Botnet Detection using Obscured Substantial Stochastic Modeling

Lav Kumar* , Dr. Pritaj Yadav**
*Department of Computer Science & Engineering
Rabindranath Tagore University, India
Email- kumarlav59@gmail.com
**Department of Computer Science & Engineering
Rabindranath Tagore University, Raisen, Madhya Pradesh, India
Email- pritaj.yadav@aisectuniversity.ac.in

Abstract-

It appears that you're describing the main objectives and approach of a thesis project focused on improving botnet detection methods. The thesis aims to address the limitations of traditional signature-based detection methods by proposing more effective solutions for detecting botnets, particularly HTTP botnets, on both Windows and Android platforms. The focus is on host-level detection of HTTP botnets, which often communicate using TCP connections. The thesis leverages data from the Management Information Base of the Simple Network Management Protocol (SNMP-MIB) to gather information about host changes.

The proposed approach involves using TCP connection-related SNMP-MIB variables as features to model system behavior at the host level. This modeling is achieved using an obscured semi-Stochastic Model (HsMM). The system conducts various experiments using HTTP botnets to validate the effectiveness of the proposed model. The results indicate that the model offers high detection accuracy, a low false-positive rate, real-time processing, and is lightweight in terms of computational resources..

Keywords-botnet, SNMP, HsMM

1. INTRODUCTION

The passage you've provided highlights the critical challenges posed by various types of attacks in today's computer and communication infrastructures. These attacks often involve the use of malicious software such as viruses, worms, and Trojan horses, which can lead to significant disruptions, data corruption, and wastage of network resources. Some attacks are even used to compromise Internet hosts and launch denial-of-service attacks, causing further harm to the network.

One method attackers use to facilitate these attacks is through the use of malware. When attackers gain access to network hosts, they can create what are called "bots." A bot is a compromised computer that an attacker can control remotely. The attackers orchestrate these compromised computers into a "botnet," which is essentially a network of controlled bots. The central control of these botnets is typically done by an intruder referred to as the "botmaster." The botmaster communicates with the bots using specialized Command and Control (C&C) channels.

This emergence of botnets as a new type of threat has significant implications for both individuals and the global network infrastructure. The growth of these botnets poses a substantial cybersecurity threat, as they can be used to carry out various malicious activities on a large scale. As a result, there is a pressing need to develop effective mechanisms to detect and counteract botnets.

2. PROPOSED MODEL

The comprehensive study of HTTP botnets, including Spyeye, Zeus, Athena, Blackenergy, and Andromeda, has revealed that certain SNMP-MIB variables undergo substantial changes during communication between an HTTP bot and its Command and Control (C&C) server. These changes are significant when the system is controlled by a botmaster, indicating a departure from normal operating conditions. To capture this behavior effectively, an obscured semi-Stochastic Model (HsMM) is proposed.

The HsMM leverages SNMP-MIB variables as observed symbols, allowing it to model the system's behavior accurately. In the training phase, SNMP-MIB variables are transformed into HsMM observation sequences using the forward-backward training algorithm. Subsequently, the HsMM is constructed based on these sequences. During the testing phase, SNMP-MIB variables are again converted into HsMM observation sequences. The HsMM is then utilized to calculate the probability of each test sequence, from which the Average Log Likelihood (ALL) is determined. This value serves as the basis for classifying incoming network traffic as either normal or indicative of botnet communication.

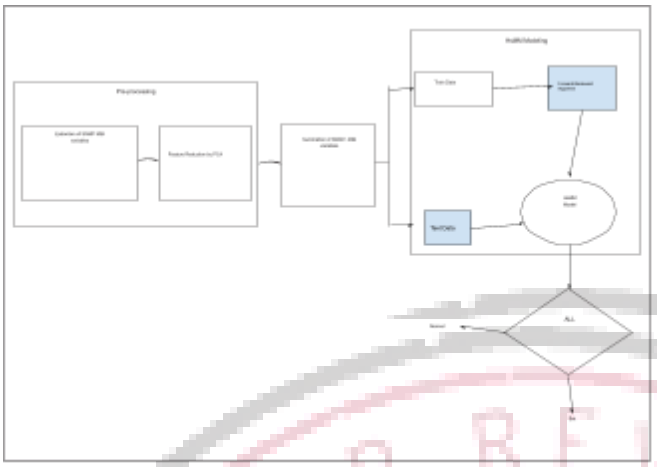

Figure 1 Block diagram of the proposed HsMM model

3. Feature Extraction

HTTP botnet sends a large number of HTTP request to the target C&C server automatically. These HTTP requests have legitimate formats and are sent via normal TCP connections and hence these requests are difficult to identify. In addition, the bots request usually is generated randomly or by repeating a few simple HTTP requests via TCP connections. Hence, in this work, we have extracted TCP connection related MIB variables for the effective detection of botnet without analyzing the packets. These MIB variables are collected through SNMP running on the host.

The study encompasses eight TCP connection-related MIB variables: tcpActiveOpens, tcpPassiveOpens, tcpAttemptFails, tcpEstabResets, tcpCurrEstab, tcpInSegs, tcpRetransSegs, and tcpInErrs. Each SNMP-MIB variable serves a distinct purpose. Specifically, tcpActiveOpens tallies instances of TCP connections transitioning directly from the CLOSED to the SYN-SENT state; tcpPassiveOpens counts transitions from the LISTEN to SYN-RCVD state; tcpAttemptFails tracks direct shifts to the CLOSED state from SYN-SENT or SYN-RCVD, including transitions from SYN-RCVD to LISTEN state; tcpEstabResets quantifies direct transitions from ESTABLISHED or CLOSE-WAIT to CLOSED state; tcpCurrEstab reveals the count of connections in ESTABLISHED or CLOSE-WAIT state; tcpInSegs represents the total segments received, including erroneous ones; tcpRetransSegs enumerates retransmitted segments containing previously transmitted data; and tcpInErrs captures segments received in error, such as those with flawed TCP checksums. These MIB variables collectively provide a comprehensive insight into various aspects of TCP behavior, aiding in effective detection and analysis of botnet activity).

The collected MIB variables undergo Principal Component Analysis (PCA) to identify the significant SNMP-MIB variables. PCA, a widely utilized dimensionality reduction technique (Li et al., 2012), transforms numerous variables into fewer uncorrelated ones by determining orthogonal linear combinations of the original variables that maximize variance. Post-PCA, the following four key MIB variables are identified: tcpActiveOpens, tcpPassiveOpens, tcpCurrEstab, and tcpInSegs.

Subsequent to exhaustive experimentation involving both normal and botnet communication traffic, it was determined that the summation of these selected MIB variables (referred to as SUM-MIB) at various time points yields intriguing outcomes. This summation approach is then adopted for further analysis, showcasing its effectiveness in distinguishing between different types of network traffic.

Reference: Li, J., Cheng, H., Hu, Y., & Jiang, W. (2012). A survey of network flow applications in intrusion detection and prevention. Journal of Network and Computer Applications, 36(2), 611-622

SUM-MIB = tcpActiveOpens + tcpPassiveOpens
+ tcpCurrEstab + tcpInSegs

4. Obscured semi-Stochastic Model

HsMM is an extension of HMM by allowing the underlying process to be a semi-Stochastic chain with a variable duration or sojourn time for each state. The important difference between HMM and HsMM is that one observation per state is assumed in HMM while in HsMM each state can emit a sequence of observations and the number of observations produced while in a state i is determined by the duration d which is the time spent in the state i as shown in Figure 2.

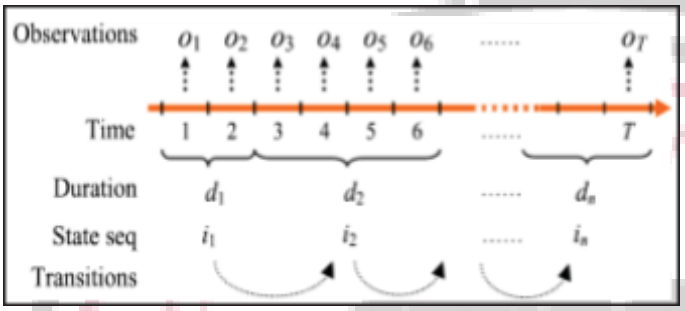

Figure 2 General structure of the HsMM (Source: Yu 2010).

5. Result and Discussion

The feature extraction component is implemented using java and SNMP setup tool is installed in the system. With the help of these components, MIB variables are collected for 12 hrs/day for seven days from the zombie machines. These MIB variables are continuously updated from the outgoing/incoming traffic. Normal traces are also collected during the system's normal activities which include web service, e-mail service, FTP service and remote service and the corresponding MIB variables are collected for 12 hrs/day for ten days from National Knowledge Network (NKN) and from our research lab network. Table 1 shows the details of the datasets.

Table 1 Description of the MIB datasets

Botnet MIB traces			
Botnet	MIB Trace size	Botnet	MIB Trace size
Spyeye	1.25 GB	Athena	2.73 GB
Blackenergy	2.96 GB	Andromeda	4.59 GB
Zeus	2.57 GB		
Normal MIB traces			
FTP service	4.95 GB	Web service	4.27 GB
E-mail service	3.28 GB	Remote service	2.90 GB

Figures from 2 to 5 show the variation of MIB variables during normal and botnet communications. All the figures are plotted at 30 seconds time intervals in x-axis.

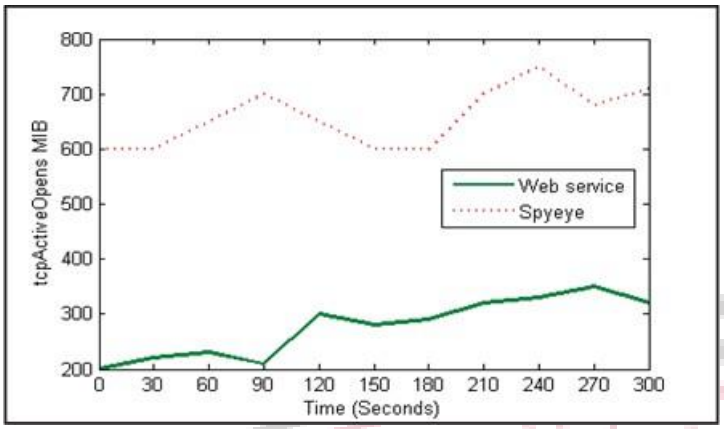

Figure 2 tcpActiveOpens MIB in web service and Spyeye

The above figure shows that there is a significant change in the tcpActiveOpens MIB variable of Spyeye botnet when compared to that if the normal web service.

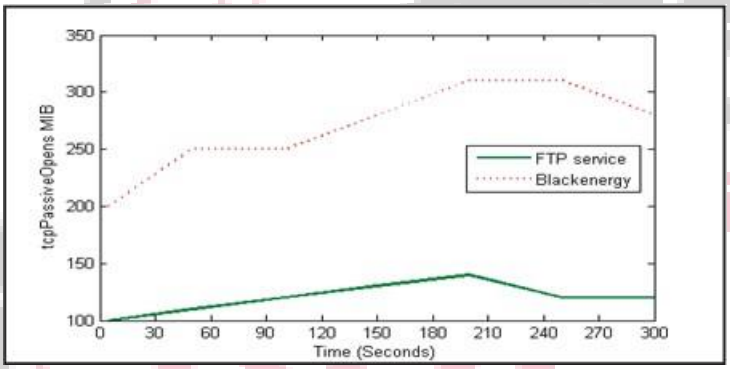

Figure 3 tcpPassiveOpens MIB in FTP service and Blackenergy

From the above figure, we can infer that, there is a significant change in the tcpPassiveOpens MIB variable values of Blackenergy botnet when compared to the values of normal FTP service tcpPassiveOpens MIB variable. During the initial stage of bot propagation, the zombie machine tried to communicate with the remote location to get the address of the C&C server or information from the botmaster. Due to this behavior, there is a significant change in the tcpPassiveOpens MIB variable.

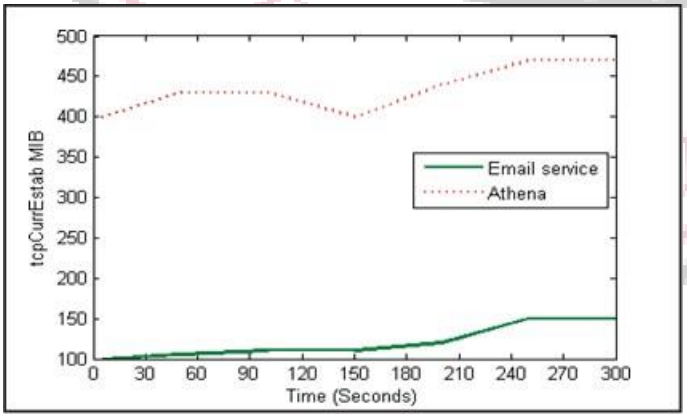

Figure 4 tcpCurrEstab MIB in Email service and Athena

Figure 4 shows the tcpCurrEstab MIB variable comparison of Email service and Athena HTTP botnet. From the figure, one can infer that the tcpCurrEstab MIB variable changes significantly during the Athena HTTP bot propagation.

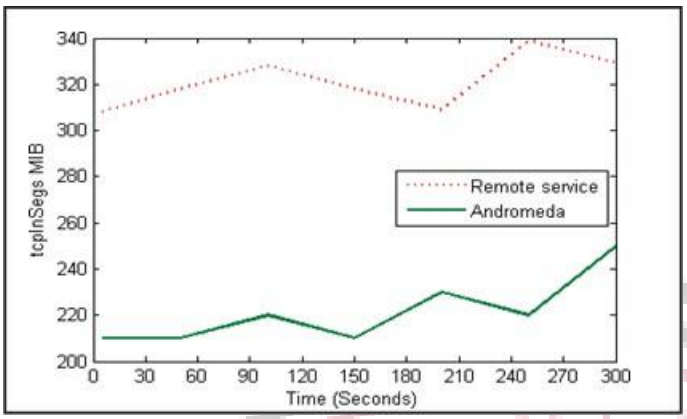

Figure 5 tcpInSegs MIB in remote service and Andromeda

The Figure 5 shows the tcpInSegs MIB variable comparison of Remote service and Andromeda HTTP botnet. From the figure, one can see that there is a significant change in the tcpInSegs MIB variable during the Andromeda HTTP bot propagation.

Web botnets do not maintain a connection with C&C server, but they periodically download the instructions using web requests on a normal interval. On the other hand, all actions of the botnet are driven by the machine and are irrelevant to the user behavior. In the bot propagation scenario, the observation sequence Ot represents the summation of selected SNMP-MIB variables count calculated during the tth second in the current state at the host machine. Here the observation period is 30 minutes and SNMP-MIB variables summation is calculated at every 30s, when the significant changes occurred in those variables in the current state. The 30s is identified as response time from the botmaster through the experimental analysis and from the study of Akiyama et al. (2007).

After extracting the significant SNMP-MIB variables from the normal and bot propagation systems, an HsMM model is trained using 70% of the both normal and botnet datasets. For example, first we trained the HsMM model with 70% of the web service and Spyeye dataset. Then the trained HsMM model is tested with the untrained Spyeye dataset. Similarly the experiments are performed with other datasets also. Accuracy is used as the metric for evaluating the proposed model. Table 2 shows the results of detection accuracy, which is really high with low false positive rate.

After obtaining the parameters of the HsMM, we first calculate the ALL of observation sequences of the normal traffic, and then the ALL of botnet observation sequences. It is found that the 5% level of confidence interval for ALL when the system is normal is [–2.5, –0.7] and ALL lies in the 5% confidence interval

[–7.5, –4.0] when the system is communicating with C&C server.

Table 2 Performance of the proposed HsMM model

Datasets	FPR	Detection accuracy	Results
Web service	0%	100%	Normal
FTP service	0%	100%	Normal
Spyeye	1.67%	98.14%	Botnet
Blackenergy	1.58%	98.72%	Botnet
Zeus	1.75%	98.02%	Botnet
Athena	1.29%	98.94%	Botnet
Andromeda	1.47%	98.62%	Botnet

We have tested our model with various real botnets and normal data in a windows machine having configurations Intel (R) Core™ i5 3317U, 1.70 GHz, and with the operating system Microsoft Windows 2000. We have used an open source statistical package – R (Statistical Package, R 2009) for evaluating the performance of the proposed model. R is a language and environment for statistical computing and graphics and it provides a wide variety of statistical techniques such as linear and nonlinear modeling, classical statistical tests, time-series analysis, classification, clustering, and graphical techniques, etc. Also it is highly extensible.

4.1 Running Time

To evaluate the efficiency of the HsMM model in terms of detection time, we conducted tests on a system configured as described above. The results indicate that the HsMM model exhibits swift performance in classifying various test instances. Specifically, the model demonstrates remarkable speed, taking only 2.05 seconds, 4.10 seconds, 2.51 seconds, 2.95 seconds, 2.73 seconds, 2.84 seconds, and 3.18 seconds to classify test instances from web service, FTP service, Spyeye, Blackenergy, Zeus, Athena, and Andromeda MIB datasets, respectively. These outcomes underscore the HsMM model's exceptional accuracy and efficiency, as it achieves optimal accuracy levels while requiring the shortest detection times for classification.

6. CONCLUSION

The proliferation and diversity of botnets have sparked a renewed urgency to develop effective botnet detection solutions. The exponential rise in the utilization of bots for malicious purposes over recent years has inflicted significant challenges on both Internet infrastructure and users. This research isdriven by the motivation to delve into this realm and provide a viable remedy to counter this evolving threat landscape. The study involves a comprehensive examination of diverse botnets, leading to the formulation of host-based and network-based detection systems tailored for HTTP botnets. Additionally, a generalized botnet detection system is proposed, independent of botnets' command and control structures. Furthermore, this research introduces an Android botnet detection system that employs structural analysis of Android APK files coupled with machine learning techniques.

The primary focus is on dissecting these botnets to comprehend their behavior and lifecycle mechanisms. Various categories of botnets are meticulously examined within a controlled environment to ascertain their potential attack vectors, topologies, command and control structures, and communication mechanisms. This chapter also offers a concise survey of existing botnet detection techniques, outlining their merits and limitations, to illuminate the imperative for novel detection mechanisms that address the evolving challenges posed by botnet proliferation

7. REFERENCES

1. Abraham, S & Chengalur-Smith, I 2019, ‘An overview of social engineering malware: Trends, tactics, and implications’, Technology in Society, vol. 32, no. 3, pp. 183-196.
2. Abu Rajab, M, Zarfoss, J, Monrose, F & Terzis, A 2018, ‘A multifaceted approach to understanding the botnet phenomenon’, Proceedings of the sixth ACM SIGCOMM conference on Internet measurement, pp. 41-52.
3. Akamai Security Blog, 2014. Botnet distributions. Available from: <https://akamaisecurity.blog/botnet_distributions/report.pdf> [25 February 2020].
4. Akiyama, M, Kawamoto, T, Shimamura, M, Yokoyama, T, Kadobayashi, Y & Yamaguchi, S 2017, ‘A proposal of metrics for botnet detection based on its cooperative behavior’, Proceedings of the IEEE International Symposium on Applications and the Internet Workshops,pp. 82-82.
5. Al-Duwairi, B, Al-Qudahy, Z & Govindarasu, M 2022 ‘A novel scheme for mitigating botnet-based DDoS attacks’, Journal of Networks, vol. 8, no. 2, pp. 297-306.
6. Alomari, E, Manickam, S, Gupta, BB, Karuppayah, S & Alfaris, R 2022, ‘Botnet-based distributed denial of service (DDoS) attacks on web servers: classification and art’, International Journal of Computer Applications, vol. 49, no. 7, pp. 24-32.
7. Anagnostopoulos, M, Kambourakis, G & Gritzalis, S 2017, ‘New facets of mobile botnet: architecture and evaluation’, International Journal of Information Security, pp. 1-19.
8. Android Market web store, Android Applications, 2016, Available from: <http://www.androidcentral.com/android-market-webstore-now-online> [14 December 2015].
9. Arbor networks - DDoS mitigation, advanced threat and APT, 2014. Available from: <https://www.arbornetworks.com/DDoS-attacks-report.html>[10 June 2014].

10. Arp, D, Spreitzenbarth, M, Hubner, M, Gascon, H & Rieck, K 2020, "DREBIN: Effective and Explainable Detection of Android Malware in Your Pocket", Proceedings of the twentieth Annual Network & Distributed System Security Symposium (NDSS), pp. 1-15.
11. Ars Technica - technology news and information 2012, DDoS Attacks. Available from: http://arstechnica.com/security/2012/10/ddos-attacks-against-major-us-banks-no-stuxnet/ [10 August 2012].
12. Arslan, B, Gunduz, S & Sagiroglu, S 2016, 'A review on mobile threats and machine learning based detection approaches', Proceedings of the fourth IEEE International Symposium on Digital Forensic and Security (ISDFS), pp. 7-13.
13. Asia cnet 2014, Hacked fridge part of botnet attack. Available from: http://asia.cnet.com/hacked-fridge-part-of-botnet-that-sent-750000-spam-emails-62223486.htm. [12 February 2014].
14. Baecher, P, Koetter, M, Holz, T, Dornseif, M & Freiling, F 2018, ' The nepenthes platform: An efficient approach to collect malware', Proceedings of the Springer international conference on Recent Advances in Intrusion Detection, pp. 165-184.
15. Barford, P & Yegneswaran, V 2017, 'An inside look at botnets', Proceedings of the springer international conference on Malware Detection, pp. 171-191.
16. Barrera, D, Kayacik, HG, van Oorschot, PC & Somayaji, A 2010, 'A methodology for empirical analysis of permission-based security
17. models and its application to android', Proceedings of the seventeenth ACM conference on Computer and communications security,
18. pp. 73-84.
19. Binkley, JR & Singh, S 2006, 'An Algorithm for Anomaly-based Botnet Detection', Proceedings of the international conference of SRUTI, pp.10-17.
20. Bouckaert, RR, Frank, E, Hall, M, Kirkby, R, Reutemann, P, Seewald, A & Scuse, D 2015, 'WEKA manual for version 3-7-12'.
21. Boutet, J 2011, 'Malicious Android Applications: Risks and Exploitation-A Spyware Story about Android Application and Reverse Engineering'
22. Buczak, AL & Guven, E 2015, 'A survey of data mining and machine learning methods for cyber security intrusion detection', IEEE Communications Surveys & Tutorials, vol.18, no.2, pp.1153-1176.