International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2016 IJSRCSEIT | Volume 1 | Issue 1 | ISSN : 2456-3307

Detecting BOT Victim in Client Networks

Abinaya. E, **Balamurugan. K**

Department of Information Technology, St Peter Engineering College, Avadi, Tamil Nadu, India

ABSTRACT

In this paper we discuss my research in detecting bot victim in client networks. Botnets are collections of Internet hosts ("bots") that, through malware infection, have fallen under the control of a single entity ("botmaster"). Botnets perform network scanning for different reasons: propagation, enumeration, penetration. One common type of scanning, called "horizontal scanning," systematically probes the same protocol port across a given range of IP addresses, sometimes selecting random IP addresses as targets. To infect new hosts in order to recruit them as bots, some botnets, e.g., Conficker perform a horizontal scan continuously using self-propagating worm code that exploits a known system vulnerability. In this project, we focus on a different type of botnet scan—one performed under the explicit command and control of the botmaster, occurring over a well-delimited interval.

Keywords: Horizontal Scanning, Botmaster, Bots, P2P, IRC, BotGraph, DPI, Clustering

I. INTRODUCTION

Existing system contains a fundamental disadvantage of centralized C&C servers are that they represent a single point of failure. In order to overcome this problem, botmasters have recently started to build botnets with a more resilient C&C architecture, using a peer-to-peer (P2P) structure or hybrid P2P/centralized C&C structures. Detecting botnets is of great importance. However, designing an effective P2P-botnet detection system is faced with several challenges. I am confident that this software package can be readily used by non-programming personal avoiding human handled chance of error.

Peer-to-peer (P2P) botnets have a random organization and operate without a C&C server. Bot software maintains a list of trusted computers (including other infected machines), information drop locations and locations where the machines can update their malware. More advanced botnets use encryption in order to hide communications between bots.

The purpose of decentralization is to help evade detection and make it harder for security researchers to access communications than is the case with a conventional botnet topology. The lack of a command-and-control server makes it less likely that detection of a single bot can lead to investigators taking down the entire network.

II. METHODS AND MATERIAL

2. Related Works

Botnets have been an active area of research for almost a decade, starting with early generation botnets that used IRC channels to implement centralized Command & Control (C&C) infrastructures.

Botnets commonly scan large segments of Internet address space, seeking hosts to either infect or compromise, or for the purpose of network mapping and service discovery. Analyzing and detecting these events can improve our understanding of evolving botnet characteristics and spreading techniques, our ability to distinguish them from benign traffic sources, and our ability to mitigate attacks.

Since Sality is one of the largest known botnets but relatively undocumented in research literature, another contribution of our study is to shed light on the scanning behavior of this new-generation botnet.

2.1 P2P as botnet command and control: a deeper insight

The research community is now focusing on the integration of peer-to-peer (P2P) concepts as incremental improvements to distributed malicious software networks (now generically referred to as botnets). While much research exists in the field of P2P in terms of protocols, scalability, and availability of content in P2P file sharing networks, less exists (until this last year) in terms of the shift in C&C from central C&C using clear-text protocols, such as IRC and HTTP, to distributed mechanisms for C&C where the botnet becomes the C&C, and is resilient to attempts to mitigate it. In this paper we review some of the recent work in understanding the newest botnets that employ P2P technology to increase their survivability, and to conceal the identities of their operators. We extend work done to date in explaining some of the features of the Nugache P2P botnet, and compare how current proposals for dealing with P2P botnets would or would not affect a pure-P2P botnet like Nugache. Our findings are based on a comprehensive 2-year study of this botnet.

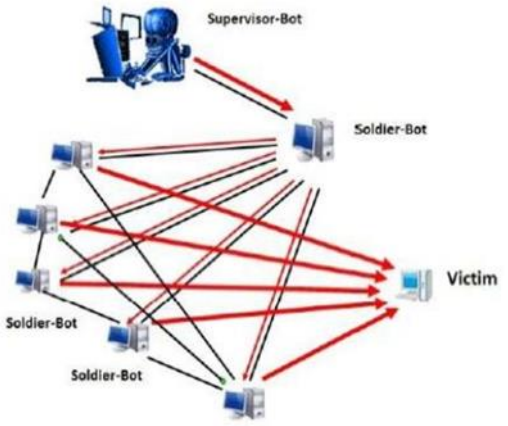


Figure 1. Structure of the Botnet

2.2 Experiences in Malware Binary DE obfuscation

Malware authors employ a myriad of evasion techniques to impede automated reverse engineering and static analysis sorts. The most popular technologies include `code obfuscators' that serve to rewrite the original binary code to an equivalent form that provides identical functionality while defeating signature-based detection systems. These systems significantly complicate static analysis, making it challenging to uncover the malware intent and the full spectrum of embedded capabilities. While code obfuscation techniques are commonly integrated into contemporary commodity packers, from the perspective of a reverse engineer, DE obfuscation is often a necessary step that must be conducted independently after unpacking the malware binary.

2.3 Internet Traffic Classification Using Bayesian Analysis Techniques

Accurate traffic classification is of fundamental importance to numerous other network activities, from security monitoring to accounting, and from Quality of Service to providing operators with useful forecasts for long-term provisioning. We apply a Na¨ve Bayes estimator to categorize traffic by application. Uniquely, our work capitalizes on hand-classified network data, using it as input to a supervised Na¨ve Bayes estimator. In this paper we illustrate the high level of accuracy achievable with the Na¨ve Bayes estimator. We further illustrate the improved accuracy of renewed variants of this estimator.

2.3.1 BotGraph : Large Scale Spamming Botnet Detection

Network security applications often require analyzing huge volumes of data to identify abnormal patterns or activities. The emergence of cloud-computing models opens up new opportunities to address this challenge by leveraging the power of parallel computing. In this paper, we design and implement a novel system called BotGraph to detect a new type of botnet spamming attacks targeting major Web email providers. Bot-Graph uncovers the correlations among botnet activities by constructing large user-user graphs and looking for tightly connected subgraph components. This enables us to identify stealthy botnet users that are hard to detect when viewed in isolation.

2.3.2 BotGraph : Large Scale Spamming Botnet Detection

Network security applications often require analyzing huge volumes of data to identify abnormal patterns or activities. The emergence of cloud-computing models opens up new opportunities to address this challenge by leveraging the power of parallel computing. In this paper, we design and implement a novel system called BotGraph to detect a new type of botnet spamming attacks targeting major Web email providers. BotGraph uncovers the correlations among botnet activities by constructing large user-user graphs and looking for tightly connected subgraph components. This enables us to identify stealthy botnet users that are hard to detect when viewed in isolation.

2.4 Understanding Churn in Peer-to-Peer Networks

The dynamics of peer participation, or churn, are an inherent property of Peer-to-Peer (P2P) systems and critical for design and evaluation. Accurately characterizing churn re- quires precise and unbiased information about the arrival and departure of peers, which is challenging to acquire? Prior studies show that peer participation is highly dynamic but with conflicting characteristics. Therefore, churn re- mains poorly understood, despite its significance.

.

2.5 Boosting the Scalability of Botnet Detection Using Adaptive Traffic Sampling

Botnets pose a serious threat to the health of the Internet. Most current network-based botnet detection systems require deep packet inspection (DPI) to detect bots. Because DPI is a computational costly process, such detection systems cannot handle large volumes of traffic typical of large enterprise and ISP networks. In this paper we propose a system that aims to efficiently and effectively identify a small number of suspicious hosts that are likely bots. Their traffic can then be forwarded to DPI-based botnet detection systems for fine-grained inspection and accurate botnet detection.

Bot Attack

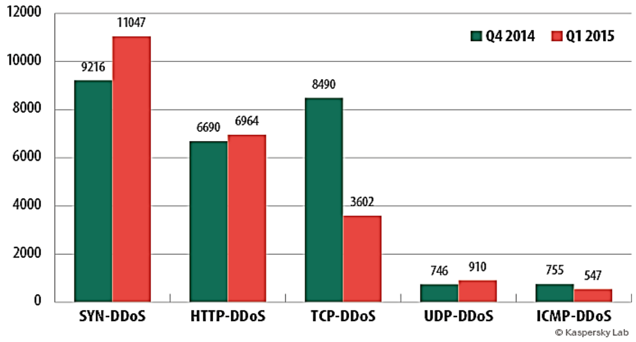


Figure 2. Bot Attack in P2P system over 2014 end and 2015 start

2.5.1 P2P Botnet Detection using Behavior Clustering & Statistical Tests

Most recent research on botnet detection focuses on centralized botnets and primarily relies on two assumptions: prior knowledge of potential C&C channels and capability of monitoring them. However, when botnets switch to a P2P (peer-to-peer) structure and utilize multiple protocols for C&C, the above assumptions no longer hold. Consequently, the detection of P2P botnets is more difficult. In this paper, we relax the above two assumptions and focus on C&C channel detection for P2P botnets that use multiple protocols (randomly chosen) for C&C.

III. RESULTS AND DISCUSSION

Proposed Work

Sality is one of the largest botnets ever identified by researchers. Its behavior represents ominous advances in the evolution of modern malware: the use of more sophisticated stealth scanning strategies by millions of coordinated bots, targeting critical voice communications infrastructure. This project offers a detailed dissection of the botnet's scanning behavior, including general methods to correlate, visualize, and extrapolate botnet behavior across the global Internet.

Since bots are malicious programs used to perform profitable malicious activities, they represent valuable assets for the botmaster, who will intuitively try to maximize utilization of bots. This is particularly true for P2P bots because in order to have a functional overlay network (the botnet), a sufficient number of peers needs to be always online.

We need flow clustering-based analysis approach to identify hosts that are mostly likely running P2P applications. Approach does not rely on any transport layer used by which can be easily violated by P2P applications.

This project offers a detailed dissection of the botnet's scanning behavior, including general methods to correlate, visualize, and extrapolate botnet behavior across the global Internet

The implementation can be done using Java, and the following codes i.e., Coarse Grained Peer-To-Peer Detection, File Uploading and Sending Bot Detection, Clustering and Eliminating, Detection of Attacker IP Address

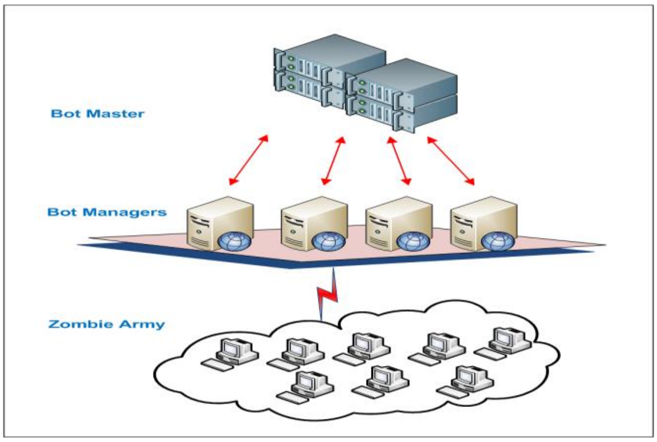


Figure 3. Project Model

3.1 Coarse Grained Peer-To-Peer Detection

This component is responsible for detecting P2P clients by analyzing the remaining network flows after the Traffic Filter component. For each host h within the monitored network we identify two flow sets, denoted as Stcp(h) and Sudp(h), which contain the flows related to successful outgoing TCP and UDP connection, respectively. We consider as successful those TCP connections with a completed SYN, SYN/ACK, ACK handshake, and those UDP (virtual) connections for which there was at least one “request” packet and a consequent response packet.

3.2 File Uploading and Sending

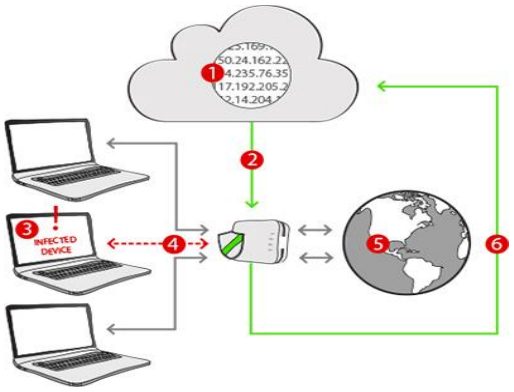
This module is used to upload required file from storage device to user account and send the file into destination account. There are many different types of files: data files, text files, program files, directory files, and so on. Different types of files store different types of information.

3.3 Bot Detection

Since bots are malicious programs used to perform profitable malicious activities, they represent valuable assets for the botmaster, who will intuitively try to maximize utilization of bots. This is particularly true for P2P bots because in order to have a functional overlay network (the botnet), a sufficient number of peers needs to be always online. In other words, the active time of a bot should be comparable with the active time of the underlying compromised system.

3.4 Clustering and Eliminating

The distance between two flows is subsequently defined as the Euclidean distance of their two corresponding vectors. We then apply a clustering algorithm to partition the set of flows into a number of clusters. Each of the obtained clusters of flows, Cj (h), represents a group of flows with similar size.



3.4 Clustering and Eliminating Bot using Coarse grained Botnet detection technique

For each Cj (h), we consider the set of destination IP addresses related to the flows in the clusters, and for each of these IPs we consider its BGP prefix (using BGP prefix announcements).

3.5 Detection of Attacker IP Address

In this module used to determine the geographical location of website visitors based on the IP addresses for applications such as fraud detection. We can find the IP address of the attacker.

IV.CONCLUSION

We also identify the performance bottleneck of our system and optimize its scalability. We presented a novel botnet detection system that is able to identify stealthy P2P botnets, whose malicious activities may not be observable.

V. FUTURE ENHANCEMENT

To summarize, although our system greatly enhances and complements the capabilities of existing P2P botnet detection systems, it is not perfect. We should definitely strive to develop more robust defense techniques, where the aforementioned discussion outlines the potential improvements of our system. Botnet developers are constantly improving their development in order to produce more and more stealthy malware for all kinds of attacks to make profit. While various approaches have been studied or used for botnet attacks, the risk of exploiting widely used browser extensions and their automatic browser extension update mechanisms for command and control channel has not been practically investigated. In this study, we show that it is not difficult to construct stealthy botnet via browser extensions.

VI.REFERENCES

[1] S. Stover, D. Dittrich, J. Hernandez, and S. Dietrich, "Analysis of the storm and nugache trojans: P2P is here," in Proc. USENIX, vol. 32. 2007, pp. 18–27.

[2] P. Porras, H. Saidi, and V. Yegneswaran, "A multi-perspective analysis of the storm (peacomm) worm," Comput. Sci. Lab., SRI Int., Menlo Park, CA, USA, Tech. Rep., 2007.

[3] P. Porras, H. Saidi, and V. Yegneswaran. (2009). Conficker C Analysis Online]. Available: http://mtc.sri.com/Conficker/addendumC/index.html

[4] G. Sinclair, C. Nunnery, and B. B. Kang, "The waledac protocol: The how and why," in Proc. 4th Int. Conf. Malicious Unwanted Softw., Oct. 2009, pp. 69–77.

[5] R. Lemos. (2006). Bot Software Looks to Improve Peerage Online]. Available: http://www.securityfocus.com/news/11390

[6] Y. Zhao, Y. Xie, F. Yu, Q. Ke, and Y. Yu, "Botgraph: Large scale spamming botnet detection," in Proc. 6th USENIX NSDI, 2009, pp. 1–14.

[7] 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, pp. 139–154.

[8] T.-F. Yen and M. K. Reiter, "Are your hosts trading or plotting? Telling P2P file-sharing and bots apart," in Proc. ICDCS, Jun. 2010, pp. 241–252.

[9] S. Nagaraja, P. Mittal, C.-Y. Hong, M. Caesar, and N. Borisov, "BotGrep: Finding P2P bots with structured graph analysis," in Proc. USENIX Security, 2010, pp. 1–16.

[10] J. Zhang, X. Luo, R. Perdisci, G. Gu, W. Lee, and N. Feamster, "Boosting the scalability of botnet detection using adaptive traffic sampling," in Proc. 6th ACM Symp. Inf., Comput.Commun. Security,