A Data Mining Analysis and Approach with Intrusion Detection / Prevention with Real Traffic

Meenakshi RM
PG Scholar, Dept of CSE,
Dr.MGR Educational and research Institute,
Maduravoyal, Chennai – 600095.

E.Saravanan,
Assistant Professor, Dept of CSE,
Dr.MGR Educational and research Institute,
Maduravoyal, Chennai – 600095.

ABSTRACT
An intrusion detection system (IDS) is a device or software application that monitors network or system activities for malicious activities or policy violations and produces reports to a Management Station.Some systems may attempt to stop an intrusion attempt but this is neither required nor expected of a monitoring system. Intrusion detection and prevention systems (IDPS) are primarily focused on identifying possible incidents, logging information about them, and reporting attempts. In addition, organizations use IDPS for other purposes, such as identifying problems with security policies, documenting existing threats, and deterring individuals from violating security policies. IDPS have become a necessary addition to the security infrastructure of nearly every organization.False positives and false negatives happen toevery intrusion detection and intrusion preventionsystem. This work proposes a mechanismfor false positive/negative assessment with multipleIDSs/IPSs to collect FP and FN cases fromreal-world traffic and statistically analyze thesecases. Over a period of 16 months, more than2000 FPs and FNs have been collected and analyzed.From the statistical analysis results, weobtain three interesting findings. First, morethan 92.85 percent of false cases are FPs even ifthe numbers of attack types for FP and FN aresimilar. That is mainly because the behavior ofapplications or the format of the applicationcontent is self-defined; that is, there is not completeconformance to the specifications of RFCs.accordingly, when this application meets anIDS/IPS with strict detection rules, its traffic willbe regarded as malicious traffic, resulting in a lotof FPs. Second, about 91 percent of FP alerts,equal to about 85 percent of false cases, are notrelated to security issues, but to management policy.For example, some companies and campuseslimit or forbid their employees and studentsfrom using peer-to-peer applications; therefore,in order to easily detect P2P traffic, an IDS/IPSis configured to be sensitive to it. Hence, thiscauses alerts to be triggered easily regardless ofwhether the P2P application has malicious trafficor not. The last finding shows that buffer overflow,SQL server attacks, and worm slammerattacks account for 93 percent of FNs, eventhough they are aged attacks. This indicates thatthese attacks always have new variations toevade IDS/IPS detection.

Keywords
IDS, FPS, FNS, FP

INTRODUCTION
The popularization of network-based services, intrusion detection systems (IDS) have become important tools for ensuring network security that is the violation of information security policy. IDS collect information from a variety of network sources using intrusion detection sensors, and analyze the information for signs of intrusions that attempt to compromise the confidentiality and integrity of networks [1]-[3]. Network-based intrusion detection systems (NIDS) monitor and analyze network traffics in the network for detecting intrusions from internal and external intruders [4]-[9]. Internal intruders are the inside users in the network with some authority, but try to gain extra ability to take action without legitimate authorization. External intruders are the outside users without any authorized access to the network that they attack. IDS notify network security administrator or automated intrusion prevention systems (IPS) about the network attacks, when an intruder try to break the network. Since the amount of audit data that an IDS needs to examine is very large even for asmall network, several data mining algorithms, such as decision tree, naïve Bayesian classifier, neural network, Support Vector Machines, and fuzzy classification, etc [10]-[20] have been widely used by the IDS community for detecting known and unknown intrusions. Data mining based intrusion detection algorithms aim to solve the problems of analyzing the huge volumes of audit data and realizing performance optimization of detection rules [21]. But there are still some drawbacks in currently available commercial IDS, such as low detection accuracy, large number of false positives, unbalanced detection rates for different types of intrusions, long response time, and redundant input attributes.

An IDS/IPS monitors the activities of a given environment and decides whether these activities are malicious or normal based on system integrity, confidentiality and the availability of information resources. As soon as a malicious or an intrusive event is detected, the IDS produces a relative alert and passes it to the network administrator promptly while the IPS not only executes what the IDS does but also blocks network traffic from the suspected malicious source. 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 (FP), or malicious traffic as normal, causing a false negative (FN). FPs and FNs cause several problems. For example, FNs generate unauthorized or abnormal activities on the Internet or in computer systems. On the other hand, a lot of FPs may easily conceal real attacks[1] and thus overwhelm the security operator. When real attacks occur true positives (real alerts) are deeply buried within FPs, so it is easy for the security operator to miss them [4].

1. Architecture of Data Mining Based IDS

An IDS monitors network traffic in a computer network like a network sniffer and collects network logs. Then the collected network logs are analyzed for rule violations by using data mining algorithms. When any rule violation is detected,

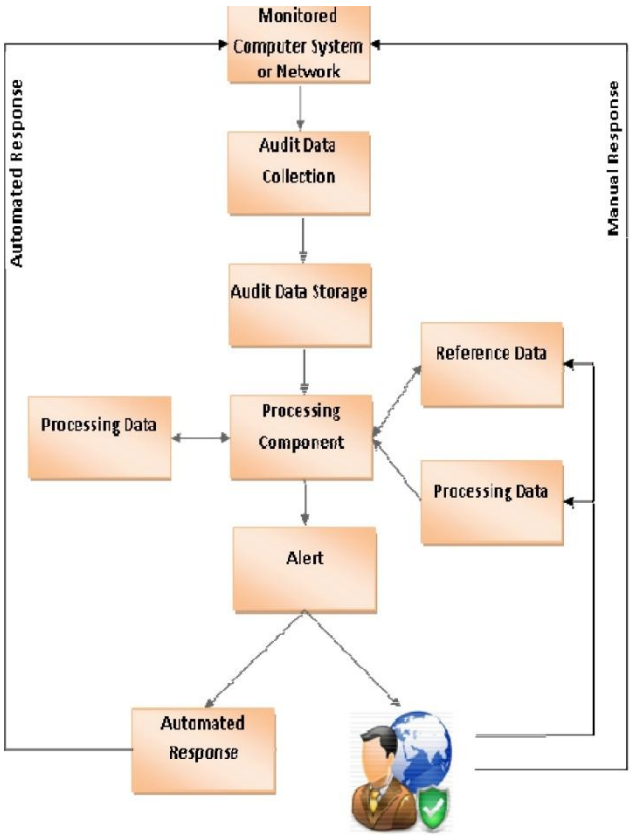

Figure 1. Organization of a generalized data mining based IDS

- Audit data collection: IDS collect audit data and analyzed them by the data mining algorithms to detect suspicious activities or intrusions. The source of the data can be host/network activity logs, command-based logs, and application-based logs.
- Audit data storage: IDS store the audit data for future reference. The volume of audit data is extremely large. Currently adaptive intrusion detection aims to solve the problems of analyzing the huge volumes of audit data and realizing performance optimization of detection rules.
- Processing component: The processing block is the heart of IDS. It is the data mining algorithms that apply for detecting suspicious activities. Algorithms for the analysis and detection of intrusions have been traditionally classified into two categories: misuse (or signature) detection, and anomaly detection.
- Reference data: The reference data stores information about known attacks or profiles of normal behaviors.

Processing data: The processing element must frequently store intermediate results such as information about partially fulfilled intrusion signatures.

ι. Alert: It is the output of IDS that notifies the network security officer or automated intrusion prevention system (IPS).

ιι. System security officer or intrusion prevention system (IPS) carries out the prescriptions controlled by the IDS.

2. Feature Selection

Feature selection becomes indispensable for high performance intrusion detection using data mining algorithms, because irrelevant and redundant features may lead to complex intrusion detection model as well as poor detection accuracy. Feature selection is the process of finding a subset of features from total original features. The purpose of feature selection is to remove the irrelevant input features from the dataset for improving the classification accuracy. Feature selection in particularly useful in the application domains that introduce a large number of input dimensions like intrusion detection. Many data mining methods have been used for selecting important features from training dataset such as information gain based, gain ratio based, principal component analysis (PCA), genetic search, and classifier ensemble methods etc [46]-[53]. In 2009, Yang et al. [54] introduced a wrapper-based feature selection algorithm to find most important features from the training dataset by using random mutation hill climbing method, and then employs linear support vector machine (SVM) to evaluate the selected subset-features. Chen et al. [55] proposed a neural-tree based algorithm to identify important input features for classification, based on an evolutionary algorithm that the feature contributes more to the objective function will consider as an important feature.

In this paper, to select the important input attributes from training dataset, we construct a decision tree by applying ID3 algorithm in training dataset. The ID3 algorithm constructs decision tree using information theory [56], which choose splitting attributes from the training dataset with maximum information gain. Information gain is the amount of information associated with an attribute value that is related to the probability of occurrence. Entropy is the quantify information that is used to measure the amount of randomness from a dataset. When all data in a set belong to a single class, there is no uncertainty then the entropy is zero. The objective of ID3 algorithm is to iteratively partition the given dataset intosub-datasets, where all the instances in each final subset belong to the same class. The value for entropy is between 0 and 1 and reaches a maximum when the probabilities are all the same. Given probabilities $p_1, p_2,..,p_s$, where $\sum_{i=1} p_i=1$;

$$Entropy: H(p_1,p_2,...p_s) = \sum_{i=1}^{S} (p_i \log(1/p_i)) \quad (1)$$

Given a dataset, *D, H(D)* finds the amount of sub-datasets of original dataset. When that sub-dataset is split into s new sub-datasets $S = \{D_1, D_2,...,D_s\}$, we can again look at the entropy of those sub-datasets. A subset is completely ordered if all instances in it are the same class. The ID3 algorithm calculates the gain by the equation "(2)".

$$Gain\ (D,S) = H(D)-\sum_{i=1}^{S} p(D_i)H(D_i) \quad (2)$$

After constructing the decision tree from training dataset, we weight the attributes of training dataset by the minimum depth at which the attribute is tested in the decision tree. The depth of root node of the decision tree is 1. The weight for an attribute is set to1 *d*, where *d* is the minimum depth at

whichthe attribute is tested in the tree. The weights of attributes that do not appear in the decision tree are assigned to zero.

i. Traffic capturing and replaying to an IPS

In this work, we replay real traffic to IPSs and identify attacks by the logs in the IPSs. Such an approach of capturing and replaying has been used for performance evaluation of IPSs [1] [2]. Extracting an attack session [3] involving multiple connections from a huge number of traffic traces is non-trivial. This work designs a method to extract an attack session based on the similarity of packets. Tcpdump captures real traffic in a PCAP file, and Tcpreplay (tcpreplay.sourceforge.net) replays the traffic trace packet by packet to IPSs at the specified speed or in the order of the timestamps that indicate the capturing time of the packets.IPSs differ in many aspects such as signature set, accuracy and logging system. The signature set affects the number of detected attacks. The accuracy affects the correctness of deciding whether an event is an attack or not. The logging system affects the name of an attack. First two properties are reasons that we want to do the integration of efforts from different IPS vendors. However, the last property is what we need to resolve to correctly compare logs of IPSs.

ii. Attack identifiers and attack types

Although IPSs may name an attack differently, most of them have a system of common identifier for attacks and that is CVE number. CVE (Common Vulnerabilities and Exposures, cve.mitre.org) is a list of security vulnerabilities and exposures that provides common names for publicly known vulnerabilities for easily sharing data across separate vulnerability capabilities (tools, repositories, and services) with this "common enumeration".

We divide attacks into three types according to the number of attackers (i.e. source IP address) and the number of connections per attacker, as presented in Table 1. An attack of the first type (i.e. 1-1) involves one attacker and a single connection per attacker. For example, the MySQL Authentication Bypass Exploit [4] allows a user to login a MySQL database without authentication. An attack of the second type (i.e. 1-N) involves one attacker and more than one connection per attacker. For example, the Blaster worm [5] establishes three connections for each victim. An attack of the third type (i.e. N-1) involves multiple attackers and a single connection per attacker. A Distributed Denial of Service (DDoS) attack belongs [6] [7] to this type. This classification will be used in the following ASE algorithm.

3. FPS AND FNS

FPs and FNs of the IDS/IPS are mystery terms that describe a situation where the IDS/IPS makes a mistake. The former means that the IDS/IPS triggers an alert when there is no malicious activity in the traffic while the latter meansthat there is no alert raised by the IDS/IPS when malicious traffic passes through it. FP and FN rates are two metrics important in measuring the accuracy of the IDS/IPS [9].

An FP of the IDS/IPS will not result in an intrusion and it may be caused by two reasons: the detection mechanism of the IDS/IPS may be faulty or the IDS/IPS detects an anomaly that turns out to be benign. Therefore, an FP may cause security analysts to expend unnecessary effort. Moreover, if a hacker launches a *snow-blind* attack, the challenge for security analysts isto somehow identify the real attack amidst the chaff caused by the hacker. This may create a potential vulnerability for the IDS. On the other hand, when an IPS has an FP, the primary concern is that legitimate traffic might be blocked. Most organizations consider blocking legitimate traffic as a much more serious problem than generating a false alert. Consequently, an FP of the IPS is a much more serious matter than that of the IDS. If the IPS blocks legitimate traffic a few times, it will be yanked out of the network.

An FN is simply a missed attack, which may put networks or computer systems in danger. Clearly an FN is undesirable, and every vendor strives to provide the most complete coverage possible. However, there is no silver bullet: no product detects all attacks. Hence, the goal becomes providing coverage against high priority attacks. Aside from lack of coverage, several other reasons may also cause an FN. For example, in order to evade the IDS or IPS, the attack may incorporate obfuscation techniques. Another possibility is overwhelming the IDS with traffic beyond its processing capacity, so the IDS will drop the packets needed to detect the attack. For an IPS, overwhelming it has a different effect: it causes traffic to be dropped. The attack doesn't succeed because attack packets are dropped, but it is also not detected. Accordingly, the attack can be tried again.

In practice, for a vendor of IDSs/IPSs, an FN is much more serious than an FP because of negative effects of an FN including reduced trust in the IDS/IPS, and because of damage caused by the intrusion. However, from a user's point of view, an FP may be more serious than an FN because an FP may cause the IPS to block the user's benign traffic. In addition, the user may allow some FNs as long as they're not too fre-quent. Therefore, it is necessary to investigate and analyze FPs and FNs with IDSs/IPSs in detail.

4. THE CAMPUS BETASITE AND THE PCAPLIB SYSTEM

The traffic source for the PCAPLib system comes from the Campus Beta-Site deployed at National Chiao Tung University, Hsinchu, Taiwan. The Campus BetaSite is used by developers to test and debug products while maintaining network quality for network users. Moreover, it is an operational network on campus and records network traffic from net-work users into packet capture (PCAP) files. The volume of network traffic on/through the BetaSite is roughly 100 Gbytes/h.

The goal of trace sharing is to preserve real-world traffic behavior in packet traces so that it can be replicated and picked up easily by researchers for network forensics.[2] To achieve this goal, the PCAPLib system consists of front-end and back-end systems. The front-end system not only extracts and classifies valuable packet traces from real-world traffic but also precisely and deeply protects the sensitive information in the packets. This is because recording the entire real-world traffic consumes storage space and searching for specific events within the huge traces is time-consuming. Therefore, recording only traffic associated with specific/special events would be better. Besides, packet anonymization protects privacy from leakage in trace sharing. On the other hand, the back-end system is responsible for storing the extracted PCAP files, whether anonymous or otherwise, and for demonstrating the usefulness of the PCAPLib system in network forensics when used in con-junction with other applications, such as FPNA.

The preprocessing component of the front-end system uses a

traffic replay tool (e.g., tcpre-play) to replay captured raw traffic to multiple devices under test (DUTs) to leverage their domain knowledge. If a DUT detects abnormal behavior in the traffic, it will trigger an alert. For the core processing component of the front-end system, there are two mechanisms, Active TraceCollection (ATC) and Deep Packet Anonymization (DPA). Based on DUT logs, the ATC finds out the anchor packets that trigger the logs, processes packets and connection associations to extract each specific/special session into packet traces, and uses supervised classification to categorize the extracted packet traces. On the other hand, the DPA parses application-level protocol identities and anonymizes sensitive fields for privacy protection of packet traces, while still maintaining their usefulness for research

5. 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 [7]. 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.

As in previous work [6, 7], the ATC leverages the domain knowledge of the DUTs of intrusion detection/prevention, antivirus, anti-spam and application classifier to collect real-world packets. The detection of DUTs may be incorrect, resulting in FPs or FNs. As a demonstration of network forensics using real-world traffic, this work assesses FP/FN cases using the FPNA mechanism as shown in Fig. 2a. FPNA has the following three procedures, majority voting, trace verification and manual analysis. First, majorityvoting 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 deci-sion-making bodies, including the legislatures of democratic nations. In this work, the voters are all DUTs 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). Majority voting's flow chart is described in Fig. 2b.

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 repro-ducible to the original DUTs. This case is a reproducible FP/FN/TP/TN when it meets the following two conditions.

1. For any DUT, it must produce an alert if it did last time.
2. The two alerts must be the same when onecame 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. In order to show these two conditions in Fig. 2c, we use "are all alerts same as before?" to represent them. Late, in order to know whether the reproducible traffic trace is a publicly malicious case, the step of manual analysis manually investigates the causes of the reproducible traffic trace and compares these causes with Common Vulnerabilities and Exposures (CVE), a dictionary of publicly known information security vulnerabilities and exposures. After this step, an FP/FN or a TP/TN is identified and the occurrences of frequent cases are also counted. Figures 2c and 2d respectively describe the flow charts of the second and third steps.

6. CONCLUSIONS

This work proposes a system to completely extract suspicious sessions from traffic traces. These suspicious sessions may cause FPs or FNs to an IPS and the extracted traffic traces can be used for analysis by signature developers to improve the accuracy of the IPS. The extraction process scans a traffic trace three times. Similarity between two packets is defined to extract a DDOS attack completely. We define "variation" and "completeness and purity" to evaluate the accuracy of ASE. 95% of the extracted attacks have low variation. Also, the average CP is up to 80%. This method could be extended to other detection system such as Anti-Virus, P2P/IM management, and Network Forensics. FPs and FNs are still the key issues for IDS, IPS which are less reliable today because of the limitations of the signature based system.

7. REFERENCES

[1] H. G. Kayacık and A. N. Zincir-Heywood, "Using Intrusion Detection Systems with a Firewall: Evaluation on DARPA 99 Dataset", Project in Dalhousie University, [Online]. Available:http://projects.cs.dal.ca/projectx/files/NIMS06-2003.pdf.

[2] DARPA 99 Intrusion Detection Data Set Attack Documentation. [Online]. Available: http://www.ll.mit.edu/IST/ideval/docs/1999/attackDB.html.

[3] V. Corey, C. Peterman, S. Shearin, M. S. Greenberg, J. V. Bokkelen, "Network Forensics Analysis," IEEE Internet Computing, vol.06, no.6, pp. 60-66, 2002.

[4] W. D. Yu, D. Aravind, P. Supthaweesuk, "Software Vulnerability Analysis for Web Services Software Systems," iscc, pp. 740-748, 11th IEEE Symposium on Computers and Communications (ISCC'06), 2006.

[5] M. Bailey, E. Cooke, F. Jahanian, D. Watson, Jose Nazario, "The Blaster Worm: Then and Now," IEEE Security and Privacy, vol. 03, no. 4, pp. 26-31, 2005.

[6] C. L. Schuba, I. V. Krsul, M. G. Kuhn, E. H. Spafford, A. Sundaram, D. Zamboni, "Analysis of a Denial of Service Attack on TCP," sp, p. 0208, 1997 IEEE Symposium on Security and Privacy, 1997.

[7] V. Paxson, "An analysis of using reflectors for distributed denial-of-service attacks" ACM SIGCOMM Computer Communication Review, 2001.

[8] M. Roesch, "Network Security: Snort - Lightweight Intrusion Detection for Networks", Proceedings of the

13th USENIX conference on System administration, November. 1999.

[9] T. H. Cormen, C. E. Leiserson, R. L. Rivest, "Introduction to Algorithms", p.p. 314-320, 1990.

[10] T. Ye, D. Veitch, G. Iannaccone and S. Bhattacharyya, "Divide and Conquer: PC-Based Packet Trace Replay at OC-48 Speeds", IEEE TRIDENTCOM, 2005.

[11] W. C. Feng, A. Goel, A. Bezzaz, W. C. Feng, and J. Walpole. "TCPivo: A high-performance packet replay engine". ACM SIGCOMM Workshop on Models, Methods and Tools for Reproducible Network Research (MoMeTools), Aug. 2003.

[12] R. W. Lucky, "Automatic equalization for digital communication," *Bell Syst. Tech. J.*, vol. 44, no. 4, pp. 547–588, Apr. 1965.