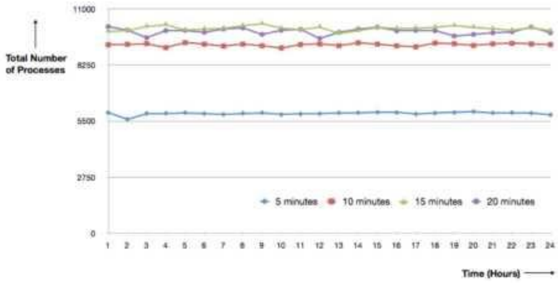

Figure 4: Graph depicting the process overhead while building the botnet topology

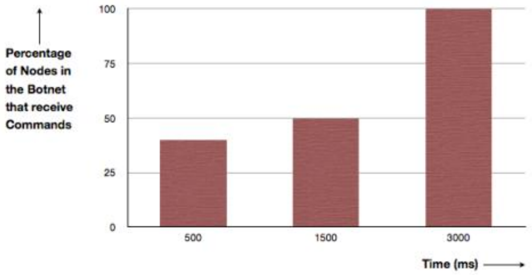

Figure 5: Percentage of Nodes in the Botnet that receive Commands

References

[1] F. Freiling, T. Holz, and G. Wicherski, Botnet tracking: "Exploring a root-cause methodology to prevent distributed denial of service attacks," Technical Report AIB-2005-07, CS Dept. RWTH Aachen Univ., Apr. 2005.

[2] [2] D. Dagon, C. Zou, and W. Lee, "Modeling botnet propagation using time zones,"Proc. 13th Ann. Network and Distributed System Security Symp. (NDSS '06), pp. 235-249, Feb. 2006.

[3] A. Ramachandran, N. Feamster, and D. Dagon, "Revealing botnet membership using DNSBL counter intelligence," Proc. USENIX Second Workshop Steps to Reducing Unwanted Traffic on the Internet (SRUTI '06), June 2006.

[4] R. Puri, "Bots & botnet: An overview," http://www.sans.org/rr/whitepapers/malicious/1299.php, 2003

[5] "The Honeynet Project, Know your enemy : GenII honeynets," http://www.honeynet.org/papers / bots, 2005.

[6] P. Wang, S. Sparks, and C. Zou, "An advanced hybrid peer-to-peer botnet,Dependable and Secure Computing," IEEE Transactions on, vol.7, no.2, April-June 2010, pp.113-127.

[7] R. Vogt, J. Aycock, and M. Jacobson, "Army of botnets," in Proceedings of 14th Annual Network and Distributed System Security Symposium (NDSS), February 2007, pp. 111-123.

[8] M. Steggink and I. Idziejczak, "Detection of peer-to-peer botnets," Research Project 1 Master of Science Program, University of Amsterdam, 2008.

[9] D. Dagon, C. Zou, and W. Lee, "Modeling botnet propagation using time zones," Proc. 13th Ann. Network and Distributed System Security Symp. (NDSS '06), pp. 235-249, Feb. 2006.

[10] J. Chu, K. Labonte, and B. Levine, "Availability and locality measurements of peer-to-peer file sharing systems," in Proceedings of SPIE ITCom: Scalability and Traffic Control in IP Networks, vol. 4868, July 2002.

[11] R. Bhagwan, S. Savage, and G.M. Voelker, "Understanding Availability," Proc. Second Int'l Workshop Peer-to-Peer Systems (IPTPS '03), Feb. 2003.

[12] E.K. Lua, J. Crowcroft, M. Pias, R.Sharma, and S. Lim, "A survey and comparison of peer-to-peer overlay network schemes," IEEE Comm. Surveys and Tutorials, vol. 7, no. 2, 2005.

[13] "Phatbot trojan analysis", http://www.lurhq.com/phatbot.html, 2008.

[14] C. Zou and R. Cunningham, "Honeypot-Aware advanced botnet construction and maintenance," Proc. Int'l Conf. Dependable Systems and Networks (DSN '06), June 2006.