International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056
Volume: 04 Issue: 03 | Mar -2017 www.irjet.net p-ISSN: 2395-0072

Detecting Victim Systems In Client Networks Using Coarse Grained Botnet Algorithm

Bharkavi.S[1], Vanmathi.V[2], Shalini.S[3], and Thangapalani.L[4]

[1,2]*Department of Computer Science and Engineering, Prince Dr. K. Vasudevan College of Engineering and Technology, Chennai, India.*

[3]*Assistant Professor, Department of Computer Science and Engineering, Prince Dr. K. Vasudevan College of Engineering and Technology, Chennai, India.*

[4]*Assistant Professor, Department of Computer Science and Engineering, Prince Dr. K. Vasudevan College of Engineering and Technology, Chennai, India.*

Abstract – *Botnets are the common vehicle for Cyber-criminal activities. They are used for spamming, phishing, denial-of-service attacks, brute-force cracking attacks, stealing private data and cyber warfare. A botnet is also called as a zombie army. This zombie army may range from net computers that the owner of the system may not know that their device is under the control of this zombie army and they transmit the viruses or spam to other co-operative computers which are cordially in link with them in the network. In our paper we propose two stages of approaches for Botnet detection. The primary stage discovers or detects and collects the anomalies related to the network which are co-related to the botnets. The second stage does the work of identifying the bots and blocking them from entering into the receiver end and also identifying the IP addresses of the Bot sender and the Bot sender will not be able to login the application any longer. The approach is generally to exploit the 2 observations: (1) Botmaster's or attacker targets are easier to find as a result of the impact with several alternative nodes, and (2) The activities of the infected machines will be similar to the Botmaster's activities but will be under the supervision of the Botmaster. In our project we are implementing a Scanner for identifying the Bots. Protection mode is implemented in every receiver end.*

***Key Words*: C&C, Decentralized server IRC, Botmaster, P-2-P, and Scanner**.

1. INTRODUCTION

The botnets are widely spread infections through internet. These botnets were found by analysing through the graphs. In the existing system the when the files are transmitted through network then there is no chance of security. While transmitting it there might be loss of data, wrong interpretations of data, and additional information of the user without the user's knowledge. So the system, say for example, the online chat application GMAIL let us consider. In day today's life the world is not ready to hear anything but people are relying on Gadgets. So the highly used one of the application is Gmail. Through this the messages are communicated and all official data in any organization is passed through an Electronic mail. But in this situation only there exist a lot of problem during transferring of mail there may be a case that the sender might unknowingly send the confidential details to the

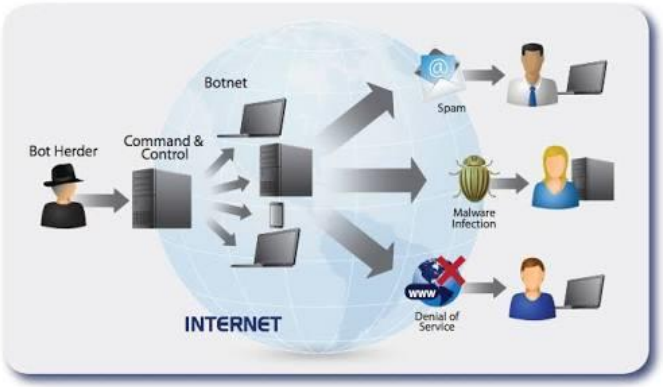


Fig 1. Structure of Botnet.

Unknown user and also the sender might get a lot of unwanted data. These unwanted data the user might initially ignore thinking it is not going to harm. But these ignoring data will only create a problem for the receiver. There are many cases where the system might be checked for antivirus that time we can scan the system for vulnerability even then, the incoming files cannot be checked for virus. The files sent through any applications will not be scanned for threats. We only have opportunity to check virus after it is received to the receiver or destination. While sharing files through external devices, files will be scanned by Anti-Virus. But files received through the application will not be

scanned by our system's anti-virus. These files when received by the system will make the system to crash. The Botmaster issues Command and control (Centralized computer) to a Bot servant and receives replies back from the servant computers (servant bots). These commands issued by the Botmaster will be harmful command so the files after received is waste to check for virus and by that time the system infected will be also serving the Botmaster. Even the user will be unaware of the system's behaviour as a Bot servant.

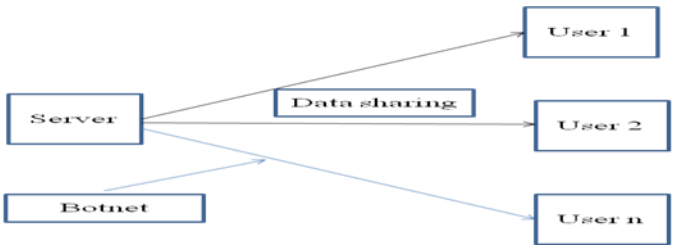


Fig 2. Existing System.

Peer-2-Peer the Bot software is used for having the information about all the computers in the network including the infected computer. The updations are made at regular intervals by the Bot software. These Bot software's will have all the information related to destination and source of the information.

The decentralization is used discover and making it difficult for researches to find out the bots in the clusters. The command and control lacks a lot of security such the owner of the system will also be the user of the infected system.

2. METHODS AND MATERIAL

Related works

In research side the ruling area is Botnets. These botnets are varied in every different situation. Many a times the botnets played very important role. These botnets maybe familiar in network side for security purpose but there are many limitations that we are unaware of. The BOT graphs techniques are used in earlier studies. The BOT graph is behavior over time graph. The type and its harmfulness are established graphically in periodic manner.

2.1 Detection and Classification of Different Botnet C&C Channels

Unlike other types of malware, botnets are characterized by their command and control (C&C) channels, through which a central authority, the Botmaster, may use the infected computer to carry out malicious activities. Given the damage botnets are capable of causing, detection and mitigation of botnet threats are imperative. In this paper, we present a host-based method for detecting and differentiating different types of botnet infections based on their C&C styles, e.g., IRC-based, HTTP-based, or peer-to-peer (P2P) based. Our ability to detect and classify botnet C&C channels shows that there is an inherent similarity in C&C structures for different types of bots and that the network characteristics of botnet C&C traffic is inherently different from legitimate network traffic. The best performance of our detection system has an overall accuracy of 0.929 and a false positive rate of 0.078.

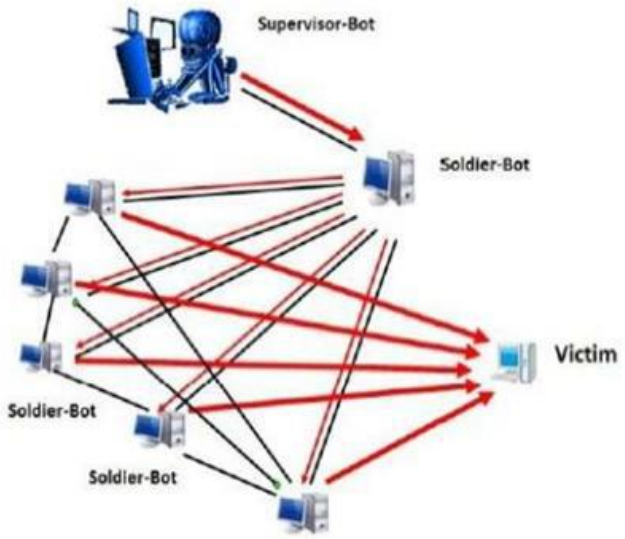
2.2 PeerShark: Detecting Peer-to-Peer Botnets by Tracking Conversations

The decentralized nature of Peer-to-Peer (P2P) botnets makes them difficult to detect. Their distributed nature also exhibits resilience against take-down attempts. Moreover, smarter bots are stealthy in their communication patterns, and elude the standard discovery techniques which look for anomalous network or communication behavior. In this paper, we propose PeerShark, a novel methodology to detect P2P botnet traffic and differentiate it from benign P2P traffic in a network. Instead of the traditional 5-tuple 'flow-based' detection approach, we use a 2-tuple 'conversation-based' approach which is port-oblivious, protocol-oblivious and does not require Deep Packet Inspection. PeerShark could also classify different P2P applications with an accuracy of more than 95%.

2.3 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.



2.4 BotMiner: Clustering Analysis of Network Traffic for Protocol- and Structure-Independent Botnet Detection

Botnets are now the key platform for many Internet attacks, such as spam, distributed denial-of-service (DDoS), identity theft, and phishing. Most of the current botnet detection approaches work only on specific botnet command and control (C&C) protocols (e.g., IRC) and structures (e.g., centralized), and can become ineffective as botnets change their C&C techniques. In this paper, we present a general detection framework that is independent of botnet C&C protocol and structure, and requires no *a priori* knowledge of botnets (such as captured Bot binaries and hence the botnet signatures, and C&C server names/addresses). We start from the definition and essential properties of botnets. We define a botnet as a *coordinated group* of *malware* instances that are *controlled* via C&C communication channels. The essential properties of a botnet are that the bots communicate with some C&C servers/peers, perform malicious activities, and do so in a similar or correlated way. Accordingly, our detection framework clusters similar communication traffic and similar malicious traffic, and performs cross cluster correlation to identify the hosts that share both similar communication patterns *and* similar malicious activity patterns. These hosts are thus bots in the monitored network. We have implemented our BotMiner prototype system and evaluated it using many real network traces. The results show that it can detect real-world botnets (IRC-based, HTTP-based, and P2P botnets including Nugache and Storm worm), and has a very low false positive rate.

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.

3. RESULTS AND DISCUSSION

Proposed system

In our proposed system, the Scanner is implemented for the identification of Bot files. The incoming files are scanned for the bot. If the file sent from the sender first the scanning is done in the sender side and it is checked if the sender or the source has also enabled the protection mode then the scanning will be done in the sender side itself.

If the sender is an adversary, then the protection mode will be disabled before sending the file and it will be sent to the receiver side. The receiver had enabled the protection mode then before receiving the file the file will be scanned in the receiver side.

Even the sent file is not a Bot file the files will be scanned for the Bot in the receiver end. This avoids the mishaps in the network. The protection mode should always be kept enabled so that the adversary trying to get control of system and also the Bot files will be blocked from entering the receiver side.

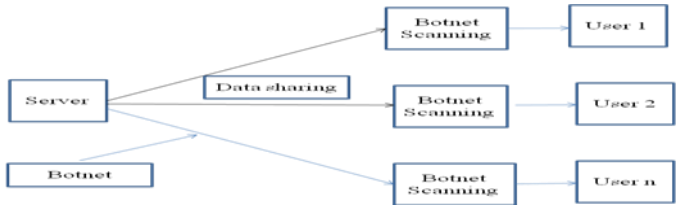


Fig 2. Proposed System

CONCLUSIONS

Through this project we can avoid the intrusion of adversary. We are also blocking the user from entering into the application. Only the admin have the access to the server and can remove the Bot sender from the table which has the Bot details, Bot sender and what type of Bot.

FUTURE ENHANCEMENT

In the network every day a new type of Bot is created and in search of new Bot we need to improve the network. Hence in our project we can implement the blocking of IP address further and find the position of the Bot sender.

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. Bot Software Looks to Improve Peerage Online. http://www.securityfocus.com/news/1130

[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.