A Linear Programming Scheme for IPS Traffic Scheduling

Jorge Crichigno†, Nasir Ghani‡

Abstract—Intrusion Prevention System (IPS) sensors represent the initial security barrier of a network. A main challenge in today’s Internet environment is the amount of traffic these devices have to inspect. This paper presents a linear program for traffic scheduling in multi-sensor environments that alleviates inspection load at sensors. The model uses a per-flow alarm rate metric which quantifies the ratio of the amount of traffic that matches the configured signatures to the amount of traffic inspected. Traffic flows can be classified based on the metric, which permits the efficient use of computational resources to inspect suspicious traffic. Numerical results demonstrate how the proposed model can be used in enterprise networks. While the linear program is not constrained to integral solutions, traffic flows are mostly scheduled for inspection to a single sensor, which facilitates the collection of state information. This feature is essential to detect malicious traffic characterized by composite signatures.

Keywords—Intrusion Prevention System (IPS), Linear Programming, Computer Networks.

I. INTRODUCTION

Traffic flows can be classified based on the metric, which permits the efficient use of computational resources to inspect suspicious traffic. Numerical results demonstrate how the proposed model can be used in enterprise networks. While the linear program is not constrained to integral solutions, traffic flows are mostly scheduled for inspection to a single sensor, which facilitates the collection of state information. This feature is essential to detect malicious traffic characterized by composite signatures.

Fig. 1 illustrates an IPS architecture, where traffic coming from the external network is scheduled and subsequently inspected by two sensors. Most small/medium size networks have a single sensor. However, with the advent of cost efficient IPS hardware module enhancements [2], security appliances can be endowed with multiple sensors. This provides an scalable, high-availability solution. Important aspects of Fig. 1 are: 1) the scheduling scheme to forward traffic to the most appropriate sensor. Consider the case where sensor 1 has twice the inspection capacity of sensor 2. The traffic scheduler, thus, should incorporate this load balancing goal into scheduling decisions. 2) The discrimination of traffic, based on reputation, to alleviate the inspection load of IPSs. Scheduling decisions based on reputation can relieve overloaded sensors by not inspecting traffic that is considered secure, in order to use the scarce computational resources to inspect suspicious traffic instead. This secure traffic can then bypass the inspection engines. E.g., networks that include Science DMZ may not afford to inspect 100% of the traffic into the network, because of the massive data transfer they are subject to. These networks may include authentication mechanisms that permit to verify the reputation of the parties. Traffic from authenticated parties could therefore be considered low-risk traffic [3].

This paper presents a linear programming scheme for traffic scheduling in a multi-sensor environment, considering load balancing. To alleviate inspection load at sensors, the model utilizes a per-flow alarm rate metric which quantifies the ratio of the amount of traffic that matches the configured signatures to the amount of traffic inspected. The paper also describes a practical scenario where the linear program is able to schedule traffic such that sensors can collect full state information.

The paper is organized as follows. Section II discusses related work. Section III formulates the problem and presents the linear and integer linear programs. Section IV presents numerical examples and Section V concludes this work.

II. RELATED WORK

Previous works dealing with load balancing in IPSs include [1], [5]. Sekar et al. [1] describe and implement an IPS prototype for network-wide deployment. The prototype

---

†College of Engineering and Technology, Northern New Mexico College, Espanola (NM), USA; jcrichigno@nnmc.edu. ‡Dept. of Electrical Engineering, University of South Florida, Tampa (FL), USA; nghani@usf.edu.
assumes that multiple IPSs are deployed, and that a centralized scheduler distributes traffic, so that no IPS is overloaded. The proposed optimization model results in an NP-hard problem. An approximation randomized algorithm is then proposed. The model, however, assumes that the incoming network traffic from multiple locations can be centrally scheduled, which may not always be achieved. Let at al. [5] formulate the load balancing problem in the context of intrusion detection as an optimization problem. The model optimizes a metric called benefit. The metric tries captures the benefits of balancing the load and at the same time avoids the loss of correlation.

Commercial IPSs [2] include high-availability and scalability products by which parallel sensors avoid the typical single point of failure. Other related works for scaling IDS/IPS use parallelization [6], [7], [8]. Zhang et al. [9] propose an SDN-based IPS deployment that supports a unified scheduling of security applications in the whole network and load balancing among IPSs. Lanier et al. [10] present the challenges for building intrusion monitoring and prevention services in the cloud. Xing et al. [11] investigate the use of OpenFlow and Snort to build an IPS called SnortFlow.

III. OPTIMIZATION MODEL

Let S be the set of sensors or IPSs. Each sensor \( s \in S \) has a processing capacity \( c_s \), measured in traffic units (e.g., pps, Mbps, Gbps). Let \( N \) be the set of traffic flows to be inspected by the sensors. A flow can be defined as a 5-tuple \( \{ \text{source IP address}, \text{source port}, \text{destination IP address}, \text{destination port}, \text{traffic rate} \} \). The latter attribute, traffic rate measured in traffic units, is denoted as \( r_n \). For a flow \( n \in N \), let \( A_n \) indicate the total number of alarm events raised by sensors per unit of time. An alarm event is raised when a signature is examined against an event (e.g., a packet), and a match exists between the signature and the event. A signature action is subsequently performed. The proposed model uses a simple metric defined as alarm rate to estimate the rate at which suspicious events or intrusions occur in flow \( n \), given by the ratio of the number of alarms generated to the traffic rate: \( p_n = \frac{A_n}{r_n} \).

Let \( x_{n,s} \) be the fraction of traffic of flow \( n \in N \) to be routed through sensor \( s \in S \). If all traffic \( r_n \) is successfully routed to and inspected by sensors, then \( \sum_{s \in S} x_{n,s} = 1 \). Let \( 0 \leq \alpha \leq 1 \) be the maximum utilization among all sensors. A similar metric is used in IP networks for load balancing [12]. A value of \( \alpha = 1 \) means that at least one sensor is operating at full capacity. Thus, \( \alpha \) indicates the degree of load balancing among sensors. The optimization model for IPS traffic discrimination and inspection in shown in Fig. 2. The objective function consists of two terms with their corresponding weights \( w_1 \) and \( w_2 \). The first term is the summation of all traffic inspected by all sensors, multiplied by corresponding alarm rates. Because \( F \) is maximized, the linear program will prioritize flows with large alarm rates. For \( n \in N \), the term \( \sum_{s \in S} p_n r_n x_{n,s} \) can be considered as the expected suspicious traffic (EST) inspected by sensors. Thus, the term

\[
\sum_{n \in N} \sum_{s \in S} p_n r_n x_{n,s} \quad (1)
\]

in Eq. (2) is the aggregate EST. The second term of Eq. (2) is the maximum utilization among all sensors. Maximizing the negative of \( \alpha \) is equivalent to minimizing it, which will result in balancing the load among sensors. Constraint (3) states that the total fraction of traffic of flow \( n \in N \) inspected by all sensors is less or equal to one. With unlimited inspection capacity, the optimal solution of LP-IPS would result in an equality in Constraint (3). Constraint (4) limits the amount of traffic inspected by sensor \( s \in S \), where the inspection capacity \( c_s \) is multiplied by \( \alpha \). Constraint (5) restricts the traffic inspected by sensors to be positive, and Constraint (6) states that \( \alpha \) is a real value between zero and one. As a linear program, LP-IPS can be solved in polynomial time in the size of the problem and its running time be upper-bounded by \( O(k^2m) \) where \( k \) is the number of variables and \( m \) the number of constraints [13].

Atomic and Composite Signatures: An atomic signature consists of a single event (e.g., packet) that is examined to determine if it matches a configured signature. Because these signatures can be matched on a single event, they do not require a sensor to maintain state information. On the other hand, a composite signature requires several pieces of data to match an attack signature, and a sensor must maintain state information. The LP-IPS model of Fig. 2 permits a flow traffic to be split and scheduled through multiple sensors. By splitting the traffic, sensors cannot have full state information. If composite signatures are predominant, maintaining state information may be desirable. Thus, all traffic \( r_n \) of a flow \( n \in N \) may need to be routed through a single sensor. LP-IPS can be modified to avoid traffic splitting by restricting variables \( x_{n,s} \) to be zero or one, i.e., by replacing Constraint (5) with Constraint (7). The new model is an integer linear program (ILP-IPS) and thus is NP-hard.

\[
\sum_{n \in N} x_{n,s} \in \{0, 1\} \quad n \in N, s \in S \quad (7)
\]

IV. NUMERICAL EXAMPLES

A. Illustrative Example 1: Single-sensor Scenario

Fig. 3(a) shows a single-sensor scenario where sensor \( s_0 \) has an inspection capacity of \( c_0 = 1000 \) traffic units. Sensor \( s_0 \) inspects two incoming flows 0 and 1. These are characterized by the traffic and alarm rates \( r_0, p_0 \) and \( r_1, p_1 \) respectively. Given the limited capacity of \( s_0 \), the goal is to find the fraction of flows 0 (\( x_{0,0} \)) and 1 (\( x_{1,0} \)) based on reputation (i.e., alarm rates) to be inspected by \( s_0 \).

Table 1 shows the solutions generated by the different schemes and the respective aggregate EST and \( \alpha \) objectives.
The performance of LP-IPS and ILP-IPS are also compared with those of LP-Equal Alarm Rate (LP-EAR) and ILP-Equal Alarm Rate (ILP-EAR), which do not take reputation into account (all flows have equal alarm rate).

### Table I. Solution Example 1.

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Solution</th>
<th>EST</th>
<th>$\alpha$</th>
</tr>
</thead>
<tbody>
<tr>
<td>LP-IPS</td>
<td>$x_{0,0} = 1.00, x_{1,0} = 0.25$</td>
<td>82</td>
<td>1.00</td>
</tr>
<tr>
<td>ILP-IPS</td>
<td>$x_{0,0} = 1.00, x_{1,0} = 1.00$</td>
<td>64</td>
<td>0.80</td>
</tr>
<tr>
<td>LP-Equal Alarm Rate</td>
<td>$x_{0,0} = 0.25, x_{1,0} = 1.00$</td>
<td>28</td>
<td>1.00</td>
</tr>
<tr>
<td>ILP-Equal Alarm Rate</td>
<td>$x_{0,0} = 0.00, x_{1,0} = 1.00$</td>
<td>8</td>
<td>0.80</td>
</tr>
</tbody>
</table>

The main performance metric is the aggregate EST inspected by $s_0$. LP-IPS and ILP-IPS prioritize the inspection of flow 0, because of its higher alarm rate. For LP-IPS, the variable $x_{0,0} = 1$ indicates a full inspection of flow 0, and variable $x_{1,0} = 0.25$ indicates that only 25% of flow 1 (200 traffic units) is inspected. The LP-IPS and ILP-IPS solutions differ in that LP-IPS is not constrained to integral solutions, and thus it partially inspects flow 1. LP-EAR and ILP-EAR produce lower aggregate EST and thus security is reduced.

### B. Illustrative Example 2: Dual-sensor Scenario

Fig. 3(b) illustrates a dual-sensor scenario where the same traffic flows as Example 1 are scheduled through two sensors $s_0$ and $s_1$ with inspection capacities $c_0 = 1000$ and $c_1 = 1500$ traffic units. In the dual-sensor scenario, sensors have sufficient capacity to process both flows. Thus, the solution of LP-IPS and ILP-IPS are insensitive to alarm rates. Moreover, the solutions of LP-IPS and ILP-IPS differ in the $\alpha$ performance metric while aggregate EST is the same. Therefore, only the solutions of LP-IPS and ILP-IPS are shown in Table II. Notice that LP-IPS is able to minimize the maximum utilization among sensors to $\alpha = 0.64$ by scheduling 80% of flow 1 through sensor 0 and the rest of flow 1 and flow 0 through sensor 1. On the other hand, ILP-IPS schedules all traffic of flow 0 to sensor $s_0$, and traffic of flow 1 to sensor $s_1$. This solution results in an utilization of 0.80 and 0.53 of sensors $s_0$ and $s_1$ respectively. Thus, $\alpha = 0.80$.

### Table II. Solution Example 2.

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Solution</th>
<th>$\alpha$</th>
</tr>
</thead>
<tbody>
<tr>
<td>LP-IPS</td>
<td>$x_{0,0} = 0.01, x_{0,1} = 1.0, x_{1,0} = 0.8, x_{1,1} = 0.2$</td>
<td>0.64</td>
</tr>
<tr>
<td>ILP-IPS</td>
<td>$x_{0,0} = 1.0, x_{0,1} = 0.0, x_{1,0} = 0.0, x_{1,1} = 1.0$</td>
<td>0.80</td>
</tr>
</tbody>
</table>

### C. Illustrative Example 3: Dynamic Scenario

The third example presents a scenario where 1000 flows arrive dynamically one after another. Flows are inspected by 5 sensors with 100 units of capacity each. The arrival time of flows is uniformly distributed between [1-60] unit times; the flow duration and traffic rate also follow a uniform distribution between [1-15] unit times and [1-50] traffic units respectively. The alarm rates are uniformly distributed between [0.5-0.0001]. For most of the simulated time, there were more than 100 active flows at a given time, which implies that LP-IPS and ILP-IPS have more than 500 variables and 200 constraints. While LP-IPS can be solved efficiently, ILP-IPS cannot. Thus, ILP-IPS was not considered.

To highlight the benefits of incorporating reputation (alarm rates) in LP-IPS, scenario 3 was also solved using LP-Equal Alarm Rate. Since the large number of flows saturate the processing capacity, $\alpha$ is not reported.

Fig. 4(a) illustrates the EST inspected by sensors. Between times 10-60 approximately, the aggregate traffic rate is larger than the overall inspection capacity, and LP-IPS prioritizes traffic with larger alarm rates. During this time interval, the average EST performance of LP-IPS is clearly higher than that of LP with equal alarm rate (LP-EAR). The normalized EST inspected by sensors, shown in Fig. 4(b), is defined as

$$\text{Normalized EST} = \frac{\text{EST}_{LP-IPS} - \text{EST}_{LP-EAR}}{\text{EST}_{LP-EAR}}$$

where $\text{EST}_{LP-IPS}$ and $\text{EST}_{LP-EAR}$ are computed according to Eq. (1). Note that the EST of LP-IPS fluctuates between 60% and 160% above that of LP-EAR.

The black curve in Fig. 4(c) shows the number of flows inspected at least partially, which includes any flow $n \in N$ such that $\sum_{s \in S} x_{n,s} > 0$. The gray curve shows the number of flows that are fully inspected (100% of the flow) by a single sensor, which include any flow $n \in N$ such that there is a single sensor $s \in S$ for which $x_{n,s} = 1$. The gray curve thus permits the identification of flows for which sensors are able to maintain full state information. An interesting observation is that even though LP-IPS relaxes the integral constraint of ILP-IPS, its solution leads to the full inspection of about 80% of all flows by a single sensor, as shown in Fig. 4(d).

Simplistic but useful approximations and upper-bounds can be derived for scenario 3. Since rates $r_n$ are relatively large (maximum rate $r_n = 50$, which is half sensor capacity) and noting that sensors operate at full capacity (total traffic demand > total inspection capacity) between times 10-60, $\text{EST}_{LP-IPS}$ can be loosely approximated as follows:

$$\text{EST}_{LP-IPS} \approx \max \{p_n\} \cdot \text{total inspec. cap.}$$

where $E[p_n] = \frac{0.0001+0.5}{2} \approx 0.25$ is the expected value of the alarm rate $p_n$, which follows a uniform distribution between
[0.0001-0.5]. The approximations of Eqs. (9) and (10) can be seen in Fig. 4(a), where the black and gray curves approach 125 and 250 between times 10-60.

Let \( \max \{p_n : n \in N\} = p_n^* \). Under a high load regime LP-IPS inspects flows \( n \in N \) with high alarm rates \( p_n \approx p_n^* \) while LP-EAR selects random flows which have average alarm rates \( E[p_n] \approx \frac{E_n}{2} \) (for the uniform distribution presented here). The normalized \( EST \) can be approximated:

\[
\text{Normalized } EST \approx \frac{p_n^*}{\frac{E_n}{2}} = 1
\]

which is consistent with Eqs. (9), (10), and Fig. 4(b) that shows the normalized \( EST \) fluctuating approximately between [60%-160%]. Fig. 4(c) shows the total number of flows inspected by sensors. An approximation for this is:

\[
160\% \approx \text{total inspec. cap.} \approx E[r_n] \approx 25.5 \approx 20
\]

where \( E[r_n] = \frac{1+50}{2} = 25.5 \) is the expected value of \( r_n \), which follows a uniform distribution in [1-50]. It can be inferred that if flows were smaller w.r.t. inspection capacities, the number of flows integrally inspected by a single sensors would approach that of the total number of flows inspected.

**D. Why is Most State Information Maintained?**

Example 3 showed that most flows are inspected by a single sensor, even when using the LP-IPS model (which does not impose integral constraints). To better understand why this result is not an accident, this section provides an analysis, using an example, of how Simplex solves LP-IPS. Consider again Fig. 3 (a). For simplicity, assume that the only objective is the maximization of the EST, and thus \( w_1 = 1, w_2 = 0 \), and \( \alpha = 1 \) (constant). The corresponding linear program, in canonical form, is given in Fig. 5.

\[
\begin{align*}
80x_{0,0} + 8x_{1,0} & = F \quad \text{(11)} \\
x_{0,0} + x_{0,0} & = 1 \quad \text{(12)} \\
x_{1,0} + x_{1,0} & = 1 \quad \text{(13)} \\
800x_{0,0} + 800x_{1,0} + x_{0,0} & = 1000 \quad \text{(14)}
\end{align*}
\]

Fig. 5. LP-IPS for Fig. 3(a), initial canonical formulation of Simplex. Basic variables \( x'_{0,0} = 1, x'_{1,0} = 1, x'_{0,0} = 1000 \) are shown in bold.

The variables \( x'_{0,0}, x'_{1,0}, \) and \( x'_{0,0} \) are slack variables used to drive the problem into canonical form. The current basic feasible solution is given by \( (x_{0,0}; x_{1,0}; x'_{0,0}; x'_{1,0}; x'_{0,0}) = (0; 1; 1; 1000) \), where the basic variables are the slack variables \( x'_{0,0}, x'_{1,0}, \) and \( x'_{0,0} \), with objective value \( F = 0 \). In Eq. (11), the coefficient of the variable \( x_{0,0} \) is positive (80); thus, Simplex will attempt to maximize the variable \( x_{0,0} \), making it a new basic variable in the next iteration. The leaving basic variable is obtained from Constraints (12) and (14):

\[
\begin{align*}
x'_{0,0} & = 1 - x_{0,0} \geq 0, \\
x'_{0,0} & = 1000 - 800x_{0,0} \geq 0.
\end{align*}
\]

The maximum value that satisfies both constraints is

\[
x_{0,0} = \text{Min} \left\{ 1, \frac{1000}{800} \right\} = 1.
\]
The leaving basic variable is $x'_{s_0}$. By setting $s_0 = 1$, Simplex scheduled flow 0 integrally. This would still be the case in a multi-sensor scenario. The ratio $\frac{1000}{200}$ in Eq. (17) is the residual capacity of sensor $s_0$ to the flow rate $r_0$ being scheduled. The new linear program in canonical form, where the objective function is expressed in terms of non-basic variables, is given in Fig. 6.

\[
\begin{align*}
8x_{1,0} - 800x'_{s_0} &= -80 + F \\
x_{0,0} + x'_{s_0} &= 1 \\
x_{1,0} + x'_{s_0} &= 1 \\
800x_{1,0} - 800x'_{s_0} &= 200
\end{align*}
\] (28)

A key observation of the above process is that the new entering basic variable $x_{e,0}$ at each iteration, where $e \in N$ is the flow to be scheduled by Simplex, takes the value

\[
x_{e,0} = \min \left\{ \frac{0}{r_e}, \frac{1}{r_e}, \frac{2}{r_e}, \frac{3}{r_e} \right\},
\]

where $r_{s,e}$ is the residual capacity of sensor $s_0$. During the initial iteration, flow 0 was scheduled integrally (Eq. (17)). The indicator that flow 0 is integrally scheduled is given by setting $x_{e,0} = 1$, which will be the general case, provided the residual capacity $r_{s,e}$ is greater than the traffic rate $r_e$.

V. CONCLUSION

This paper presented an optimization scheme used to maximize the amount of suspicious traffic inspected by IPS sensors. The scheme is intended for enterprise networks where the amount of traffic entering and/or leaving the network is greater than the processing capacity of sensors. Using the metric alarm rate to quantify the reputation of flows, the linear program discriminates traffic flows so that the amount of suspicious traffic inspected by sensors is maximized.

Numerical results and an analysis of how Simplex algorithm solves the proposed linear program showed that the solutions obtained by Simplex represent schedules where most traffic flows are integrally scheduled to a single sensor. This feature facilitates the collection of state information which is essential to avoid attacks characterized by composite signatures.

Ongoing work includes the implementation of the IPS scheme using Cisco IPS modules. The scheme follows Fig. 1. For the parameter estimation module, flow information is acquired with NetFlow [14]. This information is exported to a server that computes the solution of the optimization models using the optimization package IpSolve [15].

REFERENCES


