ISSN : 0976-8491 (Online) | ISSN : 2229-4333 (Print)

Botnet Detection based on System and Community Anomaly Detection

[1]Adisesh Mishra, [2]Jalaj Joshi
[1,2]Dept. of Computer Science, SRM University, Kattankulathur, Tamil Nadu, 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 are unaware of it, are got wind of to forward transmissions (including spam or viruses) to alternative computers on the web. During this paper, we propose a two-stage approach for botnet detection. The primary stage detects and collects network anomalies that are 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) bot masters or attack targets are easier to findbecause of several alternative nodes, and (2) the activities of infected machines are a lot in similar with one another than those of traditional machines.

Chapter 1

I. Introduction

A. General
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 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.

B. Objective
Detecting botnets and designing an effective p2p botnet detection system.

C. Existing System
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 Drawbacks:

- Fundamental disadvantage of centralized C&C.
- Servers are that they represent a single point of failure.

Chapter 2

II. Project Description

A. GENERAL
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.

B. Methodologies

Modules Name
This project having the following six modules

1. User Interface Design
2. Coarse Grained Peer-To-Peer Detection
3. File Uploading and sending
4. BOT Detection
5. Clustering and Eliminating
6. Detection of attacker IP address

User Interface Design
In this module 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.

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.

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.

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.

Clustering and Eliminating

The distance between two flows is subsequently defined as the euclidean distanceof 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).

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.

Given Input Expected Output

1. User Interface Design

Input:
Ip address
Port number
Output:
User window

2. Coarse Grained Peer-To-Peer Detection

Input:
Client Requests
Output:
Grouping the ip-address.

3. File Upload

Input:
File upload.
Output:
File sending.

4. Coarse Grained BOT Detection

Input:
1st phase Detection results.
Output:
Grouping the ip-address.

5. Coarse Grained BOT Detection

Input:
2nd phase Detection results.
Output:
Clustering and eliminating.

6. IP Detection

Input:
1st phase Detection results.
Output:
Ip-address.

Algorithm Used

Coarse-Grained Detection of P2P Bots:

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.

Chapter 3

III. Requirements Engineering

A. General

We consider the problem of detecting communities or modules in networks, groups of vertices with a higher-than-average density of edges connecting them. Previous work indicates that a robust approach to this problem is the maximization of the benefit function known as "modularity" over possible divisions of a network.

B. Hardware Requirements

The hardware requirements may serve as the basis for a contract for the implementation of the system and should therefore be a complete and consistent specification of the whole system. They are used by software engineers as the starting point for the system design. It shows what the system does and not how it should be implemented.

Hardware

Processor	: PENTIUM IV 2.6 GHz,Intel Core 2 Duo.
Ram	: 4GB DD RAM
Monitor	: 15" LED, LCD MONITOR
Hard Disk	: 40 GB

Software

Front End	: JAVA (j2ee, Servlets, jsp)
Back End	: My SQL
Operating System	: Windows 07, Mac, Linux
IDE	: Net Beans, Eclipse

1. Software Requirements

The software requirements document is the specification of the system. It should include both a definition and a specification of requirements. It is a set of what the system should do rather than how it should do it. The software requirements provide a basis for creating the software requirements specification. It is useful in estimating cost, planning team activities, performing tasks and tracking the teams and tracking the team's progress throughout the development activity.

Software

Front End	: Core Java, J2EE (Servlets, Jsp)
Back End	: SQL Server 2005.
Operating System	: Windows 07/08
IDE	: Eclipse

C. Functional Requirements
A functional requirement defines a function of a software-system or its component. A function is described as a set of inputs, the behavior, and outputs. The proposed system is achieved by suppression-based and generalization-based k-anonymous and confidential databases. The protocols rely on well-known cryptographic assumptions, and we provide theoretical analyses to proof their soundness and experimental results to illustrate their efficiency.

D. Non-Functional Requirements

1. Efficiency
To address the scalability issue, we propose an edge-centric clustering scheme to extract sparse social dimensions.In sparse social dimensions, the social dimension based approach can efficiently handle networks of millions of actors while demonstrating comparable prediction performance as other non-scalable methods.

2. Reliability
The dynamic nature of networks entails efficient update of the model for collective behavior prediction.

Chapter 4

IV. Design Engineering

A. General
Design Engineering deals with the various UML [Unified Modeling language] diagrams for the implementation of project. Design is a meaningful engineering representation of a thing that is to be built. Software design is a process through which the requirements are translated into representation of the software. Design is the place where quality is rendered in software engineering. Design is the means to accurately translate customer requirements into finished product.

B. 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.

Chapter 5

V. Development Tools

A. General
This chapter is about the software language and the tools used in the development of the project. The platform used here is JAVA. The Primary languages are JAVA,J2EE and J2ME. In this project J2EE is chosen for implementation.

B. Features of JAVA

1. The JAVA Framework
Java is a programming language originally developed by James Gosling at Microsystems and released in 1995 as a core component of Sun Microsystems' Java platform. The language derives much of its syntax from C and C++ but has a simpler object model and fewer low-level facilities. Java applications are typically compiled to bytecode that can run on any Java Virtual Machine (JVM) regardless of computer architecture. Java is general-purpose, concurrent, class-based, and object-oriented, and is specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere".
Java is considered by many as one of the most influential programming languages of the 20th century, and is widely used from application software to web applications. The java framework is a new platform independent that simplifies application development internet.Java technology's versatility, efficiency, platform portability, and security make it the ideal technology for network computing. From laptops to datacenters, game consoles to scientific supercomputers, cell phones to the Internet, Java is everywhere!

2. Objectives of JAVA
To see places of Java in Action in our daily life, explore java.com.

Why Software Developers Choose Java
Java has been tested, refined, extended, and proven by a dedicated community. And numbering more than 6.5 million developers, it's the largest and most active on the planet. With its versatility, efficiency, and portability, Java has become invaluable to developers by enabling them to:
- Write software on one platform and run it on virtually any other platform
- Create programs to run within a Web browser and Web services
- Develop server-side applications for online forums, stores, polls, HTML forms processing, and more
- Combine applications or services using the Java language to create highly customized applications or services
- Write powerful and efficient applications for mobile phones, remote processors, low-cost consumer products, and practically any other device with a digital heartbeat

Some Ways Software Developers Learn Java
Today, many colleges and universities offer courses in programming for the Java platform. In addition, developers can also enhance their Java programming skills by reading Sun's java.sun.com Web site, subscribing to Java technology-focused newsletters, using the Java Tutorial and the New to Java Programming Center, and signing up for Web, virtual, or instructor-led courses.

Object Oriented
To be an Object Oriented language, any language must follow at least the four characteristics.
1. Inheritance: It is the process of creating the new classes and using the behavior of the existing classes by extending them just to reuse the existing code and adding addition a features as needed.
2. Encapsulation: It is the mechanism of combining the information and providing the abstraction.
3. Polymorphism: As the name suggest one name multiple form, Polymorphism is the way of providing the different functionality by the functions having the same name based on the signatures of the methods.
4. Dynamicbinding: Sometimes we don't have the knowledge of objects about their specific types while writing our code. It is the way of providing the maximum functionality to a program about the specific type at runtime.

3. JavaServer Pages - An Overview

Java Server Pages or JSP for short is Sun's solution for developing dynamic web sites. JSP provide excellent server side scripting support for creating database driven web applications. JSP enable the developers to directly insert java code into jsp file, this makes the development process very simple and its maintenance also becomes very easy.
JSP pages are efficient, it loads into the web servers memory on receiving the request very first time and the subsequent calls are served within a very short period of time.
In today's environment most web sites servers dynamic pages based on user request. Database is very convenient way to store the data of users and other things. JDBC provide excellent database connectivity in heterogeneous database environment. Using JSP and JDBC its very easy to develop database driven web application.
Java is known for its characteristic of "write once, run anywhere." JSP pages are flat JavaServer Pages
JavaServer Pages (JSP) technology is the Java platform technology for delivering dynamic content to web clients in a portable, secure and well-defined way. The JavaServer Pages specification extends the Java Servlet API to provide web application developers with a robust framework for creating dynamic web content on the server using HTML and XML templates, and Java code, which is secure, fast, and independent of server platforms.
JSP has been built on top of the Servlet API and utilizes Servlet semantics. JSP has become the preferred request handler and response mechanism. Although JSP technology is going to be a powerful successor to basic Servlets, they have an evolutionary relationship and can be used in a cooperative and complementary manner.
Servlets are powerful and sometimes they are a bit cumbersome when it comes to generating complex HTML. Most servlets contain a little code that handles application logic and a lot more code that handles output formatting. This can make it difficult to separate and reuse portions of the code when a different output format is needed. For these reasons, web application developers turn towards JSP as their preferred servlets environment.

4. Evolution of Web Applications

Over the last few years, web server applications have evolved from static to dynamic applications. This evolution became necessary due to some deficiencies in earlier web site design. For example, to put more of business processes on the web, whether in business-to-consumer (B2C) or business-to-business (B2B) markets, conventional web site design technologies are not enough. The main issues, every developer faces when developing web applications, are:
1. Scalability - A successful site will have more users and as the number of users is increasing fastly, the web applications have to scale correspondingly.
2. Integration of data and business logic - The web is just another way to conduct business, and so it should be able to use the same middle-tier and data-access code.
3. Manageability - Web sites just keep getting bigger and we need some viable mechanism to manage the ever-increasing content and its interaction with business systems.
4. Personalization - Adding a personal touch to the web page becomes an essential factor to keep our customer coming back again. Knowing their preferences, allowing them to configure the information they view, remembering their past transactions or frequent search keywords are all important in providing feedback and interaction from what is otherwise a fairly one-sided conversation.
Apart from these general needs for a business-oriented web site, the necessity for new technologies to create robust, dynamic and compact server-side web applications has been realized. The main characteristics of today's dynamic web server applications are as follows:

- Serve HTML and XML, and stream data to the web client
- Separate presentation, logic and data
- Interface to databases, other Java applications, CORBA, directory and mail services
- Make use of application server middleware to provide transactional support.
- Track client sessions.

5. Benefits of JSP

One of the main reasons why the Java Server Pages technology has evolved into what it is today and it is still evolving is the overwhelming technical need to simplify application design by separating dynamic content from static template display data. Another benefit of utilizing JSP is that it allows to more cleanly separating the roles of web application/HTML designer from a software developer. The JSP technology is blessed with a number of exciting benefits, which are chronicled as follows:

- The JSP technology is platform independent, in its dynamic web pages, its web servers and its underlying server components. That is, JSP pages perform perfectly without any hassle on any platform, run on any web server, and web-enabled application server. The JSP pages can be accessed from any web server.
- The JSP technology emphasizes the use of reusable components. These components can be combined or manipulated towards developing more purposeful components and page design. This definitely reduces development time apart from the At development time, JSPs are very different from Servlets, however, they are precompiled into Servlets at run time and executed by a JSP engine which is installed on a Web-enabled application server such as BEA WebLogic and IBM WebSphere.

C. Servlets

Earlier in client- server computing, each application had its own client program and it worked as a user interface and need to be installed on each user's personal computer. Most web applications use HTML/XHTML that are mostly supported by all the browsers and web pages are displayed to the client as static documents.
A web page can merely displays static content and it also lets the user navigate through the content, but a web application provides a more interactive experience.
Any computer running Servlets or JSP needs to have a container. A container is nothing but a piece of software responsible for loading, executing and unloading the Servlets and JSP. While servlets can be used to extend the functionality of any Java- enabled server.
They are mostly used to extend web servers, and are efficient replacement for CGI scripts. CGI was one of the earliest and most prominent server side dynamic content solutions, so before going forward it is very important to know the difference between CGI and the Servlets.

D. Java Servlets

Java Servlet is a generic server extension that means a java class can be loaded dynamically to expand the functionality of a server.

Servlets are used with web servers and run inside a Java Virtual Machine (JVM) on the server so these are safe and portable.
Unlike applets they do not require support for java in the web browser. Unlike CGI, servlets don't use multiple processes to handle separate request. Servets can be handled by separate threads within the same process. Servlets are also portable and platform independent.
A web server is the combination of computer and the program installed on it. Web server interacts with the client through a web browser. It delivers the web pages to the client and to an application by using the web browser and he HTTP protocols respectively. The define the web server as the package of large number of programs installed on a computer connected to Internet or intranet for downloading the requested files using File Transfer Protocol, serving e-mail and building and publishing web pages. A web server works on a client server model.

E. Conclusion
JSP and Servlets are gaining rapid acceptance as means to provide dynamic content on the Internet. With full access to the Java platform, running from the server in a secure manner, the application possibilities are almost limitless. When JSPs are used with Enterprise JavaBeans technology, e-commerce and database resources can be further enhanced to meet an enterprise's needs for web applications providing secure transactions in an open platform. J2EE technology as a whole makes it easy to develop, deploy and use web server applications instead of mingling with other technologies such as CGI and ASP. There are many tools for facilitating quick web software development and to easily convert existing server-side technologies to JSP and Servlets.

Chapter 6

VI. Application

A. General
In this work, we study how networks in social media can help predict some human behaviors and individual preferences. In particular, given the behavior of someindividuals in a network, how can we infer the behavior of other individuals in the same social network

B. Application
- Online ticket reservation application
- Online secure chat application

C. Futire Enhancements
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.

Conclusion
In this paper, we have 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, S. Dietrich,"Analysis of the storm and nugachetrojans: P2P is here," In Proc. USENIX, Vol. 32. 2007, pp. 18–27.
[2] P. Porras, H. Saidi, 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, 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, Y. Yu,"Botgraph: Large scale spamming botnet detection", In Proc. 6th USENIX NSDI, 2009, pp. 1–14.
[7] G. Gu, R. Perdisci, J. Zhang, 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, 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, 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, N. Feamster, "Boosting the scalability of botnet detection using adaptive traffic sampling," In Proc. 6th ACM Symp. Inf., Comput. Commun. Security.

Adisesh Mishra is currently undergoing final year of his B.Tech. Course in SRM University, Kattankulathur under the discipline Computer Science. He has completed successful projects in natural language processing, linux architecture and networking domain. He has avid interests in the network security domain and is currently engaged upon implementing a successful P2P botnet detection system.

Jalaj Joshi is currently undergoing final year of his B.Tech. Course in SRM University, Kattankulathur under the discipline Computer Science. He was a former intern at Gen-X technology Gurgaon under the network security avenue for a 3 month internship course. He has avid interests in the networking domain and is currently engaged upon implementing a successful P2P botnet detection system.