Detecting Computer Network Anomaly with Data Mining Technology

HU ZhiYu[1, a], LI Li [2]

[1,2] JINGDEZHEN UNIVERSITY, JiangXi JINGDEZHEN 333000,China

[a]huzhiyu@126.com

Keywords: Data Mining; Computer Network Anomaly; Intrusion Detection System (IDS);

Abstract. With the rapid development of machine learning and Internet technology, the combination of the two methods is well appreciated recently. An anomaly and intrusion detection system is a mechanism that monitors network or system activities for malicious activities. Intrusion detection and prevention systems are primarily focused on identifying possible incidents, logging information about them and reporting attempts. As far as other usages of Intrusion detection and prevention systems are concerned, such as identifying problems with security policies and deterring individuals from violating security policies. Anomaly detection systems are becoming an important addition to the security infrastructure of nearly every organization. In this paper, we propose a novel mechanism for real-world traffic and research there cases with theoretical analysis.

Introduction

There are a multitude of malicious traffic detection techniques, and thus, vulnerabilities in common security components, such as firewalls are unavoidable. Intrusion detection system and intrusion prevention systems are commonly used today. They are used to detect different types of malicious traffic, network communications, and computer system usage with the mission of preserving systems from widespread damage; that is because other detection and prevention techniques, such as firewalls, access control, skepticism, and encryption have failed to fully protect networks and computer systems from increasingly sophisticated attacks and malware malicious traffic makes network performance inefficient and troubles users. Intrusion detection systems and intrusion prevention systems are used to detect different types of malicious traffic, network communications, the detection and prevention technologies, such as firewalls, access control and encryption fail to adequately protect the network and sophisticated computer systems from attack. However, there is no "perfect" test method, it always correct and normal activities of malicious distinguished. In other words, Intrusion Detection System/Intrusion Prevention System can determine a normal activity such as a malicious one, resulting in a false positive, or malicious traffic as normal, making false negative.

Focusing on network traffic measurement of Peer-to- Peer (P2P) [1] applications on the Internet. P2P applications supposedly constitute a substantial proportion of today's Internet traffic. The research contributions in each field are systematically summarized and compared, allowing us to clearly define existing research challenges, and to highlight promising new research directions. The results of this review should provide useful insights into the current IDS literature is a good source for people who are interested in the application of CI approach to intrusion detection systems, or related fields. In this paper, we study the effect of background traffic malicious traffic traces collected from two different locations. We show that the malicious traffic causes DNS latencies to increase by 230% and web latencies to increase by 30%. Using packet-level simulations based on an empirically derived model of the worm, we demonstrate that the effect of worm-infected hosts can be disastrous when they trigger a DDoS attack [2].

System Design and Implementation

The Existing System. Monitoring the communication protocol between a connected device because using protocol based system [3]. A host-based intrusion detection system (HIDS) identifies intrusions by analyzing system calls, application logs, file-system modifications (binaries, password files,

capability/acl databases) and other host activities and state. An example of a HIDS is OSSEC. The data extraction is not separable from the content descriptions. FP and FN rates are two important indicators to measure the accuracy of network security system, such as an IDS or IPS. It has been proven, FP's even a small rate (110,000) can produce an unacceptable number of actual detection of FP. It is important to assess the IDS / IPS developers attempt accuracy detection of FP and FNs optimized reduced, because the FP / FN rate limiting network security system, since the base rate fallacy phenomenon performance. In the statistical analysis of this work to clarify the reasons for FPS and FNs and rankings, allowing developers to avoid similar to their product development process traps. Detection of the DUT may be incorrect, FP or FNs result of. FPNA has the following three procedures, majority voting, verification and tracking human analysis.

First, majority voting is a decision which has a majority, that is, more than half of the votes. It is a binary decision voting used most often in influential decision-making bodies, including the legislatures of democratic nations. In this work, out method is based on all of the DUT voters detection / FNs's (DUT) and the potential FP definition majority vote. In other words, if only one or a few of the DUT test log generated some specific packet trace, the trace shows that the case for the FN or true negative (TN) are. On the other hand, when more than half of the DUT is alarmed when this track, track may be a FP or true positive (TP). Second, to detect potential FP / FNs's / TPS / TN after, according to the log replay this work tracking the extracted packet to the device under test. This step is called trace verification because it verifies whether this case is reproducible to the original DUTs. This case is producible FP/FN/TP/TN when it meets the two bad conditions.

Our Proposed System. IDSs/IPSs can identify a normal activity as malicious one, causing a false positive (FP), or malicious traffic as normal, causing a false negative (FN) and then a variety of commercial products, open source, and research into IDSs were proposed. When securing a network, administrators have to use many different tools. Although functionality of them is similar, administrators have to spend a considerable amount of time to read documentation and learn how to use a new tool. Our propose system holds the following advantages: (1) Network Intrusion Detection Systems gain access to network traffic by connecting to a hub, network switch configured for port mirroring, or network tap. (2) To minimize this effort a specialized tool securing network and checking available service. (3) For each operating system different applications have to be used, regardless they are doing exactly the same. (4) The ASE was expanded into a bigger system, called the PCAPLib system. The PCAPLib system not only extracted and classified the real-world traffic captured from Campus Beta Site into proper categories by leveraging multiple IDSs, but also anonymized users privacy in these FP and FN traffic traces out of security considerations.

The C4.5 Data Mining Algorithm. The C4.5 algorithm is Quinlan’s extension of his own ID3 algorithm for generating decision trees. Just as with CART, the C4.5 algorithm recursively visits each decision node, selecting the optimal split, until no further splits are possible [4]. However, there are interesting differences between CART and C4.5 Unlike CART, the C4.5 algorithm is not restricted to binary splits. Whereas CART always produces a binary tree, C4.5 produces a tree of more variable shape. - For categorical attributes, C4.5 by default produces a separate branch for each value of the categorical attribute. This may result in more “bushiness” than desired, since some values may have low frequency or may naturally be associated with other values. The C4.5 method for measuring node homogeneity is quite different from the CART method and is examined in detail below. In general, steps in C4.5 algorithm to build decision tree are: Choose attribute for root node, Create branch for each value of that attribute, Split cases according to branches, Repeat process for each branch until all cases in the branch have the same class, Choosing which attribute to be a root is based on highest gain of each attribute. To count the gain, we use formula 1-3 to calculate:

$$Gain(S,A) = Entropy(S) - \sum_{i=1}^{n} \frac{|S_i|}{|S|} \cdot Entropy(S_i) \tag{1}$$

$$Entropy(S) = \sum_{i=1}^{n} -p_i \cdot \log_2(p_i) \tag{2}$$

$$Gain(S_j, A) = Entropy(S_j) - \sum_{i=1}^{n} \frac{|S_i|}{|S_j|} \cdot Entropy(S_i) \quad (3)$$

$\{S_1, \cdots S_n\}$ is the partitions of S according to values of attribute A. n is the number of attributes A, $|S_i|$ is the number of cases in the partition S_i, $|S|$ is the total number of cases in S. S is the case set, n is the number of cases in the partition S. p_i is the proportion of S_i to S.

Multi-boosting and Data Set Analysis. The effect of combining different classifiers can be explained with the theory of bias-variance decomposition. Bias refers to an error due to a learning algorithm while variance refers to an error due to the learned model. The total expected error of a classifier is the sum of the bias and the variance. In order to reduce bias and variation, some ensemble approaches have been introduced: Adaptive Boosting, Bootstrap Aggregating (Bagging),Wagging and Multi-boosting. This is why the idea emerged of combining both in order to profit from the advantages of both algorithms and obtain an overall error reduction. A data set is a collection of data, usually presented in tabular form. Each column represents a particular variable. Each row corresponds to a given member of the data set in question. It lists values for each of the variables, such as height and weight of an object or values of random numbers. Each value is known as a datum. The data set may comprise data for one or more members, corresponding to the number of rows. Value can be a number, such as real or integer, such as representatives of a person's height in centimeters, but it might be the name of the data (ie, excluding the value), for example, on behalf of a person's race. More generally, any value can be described as a kind of measurement of the level. For each variable, the value is usually the same kind. However, it may be "missing value", which needs to somehow indicate. The Internet has seen in recent days, the continuous rise of malicious traffic, including DDoS attacks and worms. In this paper, we study the effects of background traffic malicious traffic traces collected from two different locations. We show that the malicious traffic causes DNS latencies to increase by 230% and web latencies to increase by 30%. Using packet-level simulations based on an empirically derived model of the worm, we demonstrate that the effect of worm-infected hosts can be disastrous when they trigger a DDoS attack.

Experiment and Simulation Result

A network-based IDS/IPS is an independent platform, while a host-based one consists of an agent on a host. Winpcap and jpcap captured real-world traffic and replay all functions, like antivirus, anti-spam, P2P, instant messenger (IM), streaming scan, and system logs of DUTs are enabled if possible[2]. WinPcap is an open source library for packet capture and network analysis for the Win32 platforms. Most networking applications access the network through widely used operating system primitives such as sockets. It is easy to access data on the network with this approach since the operating system copes with the low level details (protocol handling, packet reassembly, etc.) and provides a familiar interface that is similar to the one used to read and write files [5]. The purpose of Win cap is to give this kind of access to Win32 applications; it provides facilities to: (1) Capture raw packets, (2) Transmit raw packets to the network, (3) Gather statistical information on the network traffic. Jpcap is a Java class package that allows Java applications to capture and/or send packets to the network. Jpcap is based on libpcap/winpcap and Raw Socket API[4]. Therefore, Jpcap is supposed to work on any OS on which libpcap/winpcap has been implemented. Currently, Jpcap has been tested on FreeBSD 3.x, Linux RedHat 6.1, Fedora Core 4, Solaris, and Windows 2000/XP.

As shown in the following figure1, our system has a clear structure. However, there is still some draw-backs. The hackers recover the embedding data in original image because the data placed in particular bit position. To attack the hidden data using original image because referred the key value. The data extraction is not separable from the content descriptions.

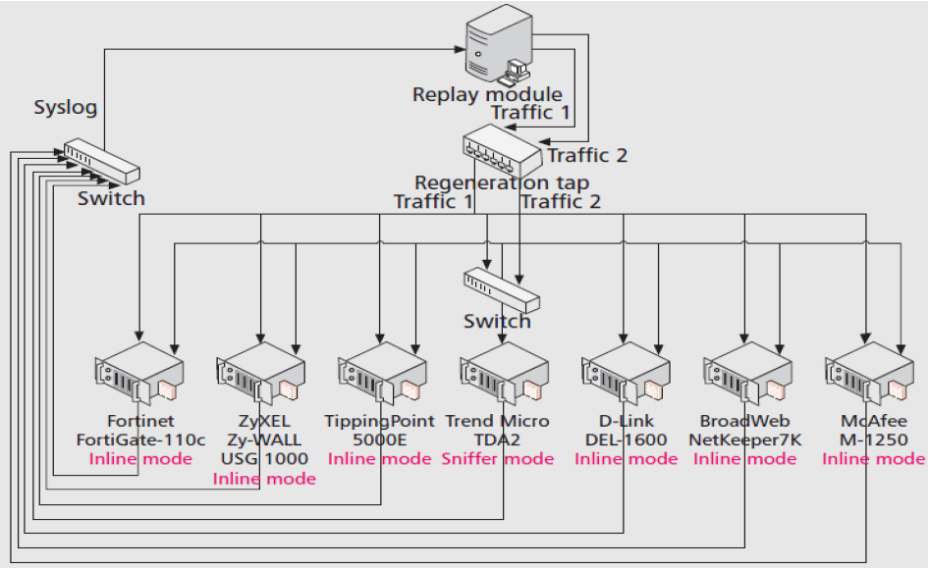

Fig. 1 The Experimental System

Conclusion and Summary

This concludes what kinds of FPs or FNs happen easily to IDS/IPS with real-world traffic and investigates their frequencies across all FPs and FNs. There are two hierarchies of classification in this work. One is by protocols, such as HTTP, FTP, NetBIOS and IRC and the other is by IDS policy types (also called “attack types”), like DDoS, buffer overflow, Web attack, scan, and so on. IDSs/IPSs are less reliable today because of the limitations of the signature-base methodology. This work proposes the C45 Algorithm which inducing classification rules in the form of decision trees from a set of given examples. The decision trees generated by C4.5 can be used for classification, and for this reason, C4.5 is often referred to as a statistical classifier. This can identify malicious traffic by using Attack System Extraction (ASE).

References

[1] K.-C. Lan, A. Hussain, and D. Dutta, “Effect of Malicious Traffic on The Network,” Proc. Passive and Active Measurement Wksp. (PAM), San Diego, CA, Apr. 2003.

[2] S.-H. Wang, “Extracting, Classifying and Anonym zing Packet Traces with Case Studies on False Positives/Negatives Assessment,” M.S. thesis, Dept. Comp. Sci., Nat’l. Chiao Tung Univ., Taiwan, 2010

[3] S.-H. Wang, “Extracting, Classifying and Anonymizing Packet Traces with Case Studies on False Positives/Negatives Assessment,” M.S. thesis, Dept. Comp. Sci., Nat’l. Chiao Tung Univ., Taiwan, 2010.

[4] Y.-D. Lin et al., “On Campus Beta Site: Architecture Designs, Operational Experience, and Top Product Defects,” IEEE Commun. Mag., vol. 48, no. 12, Dec.2010, pp. 83–91.

[5] S.-X. Wu and W. Banzhaf, “The Use of Computational Intelligence in Intrusion Detection Systems: A Review,” Detection,” Proc. 16th Int’l. Conf. Systems, Signals and Image Processing, June 2009.