Security Configuration Management in Intrusion Detection and Prevention Systems

K. Alsubhi*

David R. Cheriton School of Computer Science,
University of Waterloo,
Waterloo, ON, Canada, N2L 3G1
E-mail: kaalsubh@cs.uwaterloo.ca
*Corresponding author

Y. Alhazmi

Electrical and Computer Engineering,
POSTECH, Pohang, KB 790-784, Korea
E-mail: yalhazmi@uwaterloo.ca

N. Bouabdallah

INRIA, Campus Universitaire de Beaulieu,
Rennes Cedex 35042, France
E-mail: nizar.bouabdallah@inria.fr

R. Boutaba

David R. Cheriton School of Computer Science,
University of Waterloo,
Waterloo, ON, Canada, N2L 3G1

and

Division of IT Convergence Engineering,
POSTECH, Pohang, KB 790-784, Korea
E-mail: rboutaba@uwaterloo.ca

Abstract: This paper aims to study 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 Intrusion Detection and Prevention System performance and the rules mode selection. In particular, we analyze the IDPS rule-checking process along with its consequent action on the resulting security of the network and on the average service time per event. Simulation was conducted to validate our performance analysis study. The results demonstrate that it is desirable to strike a balance between system security and network performance.

Keywords: security performance evaluation; security configuration management; IDPS; intrusion detection and prevention systems.

Reference to this paper should be made as follows: Alsubhi, K., Alhazmi, Y., Bouabdallah, N. and Boutaba, R. (2012) 'Security configuration management in Intrusion Detection and Prevention Systems', *Int. J. Security and Networks*, Vol. 7, No. 1, pp.30–39.

Biographical notes: Khalid Alsubhi received the Bachelor's Degree with first honour degree from The Faculty of Computing and Information Technology at King Abdulaziz University in 2003. He received the Master's Degree (MMath) in Computer Science from the University of Waterloo in 2008. He is currently a PhD student in David R. Cheriton School of Computer Science at the University of Waterloo.

Yassir Alhazmi received his BSc in Electrical and Computer Engineering in 2004 from Umm-AlQura University (UQU), Makkah, Saudi Arabia. He received his MSc in Electrical and Computer Engineering in 2010 from the University of Waterloo (UW), Waterloo, Canada and presently is working towards a PhD in Electrical and Computer Engineering at the University of Waterloo. His research interests include optimisation, power quality, smart grid and Electrical Vehicles (EVs).

Nizar Bouabdallah received the BS in Telecommunications Engineering from Ecole Superieur des Communications (SupCom), Tunis, Tunisia, in 2001 and the MS and PhD in Computer Science from the University of Paris VI, France, in 2002 and 2004, respectively. He was with Alcatel Research Laboratories, Marcoussis, France, from 2002 to 2004, while working on the PhD. In 2005, he was with North Carolina State University, Raleigh, as a postdoctoral fellow. He is currently a researcher at Institut National de Recherche en Informatique et en Automatique (INRIA). In 2007, he spent six months as a visiting researcher in the School of Computer Science, University of Waterloo, Canada. His research interests include wireless mesh and sensor networks, optical networking, resource allocation under QoS, network planning and modelling, as well as performance evaluation.

Raouf Boutaba received the MSc and PhD in Computer Science from the University Pierre & Marie Curie, Paris, in 1990 and 1994, respectively. He is currently a professor of computer science at the University of Waterloo. His research interests include network, resource and service management in wired, and wireless networks. He served as the founding editor in chief of the IEEE Transactions on Network and Service Management (2007–2010) and on the editorial boards of several other journals. He has received several best paper awards and other recognitions such as the Premiers Research Excellence Award, the IEEE Hal Sobol Award in 2007, the Fred W. Ellersick Prize in 2008, the Joe LociCero award and the Dan Stokesbury in 2009.

1 Introduction

Intrusion Detection and Prevention Systems (IDPSs) are a significant mechanism in defending against various attacks, which may interfere with security and the proper operation of an enterprise information system (Karen Scarfone and Peter Mell, 2007). These systems can be anomaly-based or signature-based. IDPSs based on signatures, such as SNORT (Roesch, 1999) and BRO (Paxson, 1998), are the most common and work with the foreknowledge of attack signatures to help distinguish between traffic, which is malicious and that which is benign. IDPSs based on anomalies are different, since they learn the regular behaviour of a system and then note when unusual behaviour is detected. IDPSs can be host-based or network-based, and can function in either distributed or centralised clusters to provide better recognition of hostile traffic in a distributed networked system.

A primary requirement for the deployment of any security technology is the need to protect against a range of attacks. Another need relates to avoiding any needless performance degradation in the network when maximum safety measures are applied. This requires a balance between security, on the one hand and speed and functionality, on the other (Alsubhi et al., 2009). Current IDPSs do not tend to provide an adequate means of achieving these two contradictory needs. Network-based Intrusion Detection Systems (NIDSs) inspect copies of the packets sent over the network and raise flags whenever hostile content is found. In comparison with this method, Network-based Intrusion Prevention Systems (NIPSs) have the extra capacity of defending against the attacks. IDSs realise network performance requirements but display poor defence capabilities, as attacks succeed. On the other hand, IPSs can shield networks by rejecting packets that match any hostile pattern, but this can negatively impact network performance as malicious attacks increase.

Although many IDPSs have been proposed, their appropriate configuration and control for effective attacks detection/prevention and efficient resources consumption has always been challenging (Debar et al., 1999; Bellovin and Bush, 2009). The evaluation of the IDPS performance for any given security configuration is a crucial step for improving their real-time capability (Schaelicke et al., 2003). Another concern is related to the impact of security enforcement levels on the performance and usability of an enterprise information system.

Building upon our previous work in Alsubhi et al. (2011) and Alsubhi et al. (2011), this paper aims to study the impact of security enforcement levels on the performance and usability of an enterprise information system. In particular, we analyse 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. We also propose a rule mode selection optimisation technique that aims to determine an appropriate IDPS configuration set to maximise the security enforcement levels while avoiding any unnecessary network performance degradation. The proposed method demonstrates that the application of various sets of rules categories and configuration parameters affects service time as well as system security.

Our results show that applying different sets of rules categories and configuration parameters impacts average service time and affects system security. As a result, it is advantageous to find a balance between security and performance to acquire a satisfactory technique that

does not cause computational time to suffer. Simulation was conducted to validate our proposed technique. 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 2, then presents a background of the rule-checking process in Section 3. In Section 4, we present an analytical model to investigate the relationship between the IDPS performance and the rules mode selection. Section 5 presents the rule mode selection optimisation technique. In Section 6, we present the IDPS performance analysis related to rules mode selection problem. It also describes the impact of IDPS configuration on average service time and the relationship between the system security level and varying configuration parameters. Finally, Section 7 concludes the paper and anticipates the nature of future work.

2 Related work

A signature-based IDPS heavily relies on deep packet inspection. Studies show that the IDPS rule-checking process is a performance bottleneck (Cabrera et al., 2004; Schuff and Pai, 2007; Wu et al., 2009; Dreger et al., 2008, 2004). 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.

Lee et al. (2002) put forward a method to determine the performance of an IDS through quantifying the benefits and drawbacks of detection rules. Their goal was to establish the best-possible configuration for an overloaded IDS to prevent the dropping of information under resource constraints and to elicit adjustment to existing conditions. Their work is comparable with ours in that it measures the service time of different IDS configuration sets to establish the best one. Nevertheless, defining the cost and benefit metrics accurately is not easy and varies from one environment to another. Moreover, in view of the preventive capacity of an IDPS, the analysis offered by Lee et al. seems insufficient. This is due to violation of the stringent QoS constraint in terms of end-to-end delay attributable to the prevention services.

The authors of Chen and Yang (2004) seek to convert an IDS system into an IPS by putting forward a policy management for firewall devices incorporated with intrusion prevention capabilities. They offer an attack response matrix template, which maps intrusion types to traffic enforcement responses. Their application is, however, only at the design stage and no firm implementation or policy parameters have been given. Also, they do not reflect on the performance aspect but only on how to convert an IDS into an IPS using policies. Consequently, a balance between performance, in term of delay and prevention ought to be considered when IDPS is applied.

There have been some efforts in measuring the IDPS performance in terms of resource requirements (i.e., CPU and memory). In Dreger et al. (2008) and Dreger et al. (2004), the authors aim to fine-tune the trade-off between security level and resource consumption; nevertheless, the impact of IDPS configuration on average service time has not been conducted. Thus, our study goes a step further by studying the impact of IDPS configuration on system performance.

A study measuring the impact of the IPS operation on network performance is described in Hess et al. (2006). The authors explain the network performance degradation when intrusion prevention services are applied. Accordingly, they suggest distributing the IPS services on programmable 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.

3 Background and problem description

In this section, we describe the operation of existing IDRSs 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 (Figure 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.
- Pre-processing: examines the packet for malicious activity that cannot be captured by signature matching or performs a number of preliminary steps in the packet, i.e., normalisation, fragmentation reassembly, stream reconstruction, etc.

Table 1 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 or packet.
$\mathcal{G}$	Binary vector indicating whether a rule is a detective or preventive rule
$\mathcal{A}$	Set of attacks covered by IDPS
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
$T(r_i)$	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

- 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

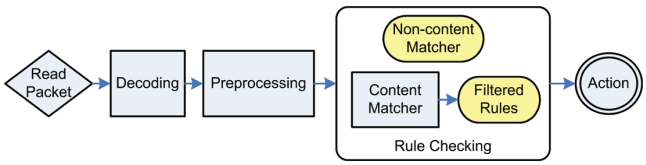
  3 an option stating which parts of the packet should be inspected for determining the presence of a particular pattern, or a collection of patterns.

  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).

Once rules are selected and initialised, they are grouped by protocol type (i.e., tcp, udp, icmp, etc.), 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 patterns 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).

Figure 1 Analysis tasks for Intrusion Detection and Prevention Systems (see online version for colours)



A rule can be either detective or preventive. The action of a detective rule is `alert` and that of a preventive rule is `drop`. The detective rule is aimed at inspecting a copy of a packet transmitted over the network, generating an alert when a hostile pattern exists. Clearly, this passive inspection mode has no impact on network performance, as it checks only for malicious activity, while genuine traffic is delivered successfully. However, since by its nature it is a passive system, this inspection mode provides poor protection. Unlike the former, the preventive rule is designed to be in-line, so that a packet will be dropped if it carries a hostile pattern. This mode can meet security requirements, but can have an adverse impact on performance, especially as malicious patterns increase. We assume that the preventive rules are applied first, so that no packet can enter the network until it is checked by all applicable rules.

3.1 Definitions and preliminaries

IDPSs are sent out with a large quantity of rules. The security administrator is responsible for including and excluding rules, in accordance with the particular needs of the protected network environment. For example, SNORT allows the enabling/disabling of rule libraries or individual rules via a set of configuration files. We let $\mathcal{R} = \{r_1, r_2, \ldots, r_N\}$ denote the set of a fixed number of rules included in the IDPS with cardinality $|\mathcal{R}| = N$. Furthermore, the security administrator can denote the form of the rules as being either detective or preventive. To categorise 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 detective or preventive (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. Obviously, a detective rule that merely examines a copy of traffic is assumed to need 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 $r_k \in \mathcal{R}$ accounts 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 each attack is independent of the others. Since the IDPS that we are considering in this case is a signature-based IDPS, the treatment of it does not incorporate the detection/prevention of the 'zero day' attacks.

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 regard to the different rules' restrictions R_i $(i = 1, 2, \ldots, N)$.

A rule r_i announces event E as malicious with regard 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 regard to attack type a_k 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 mis-announcing a malicious event that contains an attack of type a_k. We

let $\mathrm{FP}_k = \mathbb{P}\mathrm{rob}\{\mathrm{E} \overset{r_i}{\leftarrow} a_k \mid \mathrm{E} \leftarrow a_0\}$ be the false positive rate of rule r_k, i.e., the probability that rule r_k triggers an arriving event E as malicious, given that it is not malicious with regard to rule r_k.

3.2 Characterisation 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 categorises an arriving event E to be malicious. This 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 $i = 1, \ldots, N$. Clearly, the sum of this vector is equal to 1 ($\sum H(i) = 1, i = 1, \ldots, N$).

4 Performance analysis

In this section, we illustrate the means of calculating the impact of vector $\mathcal{G}$ on the resultant security of an enterprise information system and on the average response time to scrutinise an event. Once an event occurs, it goes through a sequence of detection or prevention rules consistent with the existing configuration of the IDPS represented by vector $\mathcal{G}$. The process ends if the event is dropped by a preventive rule or reported by a detective rule as a hostile event. In case an event is normal, the process terminates when all rules are checked.

4.1 Average response time

Here, we evaluate the typical response time of an IDPS. It is the time needed by the IDPS with a rule configuration $\mathcal{G}$ to effectively decide whether an arriving event is accepted as a normal event or is reported/rejected with the existence of an attack. We define $B(i)$ as the blocking probability of rule $B(i)$. It is the probability of identifying an event as hostile by a preventing rule r_i, $\forall\ i = 1, \ldots, N$. The blocking probability of rule r_i is defined by:

$$B(i) = (\mathrm{B}_{\mathrm{mal}}(i) + B_{\mathrm{safe}}(i)) \times \mathcal{G}(i) \tag{1}$$

where $\mathrm{B}_{\mathrm{mal}}$ is the probability of announcing an event as malicious by rule r_i, which 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. The term B_{safe} denotes the probability of announcing an event as hostile by rule r_i given that the event is benign and all previously evaluated rules r_j (i.e., $j < i$) mark the event as harmless.

$$\mathrm{B}_{\mathrm{mal}}(i) = \sum_{k=1}^{N} \mathrm{PB}_{\mathrm{mal}}(k,i) \times \mathrm{PE}_{\mathrm{mal}}(k,i) \tag{2}$$

$\mathrm{PB}_{\mathrm{mal}}$ represents the case when the IDPS announces the event as hostile by rule r_k given that the event arrives to rule r_i and is malicious. In this case, the probability that the IDPS correctly announces the event as hostile or mistakenly classifies it as such is calculated as follows:

$$\mathrm{PB}_{\mathrm{mal}}(k,i) = \begin{cases} 1 - \mathrm{FN}_i & \text{if } k = i \\ \mathrm{FP}_i & \text{if } k \neq i \end{cases} \tag{3}$$

$\mathrm{PE}_{\mathrm{mal}}$ represents the likelihood that the IDPS accepts the event as normal by all rules r_j, $j = 1, \ldots, i-1$, ahead of the current evaluated rule r_i where the event E is hostile. $\mathrm{PE}_{\mathrm{mal}}$ can be calculated as follows:

$$\mathrm{PE}_{\mathrm{mal}}(k,i) = \begin{cases} H(k)P_M & \text{if } i = 1 \\ \prod_{j=1}^{i-1} \big(1 - (1 - \mathrm{FN}_j)\mathcal{G}(j)\big)\mathrm{H}(k)P_M & \text{if } k \neq j \\ \prod_{j=1}^{i-1} \big(1 - \mathrm{FP}_j\mathcal{G}(j)\big)\mathrm{H}(k)P_M & \text{if } k = j \end{cases} \tag{4}$$

The first term is for the case when the existing evaluated rule r_i is the first one ($i = 1$), where no rule has been checked so far. As well, $\mathrm{PE}_{\mathrm{mal}}$ encountered the cases when there is at least one rule r_j that has been checked before rule r_i; i.e., r_i is not the first rule to be evaluated (i.e., $i > 1$)

Now let us reflect on the situation when the incident is normal in equation (1). We are interested in the likelihood of announcing an event as malicious by rule r_i given that the event is benign and that all previously evaluated rules r_j (i.e., $j < i$) mark the event as safe. B_{safe} can be calculated as follows:

$$B_{\mathrm{safe}}(i) = \mathrm{FP}_i \times \begin{cases} 1 - P_M & \text{if } i = 1 \\ \prod_{j=1}^{i-1} \big(1 - \mathrm{FP}_j\mathcal{G}(j)\big)(1 - P_M) & \text{if } i > 1 \end{cases} \tag{5}$$

Finally, the average response time can be measured as follows:

$$ART = \left[\sum_{i=1}^{N} B(i) \sum_{k=1}^{N} T(k)G(k)\right] + \left(1 - \sum_{i=1}^{N} B(i)\right) \times \sum_{i=1}^{N} T(i)G(i). \tag{6}$$

This is the time needed by an IDPS with a given configuration set to completely serve an incoming event. The accuracy of the rules and their modes (detective or preventive) play a key role in determining the response time of an IDPS.

4.2 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.

$$\begin{aligned} S &= \mathbb{P}\text{rob}\{\mathrm{E} \overset{r_j}{\leftarrow} a_i \mid \mathrm{E}{\leftarrow}a_i\} \\ &= \frac{\mathbb{P}\text{rob}\{\mathrm{E} \overset{r_i}{\leftarrow} a_i, \mathrm{E}{\leftarrow}a_i\}}{\mathbb{P}\text{rob}\{\mathrm{E}{\leftarrow}a_i\}} \\ &= \frac{\sum_{i=1}^{N} \mathbb{P}\text{rob}\{\mathrm{E} \overset{r_i}{\leftarrow} a_i, \mathrm{E} \overset{r_i}{\leftarrow} a_i\} \times \mathcal{G}(i)}{\mathbb{P}\text{rob}\{\mathrm{E}{\leftarrow}a_i\}}. \end{aligned} \tag{7}$$

Using equation (1) in equation (7) yields:

$$S = \frac{\sum_{i=1}^{N} \sum_{k=1}^{N} \mathrm{PB}_{\mathrm{mal}}(i) \times \mathrm{PE}_{\mathrm{mal}}(i) \times \mathcal{G}(i)}{P_M} \tag{8}$$

5 Optimisation of IDPS rule mode selection

In this section, we address the problem of determining the appropriate IDPS configuration set necessary to balance network security and performance. As explained before, IDPS preventive rules have the capability of blocking attacks once they have been matched. However, this induces a negative impact on network performance in terms of delay, especially when the number of preventive rules increases. Therefore, the main concern is to find the appropriate preventive rule set that maximises security enforcement levels while avoiding any unnecessary performance degradation in terms of delay. We assume that the security administrator excludes the rules that are suppose to be strictly in preventive or detective modes. The optimal solution for the Rule Mode Selection Technique (RMST) problem is the one that can maximise the prevention level and minimise system delay. Hence, the RMS problem is considered as NP-complete owing to multiobjective goals with maximal minimal matching.

5.1 Rule mode selection problem

The rule mode selection problem is formulated as follows: Given a set of IDPS rules, find a legitimate preventive rule subset that maximises the level of security, subject to the delay constraint. In our study, we assume a sequential rule-checking process where each event passes through a sequence of rules until a decision is made. For an IDPS with N rules associated with weight w_i that resembles the dominance of rule r_i in the expected value metric, we can then formalise the RMST problem as an Binary Integer Program (BIP) as follows:

$$\begin{aligned} &\max \sum_{i=1}^{N} w_i x_i \\ &\text{s.t. (a) } \sum_{i=1}^{N} t_i x_i \le D_{\max} \\ &\quad\text{(b) } x_i \in \{0,1\}, i \in \{i = 1, \ldots, N\} \end{aligned} \tag{9}$$

where x_i is a binary variable such that $x_i = 1$ if rule r_i is a preventive rule and $x_i = 0$ if rule r_i is a detective rule. The rule weight computation is explained in more detail later in this section. Inequality (a) provides an upper boundary on the expected response time. Since the expected response time for an event entering the system is proportional to the number of preventive rules, an event has to be served at a rate faster than the arrival rate to preserve the stability of the system. The delay constraint is thus translated into an upper boundary $D_{\max}$ on the number of preventive rules as the mean of the inter-arrival rate.

The RMS problem can be mapped to the 0–1 knapsack problem (Garey and Johnson, 1979) to dispose of the complexity in max-min structure. In 0–1 knapsack problem, we are given n items, each associated with a value and weight; the objective is to select a set of items that maximise the total value where the total weight is less than or equal to a given value W. The RMS problem is similar to the 0–1 knapsack problem, where the weight of the rule w_i is similar to the item's value and the upper limit response time is similar to the maximum allowed weight. Given that the 0–1 knapsack problem was proven to be also NP-complete (Garey and Johnson, 1979), the NP-completeness of the RMS problem can be proven.

5.2 RMS technique

The optimal solution of the RMS problem can be guaranteed to be obtained when performing an exhaustive search in the solution space. However, the brute force method becomes computationally impractical when the number of rules is large. The use of an approximate heuristic solution allows us to obtain a reasonably good solution in polynomial time without searching the entire solution space. Using optimisation techniques Branch & Bound and Branch & Cut for solving RMS problems is not practical when the rules number is high owing to the 2^N search

space. However, the results obtained from a simple selection method, Greedy Algorithm, can meet the computational time limit, but it suffers from the system security requirements. Hence, the proposed RMST can obtain a solution for the RMS problem in polynomial processing time and the system security level is in good agreement with the results obtained from the optimisation techniques. RMST is expressed in Algorithm 1.

Algorithm 1 OptimiseRuleSelection: rule_set, D_{max}

```
D_sum ← 0
pool_list ← empty
for r_i : i = 0 to N do
    if t(r_i) + D_sum ≤ D_max then
        prevention_list ← r_i
        D_sum ← D_sum + t(r_i)
    else if prevention_list is not empty then
        for r_j : j = 0 to N do
            if (r_i ∈ prevention_list; w_j ≥ w_i; t_j ≤ t_i) then
                remove r_i from prevention_list;
                prevention_list ← r_j;
                D_sum ← D_sum + t(r_j) - t(r_j)
                pool_list ← r_i
            end if
        end for
        if D_sum < D_max then
            Add valid rules from pool_list to prevention_list
        end if
    end if
end for
Return prevention_list
```

5.3 Rule weight computation

In this section, we describe the technique of computing rule weight with the intention of capturing the importance of every rule in the IDPS. We assign a weight to each rule in the rule configuration set that reflects its value in protecting the network by the IDPS. Two factors can be used to calculate the rule weight:

1 *potential damage* that can be prevented by a true detection

2 *operational loss* which is incurred owing to the false detection.

Potential damage $D(r_i)$: The damage prevented by rule r_i can be measured by using the severity of an attack and the accuracy of rule r_i. The attack severity measures the risk level posed by a particular attack. We let $Sev(r_i)$ denote the severity score for rule r_i that is responsible for attack a_i. There are several knowledge base sources, which provide severity scores for known attacks, including MITRE-CVE, NIST-NVD, Secunia, as well as software developer specific severity score databases. For example, the FileZilla unspecified format string vulnerability has been reported in NIST-NVD to be scored as 7.5 out of 10, where `SNORT` includes a rule accountable for this attack with multiple score references. The potential damage $D(r_i)$ can be expressed as follows:

$$D(r_i) = (1 - FN_{r_i}) \times Sev_{r_i}. \tag{10}$$

Operational loss $L(r_i)$: The operational loss incurred by rule r_i can be measured using the cost associated by the response triggered by false detection of rule r_i. For example, cost of blocking legitimate traffic or analysing false alarm. This cost can be measured/estimated from business mission as follows.

$$L(r_i) = FP_{r_i} \times Cost_{r_i}. \tag{11}$$

The rule weight $w(r_i)$ can be measured using the above-mentioned factors as follows:

$$w(r_i) = \big(\alpha \times D(r_i) - (1 - \alpha) \times L(r_i)\big) \times H(r_i) \tag{12}$$

where α is a configurable variable indicating how much of the rule weight should rely on the potential damage and operational loss. The factors used to calculate the rule weight can be obtained during the site-specific risk analysis. We are trying to benefit from information gathered by security administrators during the site-specific risk analysis process about activities that were encountered in the past. Among these are false positive rate, false negative rate, and H, and P_M.

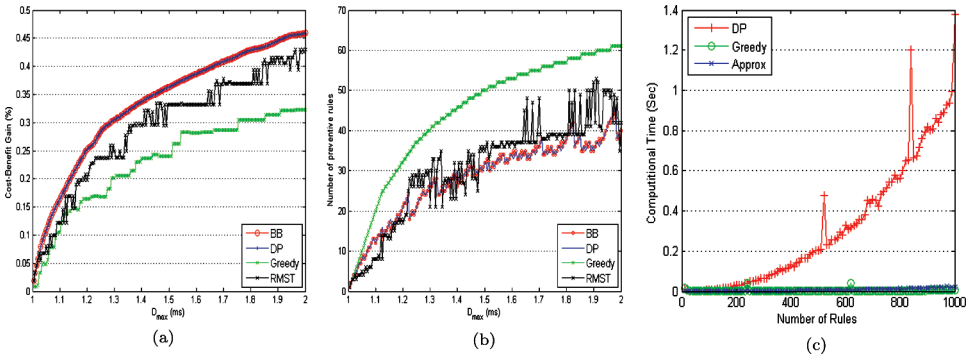
6 Performance evaluation and results

In this section, we evaluate the efficiency of the RMST. A number of simulation experiments were conducted to solve the RMS problem. We also present the optimal solution produced by solving the BIP model of the RMS problem using matlab (www.mathworks.com).

In the first experiment, we examine the accuracy of our technique in selecting the preventive rules subset. We analyse the impact of varying the maximum delay constraint ($D_{\max}$) on the resulting security and on the preventive rules selection. In this scenario, the total number of rules is 200. The weight and processing time of the rules are assigned based on zipf distribution (Zipf, 1949), which is inspired by Cabrera et al. (2004).

To validate RMST, other methods have been used for solving the RMS problem. Three techniques besides RMST are tested and all techniques' results are compared in Figure 2(a). A simple solution for the RMS problem can be obtained by a greedy algorithm, where rules are sorted by their processing time in decreasing order and chosen sequentially until the maximum allowed delay is achieved. However, the cost-benefit gain of the greedy algorithm is not desirable in terms of system security performance. Although the gain obtained using Branch and Bound (BB) and Dynamic Programming (DP) for the RMS problem is better than the proposed technique, when the number of rules increases the computational time of (BB) or (DP) is not applicable in our application, whereas the proposed RMST solves the problem in polynomial time,

Figure 2 Selected results of the maximum weight and number of preventive rules using different methods: (a) gain percentage vs $D_{\max}$, (b) number of selected rules vs $D_{\max}$ and (c) computational time (see online version for colours)



and the obtained rule set has an acceptable security level.

The qualitative and quantitative comparison of selected rules is shown in Figure 2(b). The comparison shows that the number of preventive rules selected in (BB) or (DP) is lower than that in others; however, the cost-benefit gain is maximum. On the other hand, the quality of the rules selected by the greedy algorithm is the lowest in terms of the cost-benefit gain, so the number of the preventive rules is high to satisfy the system security performance, whereas RMST selects prevention rules fairly with a reasonable number of preventive rules and cost-benefit gain.

The scale of the IDPS system affects the required computational time to find an optimal solution. In other words, 2^N combinations have to be computed to obtain the optimal solution. When the rules number is high, the search space is very large; therefore, the computational time needed is high, too. Figure 2(c) shows the relationship between the number of rules and the computational time. The BB method takes a longer time so it is not included in Figure 2(c). The greedy algorithm and RMST do not suffer from system scalability, while the number of preventive rules and the computational time is related exponentially in DP that limits applying DP to a low number of IDPS rules.

The second set of experiments studies the impact of the accuracy of the rules set in terms of FP and FN on the average response time and on the total number of preventive rules selected by BB and RMST techniques. The average response time is measured by the performance analysis model presented in Section 4 for any configuration set chosen by BB and RMST. The total number of rules is chosen to be 100 in this

Figure 3 The impact of detection rates on average response time: (a) $D_{\max}$ = high; (b) $D_{\max}$ = low; (c) $D_{\max}$ = high and (d) $D_{\max}$ = low (see online version for colours)

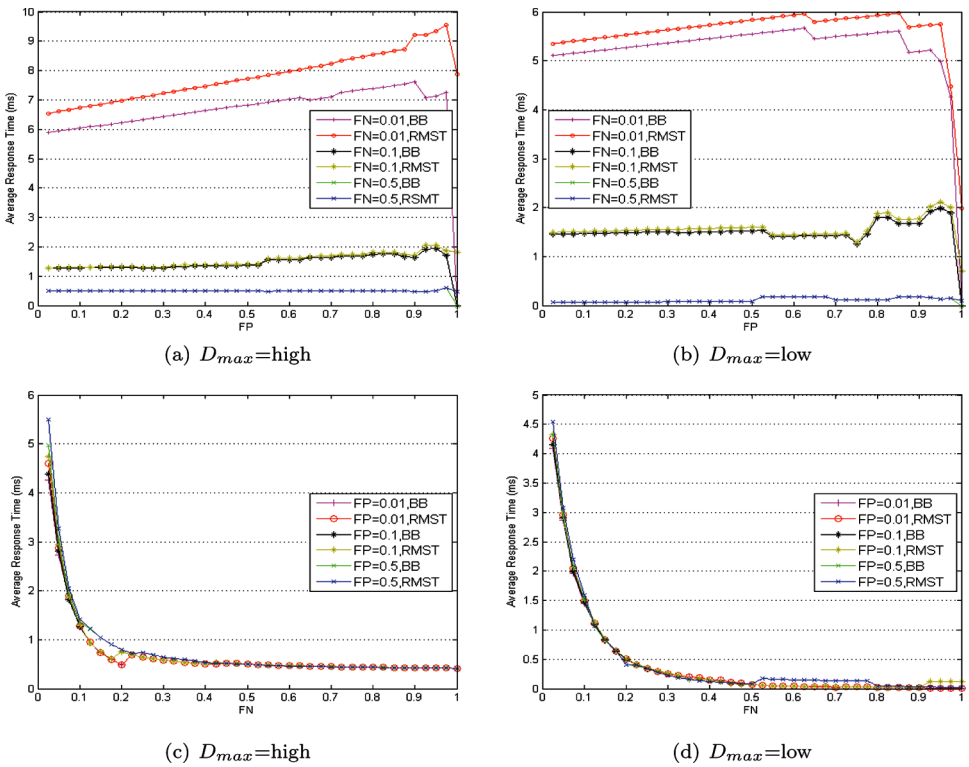
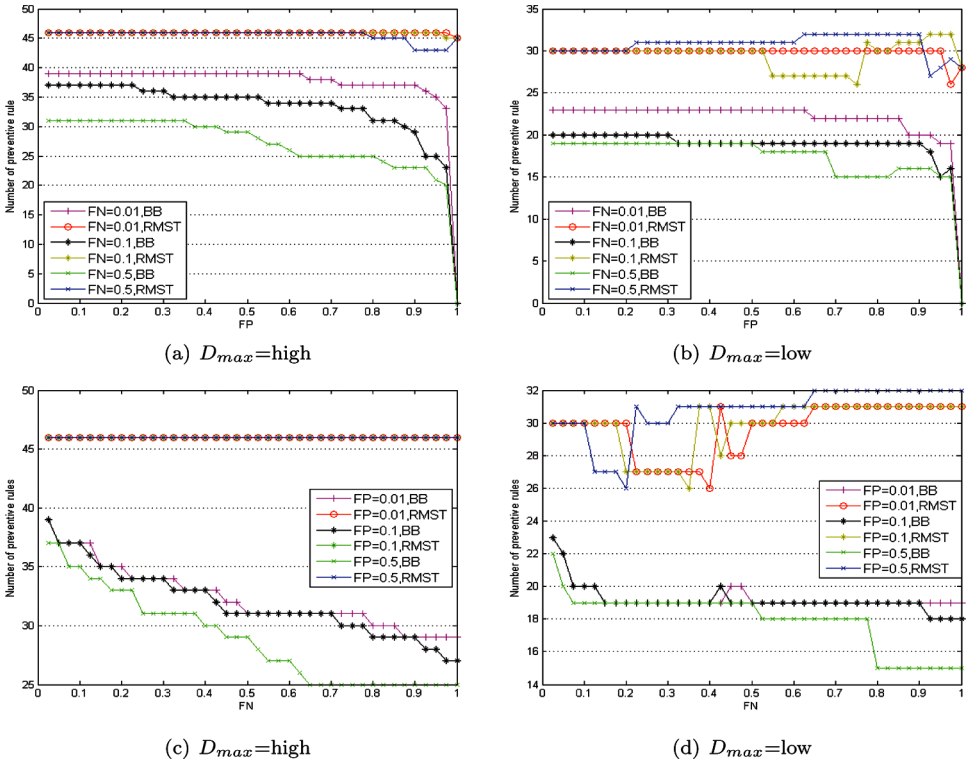


Figure 4 Selected results for impact of detection rates on preventive rule selection techniques: (a) $D_{\max}$ = high; (b) $D_{\max}$ = low; (c) $D_{\max}$ = high and (d) $D_{\max}$ = low (see online version for colours)



(a) D_{max}=high (b) D_{max}=low

(c) D_{max}=high (d) D_{max}=low

experiment. Also, we set the probability of maliciousness to be $P_M = 0.5$ and $\alpha = 0.7$.

Figure 3 plots the average response time as a function of increasing both the False Positive (FP) and False Negative (FN) rates when applying high and low values for the $D_{\max}$ constraint. Figure 4 shows the number of preventive rules chosen by BB and RMST corresponding to the configuration of the previous four figures (i.e., Figure 3).

Figure 3(a) presents the results when $D_{\max}$ is relatively high. We can see that the average response time decreases with an increase in the FN for all FP values. We can see that the average response time is longer when the IDPS becomes accurate in terms of the FN rate, no matter what the FP rates are. The number of preventive rules selected by BB and RMST for this case is presented in Figure 4(a). We can see that the BB technique adapts its selection criteria according to the change of the accuracy values while the RMST remains the same. This happens because assigning a high value to the $D_{\max}$ constraint is similar as if we are relaxing it. Figures 3(b) and 4(b) illustrate the impact of the accuracy parameters when the $D_{\max}$ is set to be low. The figures share similar results to the previous case, except that the gap between BB and RMST in average response time is reduced. This is because the low value of the $D_{\max}$ restriction makes the BB adapt its selection criteria. Overall, with a high FN factor, the average response time of both techniques is similar, although the BB has better average response time when the FN factor is very low. In other words, FN factor affects the optimisation results and thus system performance will be affected. Finally, Figures 3(c) and 4(c) show similar results as shown in Figures 3(d) and 4(d), which illustrate the deep relation between FN factor and system performance in terms of number of preventive rules and average response time.

7 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. We also propose a rule mode selection optimisation technique that aims to determine an appropriate IDPS configuration set to maximise security enforcement levels while avoiding any unnecessary network performance degradation. 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. Theoretically, with a small number of IDPS rules, the optimisation techniques are more preferable to get better results; however, when the number of IDPS rules is large, the optimisation techniques are not applicable due to 2^N search space.

Ongoing work is considering the investigation of attack graphs and 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`.

Acknowledgement

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

References

Alsubhi, K., Alhazmi, Y., Bouabdallah, N. and Boutaba, R. (2011) 'Rule mode selection in intrusion detection and prevention systems', *GLOBECOM 2011, 2011 IEEE Global Telecommunications Conference*. Houston (USA), P.1–6.

Alsubhi, K., Aib, I., François, J. and Boutaba, R. (2009) 'Policy-based security configuration management application to intrusion detection and prevention', *IEEE Conference on Communications (ICC)*, Dresden (Germany), pp.1–6.

Alsubhi, K., Bouabdallah, N. and Boutaba, R. (2011) 'Performance analysis in intrusion detection and prevention systems', *IFIP/IEEE Integrated Network Management Symposium (IM)*, Dublin (Ireland), pp.369–376.

Bellovin, S.M. and Bush, R. (2009) 'Configuration management and security', *IEEE Journal on Selected Areas in Communications JSAC*, Vol. 27, No. 3, pp.268–274.

Cabrera, J.B.D., Gosar, J., Lee, W. and Mehra, R.K. (2004) 'On the statistical distribution of processing times in network intrusion detection', *IEEE Conference on Decision and Control CDC*, Atlantis, Paradise Island, Bahamas, pp.75–80.

Chen, Y.M. and Yang, Y. (2004) 'INC WatchGuard Technologies, Policy management for network-based intrusion detection and prevention', *IEEE/IFIP Network Operations and Management Symposium, NOMS*, Seoul, Korea, pp.219–232.

Debar, H., Dacier, M. and Wespi, A. (1999) 'Towards a taxonomy of intrusion-detection systems', *Computer Networks*, Vol. 31, No. 8, pp.805–822.

Dreger, H., Feldmann, A., Paxson, V. and Sommer, R. (2004) 'Operational experiences with high-volume network intrusion detection', *Proceedings of the 11th ACM Conference on Computer and Communications Security (CCS)*, Washington, DC, USA, pp.2–11.

Dreger, H., Feldmann, A., Paxson, V. and Sommer, R. (2008) 'Predicting the resource consumption of network intrusion detection systems', *Recent Advances in Intrusion Detection (RAID)*, Springer, Boston, MA, USA, pp.135–154.

Garey, M.R. and Johnson, D.S. (1979) *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman & Co. W. H. Freeman & Co. New York, NY, USA ©1990.

Gu, G., Fogla, P., Dagon, D., Lee, W. and Skorić, B. (2006) 'Measuring intrusion detection capability: an information-theoretic approach', *Proceedings of the 2006 ACM Symposium on Information, Computer and Communications Security, ACM*, Taipei, Taiwan, pp.90–101.

Hess, A., Geerdes, H.F. and Wessäly, R. (2006) 'Intelligent distribution of intrusion prevention services on programmable routers', *Proc. of 25th IEEE INFOCOM*, Barcelona, Spain, Citeseer, pp.1–11.

Lee, W., Cabrera, J., Thomas, A., Balwalli, N., Saluja, S. and Zhang, Y. (2002) 'Performance adaptation in real-time intrusion detection systems', *Recent Advances in Intrusion Detection RAID*, Springer, RAID. pp.252–273.

Ning, P., Cui, Y., Reeves, D.S. and Xu, D. (2004) 'Techniques and tools for analyzing intrusion alerts', *ACM Transactions on Information and System Security (TISSEC)*, Vol. 7, No. 2, pp.274–318.

Paxson, V. (1998) 'Bro: a system for detecting network intruders in real-time', *SSYM'98: Proceedings of the 7th Conference on USENIX Security Symposium*, Berkeley, CA, USA, USENIX Association, pp.2435–2463.

Roesch, M. (1999) 'Snort – lightweight intrusion detection for networks', *LISA '99: Proceedings of the 13th USENIX Conference on System Administration*, Berkeley, CA, USA, USENIX Association.

Scarfone, K. and Mell, P. (2007) *Guide to Intrusion Detection and Prevention Systems (IDPS)*, National Institute of Standards and Technology (NIST) (Feb 2007), CSRC special publication SP 800-94.

Schaelicke, L., Slabach, T., Moore, B. and Freeland, C. (2003) 'Characterizing the performance of network intrusion detection sensors', *Recent Advances in Intrusion Detection: 6th International Symposium, RAID 2003*, Pittsburgh, PA, Usa, September 8–10, pp.155–172.

Schuff, D.L. and Pai, V.S. (2007) 'Design alternatives for a high-performance self-securing ethernet network interface', *IEEE International Parallel and Distributed Processing Symposium, IPDPS*.

Wu, C., Yin, J., Cai, Z., Zhu, E. and Chen, J. (2009) *A Hybrid Parallel Signature Matching Model for Network Security Applications Using Simd GPU*, Springer, Rapperswil, Switzerland, pp.191–204.

www.mathworks.com.

Zipf, G.K. (1949). *Human Behavior and the Principle of Least Effort*, Addison-Wesley, Reading MA (USA).