JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Determine Efficient and Effective Botnet Detection with Adaptive Traffic Sampling

1. Hari Krishna Deevi , 2 .Dr.B,Bhanu Prakesh

1.Research Scholar, Department of Computer Science and Engineering, Sri Satya Sai University of Technology and Medical Sciences, Sehore, Madhya Pradesh, India.

2.Professor,Department of Computer Science & Engineering, KKR & KSR Institute of Technology & Science ,Guntur, A.P

ABSTRACT

With the advent of the Internet, users have access to a broad variety of distant services in the distributed computing environment. However, there are a number of security concerns that threaten the integrity of data transmission on the distributed computing platform. For instance, harmful code is just one facet of the Internet security danger posed by the botnet issue. Distributed denial of service (DDoS) assaults, click fraud, phishing, virus distribution, spam emails, and machine construction for illegal information/material exchange are all enabled by the botnet phenomena. Consequently, it is crucial to provide a reliable system for enhancing botnet identification, analysis, and eradication. In this study, we offer a technique for quickly and accurately pinpointing a handful of potentially malicious sites that are almost certainly bots. For more precise and detailed botnet identification, their communications can be sent to DPI-based systems. Our technique drastically reduces the amount of network traffic subject to DPI by employing a unique adaptive packet sampling algorithm and a scalable spatial-temporal flow correlation approach, hence increasing the scalability of existing botnet detection systems.

Keywords: Internet, Botnet, Sampling, Detection, Network security

I. INTRODUCTION

Users are increasingly drawn to distributed computing as a result of recent advancements in communication and computer technology. Distributed services are widely used on the Internet, such as e-mail, web applications, and voice over IP applications; nevertheless, malicious software has acquired a significant role in the developing distributed computing models. Malicious software has been there from the earliest days of programmable computers, although its reach is usually localised or small. The proliferation of the Internet in recent years has served as a reporting baseline for global malware infestations that have affected millions of devices. This has led to the rise in popularity of botnets, or networks of compromised computers operated remotely. The primary objective of these decentralised coordinated networks is to launch DoS assaults, but they also engage in other forms of network-based cybercrime including phishing, click fraud, spam production, copyright infringement, key logging, and so on. It is well acknowledged that botnets pose a significant risk to online infrastructure.

Over the last decade, the botnet has evolved into a highly dangerous phenomena, demonstrating its negative impact on online communities. In an effort to counteract this menace, researchers, law enforcement agencies, corporations, and private citizens have begun to develop strategies. This problem of spotting botnets is still a work in progress for security analysts and businesses. Since all parts of botnets—their detection, mitigation, and response—evolve over time, no single method of either can provide a long-term fix. Similarly, businesses, governments, networks, and ISPs all approach the problem of botnets in various ways and with different ends in mind. In addition, as time goes on and more information becomes available, botmasters get more adept at avoiding detection by botnet detection methods and at rallying sophistication for the C&C architecture.

II. EVALUATION

We built a prototype system and tested it with data collected from actual networks and various botnets. The results demonstrate that Flow-Capture is capable of greatly increasing the sample rate for packets associated with botnets above the baseline. We compared B-Sampling to FlexSample and found that, in terms of sampling rate for botnet packets and FlowCorrelation detection rate, B-Sampling performed better than FlexSample. With only a small sample of compromised hosts, cross-epoch correlation can reliably and quickly identify bots. By narrowing its focus to packets from a small subset of potentially malicious sites, the fine-grained detector is able to achieve a high detection rate with a negligible false positive rate.

Our data was collected by attaching our monitors to a span port that was a mirror image of a backbone router on the university's network (200-300 Mbps at peak hours). We think this sort of traffic gives useful traces for evaluating our system because it spans a wide range of applications. Table 1 of the dataset includes continuous 3.5 days of TCP and UDP headers and 1.5 hours of complete packets. Since dynamic IPs assigned to wireless connections are regularly reallocated and hence cannot represent the same hosts over successive epochs, we decided to remove a B/16 subnet specifically for them. In the course of the three and a half day study, we saw 1,460 unique IP addresses. Full payload traces were also gathered, lasting for a total of 1.5 hours.

Experiment Setup and Data Collection

Table 1: Background Traces

Trace	Dur	Bots
Bot-IRC-A	4days	3
Bot-IRC-B	4days	4
Bot-HTTP-A	4days	3
Bot-HTTP-B	4days	4
Bot-HTTP-C	4days	4
Bot-P2P-Storm	4days	2
Bot-P2P-Waledac	4days	3

We gathered evidence from seven distinct botnets, whose protocols ranged from IRC to HTTP to P2P. Running bot instances ("TR/Agent.1199508.A" and "Swizzor.gen.c") on several hosts in the honeypot allowed us to gather both Bot-IRC-A and Bot-HTTP-A. Rubot is a botnet emulation framework that was used to create Bot-IRC-B and Bot-HTTPB/C. In Bot-HTTP-B, the interval between C&C server contacts was set to 10 minutes. As with Bot-HTTP-C, the bots were able to make covert communications with the C&C server by varying the time between each visit from 0 and 10 minutes. On hearing the "scan" instruction, they both launched a scanning attack. In comparison to Bot-IRC-B, Bot-IRC-A has substantially greater C&C flows because its bots send packets to the C&C server at regular intervals during the IRC session. By executing binaries in a sandbox, we were able to gather traces of two P2P-based botnets: Storm and Waledac.

We layered 3.5 days' worth of botnet traces over the traces from a random sample of client IPs in the college network after adjusting the timestamps of each botnet packet to match the time of the first packet in the background traces. Since we counted one epoch E as 12 hours, there are a total of seven time intervals. The filter includes the most popular local domain name servers (DNS), college email servers, and the IP ranges of well-known service networks and content distribution networks (such as MICROSOFT, GOOGLE, YAHOO, SUN, etc.) that are not likely to be used as Botnet C&Cs. The filter also includes the IP ranges of the top 10,000 Alexa-ranked domains (corresponding to 12230 IPs).

Evaluation of Sampling Algorithm

B-Sample was tested utilising a variety of traces with varying goal sampling speeds (0.01, 0.025, 0.05, 0.075 and 0.1). We looked at how B-Sampling stacked up against FlexSample, a cutting-edge sampling method that can be tweaked with different "conditions" for various applications. FlexSample allocated the vast majority of its funds to packets associated to "servers with high in degree of tiny flows" in order to collect botnet packets, meeting a certain requirement. However, the "high fan-in" characteristic may not hold and the botnet packets may be missed in practice because of the very limited number of infected devices.

In Table 2, we can see the results of B-Sampling and FlexSampling on the mixed dataset in terms of the overall sampling rates and the sampling rates for botnet-related packets. In the first column, labelled "SRT," we present the various aimed-for sampling rates we tried. The actual total sampling rates attained by B-Sampling and FlexSample are displayed in the second column (SRActual, B) and the third column (SRActual, Flex). Both B-Sample and FlexSample achieve very near approximations of the desired sampling rate in practise. Following this, we "zoom in" on the sampled packets and assess the real sampling rates for packets of each botnet, reporting the results in the remaining columns.

Table 2: Sampling Rate

SRT	SRActual		SRIRC−A/B		SRHT T P −A/B/C		SRStorm		SRWaledac	
	B-	Flex	B-	Flex	B-	Flex	B-	Flex	B-	Flex
0.01	0.012	0.01	0.65/0.68	0.001/0.07	0.55/0.69/0.68	0.06/0.07/0.06	0.02	0.05	0.02	0.07
0.025	0.027	0.025	0.93/0.92	0.002/0.16	0.72/0.93/0.93	0.16/0.17/0.16	0.16	0.11	0.18	0.16
0.05	0.052	0.05	0.96/0.96	0.004/0.32	0.74/0.96/0.96	0.32/0.35/0.33	0.48	0.23	0.48	0.33
0.075	0.076	0.075	0.97/0.97	0.006/0.48	0.75/0.97/0.97	0.50/0.50/0.48	0.72	0.33	0.7	0.48
0.1	0.1	0.1	0.98/0.98	0.008/0.6	0.76/0.98/0.98	0.6/0.64/0.61	0.83	0.41	0.81	0.61

Evaluation of Flow Correlation

Using the combined traces, we tested the B-Sampling method for cross-epoch correlation for two qualities: detection accuracy and scalability. Since M = N/2 (N = 7, M = 3), we suspect any pair of hosts that exhibit highly similar communication patterns for at least three of the seven epochs.

Table 3 illustrates the percentage of bots (23) and noises (1460) detected by Flow-Correlation using B-Sampling, given SRT and PerExp, in each cell. The data demonstrates that a high detection rate may be attained with a low PerExp using Flow-Correlation. For all the SRTs under evaluation, for instance, Flow-Correlation can reliably detect all the bots with a PerExp of 5%. Yet even at extremely low PerExp (2% and 3%, for example), far over half of the bots were still taken.

Table 3: Detection Rates of Cross-Epoch Correlation using B-Sampling

SRT	For each PerExp , TP(bots/23), FP(noises/1460)									
	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09	0.1
0.01	48%, 0.1%	83%, 0.5%	96%, 1%	96%, 2%	100%, 3%	100%, 4%	100%, 5%	100%, 6%	100%, 6%	100%, 8%
0.025	52%, 0%	87%, 0.5%	100%, 1%	100%, 2%	100%, 3%	100%, 4%	100%, 5%	100%, 6%	100%, 7%	100%, 8%

0.05	48%, 0.1%	100%, 0.3%	100%, 1%	100%, 2%	100%, 3%	100%, 4%	100%, 5%	100%, 5%	100%, 7%	100%, 7%
0.075	48%, 0.2%	100%, 0.3%	100%, 1%	100%, 2%	100%, 3%	100%, 4%	100%, 5%	100%, 6%	100%, 7%	100%, 8%
0.1	39%, 0.3%	78%, 0.8%	100%, 1%	100%, 2%	100%, 3%	100%, 3%	100%, 5%	100%, 5%	100%, 7%	100%, 8%
1	30%, 0.5%	65%, 0.8%	96%, 1%	100%, 2%	100%, 3%	100%, 4%	100%, 5%	100%, 5%	100%, 7%	100%, 8%

III. BOTNET DETECTION

All packets connected to malicious IP addresses identified by Flow-Correlation are examined by the fine-grained botnet detector. We tested the effectiveness and detection rate of the fine-grained detector using 1.5-hour traces combined with botnet traces.

Our detector's "IRC Message Correlation" section identified bots in Bot-IRC-A/B by comparing the content of user messages. The "Correlation" module was able to identify more robots. When scanning the local network, for instance, Bots in Bot-HTTP-B/C set off alarms. Bot-HTTPA bots send out warnings whenever they try to access new versions of servers. When Storm and Waledac find new peers, they send out notifications. In order to identify these bots, we used Flow-Correlation to compare suspicious activity/alerts with pairs of IP addresses known to be associated with them. The fine-grained detector's detection and false positive rates for a variety of SRTs and PerExps are shown in Table 4.

Table 4: Detection Rates of Fine-Grained Detectors

SRT	**For each PerExp , TP(bots/23), FP(noises/1460)**									
	0.01	**0.02**	**0.03**	**0.04**	**0.05**	**0.06**	**0.07**	**0.08**	**0.09**	**0.1**
0.01	48%, 0	83%, 0	96%, 0	96%, 0	100%, 0	100%, 0	100%, 0	100%, 0	100%, 0	100%, 0
0.025	52%, 0	87%, 0	100%, 0	100%, 0	100%, 0	100%, 0	100%, 0	100%, 0	100%, 0	100%, 0
0.05	48%, 0	100%, 0	100%, 0	100%, 0	100%, 0	100%, 0	100%, 0	100%, 0	100%, 0	100%, 0
0.075	48%, 0	100%, 0	100%, 0	100%, 0	100%, 0	100%, 0	100%, 0	100%, 0	100%, 0	100%, 0
0.1	39%, 0	78%, 0	100%, 0	100%, 0	100%, 0	100%, 0	100%, 0	100%, 0	100%, 0	100%, 0
1	30%, 0	65%, 0	96%, 0	100%, 0	100%, 0	100%, 0	100%, 0	100%, 0	100%, 0	100%, 0

Table 5 shows a comparison of the fine-grained detector's performance under two conditions: I the detector is applied directly; and ii) the detector is applied using Flow- Correlation and B-Sampling (PerExp = 0.05 and M = 3). Real-time workload reduction is dramatically aided by the use of Flow-

Correlation, a fine-grained detector that cuts down on off-line trace processing time by 95%.

Table 5: Performance of Fine-Grained Detector

	With Flow-Corr (PerE = 5%, M = 3)						direct
SRT	0.01	0.025	0.05	0.075	0.1	1	
Per of Pkts	1.7 %	2.9 %	2.1 %	3%	4.3 %	2 %	100 %
Time	33s	39s	35s	40s	49s	33 s	858s

IV. CONCLUSION

In this research, we present a method for addressing this issue by combining a scalable spatial temporal flow correlation strategy with an adaptive packet sampling technique that takes botnets into account. The adaptive packet sampling method captures more packets associated with bots by using network features of botnet C&Cs to adaptively alter the sampling probabilities while maintaining a desired sampling rate. The flow correlation method takes use of the core characteristics of botnets to uncover bots by singling out servers that exhibit always identical patterns of network traffic. Real-world network traces were used in an evaluation to demonstrate the effectiveness of our approach. The sampling method surpasses state-of-the-art adaptive sampling techniques and is able to capture more botnet packets at a given sampling rate than the baseline. It has been shown that a correlation algorithm fed with packet samples can reliably and flexibly identify different kinds of bots (including IRCbased, HTTP-based, and P2P-based). The method will improve the performance of fine-grained botnet detectors by allowing them to operate on ever-faster networks by focusing on evaluating packets comprising a smaller quantity of questionable data.

REFERENCES: -

1. M. Pour, A. Mangino, K. Friday, M. Rathbun, E. Bou-Harb, F. Iqbal, S. Samtani, J. Crichigno, and N. Ghani, “On Data-driven Curation, Learning, and Analysis for Inferring Evolving Internet-of-Things (IoT) Botnets in the Wild,” Computers & Security, vol. 91, April 2020

2. R. Vishwakarma and A. Jain, “A Honeypot with Machine Learning based Detection Framework for defending IoT based Botnet DDoS Attacks,” International Conference on Trends in Electronics and Informatics (ICOEI), Tirunelveli, India, pp. 1019-1024, 2019.

3. A. Guerra-Manzanares, H. Bahsi and S. Nõmm, “Hybrid Feature Selection Models for Machine Learning Based Botnet Detection in IoT Networks,” International Conference on Cyberworlds (CW), Kyoto, Japan, pp. 324-327, 2019.

4. X. Dong, J. Hu and Y. Cui, “Overview of Botnet Detection Based on Machine Learning,” International Conference on Mechanical, Control and Computer Engineering, Huhhot, pp. 476-479, 2018.

5. J. Jin, Z. Yan, G. Geng and B. Yan, “Botnet Domain Name Detection based on machine learning,” International Conference on Wireless, Mobile and Multi-Media (ICWMMN), Beijing, China, pp. 273-276, 2015.

6. A. Feizollah, N. B. Anuar, R. Salleh, F. Amalina, R. Ma’arof, and S. Shamshirband, “A Study Of Machine Learning Classifiers for Anomaly-Based Mobile Botnet Detection,” Malaysian Journal of Computer Science, 26(4), 2013.

7. S. Saad et al., “Detecting P2P botnets through network behavior analysis and machine learning,” International Conference on Privacy, Security and Trust, Montreal, QC, pp. 174-180, 2011.

8. G. Gu, J. Zhang, and W. Lee. Botsniffer: Detecting botnet command and control channels in network traffic. In Proc. NDSS, 2008.

9. G. Gu, R. Perdisci, J. Zhang, and W. Lee. Botminer: Clustering analysis of network traffic for protocol- and structure-independent botnet detection. In Proc. USENIX Security, 2008.