Detection of Botnets Using Honeypots and P2P Botnets

Rajab Challoo kfrc000@tamuk.edu
Dept. of Electrical Engineering & Computer Science
Texas A&M University Kingsville
Kingsville, 78363-8202, USA

Raghavendra Kotapalli raghavsan@gmail.com
Dept. of Electrical Engineering & Computer Science
Texas A&M University Kingsville
Kingsville, 78363-8202, USA

Abstract

A “botnet” is a group of compromised computers connected to a network, which can be used for both recognition and illicit financial gain, and it is controlled by an attacker (bot-herder). One of the counter measures proposed in recent developments is the “Honeypot”. The attacker who would be aware of the Honeypot, would take adequate steps to maintain the botnet and hence attack the Honeypot (Infected Honeypot). In this paper we propose a method to remove the infected Honeypot by constructing a peer-to-peer structured botnet which would detect the uninfected Honeypot and use it to detect botnets originally used by the attacker. Our simulation results show that our method is very effective and can detect the botnets that are intended to malign the network.

Keywords: Peer-to-peer network, Botnet, Honeypot, Hijacking.

1. INTRODUCTION

The Increase in the Internet malware in the recent attacks have attracted considerable amount of attraction towards botnets. Some of them include Email spamming, Key logging, click fraud and traffic sniffing [1]. Recently detected dangerous botnets include Mariposa (2008), officla (2009) and TDSS (2010). The scatter attacks done by the bot controllers using a program called bot which communicates with other botnets and receive the commands from Command and Control servers [3].

As the traditional botnets, which are designed to operate from a central source (bot-attackers machine) which can be shutdown if the source is pin-pointed by the security agencies, bot masters use or resort to peer to peer (P2P) botnets which do not have a centralized source and can grow at an alarming speed. For example, botnet Oficla can spam up to 3.6 billion targets per day [4].

In this paper we show how the use of a combination of Honeypots and Peer to Peer botnet to defend the attacks from other botnets. In order to improve the efficacy in defending against such malicious attacks, one needs to analyze the botnets from a bot-attackers perspective. This would require a study of basic structure of botnet and the network. The antivirus approach, of signature based detection of removing one bot or virus at a time works at host level but when bot-attackers use polymorphic methods creating new instances using the botcodes, evasion from antivirus becomes complicated. Security experts monitor Command and Control (C&C) traffic so as to detect an entire network which is infected, this is done to extenuate the botnet problem on a large scale by identifying the C&C channel[5]. Once a C&C channel is identified by the defenders, entire botnet could be captured by the defenders[3]. After botnet is captured, botmasters move to an advanced technique.

2. BACKGROUND

To mitigate the botnet problem, the command and control mechanism has been under study which determines the structure of C&C botnets that can monitor, hijack and shutdown the network. Defenders can however shutdown the entire C&C channel and prevent the attack [5]. In P2P botnets there is no central point for controlling the botnets. The servant bots act as client and servers [6], and accept both incoming and outgoing connections whereas the client bots do not accept incoming connections. Servant bots alone are added to the peer-lists. All bots including both client and server bots contact the servant bots to retrieve the commands [4].

2.1 Types of Botnets

Bots are basically classified into three types based on botnet topologies:
centralized, peer to peer (P2P) and random .

As described earlier, centralized bot has a point of control which shuts down the entire botnet if affected. In random botnet, one bot knows no more than the other [7]. This type of botnet has no guarantee of delivering what is required, hence the topology of random botnet is not discussed in this paper.

Peer-to-peer botnet has no central point of control and can be used by botmasters (if not in use already). Let us consider the P2P botnet constraints from a botmasters perspective [3].

1) Generate a botnet that is capable of controlling remaining bots in the network, even after considerable portion of botnet population has been removed by defenders.

2) Prevent significant exposure of the network topology when some bots are captured by defenders.

3) Monitor and obtain the complete information of a botnet by its botmaster with ease.

4) Prevent (or make it harder for) defenders from detecting bots via their communication traffic patterns.

2.2 Monitoring Botnets

Currently, there are two techniques to monitor botnets

(1) Allow Honeypots to be compromised by the botnet [2, 8], behave as normal 'bot" in the botnet, and these Honeypot spies provide all the required information to monitor botnet activities [2].

(2) To hijack the bot controllers to monitor Command and Control communications in Botnets.

The architecture of a P2P botnet is shown in Figure 1 [2].

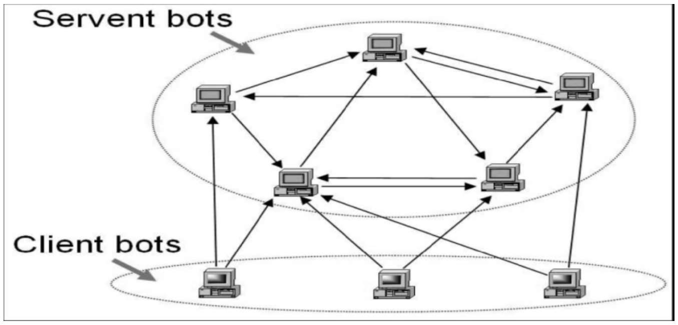

FIGURE 1: C&C Architecture of a P2P Botnet

3. PROPOSED APPROACH

Command and Control botnets, P2P botnets and Honeypots (Honeynets) are used in our approach. Consider bots A, B, C, D and E that are introduced into the network as shown in Figure 2. Honeypots are denoted as H, IH denotes the Infected Honeypot from botnets either C&C or P2P or both.

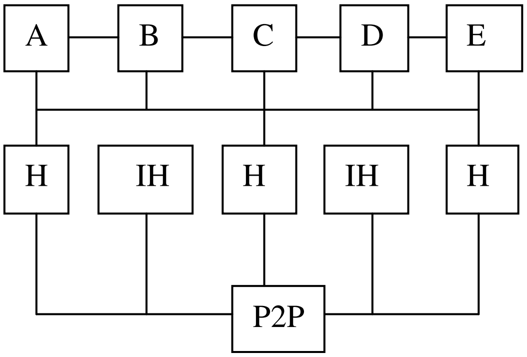

FIGURE 2: Block diagram (Using P2P botnets to detect Honeypots).

Proposed method has three steps. Explained as follows:

- Bots A, B, C, D, and E are launched into the network creating a Honeypot-aware attack,
- Bots infect the Honeypots in the network, thereby leaving infected Honeypots (IH) and uninfected Honeypots (H) in the cluster.
- Third step involves removing the Infected Honeypots (IH) using P2P botnet. Hence, uninfected Honeypots (H) can now be used in detecting bots A to E.

The P2P botnet which is constructed using peer list updating procedure, can be constructed in two parts,

The first part consists of a host which is vulnerable and later decides whether this is Honeypot or not, the second part contains the block of information and the authorization component allowing the infected host to join in the botnet.

Another honeypot-based monitoring occurrence happens during peer-list updating procedure. First, defenders could let their honeypot bots claim to be servant bots in peer-list updating. By doing this, these honeypots will be connected by many bots in the botnet, and hence, defenders are able to monitor a large fraction of the botnet. Second, during peer-list updating, each honeypot bot could get a fresh peer-list, which means the number of bots revealed to each honeypot could be doubled.

A honeypot could be configured to route all its outgoing traffic to other honeypots; at the same time, the trapped malicious code still believes that it has contacted some real machines. The P2P botnet constructed as introduced above is easy for attackers to control when facing monitoring and defense from security defenders. First, an attacker can easily learn how many zombie machines have been collected in the botnet and their IP addresses. The attacker can connect to several known infected computers, asking them to issue a command to let all bots sending a specific service request to the attacker’s sensor. On the other hand, security professionals cannot use this technique for monitoring, even if they know how to send such a command, due to their liability constraint. Second, an attacker can randomly choose any one or several bots to infill commands into the botnet—it is very hard for security defenders to cut off the control channel unless they hijack the botnet and take control of it by themselves. Such an active defense requires security professionals to issue commands to the botnet and update bot code on all (potentially hundreds or even thousands) compromised computers, which clearly puts a heavy liability burden on security professionals. Third, suppose security professionals remove many infected computers in a botnet. The attacker still has control over the remaining P2P botnet, even if the remaining botnet is broken into many separated smaller ones.

Security defenders could also try to distinguish which outgoing traffic is for honeypot detection and which outgoing traffic is for a real attack. If this could be done, then honeypots could be configured to allow the honeypot-detection traffic to be sent while blocking all other malicious traffic. For this purpose, security defenders will need to conduct more research on practically implementing automatic binary code analysis in honeypots. Internet security attack and defense is an endless war. From the attackers' perspective, there is a trade-off between detecting honeypots in their botnets and avoiding bot removal by security professionals. If an attacker conducts honeypot-aware test on a botnet frequently, honeypots in the botnet can be detected and removed quickly. But at the same time, the bots in the botnet will generate more outgoing traffic, and hence, they have more chance to be detected and removed by their users or security staff. In the end, we should emphasize that even if attackers can successfully detect and remove honeypots based on the methodology presented in the paper, there is still significant value in honeypot research and deployment for detecting the infection vector and the source of attacks. It may not be possible for honeypots to join the botnet, but the security hole used to facilitate the infection can be quickly discovered and patched.

4.0 DEFENSES AGAINST BOTNETS

Defense from Botnets can be divided to: prevention, detection and removal. In this section we provide information on how the user's system can be protected from botnet attacks. The botnet detected in the system must be removed immediately, otherwise it might cause other problems such as slow-down the performance, loss of data and leaking of information to the web.

Applying a patch if any damage occurs, is out of the context of this method. This method can recognize (fingerprint) botnets from the traffic and block the traffic, both upstream and downstream. The role of Honeypots here is to behave like servant bots to the botnets that intrude into the network. The moment the Honeypot bot is included into the botnet architecture, the monitoring of botnet activity is initiated by the defenders. The defenders can get many spying botnets into their hands so that they can monitor the commands given by the botmasters or send fake commands to the botmasters, leading to a trap since a remote code authentication (the problem of identifying a remote code program) [3] cannot distinguish the honeypot bots from the botnets from the botmaster's point of view.

4.1 Simulation and Analysis Using P2P botnets and Multiple Honeypots

Multiple Honeypots are used in synchronization with P2P botnets. It should be noted that there is a limitation of how many honeypots can be used in the cluster for efficient monitoring and detection. The number can be calculated based on an average of 50 to 100 simulation. The Simulation results are shown in Figure 3.

An analytical model can be derived by estimating the mean value of tracked bots, which is denoted by $T\ [B_{tracked}]$. There are h number of uninfected honeypots joining the cluster before the peer-list is updated [1]. Let the size of the peer-list be P, the final botnet has *l* number of botnets, and the number of servant bots used in peer-list updating procedure is Q.

Since the botnet has l bots, the average number of tracked bots can be calculated by

$$T\ [B_{tracked}] = I \left[1 - \left(1 - \frac{h}{Q}\right)^{P}\right]$$

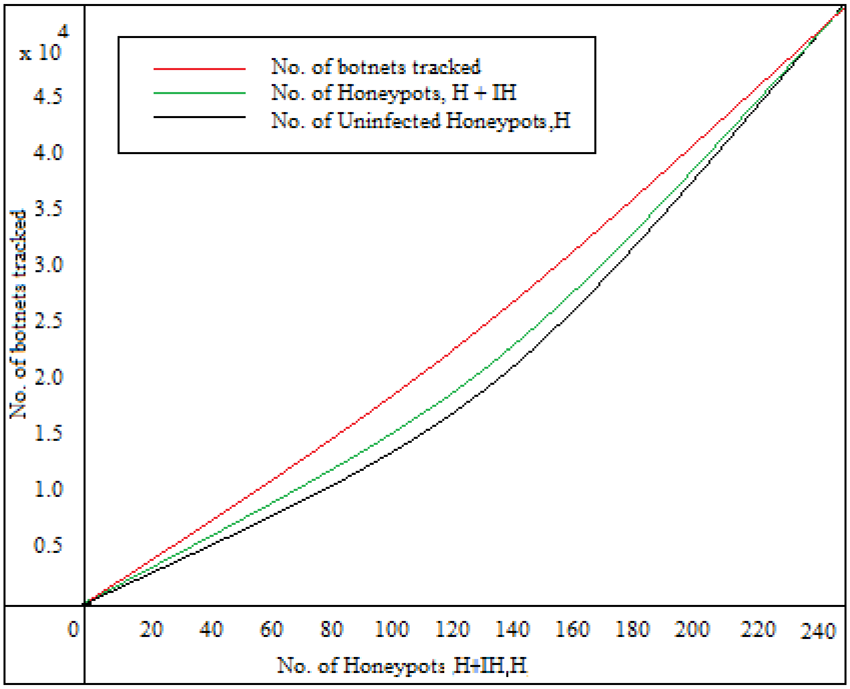

FIGURE 3: Simulation results for botnets tracked, total number of honeypots and uninfected honeypots

In Figure 3, number of Honeypots (H+IH) versus number of botnets tracked represented in the green curve and the number of uninfected honeypots denoted by H versus number of botnets tracked is represented in black curve. The number of botnets tracked versus the uninfected honeypots, H represented in red curve shows a better performance in detection after botnets are tracked.

4.2 Discussion

From the simulation and the above description on defenses against botnets, we can see that Honeypot based detection plays a vital role in the detection of botnets. The use of a robust P2P botnet to filter the infected honeypots from the network adds more to the defenders advantage as shown in Figure 3. In the future, Botmasters could design advanced ways to detect the honeypot defense system which could include fingerprinting (recognition). Attackers could also exploit the legal and ethical constraints held by the defenders [2]. The proposed method works owing to remote code authentication [3] which helps in not distinguishing the botnet from the honeypot bot.

In the future, if remote authentication method is compromised then the war between botmasters and security community would intensify. The research done until now in botnets shows “the worms” and botnets in the internet can be monitored but it gets harder to stop the attacks even after the presence of the threat is known, i.e. it should be detected without inflicting any damage to the data. Ethical and legal reasons in the prevention of botnet attacks turn out to be resource consuming [1]. The other methods which involve detection of botnets without Honeypots by using a botnet monitoring sensor is also considered which gives a clear picture on botnet activity but once the botmasters destroy the sensor, the machine or target will be infected. Our proposed method illustrates the design is practical and can be implemented by defenders with little complexities.

5 : CONCLUSION

Due to their potential for illicit financial gain, “botnets” have become popular among Internet attackers in recent years. As security defenders build more honeypot-based detection and defense systems, attackers will find ways to avoid honeypot traps in their botnets. Attackers can use software or hardware specific codes to detect the honeypot virtual environment [6, 7, 16], but they can also rely on a more general principle to detect honeypots: security professionals using honeypots have liability constraints such that their honeypots cannot be configured in a way that would allow them to send out real malicious attacks or too many malicious attacks. In this paper, we introduced a means for defending the network from botnets, when Honeypots are infected and then deploy a P2P botnet which would act as a filter to remove the infected Honeypots which remain in the Network Cluster and hence the uninfected Honeypots can be used with efficacy to defend the network. Honeypot research and deployment is important and should continue for the security community, but we hope this paper will remind honeypot researchers of the importance of studying ways to build covert honeypots, and the limitation in deploying honeypots in security defense. The current popular research focused on finding effective honeypot-based detection and defense approaches will be for naught if honeypots remain as easily detectible as they are presently.

6: REFERENCES

[1] P. Wang, S. Sparks, and Cliff C. Zou, “An Advanced Hybrid Peer-to-Peer Botnet,” IEEE; Vol. 7, No. 2, April-June 2010.

[2] Cliff C. Zou, Ryan Cunningham, “Honeypot-Aware Advanced Botnet Construction and Maintenance,” IEEE Computer society; Proceedings of the 2006 International Conference on Dependable Systems and Networks (DSN’06).

[3] Chia-Mei Chen, Ya-Hui Ou, and Yu-Chou Tsai, “Web Botnet Detection Based on Flow Information,” Department of Information Management, National Sun Yat –Sen University, Kaohsiung, Taiwan; IEEE 2010.

[4] 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.

[5] 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.

[6] J.R. Binkley and S. Singh, “An Algorithm for Anomaly-Based Botnet Detection,” Proc. USENIX Second Workshop Steps to Reducing Unwanted Traffic on the Internet (SRUTI ’06), June 2006.

[7] Sinit P2P Trojan Analysis, http://www.lurhq.com/sinit.html, 2008.

[8] Phatbot Trojan Analysis, http://www.lurhq.com/phatbot.html, 2008.

[9] F. Monrose, “Longitudinal Analysis of Botnet Dynamics,”ARO/DARPA/DHS Special Workshop Botnet, 2006.

[10] Washington Post: The Botnet Trackers, http://www.washingtonpos.com/wp-dyn/ content/article/2006/02/16AR2006021601388.html, Feb. 2006.

[11] M. Rajab, J. Zarfoss, F. Monrose, and A. Terzis, “A Multifaceted Approach to Understanding the Botnet Phenomenon,” Proc. ACM SIGCOMM Internet Measurement Conf. (IMC ’06), Oct. 2006.

[12] A. Karasaridis, B. Rexroad, D. Hoeflin, “Widescale botnet detection and characterization,” Proceedings of the first conference on First Workshop on Hot Topics in Understanding Botnets, 2007.

[13] A Taste of HTTP Botnets , team-cymru Inc, 2008, Available : http://www.team-cymru.org/ReadingRoom/Whitepapers/2008/http-botnets.pdf.

[14] Vogt R, Aycock J, Jacobson MJ. Army of botnets. In: Proc. of the 14th Annual Network & Distributed System Security Conf(NDSS). 2007.

[15] Zesheng Chen, Chao Chen, Qian Wang, "Delay-Tolerant Botnets," icccn, pp.1-6, 2009 Proceedings of 18th International Conference on Computer Communications and Networks, 2009.

[16] XF. Li, HX. Duan,W.Liu JP.Wu, "Understanding the Construction Mechanism of Botnets," uic-atc, pp.508-512, Symposia and Workshops on Ubiquitous, Autonomic and Trusted Computing, 2009.

[17] Chiang K, Lloyd L. A case study of the rustock rootkit and spam bot. In: Proc. of the 1st Workshop on Hot Topics in Understanding Botnets (HotBots 2007). 2007.

[18] R. Hund, M. Hamann, and T. Holz, "Towards Next-Generation Botnets," in Computer network Defense, 2008. EC22D 2008. European Conference on, 2008, pp. 13-40.

[19] C. Davis, S. Neville, J. Fernandez, J.-M. Robert, and J. McHugh, "Structured peer-to-peer overlay networks: Ideal botnets command and control infrastructures," In Proceedings of the 13th European Symposium on Research in Computer Security (ESORICS’08), October 2008.