# False Positives and Negatives from Real Traffic with Intrusion Detection/Prevention Systems

Cheng-Yuan Ho, Ying-Dar Lin, Yuan-Cheng Lai, I-Wei Chen, Fu-Yu Wang, and Wei-Hsuan Tai

***Abstract*—False Positives (FPs) and False Negatives (FNs) happen to every Intrusion Detection/Prevention System (IDS/IPS). This work proposes a mechanism of *False Positive/Negative Assessment* (FPNA) with *multiple* IDSs/IPSs to collect FP and FN cases from real-world traffic. Over a period of sixteen months, more than two thousand FPs and FNs have been collected and analyzed. From the statistical analysis results, we obtain three interesting findings. First, more than 92.85% of false cases are FPs even if the numbers of attack types for FP and FN are similar. Second, about 91% of FP alerts, equal to about 85% of false cases, are not related to security issues, but to *management policy*. The last finding shows that buffer overflow, SQL server attack and worm slammer attacks account for 93% of FNs, even though they are aged attacks. This indicates that these attacks always have new variations to evade IDS/IPS detection.**

***Index Terms*—False positive, false negative, intrusion detection, network security.**

## I. Introduction

During the last several years, malicious traffic detection has been an active area of network security because the Internet has witnessed a surge in malicious traffic generated by network attacks, such as denial-of-service (DoS), and propagation of botnets, viruses, worms, trojan horses, spyware and so on. Moreover, malicious traffic makes network performance inefficient and troubles users.

There are a multitude of malicious traffic detection techniques and thus vulnerabilities in common security components, such as firewalls, are unavoidable. *Intrusion detection systems* (*IDSs*) and *intrusion prevention systems* (*IPSs*) 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 [1], [2].

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 and thus overwhelm the security operator. When real attacks occur, true positives (real alerts) are deeply buried within FPs, so it's easy for the security operator to miss them [3].

Accordingly, a variety of commercial products, open source, and research into IDSs were proposed. Wu and Banzhaf [1] provided an overview of different IDS algorithms, such as artificial neural networks, swarm intelligence, evolutionary computation, artificial immune systems, fuzzy sets and soft computing, and their problems. A collaborative intelligent IDS and a fuzzy inference system were proposed to reduce FPs through fuzzy alert correlation in [2] and [4], respectively, while Sourour et al. in [3] reduced both FPs and FNs with their environmental awareness intrusion detection and prevention system. A system of *Attack Session Extraction* (*ASE*) was proposed in [5] to create a pool of traffic traces causing possible FPs and FNs to IDSs. One to two years later, the ASE was expanded into a bigger system, called the *PCAPLib* system [6]. The PCAPLib system not only extracted and classified the real-world traffic captured from *Campus BetaSite* [7] into proper categories by leveraging multiple IDSs, but also anonymized users' privacy in these FP and FN traffic traces out of security considerations. However, previous work only focused on studying how to reduce FPs and/or FNs in IDSs or how to collect and extract the FP and FN traffic traces from real-world traffic.

This work collects more than two thousand cases of FPs and FNs from the real-world traffic of Campus BetaSite by the PCAPLib system, in order to observe what kinds of FPs or FNs happen easily in which protocols and in what kind of attacks, and investigate their frequencies across all FPs and FNs. Also, the reasons behind these FP and FN cases for network forensics and trends in malicious traffic attacks are conjectured in this work. With this work, application users and developers can understand why the traffic of an application is sometimes blocked by the IPS, while IDS developers could pay attention on eliminating these FN/FP cases.

The remainder of this paper is organized as follows. The methodology of how to collect and assess FPs and FNs from real-world traffic is described in Section 2. The

Manuscript received April 10, 2012; revised May 19, 2012.

Cheng-Yuan Ho, Ying-Dar Lin, I-Wei Chen, Fu-Yu Wang, and Wei-Hsuan Tai are with National Chiao Tung University, Taiwan (e-mail ydlin@cs.nctu.edu.tw)

Yuan-Cheng Lai is with National Taiwan University of Science and Technology, Taiwan

experimental environment in this work and statistical analysis are shown in Section 3. Finally, the last section concludes this work and outlines future work.

## II. Methodology

This section first takes a look at the Campus BetaSite and the PCAPLib system (which is the traffic source), and then details how to identify and assess two thousand cases of FPs and FNs for network forensics on a set of IDSs/IPSs. Herein, the method of assessing FPs/FNs is called False Positive/Negative Assessment (FPNA).

### A. The Campus BetaSite and the PCAPLib System

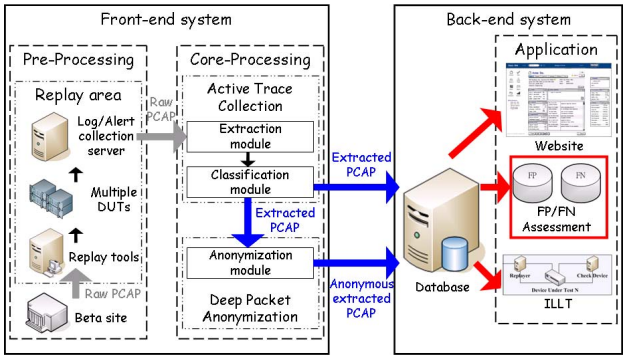
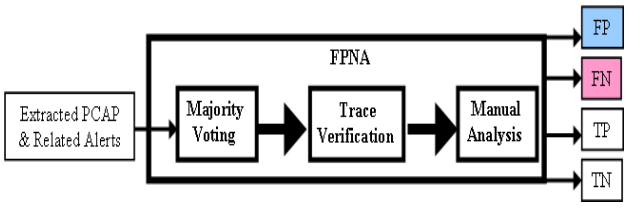



Fig. 1. Architecture and block diagram of the PCAPLib system.

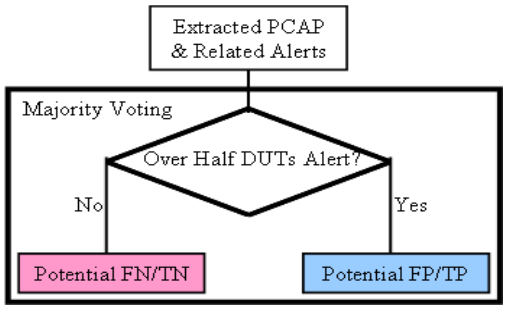
As shown in Fig. 1, the traffic source for the PCAPLib system comes from the Campus BetaSite deployed at the National Chiao Tung University, Hsinchu, Taiwan. The Campus BetaSite is used by developers to test and debug products while maintaining network quality for network users. Moreover, it is an operational network on campus and records network traffic from network users into packet capture (PCAP) files. The volume of network traffic on/through the BetaSite is roughly 100GB per hour.

The pre-processing component of the front-end system uses a traffic replay tool (e.g., tcpreplay) 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 Trace Collection (ATC) and Deep Packet Anonymization (DPA). Based on DUT alerts, the ATC finds out the anchor packets that trigger the alerts, 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.

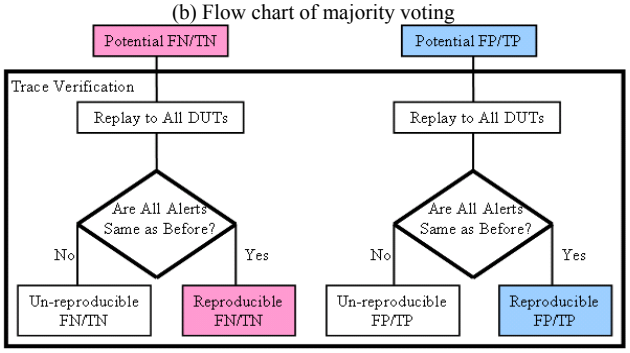
### B. False Positive/Negative Assessment



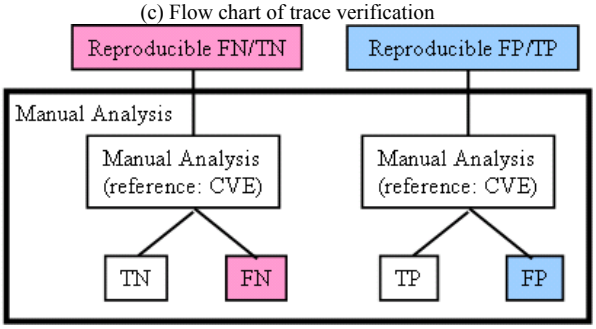

(a) Whole flow chart of FPNA mechanism




(b) Flow chart of majority voting




(c) Flow chart of trace verification




(d) Flow chart of manual analysis

Fig. 2. Details of the false positive/negative assessment mechanism.

As in previous work [5, 6], 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. 2(a). 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 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. 2(b).

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. 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. Fig. 2(c) and (d) respectively describe the flow charts of the second and third steps.

TABLE I: DETAILED INFORMATION OF 7 DUTS

| Vendor | Fortinet | ZyXEL | TippingPoint | Trend Micro | D-Link | BroadWeb |
|---|---|---|---|---|---|---|
| Device Name | FortiGate-110c | ZyWALL USG 1000 | 5000E | TDA2 | DFL-1600 | NetKeeper7K |
| IDS/IPS | IPS | IPS | IPS | IDS | IPS | IPS |
| Location | Network | Network | Network | Network | Network | Network |
| Method | Signature | Signature | Signature | Signature | Signature | Signature |
| AntiVirus | Yes | Yes | No | Yes | No | No |
| AntiSpam | Yes | No | No | No | No | No |
| P2P | Yes | Yes | Yes | Yes | Yes | Yes |
| IM | Yes | Yes | Yes | Yes | Yes | Yes |
| Streaming | No | No | No | Yes | No | No |

## III. STATISTICAL ANALYSIS

### A. Experimental Environment

The PCAP files were captured real-world traffic at the BetaSite, as shown in Fig. 1, during the period Oct. 1, 2009 to Feb. 1, 2011. Seven DUTs are used and their detailed information, such as vendor, device, name, etc. is shown in Table 1. We observe that only Trend Micro TDA2 is an IDS while the other six DUTs are IPSs. In this work, all DUTs are network-based security detection systems due to the PCAPLib system's architecture whereas they are all signature-based because a signature-based IDS/IPS is more easily implemented than an anomaly-based one. During replay, all functions, like antivirus, anti-spam, P2P, Instant Messenger (IM) and streaming scan, and system logs, of DUTs are enabled if possible. After trace verification, reproducible FPs/FNs/TPs/TNs will be passed to the manual analysis step, where all alerts are compared to the CVE in order to check whether they are FPs, FNs, TPs, or TNs.

### B. Statistical Results

This subsection analyzes 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, etc.

A. FP cases taking the most percentage of false cases

The number of FPs is thirteen times that of FNs. In other words, more than 92.85% of false cases are FPs. However, when we calculate how many kinds of attack there are in FPs and FNs, we find that the number of kinds of attack in FN cases, 27, is close to that in FP cases, 35. We guess that FP cases have many cases with traffic similarity, meaning that network traffic of a certain protocol happens to exhibit some characteristics belonging to other protocols [6]. To prove this guess, the number of each type of attack is calculated. There are dozens or hundreds of FP cases as compared to only a few FN cases.

About 91% of FP alerts, equal to about 85% of false cases, are not related to security issues, but to management policy. Policy here means some configuration arguments are artificially constructed for some reason. For instance, some companies and campuses limit or forbid their employees and students from using peer to peer (P2P) applications, and therefore, thresholds of P2P traffic in an IDS/IPS will be configured very low. Hence, this causes alerts to be easily triggered regardless of whether the P2P application has malicious traffic or not.

### C. Policy and Self-Defined formats Causing FPs

Here we raise several real cases. The "HTTP-Inspection" alert results from application clients using their self-defined formats, not defined by RFCs, and the traffic happens to be similar to an ASCII-encoding attack, apache-whitespace attack, and so on. The "SQL Injection comment attempt" alert results from BitTorrent clients who happen to bind port 80, and the traffic happens to be similar to an injection attempt. Then "TCP port scan" alert results from applications which test how many free ports there are in order to establish many connections at the same time. The "FTP wu-ftp bad file completion attempt" alert results from the "[" character which appears often in FTP transfer data. The "Veritas Backup Agent DoS attempt" alert results from BitTorrent clients who bind port 10000 (the port monitored by the rule), and the traffic happens to be similar to a DoS attempt.

### D. Many Aged Attacks Having new Variations

Some representative cases deserve our attention. The "Buffer Overflow" alert results from Windows being vulnerable to buffer overflow when handling certain types of Remote Procedure Call (RPC) traffic, and this flaw occurs within the 'netapi32.dll' component of the Server service with NetPathCanonicalize requests. The "SQL Server Attack" alert results from a login that fails for user 'sa'. The "MS-SQL Worm Slammer" alert is caused by DoS on some Internet hosts. In sum, the buffer overflow and the MS-SQL worm slammer, totaling 103 FN cases, are aimed at Microsoft products because Microsoft is estimated to make up nearly 90% of the OS marketshare. Moreover, although buffer overflow, SQL server attacks and worm slammer attacks are aged attacks, they still account for 93% of FNs. This may indicate that these attacks always have

new variations to avoid IDS/IPS detection.

## IV. Conclusion

This work proposes the False Positive/Negative Assessment (FPNA) mechanism in the PCAPLib system to provide statistical analysis of FP and FN cases. The FPNA collected more than two thousand FPs and FNs during sixteen months. 92.85% of false cases were FPs and 7.15% were FNs. Out of numerous FPs, about 91% of FP alerts occur because of IDS's or IPS's policy, not due to security issues. The distribution of the collected FPs shows that 90% are using HTTP and 57% of FPs are thought to be HTTP inspection attacks. NetBIOS accounts for 68% of FNs and about 67% of FN cases are aimed at Microsoft products. From the statistical analysis, we also observe that traffic similarity is the main cause of FP cases, and missing attack signatures in the signature design is the cause of FN cases.

## Reference


[1] S. X. Wu and W. Banzhaf, "The Use of Computational Intelligence in Intrusion Detection Systems: A Review," *Elsevier Applied Soft Computing*, vol. 10, pp. 1-35, January 2010.

[2] H. T. Elshoush and I. M. Osman, "Reducing False Positives through Fuzzy Alert Correlation in Collaborative Intelligent Intrusion Detection Systems — A Review," in *Proc. of IEEE International Conference on Fuzzy Systems (FUZZ)*, pp. 1-8, July 2000.

[3] M. Sourour, B. Adel and A. Tarek, "Environmental Awareness Intrusion Detection and Prevention System toward Reducing False Positives and False Negatives," in *Proc. of IEEE Symposium on Computational Intelligence in Cyber Security* (*CICS'09*), April 2009.

[4] G. P. Spathoulas and S. K. Katsikas, "Using a Fuzzy Inference System to Reduce False Positives in Intrusion Detection," in *Proc. of 16th International Conference on Systems, Signals and Image Processing* (*IWSSIP 2009*), June 2009.

[5] I. W. Chen, P. C. Lin, C. C. Luo, T. H. Cheng, Y. D. Lin, Y. C. Lai, and F. C. Lin, "Extracting Attack Sessions from Real Traffic with Intrusion Prevention Systems," in *Proc. IEEE Intl. Conference on Communications (ICC)*, June 2009.

[6] S.-H. Wang, "Extracting, Classifying and Anonymizing Packet Traces with Case Studies on False Positives/Negatives Assessment," M.S. thesis, *Dept. Computer Science*, *National Chiao Tung University*, *Taiwan*, 2010.

[7] Y. D. Lin, I. W. Chen, P. C. Lin, C. S. Chen, and C. H. Hsu, "On Campus Beta Site: Architecture Designs, Operational Experience, and Top Product Defects," *IEEE Communication Magazine*, vol. 48, pp. 83-91, December 2010.