ARCHITECTURE FOR APPLYING DATA MINING AND VISUALIZATION ON NETWORK FLOW FOR BOTNET TRAFFIC DETECTION

Alireza Shahrestani[1, 2], Maryam Feily[2], Rodina Ahmad[1], Sureswaran Ramadass[2]

[1]*Faculty of Computer Science and Information Technology*
University of Malaya (UM)
Kuala Lumpur, Malaysia

[2]*National Advanced IPv6 Centre of Excellence (NAv6)*
Universiti Sains Malaysia (USM)
Penang, Malaysia

{shahrestani, maryam, sures}@nav6.org, rodina@um.edu.mu

Abstract—**Botnet is one of the most recent tools used in cyber-crime including Distributed Denial of Service attacks, phishing, spamming, and spying on remote computers. These days, governments, business, and individuals are facing catastrophic damages caused by hackers using malicious botnets. It is a major challenge for cyber-security research community to combat the emerging threat of botnets. Current network intrusion detection methods based on anomaly detection approaches suffer from fairly high error rate and low performance. The proposed flow based botnet detection system tackles these issues by combining data mining and visualization. The anomalous data is passed to several trust models, and the flows are re-evaluated to obtain their trustfulness, which is then aggregated to detect malicious traffic via visualization. The visualized information will be analyzed by human intellectual and conceptual ability to gain useful knowledge about botnet activities for further precaution and validation.**

Keywords: *Botnet; Botnet Detection; Data Mining; Visualization;*

1. Introduction

Botnets are emerging as "the biggest threat facing the internet today" [1] due to their enormous volume and sheer power. According to explanation in [2, 3], malicious botnet is a network of compromised computers called "Bots" under the remote control of a human operator called "Botmaster". The term "Bot" is derived from the word "Robot"; and similar to robots, bots are designed to perform some predefined functions in automated way. In other words, the individual bots are software programs that run on a host computer allowing the botmaster to control host actions remotely [2, 3]. The botmaster can instruct these bots to recruit new bots, launch coordinated DDoS attack against specific hosts, steal sensitive information from infected machines, send mass spam emails, and so on [4].

The bot program establishes a command and control (C&C) channel, and connects the zombie to the command and control (C&C) server. Upon the establishment of C&C channel, the zombie becomes a part of attacker's botnet army. Afterward, the actual botnet command and control activities will be started. The botmaster uses a rendezvous point which is usually a public IRC server or a compromised host to issue malicious commands. These commands are disseminated to the bot army through the established C&C channel. Accordingly, bot programs receive and execute commands sent by botmaster. Furthermore, the C&C channel enables the botmaster to remotely control the action of large number of bots to conduct various illicit activities [5, 6, 7].

Botnets use different topologies and protocols for their command and control communications. The most widespread botnets are based on Internet Relay Chat (IRC). IRC botnets have centralized architecture, and therefore there is a single point of failure. There are many approaches available to detect and take down IRC botnets. On the other hand, Peer to Peer (P2P) is a relatively newer technology used in botnets. P2P botnets use P2P protocols for their malicious communication among other bots and the botmaster. Since P2P botnets are distributed, there is no central point of failure. Besides, P2P botnets are relatively smaller than their IRC counterparts. As a result, these botnets are more difficult to detect and destroy than the IRC botnets [6, 7].

In general, network flow defines both attack and attack mechanism; but, in the case of botnet, mechanism is constructed and maintained independently of how it will be used for future attacks. Most flow-based trace back systems adopt a reactive approach. Thus, the process of tracing packets back to their origin hosts is triggered only after an attack is detected. Nevertheless, it is possible to detect botnets prior to the attack in that botnets usually remain in a benign state for some times before they are used for a specific attack [5]. Botnet detection and disruption has been a major research topic in recent years. There have been several proposals on using data mining for counter-terrorism and cyber-security applications. For instance, data mining can be used to detect unusual patterns, terrorist activities and fraudulent behavior. In addition, data mining can also be used for intrusion detection and malicious code detection.

Our goal is to find evidence of botnet activity by passive network traffic monitoring and without examining the traffic content. Moreover, we intend to gain useful knowledge about botnet traffic through visualization to facilitate botnet traffic detection, since visualized information is easier to comprehend for human beings. The information about botnet traffic which will be visualized is discovered through multi layers of data mining techniques applied to network traffic data. Combination of data mining and visualization will assist network administrators to recognize the threats more easily and efficiently.

978-0-7695-3892-1/09 $26.00 © 2009 IEEE
DOI 10.1109/ICCTD.2009.82

The remainder of this article is structured as follows: in Section 2, we provide a fundamental description of network flow and its characteristics. Next, in Section 3, related data mining approaches on the flow data will be discussed. The importance of visualization for botnet detection will be illustrated in Section 4. The proposed architecture for combination of data mining and visualization will be explained in Section 5. Finally, the paper will be concluded in Section 6.

2. Network Flow Characteristics

A specific network flow is defined as a series of packets that belong to the same instance of communication between an application at a source host, and an application at a destination host. Subsequently, to identify a particular TCP or UDP flow we can use a 5-tuple comprised of five values from packet headers including: source and destination IP addresses, source and destination port numbers, and protocol identifier number. These five values definitively identify a particular instance of communication between a source host application and destination host application.

Furthermore, we can categorize flow characteristics into two categories: Static characteristics and Dynamic characteristics. Static characteristics remain constant over the lifetime of the flow, whereas dynamic characteristics tend to change as the flow progress through time. Several static characteristic of flow can be retrieved from packet headers. Moreover, there are other static characteristics such as the start time, stop time, and duration of flow which are not carried inside the packets. On the other hand, by looking outside the packet we can draw dynamic features of the flow. For instance, arrival and departure time of the packets can be added as supplementary information. Besides, it is possible to derive additional dynamic features such as throughput and burst time from the payload information of packets [8].

Useful knowledge about flow can be discovered from flow characteristics. The first step for knowledge discovery from flow characteristic is to define a unique flow as an object. A unique flow can be defined by choosing and quantifying a specific set of flow characteristics. Doing so, we will be able to derive useful knowledge by comparing different instances of flows in the form of objects. Flow based knowledge discovery increases the potential of data availability, and facilitate handling of transferred data as well as privacy issues. However, this approach endures few drawbacks that should be considered. Namely, in a complex context like security, where every arbitrary detail may be crucial for accurate analysis, reducing the amount of information may result in lower level of confidence due to the higher false positive rates. In addition, knowledge about a specific flow will be discovered at the end of the flow. This is considered as delay and reduces real-time functionality. To overcome these problems additional metadata is required. Moreover, sometimes the existing data should be converted into a more convenient data types. In order to handle all these data effectively, data mining techniques will be employed [9].

3. Data Mining for Botnet Detection

Data mining aims to recognize useful patterns to discover regularities and irregularities in large data sets [10]. In flow based knowledge discovery, it is usual to use a full packet trace for flow identification. Although packet trace files provide complete description of flows, they are too large for analysis. Therefore, data mining techniques can be applied for optimization purposes. In fact, data mining techniques enables to extract sufficient data for analysis. Consequently, instead of storing the trace files of each packet, only a specific set of flow characteristics will be considered for flow identification, and each flow will be stored as an object in a database for analysis.

Moreover, it is important to note that malicious botnet activities are coordinated in that a group of compromised hosts react to each command sent by the botmaster. Besides, botnet command and control traffic appears as legitimate traffic. Thus, individual connection records in an attack through botnet are not malicious by themselves, unless they prove to be part of a series of synchronized connections. Therefore, basic attributes which give a good description of individual connections, are not sufficient to identify malicious botnet traffic, since they cannot provide larger overview of a group of connections and network flows. To overcome this, we might need to drive extra knowledge and values through different data mining techniques. A wide range of data mining techniques including correlation, classification, clustering, statistical analysis, and aggregation can be used for knowledge discovery about network flows [8, 10].

Flow correlation algorithms are useful to compare flow objects based on some characteristic other than packet content. This is very effective when the content of packets is not available or is encrypted. Most correlation algorithms only use a single characteristic for comparison. For example, an algorithm might compare flows based on the packet inter-arrival times. To identify related connections, any of the characteristics may be chosen so that they can be compared. 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. If the correlation between two flows is strong enough, we can derive that the flows belong to same alliance. This decision is often made by comparing the metric to a threshold [5].

Classification is another way for detecting abnormal activities in a network. Classification algorithms assume that incoming packet will match one of the previous patterns. Therefore, it is not an appropriate approach to detect new attacks. In other words, detecting new attacks cannot depend on the current set of classification rules. Clustering and statistical analysis that are primarily concerned with one or few attributes are more efficient for anomaly detection [11, 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. The clustering task does not try to classify, estimate, or predict the value of a target variable. Instead, 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 [13]. In a flow based detection system, clustering refers to the grouping of records, observations, or cases into classes of similar objects. In this case, cluster is a collection of objects that are similar to one another and dissimilar to objects in other clusters. In each cluster, the similarity of the objects within the cluster is maximized, whereas the similarity to objects outside this cluster is minimized. The proposals in [14, 15] are good clustering approaches for outlier detection based on distance.

On the other hand, in circumstances that we have a hypothesis about flow characteristics, statistical techniques can be employed to quantify our investigation. In statistical analysis, estimation and prediction methods are used for statistical inference. Estimation and testing hypotheses about flow characteristics are based on the information contained in each object. This technique is also known as top –down learning [16].

There is also another data mining technique to obtain a summary count of traffic matching a particular pattern. This method is called data aggregation that enables to get a more complete picture of the information by collecting and analyzing several types of records from different channels simultaneously. Furthermore, aggregation summarizes data by combining two or more attributes or objects into a single one. For example, we might want to know, for a particular source IP address X, and a particular IP address Y, how many unique destination IP addresses were contacted in a specific time window Z. A high value of this measure could give an indication of IP mapping, which is a pre-attack inspection of the network [10]. Comparing to other data mining techniques, aggregation is generally more expensive in terms of computation as two or more attribute are involved. Nevertheless, efficiency can be improved by selecting appropriate aggregations of attributes and statistics. Regardless of using more fields, aggregation reduces the downstream volume of data which is helpful for analyzing large datasets [17, 18, 19].

4. Visualization

There is a common belief that automation is a complete problem solving solution that is able to counter all problems without human involvement. However, this is not the case in botnet detection. In this case, human analysts will always be needed to monitor that the automated system is performing as desired, to identify new botnet activities, and to analyze the more sophisticated malicious behaviors of botnets.

A data mining system for botnet detection will ideally offer various data visualizations to aid human analysts in identifying trends that are overlooked by automated methods, and also validate those identified by the automated system. Consequently, large abnormal changes, which are difficult to detect in an automated fashion, can be easily identified by an analyst. Additionally, such an analyst should quickly learn the visual patterns of certain types of anomalous activities which will be helpful for future detections [13].

Typical network traffic generates a huge amount of data for analysis. Determining if traffic is suspicious can take a significant amount of time and effort. Due to the number of networks and attacks represented with existing intrusion detection systems, it is often difficult to determine where legitimate problems are actually occurring. Several tools have been created to analyze and sift this data. These security tools often require a great deal of configuration in order to provide accurate alerts for malicious activity. In fact, it is desired to avoid information overload and alarm fatigue. Visualization can provide notification of attacks without distracting the user with huge volumes of data that can result in alarm fatigue. The visualization provides a quick view of the informational goal. In addition, details on demand allow the system administrator access to a deeper level of information regarding individual attack details [20].

5. Proposed Architecture

Mainly, we are targeted to create a *"Visual Threat Monitor"* which can detect botnet activities effectively. We propose a solution that combines data mining and visualization to enhance botnet traffic detection. Since visualizing the huge volume of raw data requires a large amount of time and efforts, in our solution we propose an optimized visualization technique that visualizes processed and selected data rather than raw data. We will apply data mining techniques iteratively to multiple log files to discover useful information about malicious botnet activities. The discovered information will be then visualized by grid visualization, scatter plot and histograms. These visualization techniques are easy to interpret and good for visualizing large datasets. The processing pipeline of the proposed visual threat monitor consists of correlation, statistical analysis, clustering, aggregation, and visualization.

Combination of data mining and visualization will assist the security personnel to recognize the threats more easily and effectively. Data mining is useful for knowledge discovery about malicious behavior of bots and finding evidence of botnet existence. On the other hand, the visualized information is easier to comprehend for human beings to gain useful knowledge. The proposed visual threat monitor will be highly useful to detect botnet activities. It will enhance the capability to detect, defend and respond to cyber-threats through botnets. The architecture of the proposed visual threat monitor is shown in Figure 1.

The core component of the system is the visualization module which represents the result passed through other layers of the system. The first layer is network flow layer that is responsible for capturing the trace packet data, draw

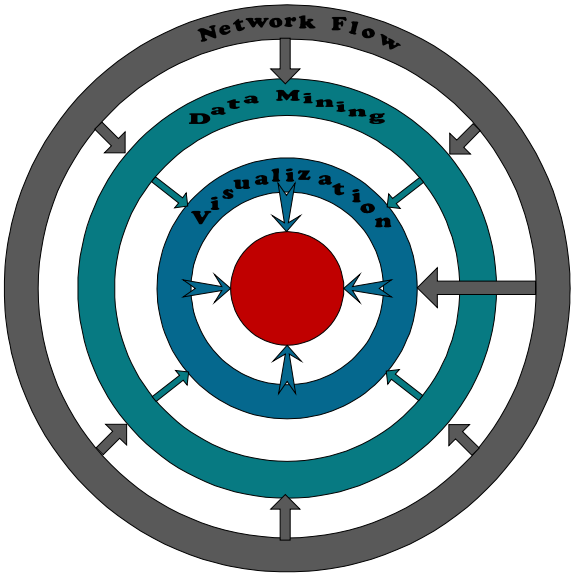

Figure. 1. Architecture of Proposed Visual Threat Monitor

new data according to time and size of packets, and store them in the data base. Next, we move to one layer deeper to discover knowledge from the database. We can use various data mining techniques including correlation, statistical analysis, clustering and aggregation to discover knowledge of malicious botnet activities. Each technique obtains different result based on different aspects of flow objects. The results will be passed to core component for visualization. As it is depicted in Figure 1, it is also possible to send the captured data in network flow layer directly to the visualization module to get an over view of the activities, without any drilling on the database. In the core layer, discovered information will be visualized by grid visualization, scatter plot and histograms. As mentioned earlier, these visualization techniques are easy to interpret and good for visualizing large datasets.

Since each layer in this architecture will be developed and maintained separately, any modification and update of layers can be done easily without affecting other layers. This provides a great flexibility and scalability for the system. For instance, if additional data is needed in future, we can modify the network flow layer to capture new series of attributes. Moreover, in data mining layer, the analyzer can define new functions to compare the objects and later be added to this layer as updates to the system. Accordingly, new methods of visualization can be declared to improve the overall capability of the system for threat visualization.

6. Conclusion

With the sharp rise in computer network attacks through botnets, current security monitoring tools will not be insufficient for efficient botnet traffic detection. Therefore, an effective tool that can detect malicious botnet traffic and alert the user is highly demanded. In order to quickly react to threats, notifications of malicious activities should be immediately sent to the security personnel. Security administrators are most interested to find out how the machines were compromised and how to protect the remainder of their networks from attacks.

The proposed flow based botnet detection system which is called *"Visual Threat Monitor"* tackles these issues by combining data mining and visualization. The anomalous data is passed to several trust models, and the flows are re-evaluated to obtain their trustfulness, which is then aggregated to detect malicious traffic via visualization. The visualized information will be analyzed by human intellectual and conceptual ability to gain useful knowledge about botnet activities for further precaution and as well as validation. The proposed visual threat monitor is a flexible and scalable approach for botnet traffic detection that can be adjusted according to future security demands.

7. Acknowledgment

The authors graciously acknowledge the funding from the Universiti Sains Malaysia (USM) through the USM Fellowship to Maryam Feily, as well as the support from the University of Malaya (UM) through the Postgraduate Grant to AliReza Shahrestani.

8. References

[1] T. Ferguson, "Botnets threaten the internet as we know it," ZDNet Australia, April 2008.

[2] M. Rajab, J. Zarfoss, F. Monrose, and A. Terzis, "A multifaceted approach to understanding the botnet phenomenon," in *Proc. 6th ACM SIGCOMM Conference on Internet Measurement (IMC'06)*, 2006, pp. 41–52.

[3] N. Ianelli, A. Hackworth, "Botnets as a vehicle for online crime," *CERT Request for Comments (RFC) 170*0, December 2005.

[4] M. M. Masud, T. Al-khateeb, L. Khan, B. Thuraisingham , and K. W. Hamlen, "Flow-based Identification of Botnet Traffic by Mining Multiple Log Files," in *Proc. International Conference on Distributed Frameworks & Applications (DFMA)*, Penang, Malaysia, October 21-22, 2008.

[5] W. Strayer, D. Lapsley, B. Walsh, and C. Livadas, *Botnet Detection Based on Network Behavior* ser. Advances in Information Security. Springer, 2008, PP. 1-24.

[6] Z. Zhu, G. Lu, Y. Chen, Z. J. Fu, P.Roberts, and K. Han, "Botnet Research Survey," in *Proc. 32nd Annual IEEE International Conference on Computer Software and Applications (COMPSAC '08)*, 2008, pp.967- 972.

[7] K. K. R. Choo, "Zombies and Botnets," Trends and issues in crime and criminal justice, no. 333, Australian Institute of Criminology, Canberra, March 2007.

[8] G. Schaffrath and B. Stiller, *Conceptual Integration of Flow-Based and Packet-Based Network Intrusion Detection* LNCS, Springer, 2008, PP. 190–194.

[9] H. Weststrate, "Botnet detection using netflow information Finding new botnets based on client connections" in *Proc. 10th Twente Student Conference on IT*, 2009.

[10] The MITRE Corporation, "Data Mining for Network Intrusion Detection: How to Get Started," [Online]. Available: http://www.mitre.org/work/tech_papers/tech_papers_01/bloedorn_datamining/bloedorn_datamining.pdf. [Accessed: March. 12, 2009].

[11] J. Goebel and T. Holz, "Rishi: Identify bot contaminated hosts by irc nickname evaluation," *in Proc. 1st Workshop on Hot Topics in Understanding Botnets*, 2007.
[12] W. Strayer, D. Lapsley, B. Walsh, and C. Livadas, Botnet Detection Based on Network Behavior, ser. Advances in Information Security. Springer, 2008, PP. 1-24.
[13] P. Dokas, L. Ertoz, V. Kumar, A. Lazarevic, J. Srivastava, P. Tan, "Data Mining Methods for Network Intrusion Detection" in *Proc. NSF Workshop on Next Generation Data Mining*, 2002.
[14] E.M. Knorr and R. T. Ng [1998]. "Algorithms for Mining Distance-Based Outliers in Large Datasets", in *Proc. 24th Int. Conference on Very Large Databases*, 1998, pp. 392-403.
[15] S. Ramaswarny, R. Rastogi, and K. Shim. "Efficient Algorithms for Mining Outliers from Large Data Sets", in *Proc. ACM Sigmod 2000 Int. Conference on Management of Data*, 2001.
[16] R. Johnson and P. Kuby, *Elementary Statistics*, Brooks-Cole, Toronto, Ontario, Canada, 2004.
[17] G. Gu, R. Perdisci, J. Zhang, and W. Lee, "Botminer: Clustering analysis of network traffic for protocol- and structure independent botnet detection," in *Proc. 17th USENIX Security Symposium*, 2008.
[18] C. Skorupka, , J. Tivel, L. Talbot, D. Debarr, W. Hill, E. Bloedorn, and A. Christiansen. "Surf the Flood: Reducing High-Volume Intrusion Detection Data by Automated Record Aggregation," in *Proc. SANS 2001 Technical Conference*, 2001.
[19] A. Giani, I. G. De Souza, V. Berk, and G. Cybenko, "Attribution and Aggregation of Network Flows for Security Analysis," White Paper. January 2008.
[20] J. E. Clark, C. P. Lee, R. Menon, and V. Rood, "HoneyTrap: Visualization for Monitoring Honeynets," [Online]. Available: http://idt.gatech.edu/~rmenon/newfolio/
files/honeytrap/honeytrap.pdf. [Accessed: March. 17, 2009].