Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 2, Issue. 4, April 2013, pg.610 – 614

RESEARCH ARTICLE

New Data Mining Algorithm for Intrusion Detection in Networks

M. Penchala Praveen[1], M. Sambath[2], S. Ravi[3]

[1]Computer Science and Engineering, Hindustan University, India
[2]Computer Science and Engineering, Hindustan University, India
[3]Computer Science and Engineering, Hindustan University, India

[1] praweenkrishna@gmail.com; [2] sambath1980@gmail.com; [3] sravi@hindustanuniv.ac.in

Abstract— An intrusion detection system is a mechanism that monitors network or system activities for malicious activities. Intrusion detection and prevention systems (IDPS) are primarily focused on identifying possible incidents, logging information about them and reporting attempts .In organizations use IDPS for other purposes, such as identifying problems with security policies and deterring individuals from violating security policies. Intrusion detection systems have become a necessary addition to the security infrastructure of nearly every organization. Some systems may attempt to stop an intrusion attempt but this is neither required nor expected of a monitoring system, this work proposes a mechanism for real-world traffic and statistically analyzes these cases.

Key Terms: - Java Capturing Packets; Windows Capturing Packets; Classifiers 45; Intrusion Detection System; False positive; False Negative; peer to peer Application

I. 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 techniques, such as firewalls, access control and encryption have failed to fully protect networks and computer systems from sophisticated attacks. However, there is no “perfect” detection approach, which can always correctly distinguish between malicious and normal activities. In other words, IDSs/IPSs can identify a normal activity as a malicious one, causing a false positive, or malicious traffic as normal, causing a false negative.

II. LITERATURE REVIEW

Focuses on network traffic measurement of Peer-to- Peer (P2P)[2] 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 findings of this review should provide useful insights into the current IDS literature and be a good source for anyone who is interested in the application of CI approaches to IDSs or related fields.

The Internet has seen, in recent days, a continuous rise in malicious traffic including DDoS and worm attacks. In this paper, we study the effect of malicious traffic on the background traffic by gathering traces 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 a 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.[3]

III. Experimental Setup

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].

A. *WinPcap*

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[4]. The purpose of Win cap is to give this kind of access to Win32 applications; it provides facilities to

- Capture raw packets.
- Transmit raw packets to the network.
- Gather statistical information on the network traffic

B. *Jpcap*

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 Microsoft Windows 2000/XP.

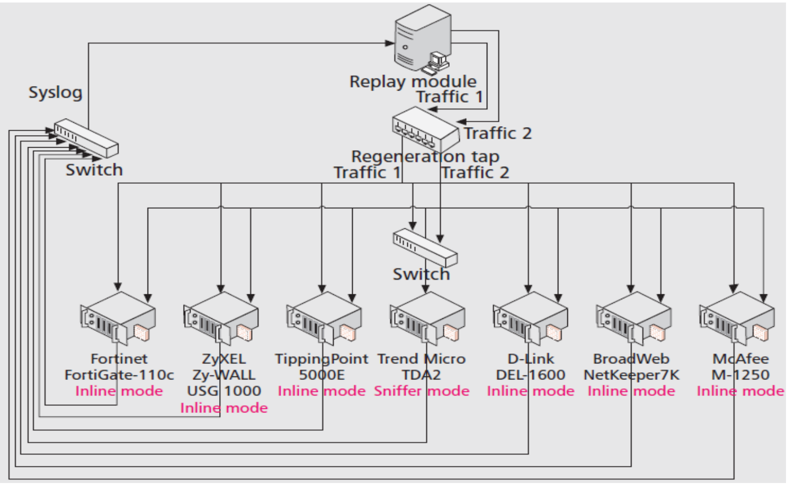

Fig 1 Experimental Setup for the IDS

IV. Existing System

Monitoring the communication protocol between a connected device (a user/PC or system) because using protocol based system[5] . 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.

A. *False Positive/Negative Assessment*

FP and FN rates are two important metrics in measuring the accuracy of a network security system, such as an IDS or IPS. It has been demonstrated that even a small rate (1 in 10,000) of FPs could generate an unacceptable number of FPs in practical detections. The assessment is important to IDS/IPS developers trying to optimize the accuracy of detection by reducing both FPs and FNs, because the FP/FN rate limits the performance of network security systems due to the base-rate fallacy phenomenon. The statistical analyses in this work can elucidate the causes and rankings of FPs and FNs, thus allowing developers to avoid similar pitfalls during their product development. The detection of DUTs may be incorrect, resulting in FPs or FNs. FPNA has the following three procedures, majority voting, trace verification and manual 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, the voters are all DUTs (device under test) and potential FPs/FNs are detected under the definition of majority voting. In other words, if only one or a few DUTs generate a detection log for some specific packet trace, this trace appears as an FN or a true negative (TN) case. On the other hand, when more than half of the DUTs have alerts for this trace, the trace is likely to be an FP or a true positive (TP). Second, after detecting the potential FPs/FNs/TPs/TNs, this work replays the extracted packet trace according to the log to the DUTs again. 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 following two conditions.

- For any DUT, it must produce an alert if it did last time
- The two alerts must be the same when one came from some DUT last time and the other is produced by the same DUT this time Otherwise, this case is un-reproducible. For example, there are one traffic flow and three DUTs, A, B and C. After this traffic flow passes through the PCAPLib system, we get an extracted packet trace from this traffic and two alerts from A and C. Two alerts are named A1 and C1, respectively. Then, we replay this extracted packet to A, B and C again. If A and C produce alerts, called A2 and C2, and the content of A2 and C2 are same as that of A1 and C1, respectively, this extracted packet trace is reproducible.

B. *Limitation of the existing system*

1. The hackers recover the embedding data in original image because the data placed in particular bit position.
2. To attack the hidden data using original image because referred the key value.

The data extraction is not separable from the content descriptions.

V. System Design

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. To create a pool of traffic traces causing possible FPs and FNs to IDSs because using Attack System Extraction (ASE). 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.

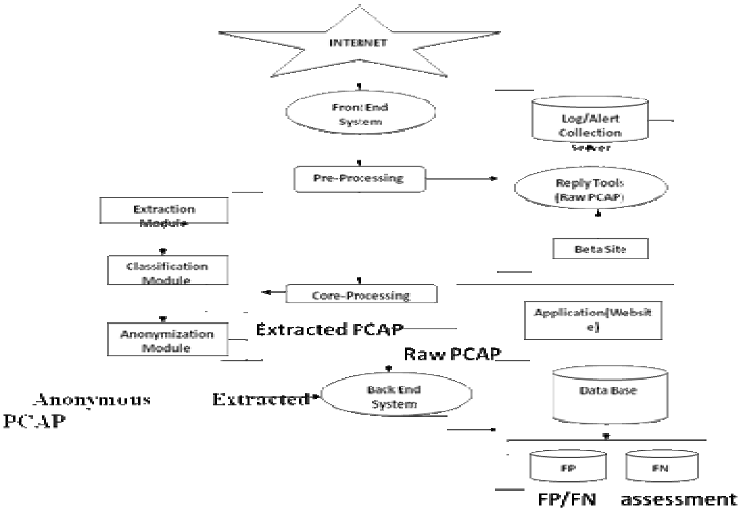

Fig 2 Proposed System Architecture

A. Advantages of proposed framework

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.

B. *C45 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 [6]. 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,below:

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

{S1, ..., Si, …, Sn} = partitions of S according to values of attribute A , n = number of attributes A,
|Si| = number of cases in the partition Si , |S| = total number of cases in S

$$Entropy(S) = \sum_{i=1}^{n} - pi * \log_2 pi \text{ ....}(2)$$

S : Case Set , n : number of cases in the partition S, pi : Proportion of Si to S

C. *Multi boosting*

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 (AdaBoost) ,Bootstrap Aggregating (Bagging),Wagging and Multiboosting. 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.

D. *Analyzing the Data set*

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. The values may be numbers, such as real numbers or integers, for example representing a person's height in centimeters, but may also be nominal data (i.e., not consisting of numerical values), for example representing a person's ethnicity. More generally, values may be of any of the kinds described as a level of measurement. For each variable, the values will normally all be of the same kind. However, there may also be "missing values", which need to be indicated in some way. The Internet has seen, in recent days, a continuous rise in malicious traffic including DDoS and worm attacks. In this paper, we study the effect of malicious traffic on the background traffic by gathering traces 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.

VI. Conclusion and Future Enhancement

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.-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.

[3] I.-W. Chen et al., “Extracting Attack Sessions from Real Traffic with Intrusion Prevention Systems,” Proc. IEEE ICC, June 2009.

[4] 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.

[5] 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.

[6] I.-W. Chen et al., “Extracting Attack Sessions from Real Traffic with Intrusion Prevention Systems,” Proc. IEEEICC, June 2009.

[7] 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.

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

[9] TippingPoint Technologies, “IPS vs. IDS: Similar on the Surface, Polar Opposites Underneath,” Whitepaper,http://rovingplanet.net/resources_whitepapers.html.

[10] “Global Market Share Statistics and News,” http://marketshare. hitslink.com/os-market-share.aspx?qprid=9.