Performance Analysis in Intrusion Detection and Prevention Systems

Khalid Alsubhi⋆, Nizar Bouabdallah†, Raouf Boutaba⋆‡
⋆David R. Cheriton School of Computer Science, University of Waterloo, Ontario, Canada
†INRIA, Campus universitaire de Beaulieu; 35042 Rennes Cedex, France
‡Division of IT Convergence Engineering, POSTECH, Republic of South Korea
E-Mail:{kaalsubh,rboutaba}@uwaterloo.ca; nizar.bouabdallah@inria.fr

***Abstract*—Intrusion Detection and/or Prevention Systems (IDPS) represent an important line of defense against a variety of attacks that can compromise the security and proper functioning of an enterprise information system. Although many IDPS systems have been proposed, their appropriate configuration and control for effective attacks detection/prevention and efficient resources consumption has always been challenging. The evaluation of the IDPS performance for any given security configuration is a crucial step for improving real-time capability. This paper aims to analyze the impact of security enforcement levels on the performance and usability of an enterprise information system. We develop a new analytical model to investigate the relationship between the IDPS performance and the rules mode selection. In particular, we analyze the IDPS rule-checking process along with its consequent action (*i.e.*, alert or drop) on the resulting security of the network, and on the average service time per event. Simulation was conducted to validate our performance analysis study. Our results show that applying different sets of rules categories and configuration parameters impacts average service time and affects system security. The results demonstrate that it is desirable to strike a balance between system security and network performance.**

***Index Terms*—Security Performance Evaluation, Security Management, Security Configuration.**

I. Introduction

Among other security enforcement tools and mechanisms, Intrusion Detection and/or Prevention Systems (IDPSs) represent an important line of defense against a variety of attacks that can compromise the security and proper functioning of an enterprise information system [1]. IDPSs can be signature-based or anomaly-based. Signature-based IDPSs, such as SNORT [2] and BRO [3], are the most popular and are based on the pre-knowledge of attack signatures which help distinguish between malicious and benign traffic. Anomaly-based IDPSs work differently in that they learn about the normal behavior of a system and then raise alerts whenever an abnormal behavior is detected. IDPSs can be network or host-based and can operate in centralized or distributed clusters in order to provide better detection of malicious traffic across a distributed networked system.

One of the major requirements for deploying any security technology is to defend against the variety of attacks. Another requirement is related to the avoidance of any unnecessary network performance degradation when maximum security is applied. This results in a tradeoff between security enforcement levels on one side and the performance and usability of an enterprise information system on the other [4]. Existing IDPSs do not seem to provide a satisfactory method of achieving the two conflicting goals mentioned above. Network-based Intrusion Detection Systems (NIDSs) inspect copies of the packets that are transmitted over the network and generate alerts whenever malicious contents are found. In contrast, Network-based Intrusion Prevention Systems (NIPSs) have an extra ability to prevent the attacks from being successful. IDSs fulfill the network performance requirement (in terms of delay) but exhibit a poor protection as the attacks have already succeeded. On the other hand, IPSs can protect the network by dropping the malicious packets that match any attacking pattern; however, this can have a negative impact on the network performance in terms of delay as the attacking patterns increase.

Although many IDPS systems have been proposed, their appropriate configuration and control for effective attacks detection/prevention and efficient resources consumption has always been challenging [5], [6]. The evaluation of the IDPS performance for any given security configuration is a crucial step for improving their realtime capability [7]. Another concern is related to the impact of security enforcement levels on the performance and usability of an enterprise information system. In this paper, we study the impact of security enforcement levels on the performance and usability of an enterprise information system. In particular, we analyze the impact of configuring an IDPS rule-checking process along with its consequent action (i.e. alert or drop) on the resulting security of the network, and on the average service time per event. We develop a new analytical model to investigate the relationship between the IDPS performance and its configuration. Our results show that applying different sets of rules categories and configuration parameters impacts average service time and affects system security. The results demonstrate that it is desirable to strike a balance between system security and network performance.

The paper proceeds with an overview of related work in Section II, then presents a background of the rule-checking process in Section III. In Section IV, we present an analytical model to investigate the relationship between the

IDPS performance and the rules mode selection. Section V presents the performance analysis study of the impact of IDPS configuration on average service time. It also describes the relationship between the system security level and deferring configuration parameters. Section VI, discusses the challenges encountered in IDPS performance analysis. Finally, Section VII concludes the paper and anticipates the nature of future work.

II. Related Work

A signature-based IDPS heavily relies on deep packet inspection. Studies show that the IDPS rule checking process is a performance bottleneck [8], [9], [10]. Accordingly, researchers focus on finding solutions and algorithms, either software or hardware, to improve the performance of the content-matching process. However, very little work has addressed the problem of dynamic adaptation for the sake of balancing system performance and security.

There have been some efforts in measuring the IDPS performance in terms of resource requirements (*i.e.*, CPU and memory). In [11], [12], the authors aim to fine tune the trade-off between security level versus resource consumption. Our study goes a step further by analyzing the impact of IDPS configuration on average service time.

Lee et al. [13] propose a technique to measure the performance of an IDS by quantifying the benefits and costs of detection rules. They aim to dynamically determine the optimal configuration for an overloaded IDS to prevent data dropping under resource constraint and to trigger adaptation to current conditions. Their work is similar to ours in that it measures the expected service time of different IDS configuration sets to determine the optimal one. However, defining the cost and benefit metrics precisely is not an easy task and it varies from one environment to another. Furthermore, considering the preventive capability of an IDPS, the analysis presented by Lee et al. seems inadequate. This is due to violation of the strict QoS requirement in terms of end-to-end delay caused by the prevention services.

A study measuring the impact of the IPS operation on network performance is described in [14]. The authors explain the network performance degradation when intrusion prevention services are applied. Accordingly, they suggest distributing the IPS services on programable routers to mitigate this issue. In fact, adding a deep packet inspection operation to routers will certainly cause longer delay since they are not designed for this purpose.

The authors of [15] seek to transform an IDS system into an IPS by proposing a policy management for firewall devices integrated with intrusion prevention capabilities. They propose an attack response matrix model which maps intrusion types to traffic enforcement actions. Their proposal is, however, only at the design level and no concrete implementation or policy specifications have been provided. In addition, they do not consider the performance aspect but only how to transform an IDS into an IPS using policies.

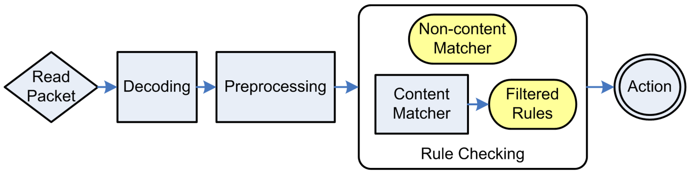


Fig. 1. Analysis Tasks for Intrusion Detection and Prevention Systems

III. Background and Problem Description

In this section, we describe the operation of existing intrusion detection and prevention systems and some of the weaknesses inherent in them. Generally, IDPSs perform a number of analysis tasks to identify malicious traffic. `SNORT`, for example, carries out the following tasks (Fig. 1):

- Data decoding: decodes the header information of the packet and translates specific protocol elements into a data structure, for the use of the following tasks.
- Preprocessing: examines the packet for malicious activity that can not be captured by signature matching or performs a number of preliminary steps in the packet, *i.e.*, normalization, fragmentation reassembly, stream reconstruction, etc.
- Rule checking: examines the packet to determine if it is associated with an intrusion. There are two types of rules an IDPS can handle: content-based and non-content-based. The former is divided into three main sections: 1) action to be taken, 2) header specifying protocol, IP addresses, and ports information, and 3) an option stating which parts of the packet should be inspected for determining the presence of a particular pattern, or a collection of patterns. The non-content-based rule is similar to the content-based one except that there is no pattern to look for.
- Action execution: the action describes what response an IDPS can perform when a packet matches a specified rule. The main actions include (but are not limited to): logging a packet (log), generating an alert (alert), dropping a packet (drop), terminating a connection (reject), and ignoring a packet (pass).

Our analysis will be limited to the rule-checking process along with the action associated with each rule. Once rules are selected and initialized, they are grouped by protocol type (*i.e.*, tcp, udp, icmp, ect.), and then by ports, then by those with content and those without. For each content-based group, a multi-pattern matcher is constructed for all rules by choosing a single pattern from all patterns in each rule option (*e.g.*, `SNORT` uses longest pattern). Clearly, there is no pattern matcher for non-content-based rules. When a packet arrives at the rule-checking engine, the corresponding multi-pattern matcher will be called on to filter out (for further evaluation) the rules whose single pattern are matched. The filtered rules can be large depending on the chosen patterns for the multi-pattern matcher and on the number of rules within a group(*i.e.*, http). The evaluation of these filtered rules and

TABLE I
SUMMARY OF NOTATIONS

Symbol	Meaning
$\mathcal{R}$	Set of Detection and Prevention rules in IDPS.
$\mathcal{N}$	Number of rules contained by IDPS.
E	An arriving event.
$\mathcal{G}$	Binary vector indicating whether a rule is a detective or preventive rule
$\mathcal{A}$	Set of attacks covered by IDPS
$\mathrm{P_M}$	Prior probability of attack occurrence
FP	False positive probability for the detection and prevention of IDPS
FN	False negative probability for the detection and prevention of IDPS
$\mathrm{T}(r_i)$	Processing time for rule r_i
H(k)	Vector indicating the proportion of malicious event of type i.
$B(i)$	The blocking probability of a preventing rule r_i

the non-content-based rules are applied sequentially. Once a rule matches a packet the corresponding action will be taken.

A rule can be either a detective or a preventive rule. A detective rule's action is `alert` and a preventive rule's action is `drop`. The detective rule aims to inspect a copy of a packet transmitted over the network and generate an alert when a malicious pattern exists in the packet content. Clearly, this passive inspection mode has no impact on the network performance in terms of delay, as it checks only a copy of traffic for malicious activity, while the actual traffic is delivered successfully. However, this inspection mode exhibits a poor protection as it does not prevent an attack from succeeding. Unlike the detective rule, the preventive rule is configured to be in-line mode so that traffic will be dropped if it carries a malicious pattern that matches the rule. This preventive mode can meet the security requirement but it can have a negative impact on the network performance, especially when the attacking patterns increase.

A. Definitions and Preliminaries

IDPS rules (or signatures) are classified into libraries. Each library contains a number of rules that are related to a known attack type. For example, the FTP library contains all rules related to attacks on FTP servers (i.e., SNORT (v 2.8.6) has a set of 58 libraries). Let $\mathcal{L} = \{l_1, l_2, \ldots, l_M\}$ be the set of all available libraries. We let $\mathcal{R}(l_i) = \{r_1, r_2, \ldots, r_{N_i}\}$ denote the set of a finite number of rules included in a library l_i. The total number of rules included in an IDPS is $N = |\mathcal{R}| = \sum N_i$, where $i = 1, 2, \ldots, M$.

IDPSs are shipped with a large number of rules. The security administrator is responsible for including and excluding rules according to the specific needs of the protected network environment. For instance, SNORT allows the enabling/disabling of rule libraries or individual rules through a set of configuration files. Furthermore, the security administrator can specify the mode of the rules as either in a detective or a preventive mode. To classify the rule as to which group it belongs to, we define a binary vector $\mathcal{G} = \{g_1, g_2, \ldots, g_N\}$ that indicates whether a rule is a detective or preventive rule (i.e., detection mode if $G(k) = 0$, prevention mode if $G(k) = 1$, where $k = 1, 2, \ldots, N$). This binary vector is defined as corresponding to rules vector $\mathcal{R}$ with $\mathcal{N}$ rules.

Each rule r_k has a processing time T_k. We consider only the time that it takes a rule to process an actual packet. Clearly, a detective rule that simply examines a copy of traffic is assumed to require no processing time on the actual traffic. The processing time t_k will be considered only if the rule r_k is in a preventive mode ($G(k) = 1$).

Each rule $r_k \in \mathcal{R}$ is responsible for only one type of malicious event. We let $\mathcal{A} = \{a_1, a_2, \ldots a_N\}$ be the set of different attacks covered by the IDPS, assuming that the occurrence of each attack is independent of the others.

We denote E as an arriving event or flow. The event E is malicious with attack of type k where $k \in \{0, 1, \ldots, N\}$ and is denoted as $\mathrm{E} \leftarrow a_k$. Note that an event contains at most only one type of maliciousness. We denote by $\mathrm{E} \leftarrow a_0$ a benign event which does not contain any malicious content with regards to the different rules' restrictions R_i ($i = 1, 2, \ldots, N$).

A rule r_i announces event E as malicious with regards to attack type a_i is defined as $\mathrm{E} \overset{r_i}{\leftarrow} a_i$. Similarly, we define $\mathrm{E} \overset{r_i}{\leftarrow} a_0$ to indicate that the event E is announced as normal when no rule r_i reports the presence of attack a_i in it for all $i = 1, 2, \ldots, N$. The probability that rule r_k triggers an arriving event E as malicious, given that it is malicious with regards to attack type a_k is defined by: $\mathbb{P}\mathrm{rob}\{\mathrm{E} \overset{r_i}{\leftarrow} a_k \mid \mathrm{E} \leftarrow a_k\}$ which is equal to the true positive probability TP_k =1-FN_k. FN_k represents the false negative rate of rule r_k when miss-announcing a malicious event that contains an attack of type a_k. We let FP_k=$\mathbb{P}\mathrm{rob}\{\mathrm{E} \overset{r_i}{\leftarrow} a_k \mid \mathrm{E} \leftarrow a_0\}$ be the false positive rate of rule r_k, that is, the probability that rule r_k triggers an arriving event E as malicious, given that it is not malicious with regards to rule r_k.

B. Characterization of Traffic

A site-specific risk analysis provides information about the malicious activities that were encountered in the past. We believe that the risk analysis process is an important step to quantitatively measure the network security. However, our focus is not on developing a risk analysis model rather we are trying to benefit from information gathered by security administrators during the site-specific risk analysis process which includes the proportion of malicious events among all detected events, prior probability of maliciousness, false positive rate, and false negative rate. We mentioned the risk analysis model here for the sake of showing the feasibility of obtaining such parameters.

We denote P_M as the probability of maliciousness that categorizes an arriving event E to be malicious. This prior probability can be used to estimate future attacks. We denote by H(k) the vector indicating the proportion of malicious event of type i among all the malicious events for all i=1,...,N. Clearly, the sum of this vector is equal 1 ($\sum$H(i)=1, $i = 1, \ldots, N$).

IV. Analytical Model

In this section, we develop an analytical model to study the impact of the vector $\mathcal{G}$ on the resulting security of an enterprise information system and on the average service time to inspect an event. We assume that the IDPS processes one event at a time. Once an event arrives, it goes through a sequence of detection and/or prevention rules according to the current configuration of the IDPS represented by vector $\mathcal{G}$. The process terminates if the event is dropped by a preventive rule or reported by a detective rule as a malicious event. In case an event is normal, the process ends when all rules are checked.

A. Average Processing Time

In this section, we evaluate the average service time of an IDPS. It is the time required by the IDPS with a rule configuration $\mathcal{G}$ to successfully determine whether an arriving event is accepted as a normal event or reported/rejected with the presence of an attack in it. In the rule analysis process, preventive rules have a great impact on the service time of an IDPS. For example, a significant improvement in processing time can be achieved if a frequently triggered preventive rule is checked as early as possible because unnecessary analysis is avoided. We define $B(i)$ as the blocking probability of rule r_i. It is the probability of announcing an event as malicious by a preventing rule r_i, $\forall\ i = 1,\ldots, N$. The blocking probability of rule r_i is defined by:

$$B(i) = \mathbb{P}\text{rob}\{\text{E} \overset{r_i}{\leftarrow} a_i, \mathcal{G}(i)\} \tag{1}$$

where an event E is announced as malicious with an attack of type a_i by a rule r_i and the rule is a preventive rule, $\mathcal{G}(i)$=1.

In order for rule r_i to announce an event as malicious, all previous rules have to announce it as safe. In other words, an event should arrive at rule r_i before any decision is taken on it. This can be expressed as follows:

$$B(i) = \mathbb{P}\text{rob}\{\text{E} \overset{r_j}{\leftarrow} a_0(\forall j < i), \text{E} \overset{r_i}{\leftarrow} a_i\}\mathcal{G}(i) \tag{2}$$

Given that the event space consists of a malicious event of attack type k ($\text{E} \leftarrow a_k$) and benign event $\text{E} \leftarrow a_0$, we rewrite $B(i)$ as follow:

$$\begin{aligned} B(i) = \mathbb{P}\text{rob}\{&\text{E} \leftarrow a_k, \text{E} \overset{r_j}{\leftarrow} a_0(\forall j < i), \text{E} \overset{r_i}{\leftarrow} a_i\}\mathcal{G}(i) \\ &+ \mathbb{P}\text{rob}\{\text{E} \leftarrow a_0, \text{E} \overset{r_j}{\leftarrow} a_0(\forall j < i), \text{E} \overset{r_i}{\leftarrow} a_i\}\mathcal{G}(i) \end{aligned} \tag{3}$$

Let us consider the situation when the event is malicious. Clearly, the probability of announcing an event as malicious by rule r_i depends on the probability that the event is malicious and on the probability of accepting the event as normal by all the rules previously checked. We let the first term of Equation 3 be $B_{mal}(i)$ and using the theorem of total and conditional probability, $B_{\text{mal}}(i)$ can be written as:

$$\begin{aligned} B_{\text{mal}}(i) = &\sum_{k=1}^{N} \mathbb{P}\text{rob}\{\text{E} \overset{r_i}{\leftarrow} a_i \mid \text{E} \overset{r_j}{\leftarrow} a_0(\forall j < i), \text{E} \leftarrow a_k\} \\ &\times \mathbb{P}\text{rob}\{\text{E} \overset{r_j}{\leftarrow} a_0(\forall j < i), \text{E} \leftarrow a_k\} \end{aligned} \tag{4}$$

The first term in Equation 4 represents the case when the IDPS announces the event as malicious by rule r_k given that the event arrives to rule r_i and it is malicious. In this case, the probability that the IDPS correctly announces the event as malicious or mistakenly classifies it as malicious is defined as PB_{mal}. This can be calculated as follows:

$$\text{PB}_{\text{mal}}(k,i) = \begin{cases} 1 - \text{FN}_i & \text{if } k = i \\ \text{FP}_i & \text{if } k \neq i \end{cases} \tag{5}$$

We let PE_{mal} stand for the second term of Equation 4, which represents the probability that the IDPS accepts the event as normal by all rules r_j, j=1,. . . ,i-1, earlier than the current evaluated rule r_i where the event E is malicious. We have two cases in this situation. In the first case, the current evaluated rule r_i is the first one (i=1), where no rule has been checked so far. PE_{mal} can be calculated as:

$$\text{PE}_{\text{mal}}(k,i) = H(k)P_M \quad \text{where } i = 1. \tag{6}$$

The second case of PE_{mal} may be encountered when there is at least one rule r_j that has been checked before rule r_i; that is, r_i is not the first rule to be evaluated (*i.e.*, $i > 1$). Accordingly, PE_{mal} can be calculated by:

$$\text{PE}_{\text{mal}}(k,i) = \begin{cases} \prod_{j=1}^{i-1} \Big(1 - (1 - \text{FN}_j)\mathcal{G}(j)\Big)\text{H}(k)P_M & \text{if } k \neq j \\ \prod_{j=1}^{i-1} \Big(1 - \text{FP}_j\mathcal{G}(j)\Big)\text{H}(k)P_M & \text{if } k = j \end{cases} \tag{7}$$

Now let us consider the situation when the event is normal. We are interested in the probability of announcing an event as malicious by rule r_i given that the event is safe and all previously evaluated rules r_j (*i.e.*, $j < i$) mark the event as safe. This can be written as:

$$\begin{aligned} B_{\text{safe}}(i) = \mathbb{P}\text{rob}\{&\text{E} \overset{r_i}{\leftarrow} a_i \mid \text{E} \overset{r_j}{\leftarrow} a_0(\forall j < i), \text{E} \leftarrow a_0\} \\ &\times \mathbb{P}\text{rob}\{\text{E} \overset{r_j}{\leftarrow} a_0(\forall j < i), \text{E} \leftarrow a_0\} \end{aligned} \tag{8}$$

Applying the same steps used for the malicious case yields the following equation in a safe case:

$$B_{\text{safe}}(i) = \text{FP}_i \times \begin{Bmatrix} 1 - P_M & \text{if } i = 1 \\ \prod_{j=1}^{i-1} \Big(1 - \text{FP}_j\mathcal{G}(j)\Big)(1 - P_M) & \text{if } i > 1 \end{Bmatrix} \tag{9}$$

Given the Equations 3, 4, and 9, we can calculate the blocking probability $B(i)$ of rule i as follows:

$$B(i) = \text{B}_{\text{mal}}(i) \times \mathcal{G}(i) + B_{\text{safe}}(i) \times \mathcal{G}(i) \tag{10}$$

where

$$\text{B}_{\text{mal}}(i) = \sum_{k=1}^{N} \text{PB}_{\text{mal}}(k,i) \times \text{PE}_{\text{mal}}(i) \tag{11}$$

Finally, we measure the average service time of an IDPS as follows:

$$\begin{aligned} Avg = &\left[\sum_{i=1}^{N} B(i) \sum_{k=1}^{N} T(k)G(k)\right] + \Big(1 - \sum_{i=1}^{N} B(i)\Big) \\ &\times \sum_{i=1}^{N} T(i)G(i) \end{aligned} \tag{12}$$

B. Level of Security

The main objective of deploying any security tool is to protect the network from any malicious activities. Measuring the impact of security configurations can help security administrators in making optimal decisions about how to strengthen network security. In IDPSs, rules in preventive mode have the capability of blocking attacks once they have been matched. However, this induces a negative impact on network performance (*i.e.*, E2E delay, throughput, service usability, jitter, etc.) especially when the number of preventive rules increases. Therefore, the main concern is to find the appropriate balance between security enforcement levels and the performance and usability of an enterprise information system. Here, we evaluate the impact of a chosen IDPS configuration on the resulting security of the system. In particular, we are interested in measuring the probability of blocking an event given that it is malicious.

$$\begin{aligned} S &= \mathbb{P}\text{rob}\{\text{E} \overset{r_j}{\leftarrow} a_i \mid \text{E}{\leftarrow}a_i\} \\ &= \frac{\mathbb{P}\text{rob}\{\text{E} \overset{r_i}{\leftarrow} a_i, \text{E}{\leftarrow}a_i\}}{\mathbb{P}\text{rob}\{\text{E}{\leftarrow}a_i\}} \\ &= \frac{\sum_{i=1}^{N} \mathbb{P}\text{rob}\{\text{E} \overset{r_i}{\leftarrow} a_i, \text{E} \overset{r_i}{\leftarrow} a_i\} \times \mathcal{G}(i)}{\mathbb{P}\text{rob}\{\text{E}{\leftarrow}a_i\}} \end{aligned} \tag{13}$$

Using Equation 3 in Equation 13 yields:

$$S = \frac{\sum_{i=1}^{N} \sum_{k=1}^{N} \text{PB}_{\text{mal}}(i) \times \text{PE}_{\text{mal}}(i) \times \mathcal{G}(i)}{P_M} \tag{14}$$

C. Accuracy of Action

The capability of an IDPS to apply different rule modes (*i.e.*, alert or block) motivated the need for measuring action accuracy. Therefore, we study the action that is taken by the IDPS against an arriving event. The action of the IDPS could be either accepting or blocking an event. The accuracy of action is defined as taking the right action with regards to an arriving event. That is, the action of the IDPS is accurate if it accepts an event that is normal and/or blocks a malicious event. We define the accuracy of action A_{acc} as the probability of either accepting a benign event or blocking a malicious one. A_{acc} can be written as follows:

$$A_{acc} = \mathbb{P}\text{rob}\{\text{E}{\leftarrow}a_i, \text{E} \overset{r_i}{\leftarrow} a_i\} + \mathbb{P}\text{rob}\{\text{E}{\leftarrow}a_0, \text{E} \overset{r_i}{\leftarrow} a_0\} \tag{15}$$

Using the conditional probability theorem yields:

$$\begin{aligned} A_{acc} &= \mathbb{P}\text{rob}\{\text{E} \overset{r_i}{\leftarrow} a_i \mid \text{E}{\leftarrow}a_i\}\mathbb{P}\text{rob}\{\text{E}{\leftarrow}a_i\} \\ &+ \mathbb{P}\text{rob}\{\text{E} \overset{r_i}{\leftarrow} a_0 \mid \text{E}{\leftarrow}a_0\}\mathbb{P}\text{rob}\{\text{E}{\leftarrow}a_0\} \end{aligned} \tag{16}$$

By substituting 14 in 16, the action accuracy taken by the IDPS for an arriving event is as follows:

$$A_{acc} = S \times P_M + \prod_{i=1}^{N} \Big(1 - FP_i\mathcal{G}(i)\Big) \times (1 - P_M) \tag{17}$$

D. Accuracy of Decision

In this section, we analyze the decision accuracy made by the IDPS. This refers to the decision to announce an arriving event as malicious or not, regardless of the action taken as a result of the announcement. The decision is accurate when announcing an arriving malicious event as malicious while not doing so with the benign one. Therefore, the accuracy of the decision is defined as the probability of either triggering an event as malicious while it is malicious or not triggering the event when it is normal. This is equivalent to the complement of making a wrong decision with regards to an arriving event. That is, the decision accuracy of the IDPS is the complement of announcing an event as malicious where it is not malicious or announcing a benign event as malicious. The inaccuracy of the decision can be written as follows:

$$\overline{D}_{acc} = \mathbb{P}\text{rob}\{\text{E}{\leftarrow}a_i, \text{E} \overset{r_i}{\leftarrow} a_0\} + \mathbb{P}\text{rob}\{\text{E}{\leftarrow}a_0, \text{E} \overset{r_i}{\leftarrow} a_i\} \tag{18}$$

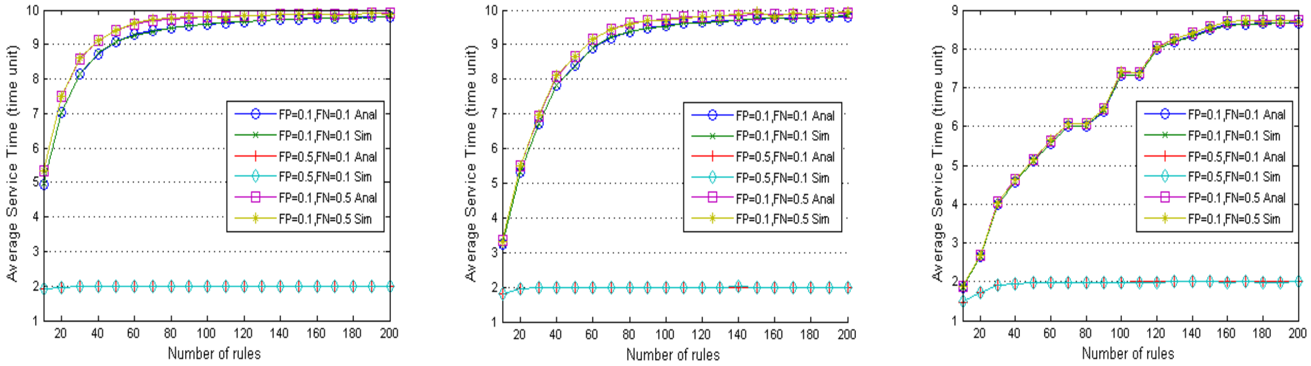
Solving this equation results in:

$$\begin{aligned} \overline{D}_{acc} &= \Big(1 - \prod_{i=1}^{N} \big(1 - FP_i\big)\Big) \times (1 - P_M) \\ &+ \sum_{k=1}^{N} FN_k \prod_{\substack{i=1 \\ i \neq k}}^{N} \big(1 - FP_i\big) \times H(k)P_M. \end{aligned} \tag{19}$$

The difference between the action and decision accuracy is that the former concerns the response taken by a triggered rule to either block an arriving event or to accept it. The latter concerns the decision of announcing an arriving event as malicious or benign. Clearly, the rule's mode of an IDPS has no impact on the decision accuracy but it affects the action accuracy. For instance, the action of an IDPS is considered to be inaccurate if a malicious event matches a specified rule which is in a detective mode (alert). This is because the malicious traffic has not been blocked by the detective rule. The action accuracy can be identical to the decision accuracy when the IDPS is configured to operate entirely in IDS mode (i.e., the action of all the rules is alert) or entirely in IPS mode (see Table II when IPS=100%).

V. Performance Evaluation and Results

In this section, we study the impact of the IDPS configuration on the average service time. We also measure the security level of the system when choosing different configuration parameters. The results are derived using both analytical and simulation approaches. The simulations are performed using a new discrete-event simulation tool developed under Matlab [16]. In order to test the validity of our work while keeping the case simple, we assign equal processing time for all the rules (one unit of time), we let the proportion of maliciousness be equally distributed $H(i)/\mathcal{N}$, $i = 1, \ldots, N$, and we set the probability of maliciousness to be P_M=0.5. For simplicity, we assume that the false detection rates, FP and FN, are measured for the entire rule-checking engine of an IDPS.



(a) (P_M,IPS)=(0.5,75%)

(b) (P_M,IPS)=(0.5,50%)

(c) (P_M,IPS)=(0.5,10%)

Fig. 2. Selected Results of Average Service Time for Number of Rules with Different Preventive Percentage

A. Average Service Time

The average service time is calculated as the time required for an event to be completely served. An event is served once a detective/preventive rule finds a match and triggers its action or once the event is identified as normal. Figure 2 shows the impact of the increasing number of rules $\mathcal{N}$ on the average service time when processing an event using different preventive rule percentages. For each preventive percentage, we plot the average service time using different detection rates. Figures 2(a), 2(b) and 2(c) show the results when selecting the percentage of preventive rules to be 75%, 50%, and 10% respectively.

Figure 2(a) illustrates the impact of increasing the number of rules on the average service time when 75% of the rules are in preventive mode. Figure 2(a) shows that the larger $\mathcal{N}$ is, the longer service time becomes. When we reach approximately 100 rules, the system reaches the saturation state and additional rules result in very little further impact. Prior to the saturation region, a reduction in the number of rules results in an improvement in average service time. In this situation, a malicious event is likely to be missed by the rule accountable for it and accordingly is examined by other rules that are not responsible for it. Thus, a reduction in the false negative rate yields no appreciable improvement in average service time. The false negative rate has little impact on the average service time. However, this is not the case when the value of FP increases. Rather, there is a notable reduction in average service time when the rate of false positives increases. This occurs because of an early decision made as a result of a wrong diagnosis. In this case, an increase in the number of rules has only a slight impact on average service time. When the percentage of preventive rules is 50%, Figure 2(b) appears quite similar to Figure 2(a). Increasing the number of rules, false positives, and false negatives impacts the average service time in a way very similar to that illustrated in Figure 2(a). However, the average service time approaches the saturation state more slowly than in the previous case when 75% of the rules are preventive. That is, the improvement in average service time is limited once the saturation point of approximately 130 rules is reached (out of 200 rules).

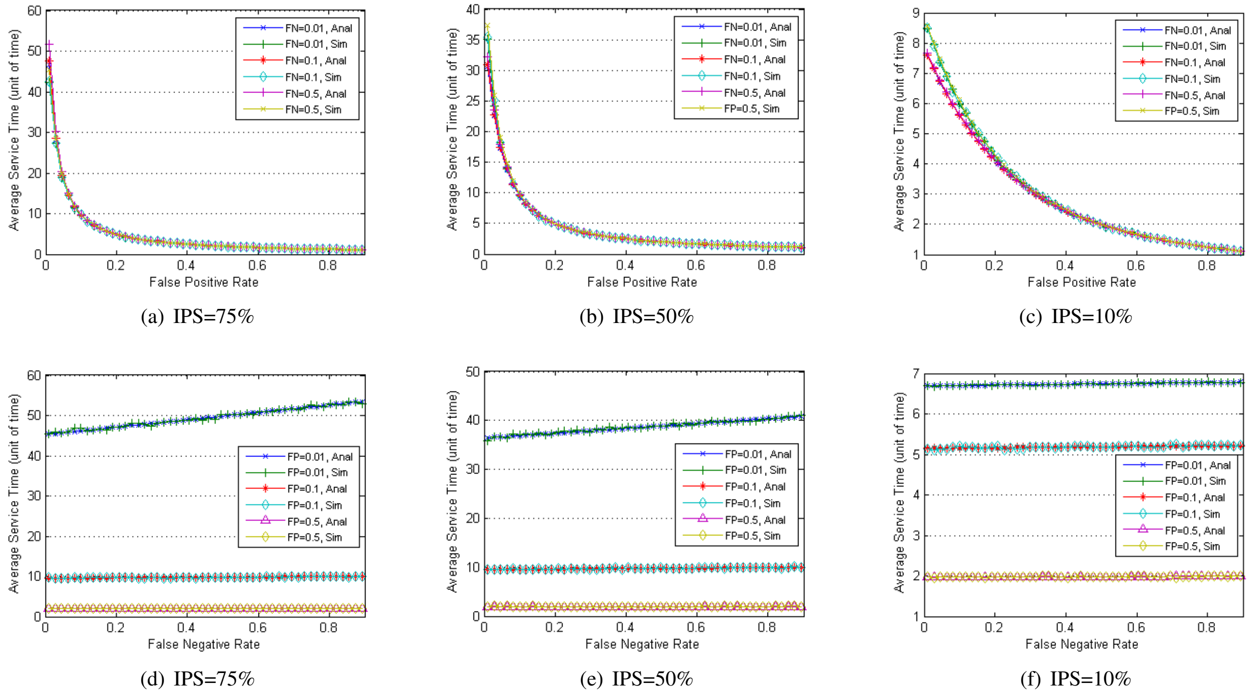
In Figure 2(c), when 10% of the rules are in preventive mode, we observe a somewhat different impact on the average service time. Figure 2(c) shows that the average service time increases linearly with the number of rules. Improvement in average service time in this case is obviously the most advantageous of the three scenarios, because the saturation point is not reached until approximately 180 rules.

To conclude, the potential for improvement in average service time increases as we reduce the percentage of preventive rules. Of course, the price one has to pay for reduction in the number of preventive rules is a corresponding decrease in enterprise network security.

Figure 3 plots the average service time as a function of increasing both the false positive and false negative rates. We are interested in understanding the impact of the detection rates on the average time required to completely inspect an arriving event. The number of rules in this case is chosen to be 100. Figures 3(a), 3(b), and 3(c) show the impact of varying the false positive rate while fixing the false negative rate and using different preventive rule percentages.

Figure 3(a) presents the results when 75% of the rules are preventive. We can see that the average service time decreases with an increase in the false positive rate for all false negative rate values. We can see that the average service time is longer when the IDPS becomes accurate in terms of the false positive rate, no matter what the false negative rates are. Indeed, the IDPS consumes more time to correctly distinguish the malicious events from the benign ones rather than just mistakenly identifying a malicious event at an early stage.

Figures 3(b) and 3(c) show similar results, except that a dramatic decrease in average service time results from the reduction of the percentage of preventive mode services. However, with the use of different preventive mode percentages, the average service time approaches saturation differently. That is, when the IPS percentage is large, the average service time reaches saturation quickly. In contrast, the saturation state is reached more slowly as the IPS percentage decreases. Clearly,



(a) IPS=75%

(b) IPS=50%

(c) IPS=10%

(d) IPS=75%

(e) IPS=50%

(f) IPS=10%

Fig. 3. Selected Results for Impact of Detection Rates on Average Service Time

the impact on average service time will be no more than 10 units of time when 10% of the rules are in IPS mode (see Figure 3(c)).

Figures 3(d), 3(e), and 3(f) plot the impact of changing the false negative detection rate (FN) for different false positive rates and with the use of different preventive rule percentages. Clearly, a reduction in the percentage of preventive rules results in a significant reduction in average service time. Furthermore, an increase in the rate of false negatives produces very little change in average service time.

B. Level of Security and Accuracy

In this section, we intend to study the impact of choosing different configuration parameters on the security of the system and on the action and decision accuracy. The results are achieved using both the analytical and the simulation models that we have developed. For all results in this section, we based our study on a total of 100 rules. Table II presents the impact of varying four configuration parameters, including FP, FN, P_M, and IPS% on security and accuracy of the system. Clearly, an increase in the preventive rule percentage yields a corresponding improvement in system security. Nevertheless, a system still has a good level of security even though the percentage of preventive rules is relatively low. For example, when FP is 0.5, FN is 0.1, and IPS is 10%, the security result reach 79%. However, the accuracy of action and decision for this case are not satisfactory.

TABLE II
SELECTED RESULTS OF SECURITY LEVEL, DECISION ACCURACY, AND ACTION ACCURACY WITH DIFFERENT CONFIGURATION SETS

Configuration Parameters			Analytical Results			Simulation Results		
FP	FN	P_M	Security	Action Accuracy	Decision Accuracy	Security	Action Accuracy	Decision Accuracy
10% IPS								
0.1	0.1	0.1	0.3340	0.7624	0.4099	0.3263	0.7611	0.4104
0.1	0.1	0.5	0.3340	0.5720	0.6550	0.3310	0.5687	0.6565
0.5	0.1	0.1	0.7900	0.3040	0.1009	0.7939	0.3056	0.1007
0.1	0.5	0.1	0.2620	0.7552	0.3944	0.2654	0.7548	0.3942
0.1	0.5	0.5	0.2620	0.5360	0.5775	0.2597	0.5354	0.5778
0.5	0.5	0.1	0.7500	0.3000	0.1008	0.7493	0.2991	0.1003
50% IPS								
0.1	0.1	0.1	0.6720	0.5986	0.4099	0.6650	0.5956	0.4094
0.1	0.1	0.5	0.6720	0.6312	0.6550	0.6723	0.6315	0.6555
0.5	0.1	0.1	0.9813	0.1263	0.1009	0.9790	0.1272	0.1028
0.1	0.5	0.1	0.5407	0.5855	0.3944	0.5461	0.5862	0.3959
0.1	0.5	0.5	0.5407	0.5656	0.5775	0.5373	0.5631	0.5774
0.5	0.5	0.1	0.9688	0.1250	0.1008	0.9662	0.1233	0.0995
100% IPS								
0.1	0.1	0.1	0.9613	0.4099	0.4099	0.9578	0.4113	0.4113
0.1	0.1	0.5	0.9613	0.6550	0.6550	0.9616	0.6559	0.6559
0.5	0.1	0.1	0.9998	0.1009	0.1009	0.9998	0.0993	0.0993
0.1	0.5	0.1	0.8063	0.3944	0.3944	0.8068	0.3952	0.3952
0.1	0.5	0.5	0.8063	0.5775	0.5775	0.8081	0.5771	0.5771
0.5	0.5	0.1	0.9990	0.1008	0.1008	0.9991	0.1006	0.1006

VI. DISCUSSION

The challenge involved in performance analysis of a security system so as to reduce resource utilization while preserving a good level of security enforcement is the need to obtain estimates for the various parameters used in the analysis. Finding the false positive (FP) and false negative (FN) rates can be accomplished by either using proper training data sets or by analyzing the past behavior of the system [17]. The rule processing time T can be measured experimentally. SNORT, for instance, provides statistics on rule performance through a simple configuration option (*i.e.*, `profile rules`). For each rule, SNORT provides an estimate of how much it takes to process a packet. The prior probability of attack occurrence (P_M) and the proportion of attacks (H) can be initially estimated using a site-specific risk analysis approaches and updated with new attacks accordingly. Although dealing with a prior probability is a challenging task, our target is rather measuring the performance of the security system.

An additional concern which affects the performance analysis is the selection of categories (sets of rules) for each preventive level. The selection of categories may vary form one environment to another. For instance, the maximum preventive level for protecting the web server of a company should include not only all the rules which prevent web-server-specific attacks, but also those related to potential preliminary steps of these attacks, such as scanning. A possible solution for choosing the categories for each detection level can be based on common attack graphs [18] where the early steps of the attacks are included in the minimum prevention level.

VII. Conclusion

In this paper we studied how choosing which rules are preventive or detective has an impact on the security of the system, on the average service time, and on the decision and action accuracy of an IDPS. We developed a new analytical model to investigate the relationship between IDPS performance and its configuration. Simulation was conducted to validate our performance analysis study. Our results show that applying different sets of rules categories and configuration parameters impacts average service time and affects system security. The results demonstrate that it is desirable to strike a balance between system security and network performance in terms of delay. Ongoing work is considering the investigation of attack graphs, attack statistical relationships, as well as learning mechanisms. The intent is to determine an appropriate IDPS configuration that will balance network security and performance. We also plan to validate our analysis using real IDPS systems such as SNORT and BRO.

Acknowledgment

This work was supported in part by the Natural Science and Engineering Council of Canada (NSERC) under its Discovery program, and in part by the WCU (World Class University) program through the Korea Science and Engineering Foundation funded by the Ministry of Education, Science and Technology (Project No. R31-2008-000-10100-0).

References

[1] K. Scarfone and P. Mell, “Guide to intrusion detection and prevention systems(idps),” *National Institute of Standards and Technology (NIST)*, no. CSRC special publication SP 800-94, Feb 2007.

[2] M. Roesch, “Snort - lightweight intrusion detection for networks,” in *LISA ’99: Proceedings of the 13th USENIX conference on System administration*. Berkeley, CA, USA: USENIX Association, 1999.

[3] V. Paxson, “Bro: a system for detecting network intruders in real-time,” in *SSYM’98: Proceedings of the 7th conference on USENIX Security Symposium, 1998*. Berkeley, CA, USA: USENIX Association, 1998.

[4] K. Alsubhi, I. Aib, J. François, and R. Boutaba, “Policy-based security configuration management application to intrusion detection and prevention,” in *Proceedings of the 2009 IEEE international conference on Communications*, ser. ICC’09, 2009.

[5] H. Debar, M. Dacier, and A. Wespi, “Towards a taxonomy of intrusion-detection systems,” *COMPUTER NETWORKS*, vol. 31, no. 8, 1999.

[6] S. Bellovin and R. Bush, “Configuration management and security,” *IEEE Journal on Selected Areas in Communications JSAC*, vol. 27, no. 3, 2009.

[7] L. Schaelicke, T. Slabach, B. Moore, and C. Freeland, “Characterizing the Performance of Network Intrusion Detection Sensors,” *Recent Advances in Intrusion Detection: 6th International Symposium, RAID 2003, Pittsburgh, PA, Usa, September 8-10, 2003.*

[8] J. Cabrera, J. Gosar, W. Lee, and R. Mehra, “On the statistical distribution of processing times in network intrusion detection,” in *43rd IEEE Conference on Decision and Control, Atlantis, Paradise Island, Bahamas*, 2004.

[9] D. Schuff and V. Pai, “Design alternatives for a high-performance self-securing ethernet network interface,” in *IEEE International Parallel and Distributed Processing Symposium, IPDPS 2007.*

[10] C. Wu, J. Yin, Z. Cai, E. Zhu, and J. Chen, “A hybrid parallel signature matching model for network security applications using simd GPU.” Springer, 2009, pp. 191–204.

[11] H. Dreger, A. Feldmann, V. Paxson, and R. a. Sommer, “Predicting the resource consumption of network intrusion detection systems,” in *Recent Advances in Intrusion Detection*. Springer, 2008.

[12] H. Dreger, A. Feldmann, V. Paxson, and R. Sommer, “Operational experiences with high-volume network intrusion detection,” in *Proceedings of the 11th ACM conference on Computer and communications security*. ACM, 2004.

[13] W. Lee, J. Cabrera, A. Thomas, N. Balwalli, S. Saluja, and Y. Zhang, “Performance adaptation in real-time intrusion detection systems,” in *Recent Advances in Intrusion Detection*. Springer, RAID, 2002.

[14] A. Hess, H. Geerdes, and R. Wessäly, “Intelligent distribution of intrusion prevention services on programmable routers,” in *Proc. of 25th IEEE INFOCOM, Barcelona, Spain*. Citeseer, 2006.

[15] Y. Chen, Y. Yang, and I. WatchGuard Technologies, “Policy management for network-based intrusion detection and prevention,” in *Network Operations and Management Symposium, NOMS 2004. IEEE/IFIP.*

[16] www.mathworks.com.

[17] G. Gu, P. Fogla, D. Dagon, W. Lee, and B. Skorić, “Measuring intrusion detection capability: An information-theoretic approach,” in *Proceedings of the 2006 ACM Symposium on Information, computer and communications security*. ACM, 2006.

[18] P. Ning, Y. Cui, D. Reeves, and D. Xu, “Techniques and tools for analyzing intrusion alerts,” *ACM Transactions on Information and System Security (TISSEC)*, vol. 7, no. 2, 2004.