INTERNATIONAL RESEARCH JOURNAL IN ADVANCED ENGINEERING AND TECHNOLOGY (IRJAET) E - ISSN: 2454-4752 P - ISSN : 2454-4744
VOL 2 ISSUE 5 (2016) PAGES 1165 - 1168
RECEIVED : 27/08/2016. PUBLISHED : 02/09/2016
September 2, 2016

DETECTING VICTIM SYSTEM IN CLIENT AND CLIENT NETWORKS

[1]Saru latha.D, [2]Srinivasan.J,
[1]M.Phil Scholar, Dept of Computer Science
Adhiparasakthi College of Arts and Science, Vellore
[2]Asst.Prof, Dept of Computer Science, Adhiparasakthi College of Arts and Science, Vellore.

ABSTRACT
Botnets are the most common vehicle of cyber-criminal activity. They are used for spamming, phishing, denial-of-service attacks, brute-force cracking, stealing private information, and cyber warfare. 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 paper offers a detailed dissection of the botnet's scanning behavior, including general methods to correlate, visualize, and extrapolate botnet behavior across the global Internet.

Keywords – Botnet, Sality, Global internet, Malware.

1. INTRODUCTION

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 known system vulnerability. In this paper, 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.

2. LITERATURE REVIEW

It is worth noting that the legitimate P2P application running on a bot-compromised host may present a significant challenge for the existing detection method such as It is mainly due to the fact that the traffic profile of a bot-compromised host might be completely distorted by the legitimate P2P application running on it simultaneously. For instance, in our experiments, when a host is running a Waledac and a Bitorrent application simultaneously. 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.

Malware authors employ a myriad of evasion techniques to impede automated reverse engineering and static analysis eorts. 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 signicantly 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, deobfuscation is often a necessary step that must be conducted independently after unpacking the malware binary.

Accurate trafc classication 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 trafc by application. Uniquely, our work capitalizes on hand-classied 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 rened variants of this estimator. 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.

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.

The dynamics of peer participation, or churn, are an inher-ent 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.

3. PROPOSED APPROACH

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

INTERNATIONAL RESEARCH JOURNAL IN ADVANCED ENGINEERING AND TECHNOLOGY (IRJAET) E - ISSN: 2454-4752 P - ISSN : 2454-4744
VOL 2 ISSUE 5 (2016) PAGES 1165 - 1168
RECEIVED : 27/08/2016. PUBLISHED : 02/09/2016
September 2, 2016

3.1. COARSE GRAINED BOTNET 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.

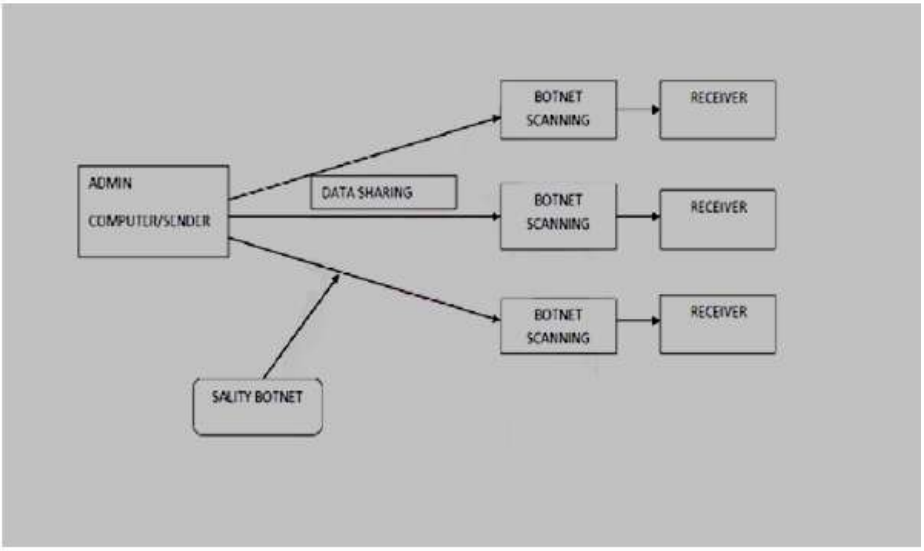

Fig.1.Block diagram

4. EXPERIMENTAL RESULTS AND DISCUSSIONS

4.1 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.

4.2 FILE UPLOAD AND FORWARD

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.

4.3 BOT DETECTION

Since bots are malicious programs used to perform profitable malicious activities, they represent valuable assets for the bot master, 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.

4.4 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. 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).

4.5 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.

5. CONCLUSION AND FUTURE WORKCONCLUSION

In this paper, we presented a novel botnet detection system that is able to identify stealthy P2P botnets, whose malicious activities may not be observable. 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.

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. 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*.