ISSN 2250-3153

A Study on Botnet Detection Techniques

Haritha.S.Nair[1], Vinodh Ewards S E[2]

[1]Department of Computer Science and Engineering, Karunya University, Coimbatore, India
[2]School of Computer Science and Technology, Karunya University, Coimbatore, India

Abstract- A botnet is a network of compromised computers, termed bots that are used for malicious purposes. When a computer becomes compromised typically through a drive-by download, that has embedded malicious software, that computer becomes a part of a botnet. A bot typically runs hidden and uses a covert channel to communicate with its command and control server. Botnets are controlled through protocols such as IRC and HTTP and in protocol-conforming manners. This makes the detection of botnet command and control a challenging problem. In this paper we discuss some of the botnet detection techniques and compare their advantages, disadvantages and features used in each technique.

Index Terms- botnet, command and control, internet relay chat (IRC), nickname, passive anomaly analysis, spam.

I. INTRODUCTION

The recent growth of botnet activity in cyberspace has attracted in a significant way the attention of the research community. Botnets are one of the most dangerous species of network-based attacks today , responsible for a large volume of malicious activities from distributed-denial-of-service (DDoS) attacks to spamming, phishing, identify theft and DNS server Spoofing. The concept of botnet refers to a group of compromised computers remotely controlled by one attacker or a small group of attackers working together called a ‘‘botmaster”. These large groups of hosts are assembled by turning vulnerable hosts into so-called zombies, or bots, after which they can be controlled from afar. A collection of bots, when controlled by a single command and control (C2) infrastructure, form what is called a botnet. The botmaster’s ability to carry out an attack from hundreds or even tens of thousands of computers means increased bandwidth, increased processing power, increased memory for storage and a large number of attack sources making botnet attacks more malicious and difficult to detect and defend against.

The botnet detection techniques can be classified into three, namely,

- honeypot
- passive anomaly analysis and
- based on traffic application.

A honeypot [1] is a trap set to detect, deflect, or in some manner counteract attempts at unauthorized use of information systems. Generally it consists of a computer, data, or a network site that appears to be part of a network, but is actually isolated and monitored, and which seems to contain information or a resource of value to attackers.

The passive anomaly based detection is done by monitoring system activity and classifying it as either normal or anomalous. The classification is based on heuristics or rules, rather than patterns or signatures, and will detect any type of misuse that falls out of normal system operation. This is as opposed to signature based detection which can only detect attacks for which a signature has previously been created. In order to determine what traffic attack is, the system must be taught to recognize normal system activity PAYL [2] http://en.wikipedia.org/wiki/Anomaly based_intrusion_detection_system - cite_note-Wang2004-0 and MCPAD [3] are two anomaly based intrusion detection techniques that reduces the high false positive rate.

Botnet detection techniques based on traffic application classification are usually guided by botnet C&C control protocol e.g. if one is only interested in IRC-based botnets then traffic will be classified into IRC and non-IRC groups [4].

The rest of the paper is organized as follows. Section 2 will describe the traffic application classification based detection techniques. Section 3 will describe the anomaly based detection techniques and section 4 describes some botnet detection techniques independent of C&C protocol. Section 5 makes some concluding remarks.

II. TRAFFIC APPLICATION CLASSIFICATION BASED DETECTION METHODS

In [5,6], Strayer et al. use statistical flow characteristics and supervised classifiers to classify traffic into IRC or non-IRC groups. Once IRC traffic is identified, flows that were active at same time are correlated. The last stage detects malicious botnet by finding common IP address endpoint and any evidence of communication between botmaster and the C&C server.

In [5] Strayer et al. use an approach where the traffic that is unlikely to be a part of a botnet is eliminated first, then classifies the remaining traffic into a group that is likely to be part of a botnet, then correlates the likely traffic to find common communications patterns that would suggest the activity of a botnet. The technique begins with simply looking for chat sessions and then examining the content for botnet command. The freely available bot-building source code is used for text-based interaction to implement an IRC. The traces collected through this are used as an initial proxy for botnet traffic. These recorded traces are then fed into a series of quick reduction filters where the traces are classified into good sites (whitelists) and bad sites (blacklists) and also the flow attributes are examined. After the initial filters, the remaining flows are passed through a flow classification engine based on machine learning techniques. The classifiers attempt to group flows into broadly defined categories. Those flows that appear to have chat-like characteristics are passed on to the correlator stage where correlator performs a pair wise examination looking for flows behaving in similar manner

as two flows can be generated from the same application. Flows that are correlated are then passed on to topological analysis to determine which flows share a common controller. The result of this pipeline is a (hopefully) small set of flows that show a fair amount of evidence that they are related and are part of a botnet. This technique is well suited for real-time analysis of traffic data but it requires both legitimate and malicious training traffic and an accurate manner to label it.

Another approach is to use machine learning technique to identify botnet traffic [6] where the detection is done as a two stage process. Initially the IRC and non-IRC traffic are distinguished and then the botnet IRC and real IRC traffics are secernated. In [6] Strayer et al. proposes an approach to identify and detect botnet prior to them being used in cyber attack. This is achieved through a two stage approach where in first stage the communication flows are classified into chats and non-chats and in the second stage the classification of real and botnet chat is performed. For this the flows that are likely to compromise botnet C2 traffic is detected and then these flows are correlated to identify groups of flows that pertain to same botnets. Finally the C2 host is identified which leads to the attack host. One of the challenges faced is obtaining the ground truth. Data modeling is performed by retaining only the TCP packets and characterization of flows are done based on TCP and IP packet headers. These flows are then reduced to machine learning by using a set of heuristics and the flows that are not botnets are discarded.

III. ANOMALY BASED DETECTION TECHNIQUES

In [2] Wang et al. use an n-gram feature on payload for detection purpose. N-gram is a sequence on n adjacent bytes. For modeling a payload it is first classified into clusters according to some criteria like using port numbers or length. A payload model is computed for payloads of different lengths from same port for each direction of payload flow. A sliding window of size n is passed over the payload and the occurrence of each n-gram is counted. When n=1, the average byte frequency of each ASCII character 0-255 is obtained. In addition to this mean value, the standard deviation and variance is also calculated. Thus a set of payload models M_{xy} is computed where M_{xy} stores the average byte frequency and standard deviation of each byte's frequency of a payload of length x and port y. During detection, each incoming payload is scanned and its byte value distribution is computed. This newly computed payload distribution is then compared against model M_{xy}; if the distribution of the new payload is significantly different from the normal, the detector flags the packet as anomalous and generates an alert.

BotSniffer by Gu [7] is a technique that does not require any prior knowledge of signatures or C&C server addresses and can identify both the C&C servers and infected hosts in the network. The approach makes use of the fact that bots within the same botnet are likely to demonstrate spatial-temporal correlation. Botsniffer has two main components the monitor engine and the correlation engine. The monitor engine is implemented on top of the open-source system Snort [8]. The monitor engine examines network traffic and collects many attributes from the monitored network. BotSniffer first performs preprocessing to filter out irrelevant traffic to reduce the traffic volume, this is not essential but can improve the efficiency of BotSniffer. Two whitelists are generated in the process. The hard whitelist contains the normal servers that are less likely to serve as botnet C&C servers and the soft whitelist contains those addresses that are declared as "normal" in analysis phase. The soft whitelist is dynamically generated and is for a certain amount of time. The monitor engine also matches the protocol used by the clients that are similar to C&C and is port independent as the bots may use different ports. For correlation purpose the monitor engine also monitors the message and activity response of a bot using Response-Crowd-Density-Check algorithm and Response-Crowd-Homogeneity-Check algorithm respectively. The events observed by the monitor engine are analyzed by the correlation engine where the clients are grouped according to their destination IP and port and within each group perform group analysis of spatial-temporal correlation and similarity of activity or message response behaviors send to or from same server. If any suspicious C&C is found an alarm is triggered. Even for a detection of a single bot the alarm is triggered.

IV. DETECTION TECHINIQUE INDEPENDENT OF C&C CONTROLS

In BotMiner [9] the authors present a botnet detection method which clusters: network flows (C-Plane), which records network flows, and activity traffic (A-Plane), which identifies the activities of each host. A C-Plane flow (C-flow) contains all of the flows over a given time period between a particular internal IP and destination IP and port which use the same transport layer protocol. Some flows are excluded from consideration such as internal flows, and those to trustworthy legal servers such as Google. Certain C-flow characteristics are extracted like flows per hour (fph), packets per flow (ppf), bytes per packet (bpp), and bytes per second (bps). The A-Plane identifies hosts which demonstrate an abnormally high scan rate or weighted failed connection rate, spamming and downloading any Portable Executable binary. Clustering algorithms are applied to group hosts with similar communication patterns and activities patterns. Finally performs cross-plane correlation to identify hosts with similar communications and activities patterns.

V. TECHINIQUES BASED ON SPAM EMAILS

The techniques based on spam emails usually analyses the email patterns and may also derive certain features of these emails such as sender/recipient address etc. In [10] the authors propose a method called EsBod which is an email shape based botnet detector which will classify the email into spam or real email. The shape generator of the Esbod will extract the skeleton of the email that is fed into the system. Using Gaussian kernel density estimator the shape (or template) of the email is derived from the skeleton. The classifier of the EsBod will takes in this derived shape and matches with the botnet signature repository using Hellinger distance.

In [11], Brodsky et al proposes a distributed content independent spam classification method called Trinity which uses source identification along with a peer-to-peer based distributed database. Trinity first determines the source IP of the received email and then updates the database using this IP. The database is checked for past traces of email sources and number of emails that source recently send within a fixed period and a

score is obtained. This score can be used for classification by the MUA (mail user agent). If the score is high and if the sender is not in the sender/recipient address book, then the email must be a spam.

A graph based approach is proposed in [12] by the authors. In this approach large user-user graphs and tightly connected subgraphs are drawn which detects botnet spamming attacks targeting major Web email providers.

VI. NICK NAME BASED DETECTION TECHNIQUE

All the bots communicating with its botmaster will have a nickname. The first step in establishing a botnet connection between the botmaster and the bot is assigning a unique nickname for the bot. Analyzing such nicknames and the similarities between the nicknames can be a useful technique for detection of botnets. In [13] Wang et al. use the similarity between the nicknames and is calculated using channel distance. The nicknames within one channel will have same structure. The channel distance is the Euclidean distance between two similar nicknames.

The detection methods using the nickname will rely on the communication channel between the bot and the Command and Control. The method proposed by Goebel in[14] is a similar one. In [14] Goebel use n-gram feature along with a scoring function for detection of bots that use uncommon communication channels by monitoring network traffic for unusual or suspicious IRC nicknames. The method generates warning emails to administrator to report infected machines. The captures packets are analyzed and some features like time, IP address and port of source and destination, nicknames etc. Then the scoring function checks for occurrence of several criteria. For each successful hit, scoring function points of the nickname are incremented. The final points of the nickname along with other information are stored in the connection object. The higher the connection object, the probability that it is a spam is higher. If the scoring function crosses a particular threshold the system will trigger an alarm.

VII. CONCLUSION

In this paper most of the techniques used for botnet detection so far are reviewed. Other than above mention techniques there are techniques that are based on log correlation which is mentioned in [15] by Masud.

REFERENCES

[1] Honey net webpage http://www.honeynet.org/project

[2] K. Wang, S. Stolfo, "Anomalous payload-based network intrusion detection", in: *Proceedings of the 7th International Symposium on Recent Advances in Intrusion Detection (RAID)*, (Sophia Antipolis, France, 2004).

[3] Perdisci, Roberto; Davide Ariu, Prahlad Fogla, Giorgio Giacinto, and Wenke Lee. "McPAD : A Multiple Classifier System for Accurate Payload-based Anomaly Detection". *Computer Networks, Special Issue on Traffic Classification and Its Applications to Modern Networks* **5** (6) (2009): 864–881.

[4] Wei Lu, Goaletsa Rammidi, Ali A. Ghorbani, "Clustering botnet communication traffic based on n-gram feature selection", in *ELSEVIER Computer Communications* 34 (2011) 502–514.

[5] T. Strayer, D. Lapsley, R. Walsh, C. Livadas, "Botnet Detection: Countering the Largest Security Threat", *Springer, Chapter Botnet Detection Based on Network Behavior* , vol. 36, 2008.

[6] C. Livadas, R. Walsh, D. Lapsley, T. Strayer, "Using machine learning techniques to identify botnet traffic", in: *Proceedings 2006 31st IEEE Conference on Local Computer Networks*, 2006, pp. 967–974.

[7] G. Gu, J. Zhang, W. Lee, "BotSniffer: Detecting botnet command and control channels in network traffic", in: *Proceedings of the 15th Annual Network and Distributed System Security Symposium* (NDSS'08), 2008.

[8] M. Roesch. "Snort - lightweight intrusion detection for networks". In *Proceedings of USENIX LISA'99*, 1999.

[9] G. Gu, R. Perdisci, J. Zhang, W. Lee, "BotMiner: clustering analysis of network traffic for protocol- and structure-independent botnet detection", in: *Proceedings of the 17th USENIX Security Symposium (Security'08),* 2008.

[10] P. Sroufe, S. Phithakkitnukoon, R. Dantu, J. Cangussu, "Email shape analysis for spam botnet detection", in: *Sixth IEEE Consumer Communications and Networking Conference*, (Las Vegas, NV), January 2009, pp. 1–2.

[11] A. Brodsky, D. Brodsky, "A distributed content independent method for spam detection", in: *Proceedings of the First Conference on First Workshop on Hot Topics in Understanding Botnets*,(Cambridge, MA, 2007), p. 3.

[12] Y. Zhao, Y.L. Xie, F. Yu, Q.F. Ke, Y. Yu, Y. Chen, E. Gillum, "BotGraph: large-scale spamming botnet detection", in: *Proceedings of the 6th USENIX Symposium on Networked Systems Design and Implementation*, 2008, pp. 321–334.

[13] W. Wang, B. Fang, Z. Zhang, C. Li, "A novel approach to detect IRC-based botnets", in: *International Conference on Networks Security, Wireless Communications and Trusted Computing*, (Wuhan, Hubei, 2009), pp. 408–411.

[14] J. Goebel, T. Holz, "Rishi: identify bot contaminated hosts by irc nickname evaluation", in: *HotBots'07: Proceedings of the First Conference on First Workshop on Hot Topics in Understanding Botnets*, (Berkeley, CA, USA, 2007) (USENIX Association).

[15] M.M. Masud, J. Gao, L. Khan, B. Thuraisingham, "Peer to peer botnet detection for cyber-security: a data mining approach", in: Proceedings of the 4th Annual Workshop on Cyber Security and Information Intelligence Research: Developing Strategies to Meet the Cyber Security and Information Intelligence Challenges Ahead, (Oak Ridge, Tennessee, 2008).

AUTHORS

First Author – Haritha.S.Nair, Pursuing MTech in Software Engineering, Karunya University.
Email id - harithasnair.1186@gmail.com

Second Author – Vinodh Ewards S.E, Assistant Professor / CSE, Karunya University.
Email id - ewards@karunya.edu