BOTNET DETECTION BASED ON COARSE GRAINED PEER-TO-PEER TECHNIQUE

M.Muthulakshmi & M.Grace Ananthi
UG Scholar, P.S.R. Rengasamy College of Engineering for Women, Sivakasi, Tamilnadu India

S.Suganya & P.Raghavan
Associate Professor, P.S.R. Rengasamy College of Engineering for Women, Sivakasi, Tamilnadu, India

Abstract

Botnets are the foremost common vehicle of cyber-criminal activity. They're used for spamming, phishing, denial-of-service attacks, brute-force cracking, stealing non-public data, and cyber warfare. A botnet (also referred to as a zombie army) may be a range of net computers that, though their homeowners ar unaware of it, are got wind of to forward transmissions (including spam or viruses) to alternative computers on the web. During this paper, we have a tendency to propose a two-stage approach for botnet detection. the primary stage detects and collects network anomalies that ar related to the presence of a botnet whereas the second stage identifies the bots by analyzing these anomalies. Our approach exploits the subsequent 2 observations: (1) botmasters or attack targets ar easier to find as a result of the impart with several alternative nodes, and (2) the activities of infected machines are a lot of correlative with one another than those of traditional machines.

Keywords: *Botnet, User interface, Traffic filter component, spam, IP address, Blocking the user.*

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, umeration, 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 scancontinuously using self-propagating worm code that exploits a known system vulnerability. Inthis paper, we focus on a different type of botnet scan—one performed under the explicitcommand and control of the botmaster, occurring over a well-delimited interval. A botnet is acollection of compromised hosts that are remotely controlled by an attacker (the botmaster)through a command and control (C&C) channel. Botnets serve as the infrastructuresresponsible for a variety of cyber-crimes, such as spamming, distributed denialof-service(DDoS) attacks, identity theft, click fraud, etc. The C&C channel is an essential component of a botnet because botmasters rely on the C&C channel to issue commands to their bots andreceive information from the compromised machines. Botnets may structure their C&Cchannels in different ways.

Architecture Diagram

Existing System

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 .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. A bot-compromised host might be completely distorted by the legitimate P2P application running on it simultaneously. Servers are that they represent a single point of failure. Fundamental disadvantage of centralized C&C

Proposed System

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. 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. P2P botnets before they are detected in contrast to our approach can detect and profile various P2P applications. 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 botnets, whose malicious activities may not be observable.

Related Works

To define a general private-key encryption scheme in which a transmitter distributes the same encoded measurements to receivers of different classes. We perform a statistical analysis of the measurements to show that, although not perfectly secure, compressed sensing grants some level of security that comes at almost zero cost and thus may benefit resource-limited applications. We perform a statistical analysis of the measurements to show that, although not perfectly secure, compressed sensing grants some level of security that comes at almost zero cost and thus may benefit resource-limited applications. In addition to this, we report some exemplary applications of multiclass encryption by compressed sensing of speech signals, electrocardiographic tracks and images, in which quality degradation is quantified as the impossibility of some feature extraction algorithms to obtain sensitive information from suitably degraded signal recoveries.

[1] User Interface Design

We design the windows for the project. These windows are used to send a message from one peer to another. We use the Swing package available in Java to design the User Interface. Swing is a widget toolkit for Java. It is part of Sun Microsystems' Java Foundation Classes (JFC) — an API for providing a graphical user interface (GUI) for Java programs.

[2] 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] File Uploading and Sending

It 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] 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

[5] 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).

[6] Detection of Attacker IP Address

It is 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.

Conclusion

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

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
3. (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*