Journal of Information Engineering and Applications
ISSN 2224-5782 (print) ISSN 2225-0506 (online)
Vol.3, No.12, 2013

A Review-Botnet Detection and Suppression in Clouds

Namrata A. sable
M.E.(CSE)
G. H. Raisoni College of Engineering & Management, Amravati
SGBAU, Amravati University
Amravati(MS), India
E-mail: namratasable10@gmail.com
Prof. D. S. Datar
G. H. Raisoni College of Engineering & Management, Amravati
SGBAU, Amravati University
Amravati(MS), India
E-mail: dineshdatar@raisoni.net

Abstract
Internet security problems remain a major challenge with many security concerns such as Internet worms, spam, and phishing attacks. Botnets is well-organized distributed network attacks, consist of a large number of bots that generate huge volumes of spam or launch Distributed Denial of Service (DDoS) attacks on victim hosts. Botnet attacks degrade the status of Internet security. Clouds provide botmaster with an ideal environment of rich computing resources where it can easily deploy or remove C&C server and perform attacks. It is of vital importance for cloud service providers to detect botnet, prevent attack, and trace back to the botmaster. It also becomes necessary to detect and suppress these bots to protect the clouds. This paper provides the various botnet detection techniques and the comparison of various botnet detection techniques. It also provides the botnet suppression technique in cloud.
Keywords: Cloud computing, network security, botnet, botmmaster, botnet detection, botnet suppression

1. Introduction

Internet is playing a very important role as our information infrastructure, the e- business and e-pay sector is booming due to its convenience and benefits for users. However, Internet security remains a big challenge as there are many security threats. Attackers increasingly initiate e-crime attacks and abuses, such as spam, phishing attacks, and Internet worms. Botnet is a collection of internet-connected computers whose security defences have been breached. Also botnet are the group of compromised computers controlled by one or group of attacker known as "Botmaster". The most of botmasters usually don't use their PC to control botnet, instead they use public server to transfer command and control to every bot. this server is called command and control server (C&C). The architecture of botnets is either centralized or distributed. The centralized botnets have one or more command and control(C&C) servers, and the distributed include all the botnets that do not operate in a centralized manner, which can hide botmaster very well. Peer-to-Peer botnet is a well-known example of the distributed type. Centralized architecture increases the efficiency over the traditional C&C botnets, but at the same time, they are largely limited by the total number of bots they could control at one time, Because of the lack of the central server, the Botmaster cannot directly control all the bots.

The communication protocols of botnets are Internet Relay Chat (IRC) and Hypertext Transport Protocol (HTTP). IRC is a wide-used botnets protocol, because IRC servers are extremely popular so that botmaster can easily find a public IRC server for use . However, the disadvantage of IRC is also obvious, once the botnet is detected, the IRC server can be easily found and taken down, in addition. A better option for the botmaster is use HTTP, because the port of HTTP is almost always permitted by firewalls. Whatever for the C&C server or normally bot, there always has huge HTTP traffic, it's very hard to find something in this lager data. Botnet operators can use the aggregated power of many bots to exponentially raise the impact of those dangerous activities. The major attacks under botnet are, DDos, Scanning, Phishing, Click fraud, spamming.

In this paper, section 1 gives introduction to botnet. Section gives the literature review and related work on botnet detection and suppression. Section 3 introduces various botnet detection techniques and comparison. Section 4 introduces botnet suppression technique in cloud. In section 5 proposed work and objectives are given.

2. Literature Review and Related Work

Extensive research has been done in botnet detection and suppression .In this section we mainly focuses on the different botnet detection technique and botnet suppression technique.

The mechanism of various botnet detection techniques are given by Jignesh Vania, Arvind Meniya, H. B. Jethva [1] Haritha S. Nair, Vinodh Ewards [2]. Inspection of the network traffic respective to anomalies or specific singularities of the IRC protocol which can be a hint for a botnet called the network based botnet

detection techniques are given by W. T. Strayer, R. Walsh, C. Livadas, D. Lapsley [3], J. Goebel, T. Holz. Rishi [4], A. Karasaridis, B. Rexroad, D. Hoeflin. Widescale [5]. BotHunter a method for botnet detection which entails correlating alarms from different network intrusion detection system (NIDS) elements which reside at the egress boundary and BotMiner a botnet detection method which clusters: communications traffic (C-Plane), which identifies which hosts are talking to which other hosts, and activity traffic is given by Gu, R. Perdisci, J. Zhang, and W. Lee [6]. The binder, host-based procedures botnet detection technique includes detection of possible anomalies or modification of the file system is proposed by W. Cui, R. H. Katz, W. Tan [7]. The botSwat is a techmique that characterizes the remote control behavior of bots via identifying when selected system call arguments contain data received over the network, is given by E. Stinson, J. C. Mitchell [8].

The Garlic architecture which is distributed botnet suppression system which suppresses the botnets in clouds is proposed by F. Han, Z. Chen, H. Xu, and Y. Liang [9]. Based on an overlay network, Collaborative Network Security System which automatically collects network traffic from every collaborative UTM in a distributed mode and then processing these collected data in the security center is proposed by F. Han, Z. Chen, H. Xu, and Y. Liang [9].

3. Botnet Detection and Comparison

3.1 Classification based on behavior

3.1.1 Active analysis

Active approaches in botnet analysis cover all kinds of analysis techniques which makes bot master, directly or indirectly informed about botnet detection activity. Capturing bot malware and deactivating its malicious parts is a well-known active analysis type. Honeypots and honeynets are other active analysis methods performed in botnet detection and prevention.

i) Honeypots and honeynets: A honeypot defined in environment where vulnerabilities have been deliberately introduced to observe attacks and intrusions. They have a strong ability to detect security threats, to collect malware signatures and to understand the motivation and technique behind the threat used by perpetrator. In a wide-scale network, different size of honeypots form honeynet. Honeypots are classified as high-interaction and low-interaction according to their emulation capacity. A high-interaction honeypot can simulate almost all aspects of a real operating system. It gives responses for known ports and protocols as in a real zombie computer. On the other hand, low-interaction honeypots simulate only important features of a real operating system. High-interaction honeypots allow intruders to gain full control to the operating system; however low-interaction honeypots do not. Honeypots are also classified according to their physical state. Physical honeypot is a real machine running a real operating system. Virtual honeypot is an emulation of a real machine on a virtualization host. The value of a honeypot is determined by the information obtained from it. Monitoring the network traffic on a honeypot lets us gather information that is not available to network intrusion detection systems (NIDS).

3.1.2 Passive analysis

Passive approaches analyze traffic which the botnet generates without corrupting or modifying it. The analysis mainly focuses on secondary effects of botnet traffic such as broken packets resulting from a distant DDoS attack. Passive systems are more complex to implement but in the other hand they have the big advantage that they cannot be detected by intruder; because if perpetrator sends a message to a darknet, he will not get a SYN response. So a darknet is absolutely gives the same sense as an unused IP address to an intruder.

3.2 *Classification based on used data*

3.2.1 Analysis Based on IDS Data

Intrusion detection [11] is “the process of identifying and responding to malicious activities targeted at computing and network resources”. An intrusion attempt, also named as attack, denotes the sequence of actions to gain control of the system. Intrusion Detection System (IDS) discriminates intrusion attempts from normal system usage. Intrusion detection systems are basically classified into two categories:

i) Misuse-based IDS and Anomaly-based IDS [11]. A misuse-based IDS, also known as signature-based or knowledge-based IDS, detects malicious traffic by comparing new data with a knowledge base or signatures of known attacks. The system delivers an alarm if a previously known intrusion pattern is detected. Misuse-base systems like Snort use pattern matching algorithms in packet payload analysis. It is obvious that misuse-based systems analyze not only the traffic flow of the network; they also analyze payload data of the flow. Misuse based intrusion detection systems are highly accurate systems. But they need to pay attention on up to date the signature base of the system. They are also ineffective for detecting new intrusion types and zero day threats. Rule based intrusion detection systems like Snort are running by using known malware signatures. They monitor the network traffic and detect sign of intrusions. It is obvious that payload information of network traffic is transformed and embedded into the signature or rule. The IDS detects malicious traffic fitting the communication parameters defined by the rule.

ii) Anomaly Based Botnet Detection also known as behavior-based IDS, compare input data with the expected behavior of the system. However behavior based systems can detect unknown attacks because of their anomaly based nature; they may give false positive alarms.

This approach tries to detect Botnet based on number of network traffic anomalies such as high network latency, high volumes of traffic, traffic on unusual ports, and unusual system behavior that could show existence of bots in the network. This approach can detect unknown Botnets. It can be categorized as Host based and Network Based Detection. In host based detection technique, a detection strategy which monitors and analyzes the internals of a computer system instead of network traffics on its external interfaces.

iii) A network-based technique: It is a detection strategy which tries to detect Botnets by monitoring network traffics. We can classify Network-based techniques into two categories: Active monitoring and Passive monitoring. In Active monitoring, it injects test packets in network to measure the reaction of network such that gaining extra traffic on network.

3.2.2 DNS Based detection technique

In order to access the C&C server bots carry out DNS queries to locate the particular C&C server that is typically hosted by a DDNS (Dynamic DNS) provider. So DNS monitoring will be easy approach to detect botnet DNS traffic and detect DNS traffic anomalies. This is most famous and easy technique to botnet detection but it will be tough to detect recent advanced botnet through this technique.

3.3 *Data Mining Based Detection Technique*

Data mining aims to recognize useful patterns to discover regularities and irregularities in large data sets. Packet flow provides full information of flow data but in large file type. Anomaly based techniques are mostly based on network behavior anomalies such as high network latency, activities on unused ports [11]. Data mining technique can be applied for optimization purpose. It enables to extract sufficient data for analysis from network log file. Most useful data mining techniques includes correlation, classification, clustering, statistical analysis, and aggregation for efficiently knowledge discovery about network flows [12]. Flow correlation algorithms are useful to compare flow objects based on some characteristic other than packet content. This technique is very effective when content of packet is not available of encrypted, e.g. might compare arrival time. These kinds of algorithms utilize the characteristic values as inputs into one or more functions to create a metric used to decide if the flows are correlated [12]. Classification algorithms assume that incoming packet will match one of the previous patterns. Therefore, it is not an appropriate approach to detect new attacks [12]. Clustering is a well-known data mining technique where data points are clustered together based on their feature values and a similarity metric. Clustering differs from classification, in that there is no target variable for clustering. Clustering algorithms divide the entire data set into subgroups or clusters containing relatively identical features. Thus, clustering provides some significant advantages over the classification techniques, since it does not require a labeled data set for training. To find particular pattern from large dataset is known as aggregation method, collecting and analyzing several types of records from different channels simultaneously. Association rule is to find the correlation of different items appeared in the same event. Association rule mining is to derive the implication relationships between data items under the conditions of a set of given project types and a number of records and through analyzing the records, the commonly used algorithm is A priori algorithm. As shown in table Data Mining approach is almost the best one among other technique.

At this point of view pattern recognition and machine learning based data mining techniques are very useful to extract unexpected network patterns. Firstly it can be useful to introduce a research of preprocessing tasks of anomaly and data mining based botnet detection systems.

Fig. 1 shows the classification of botnet detection techniques.

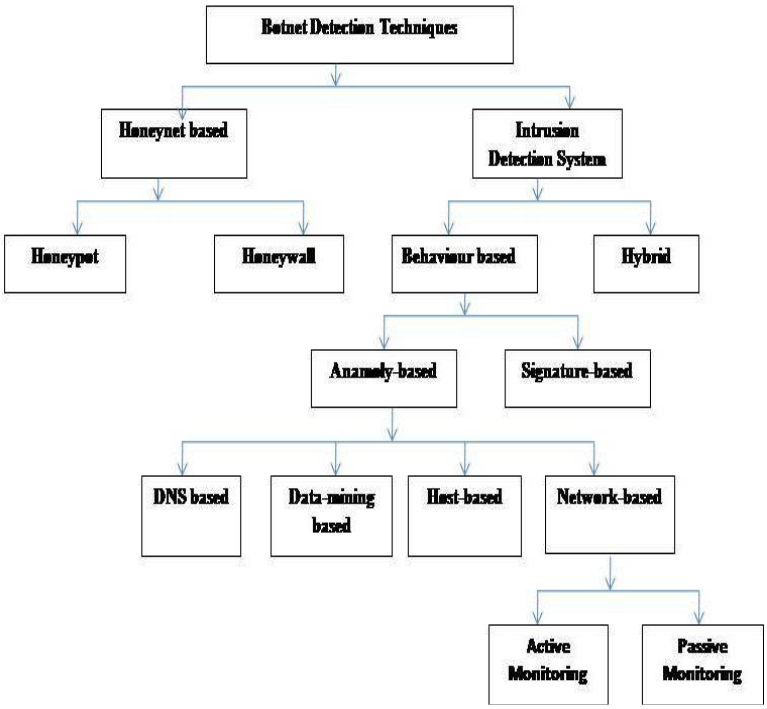

Fig. 1 Botnet Detection Techniques.

4. Botnet Supression In Clouds

4.1 *Garlic architecture*

The structure is important sign of garlic system [13], as its name, it is distributed and consists of terminal node and control node. The system may have multiple terminal nodes and one control node, every node can be collaborated with each other [13], as illustrated in Fig. 2

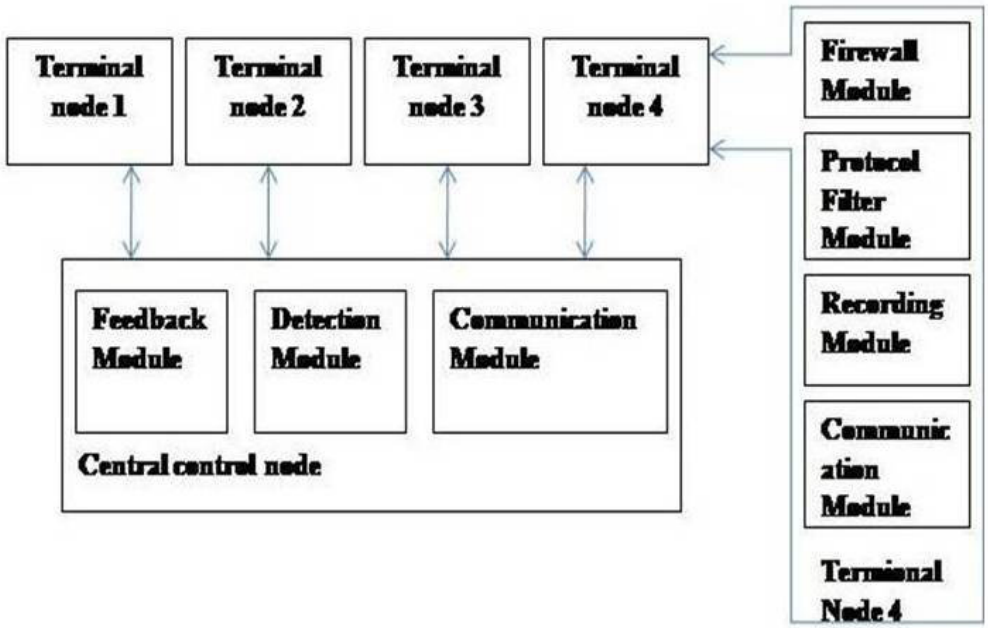

Fig. 2 System Architecture

In Fig. 2, we can see the garlic system has a central control node, which has three modules. The control node connects with four terminal nodes. The terminal node is based on the Unified Threat Management (UTM) system [14], which has four functional modules: the firewall module, the protocol filter module, the recording module and the communication module. The firewall module can pass or block network traffic according to firewall rules. Similarly, the protocol filter module filters network traffic based on regular expressions of rules.

The recording module can be set to record all of the traffic or just partial traffic of each connection, this module use technology of TIFA [15]. Communication module exchanges information with the control node. Central control node is a server that has three functional modules, feedback module, detection module and communication module. The central control node collects traffic from all the terminal nodes controlled by itself, process the traffic using detection module, which is based on cloud computing technology, generally, when botnets are detected, the detection module will report IP address and port, according to these information, rules can be generated. Then central control node distributes rules to all the terminals. After distribution, Garlic checks whether the rules have feedback indicating the rules are effective or not. Garlic system will continuously collect feedback for each rule, according to these, it can regenerate rules and distribute them, and these second-generation rules can also produce feedbacks, forming a recursive process.

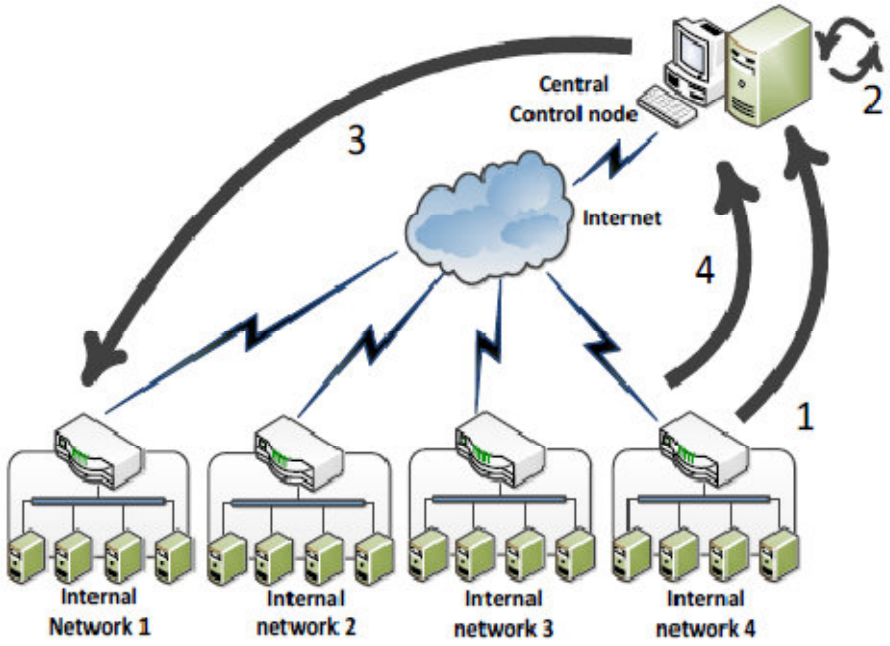

Fig. 3 The workflow of Garlic.

Fig. 3 is the workflow of Garlic. Step 1 indicates terminal node transfers traffic data to central control node, traffic data is recorded at the terminal node. When the traffic load is low, the recording module can be set to record all the traffic; when the network is overloaded, it can be set to just record the first 10-20 KB of each connection. However, this header should contain the essential information. In step 2, the control node processes huge traffic which is collected in Step 1. Due to the burst nature of network traffic and the fact that most networks can produce thousands and even millions of packets per second, in addition, the traffic of botnet must be huge, it is very challenging to process the data in real time, which is solved by LARX. LARX is a cloud computing platform that can process multiple tasks in parallel. addreses, the source port and destination port. The final field is used to mark the rules, facilitate feedback for viewing. Then the third step is to load the rules to each terminal node, so the advantage of this approach is combining prevention, control and sharing rules together.

5. Proposed Work

We can apply the data mining botnet detection Technique in collaborative network security management system (CNSMS). In CNSMS using cloud storage to keep collected traffic data and then processing it with cloud computing platforms to find the malicious attacks and to secure the cloud from botnet.

References

Jignesh Vania, Arvind Meniya, H. B. Jethva, A Review on Botnet and Detection Technique, in *International* Journal of Computer Trends and Technology- volume4Issue1- 2013,Page 23.

Haritha S. Nair, Vinodh Ewards S E *A Study on Botnet Detection Techniques*, International Journal of Scientific and Research Publications- Volume 2, Issue 4, April 2012.

W. T. Strayer, R. Walsh, C. Livadas, D. Lapsley. Detecting Botnets with Tight Command and Control. In IEEE Conference on Local Computer Networks, Nov. 2006.
J. Goebel, T. Holz. Rishi: Identify Bot Contaminated Hosts by IRC Nickname Evaluation. In Workshop on Hot Topics in Understanding Botnets, April 2007.
A. Karasaridis, B. Rexroad, D. Hoeflin. Widescale Botnet Detection and Characterization. In Workshop on Hot Topics in Understanding Botnets, April 2007.
G. Gu, R. Perdisci, J. Zhang, and W. Lee. Bot- Miner: Clustering Analysis of Network Traffic for Protocol- and Structure-Independent Botnet Detection. In USENIX Security Symposium, July 2008.
W. Cui, R. H. Katz, W. Tan. Design and Implementation of an Extrusion-based Break-In Detector for Personal Computers. In Annual Computer Security Applications Conf., Dec. 2005.
E. Stinson, J. C. Mitchell. Characterizing Bots' Remote Control Behavior. In Detection of Intrusions & Malware, and Vulnerability Assessment, July 2007.
F. Han, Z. Chen, H. Xu, and Y. Liang, Garlic: A distributed botnets suppression system, in Proc. IEEE ICDCS workshop on the First International Workshop on Network Forensics, Security and Privacy (NFSP), Macau, China, 2012, pp. 634-639.
F. Han, Z. Chen, H. Xu, and Y. Liang, A collaborative botnets suppression system based on overlay network, International Journal of Security and Networks, vol. 7, no. 4, 2012.
Erdem Alparslan, Adem Karahoca and Dilek Karahoca. *BotNet Detection: Enhancing* Analysis by Using Data Mining Techniques, Downloaded from http://dx.doi.org/10.5772/48804 (BOOK).
Alireza Shahrestani, Maryam Feily, Rodina Ahmad, Sureswaran Ramadass. Architecture for applying data mining and Visualization on network flow for botnet traffic Detection, International Conference on Computer Technology and Development,IEEE, Pages 33-37 2009.
Beipeng Mu, Xinming Chen and Zhen Chen. A Collaborative Network Security Management System in Metropolitan Area Network. In:CMC 2011.
Ying Zhang, Fachao Deng, Zhen Chen, Yibo Xue and Chuang Lin. UTM-CM: A Practical Control Mechanism Solution for UTM System. In:CMC 2010.
Jun Li, Shuai Ding, Ming Xu, Fuye Han and Zhen Chen. TIFA: Enabling Real-Time Querying and Storage of Massive Stream Data. In:ICNDC 2011.