Optimal Traffic Scheduling for Intrusion Prevention Systems

J. Crichigno[1]*, M. Pourvali[2], F. Shaikh[2], A. Rayes[3], E. Bou-Harb[4], N. Ghani[2]

***Abstract*—A major challenge for intrusion prevention system (IPS) sensors in today's Internet is the amount of traffic these devices have to inspect. Hence this paper presents a linear program (LP) for traffic scheduling in multi-sensor environments that alleviates inspection loads at IPS sensors. The model discriminates traffic flows so that the amount of inspected suspicious traffic is maximized. While the LP is not constrained to integral solutions, traffic belonging to a flow is mostly scheduled for inspection to a single sensor, which facilitates the collection of state information. An analysis of how the Simplex algorithm solves the model and numerical results demonstrate that state information can be preserved without imposing integral constraints. This benefit prevents the LP from becoming an integer linear program (ILP), which is essential for efficiently implementing the proposed model. The paper also shows that the ratio of the total number of flows integrally inspected by a single sensor to the total number of flows inspected in a multi-sensor environment depends on the ratio of IPS sensor capacity to flow traffic rate. Finally, some practical deployment observations are presented.**

***Keywords*—IPS, Linear Programming, Computer Networks.**

I. Introduction

Intrusion prevention systems (IPSs) are critical components for detecting and blocking malicious traffic. IPS sensors are deployed at the perimeter of a network, where traffic is analyzed from layer two header to application layer data (layer four payload). This analysis permits IPSs to identify, stop, and block malicious attacks [1]. Most IPSs are deployed at a fixed location. However this approach is facing many challenges owing to the increasing amount of traffic. Hence some scaling solutions have been proposed in the literature, including IPS clusters, distributed IPSs [1], [2], and high-availability and scalability products using parallel sensors [3].

Fig. 1 shows a sample high-availability/scalability IPS architecture with several key features. One of them is a scheduling scheme to forward traffic to the most appropriate sensor, which receives feedback from a parameter estimator. A common parameter used to discriminate traffic flows based on reputation is the rate of alarms generated by a given flow. Consider the case where sensor 1 has twice the inspection capacity of sensor 2. The scheduler should clearly incorporate load balancing into the scheduling decisions. However, although scheduling/splitting traffic from a single flow across multiple sensors can help balance inspection loads, it also prevents sensors from collecting critical state information (i.e., correlating data across multiple packets with specific composite signatures). The scheduling scheme must balance *aggregate* traffic flows across different sensors and simultaneously schedule individual flows through a single sensor. This architecture also supports traffic discrimination based on flow reputation to alleviate inspection load at IPSs. Scheduling decisions based on reputation can relieve overloaded sensors by avoiding inspection of traffic considered secure. This traffic can then bypass inspection engines. This scheme is useful in large data transfer scenarios as those observed in science demilitarized zones (SDMZs) [4].

In light of the above, this paper presents an LP scheme for traffic scheduling in multi-sensor environments, considering load balancing. The paper also presents an analysis of how the Simplex algorithm solve the proposed LP. The analysis demonstrates that state information can be preserved without imposing integral constraints to the LP.

The paper is organized as follows. Section II discusses related work. Section III formulates the problem and describes two models: an LP and an ILP. Section IV presents numerical results. Section V describes the preservation of state information. Section VI makes some practical observations, and Section VII concludes the paper.

II. Related Work

This section summarizes previous work related to IPSs. Load balancing in IPS systems is addressed by [1], [2]. Sekar *et al.* [1] described and implemented an IPS prototype for network-wide deployment. This scheme assumes that multiple IPSs are deployed, and that a centralized scheduler distributes traffic to avoid IPS overloading. Since the optimization model results in an NP-hard problem, a randomized algorithm is proposed. The model assumes that incoming network traffic from multiple locations can be centrally scheduled. Le et al. [2] formulated the load balancing problem in the context of intrusion detection as an optimization problem. This model optimizes a metric called benefit, which captures the gain of balancing the traffic load while avoiding correlation losses.

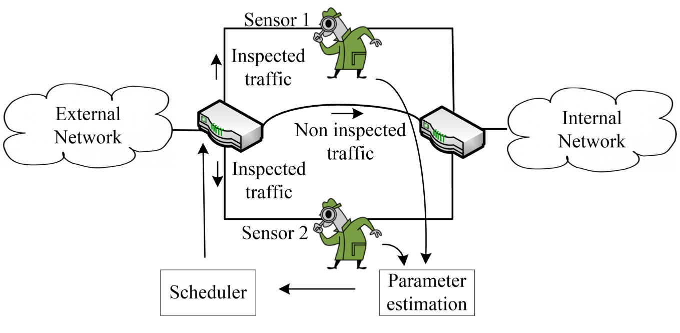

Fig. 1. Multi-sensor IPS architecture. IPS representation from [6].

Manuscript received October 13, 2016, revised July 11, 2017.

[1]University of South Carolina, Columbia (SC), USA; [2]University of South Florida, Tampa (FL), USA; [3]Cisco Systems, San Jose (CA), USA; [4]Florida Atlantic University (FL), USA. *Corresponding author: jcrichigno@ieee.org

doi: 10.11601/ijates.v6i2.201

Commercial IPSs also deploy high-availability and scalability products which use parallel sensors to avoid the typical single point of failure. Other related works for scaling IDS/IPS use parallelization [7],[8],[9]. These schemes exploit the computational power of supercomputers (or multi-core computers) by adopting a parallel architecture combining data and pipeline parallelism [7]. Zhang *et al.* [10] proposed an SDN-based IPS deployment that supports unified scheduling of security applications in the whole network (and also load balancing among IPSs). Laniepce *et al.* [11] highlighted the challenges associated with designing intrusion monitoring and prevention services in the cloud. Finally, Xing *et al.* [12] used OpenFlow and Snort to build an IPS called SnortFlow. This system can be reconfigured on-the-fly based on dynamic conditions.

III. Problem Formulation

This section is organized as follows. Section III-A presents an LP for traffic scheduling in multi-sensor environments. Section III-B incorporates integral constraints to the previously defined LP and presents the resulting ILP, and Section III-C discusses the complexity of both LP and ILP models.

A. Linear Program (LP) 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 is defined as a 5-tuple given by $\{$source IP address, source port, destination IP address, destination port, traffic rate$\}$. The latter is denoted by r_n, $n \in N$. Let A_n be the total number of traffic units of flow $n \in N$ that triggered alarm events. An alarm event is raised when a signature is examined against an event (e.g. a packet), and a match is found between the signature and the event (i.e. potential security violation). The proposed model uses a dimensionless metric defined as the *alarm rate* to estimate the rate at which suspicious events occur in flow n. This metric is given by the following ratio:

$$p_n = \frac{A_n}{r_n}. \tag{1}$$

where $0 \leq p_n \leq 1$. If the traffic r_n does not raise any alarm, $p_n = 0$. If all traffic r_n generates alarms, $p_n = 1$.

Let $x_{n,s}$ be the fraction of traffic from flow $n \in N$ to be scheduled at sensor $s \in S$. If all traffic r_n is successfully scheduled 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. Note that a similar metric is also used in IP networks for load balancing [13]. A value of $\alpha = 1$ means that at least one sensor is operating at full capacity. Based on the above definitions, the proposed LP is defined by:

$$\text{Max } F = w_1 \sum_{n \in N} \sum_{s \in S} p_n r_n x_{n,s} - w_2 \alpha. \tag{2}$$

$$\sum_{s \in S} x_{n,s} \leq 1 \qquad n \in N. \tag{3}$$

$$\sum_{n \in N} r_n x_{n,s} \leq c_s \alpha \qquad s \in S. \tag{4}$$

$$x_{n,s} \geq 0 \qquad n \in N, s \in S. \tag{5}$$

$$0 \leq \alpha \leq 1. \tag{6}$$

The objective function of the LP, referred as LP-IPS in the rest of the paper, consists of two terms with weights w_1 and w_2. The first term is the summation of all traffic inspected by all sensors, multiplied by their corresponding alarm rates. Since F is maximized, the linear program prioritizes flows with high alarm rates. For $n \in N$, the term $\sum_{s \in S} p_n r_n x_{n,s}$ can be considered as suspicious traffic. Thus, the first term of Eq. (2) is the aggregated expected suspicious traffic (EST):

$$EST = \sum_{n \in N} \sum_{s \in S} p_n r_n x_{n,s}. \tag{7}$$

The second term in Eq. (2) is the maximum utilization among all sensors. Maximizing the negative of α is equivalent to minimizing it. Constraint (3) states that the total fraction of traffic from flow $n \in N$ inspected by all sensors should be less or equal to unity. Constraint (4) limits the amount of traffic inspected by sensor $s \in S$, where the inspection capacity c_s is multiplied by α.

B. Integer Linear Program (ILP) Model

An atomic signature consists of a single event (e.g. packet) that is examined to determine if it matches a configured signature. Since 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 (e.g., packets) to match an attack signature, and hence a sensor must maintain state information. The LP-IPS model permits an individual traffic flow to be split and scheduled across multiple sensors. However splitting traffic implies that sensors cannot collect full state information. Therefore if composite signatures are predominant, maintaining state information will be critical. As a result, all traffic r_n of a flow $n \in N$ may need to be routed through a single sensor.

The LP-IPS can be modified so that traffic from a single flow $n \in N$ is not split. This requirement can be added by restricting variables $x_{n,s}$ to be binary integers (0 or 1). The associated program is defined by:

$$\text{Max } F = w_1 \sum_{n \in N} \sum_{s \in S} p_n r_n x_{n,s} - w_2 \alpha. \tag{8}$$

$$\sum_{s \in S} x_{n,s} \leq 1 \qquad n \in N. \tag{9}$$

$$\sum_{n \in N} r_n x_{n,s} \leq c_s \alpha \qquad s \in S. \tag{10}$$

$$x_{n,s} \in \{0, 1\} \qquad n \in N, s \in S. \tag{11}$$

$$0 \leq \alpha \leq 1. \tag{12}$$

Note that restricting variables to take integral values converts the LP-IPS into an ILP. This ILP will be referred as ILP-IPS in the rest of the paper. Also, note that Constraint (9) is either satisfied with equality when a single sensor is used to inspect the traffic of a flow $n \in N$ (i.e. only one variable $x_{n,s}$ is unity), or satisfied with inequality when the traffic is not inspected by any sensor (i.e. all variables $x_{n,s}$ are zero).

C. Complexity of LP-IPS

Since the LP-IPS is a linear program, it can be solved in polynomial time in the size of the problem. The size of LP-IPS is given by the number of constraints and variables. From (5) and (6), the total number of variables is $k = |N| \cdot |S| + 1$. Similarly, from (3),(4),(5) and (6), the number of constraints is $m = |N| + |S| + |N| \cdot |S| + 2$. Both k and m are polynomial in the number of variables and constraints. In practice the Simplex method runs in polynomial time of k and m. Similarly, the Interior Point method can solve LP-IPS problem in a polynomial time that is upper-bounded by $O(k^2 m)$ [14]. On the other hand, the ILP-IPS problem is an integer linear program and is therefore NP-hard, i.e. the running time is exponential in the size of the problem.

IV. Illustrative Examples

This section presents first a small illustrative example in a dual-sensor scenario (Section IV-A), followed by a dynamic example where traffic flows arrive one by one (Section IV-B).

A. Illustrative Example 1: Dual-Sensor Scenario

Fig. 2(a) illustrates a scenario where sensors s_0 and s_1 have inspection capacities of $c_0 = 1000$ and $c_1 = 1500$ traffic units, respectively. These sensors have to inspect two traffic flows 0 and 1, which are characterized by traffic and alarm rates r_0, p_0, and r_1, p_1, respectively. Given that the aggregate traffic rate is less than the aggregate inspection capacity, the solution for the LP-IPS should perform load balancing. Similarly, ILP-IPS should also perform load balancing by scheduling flows 0 and 1 through different sensors.

The solutions for the LP-IPS and ILP-IPS models are shown in Table I. The respective objective functions only differ in the α performance metric, while EST is the same. LP-IPS is able to minimize the maximum utilization among the sensors to $\alpha = 0.64$ by scheduling 80% of flow 1 via sensor 0 and the remainder of flow 1 and flow 0 through sensor 1. ILP-IPS schedules all traffic from flow 0 to sensor s_0, and traffic from flow 1 to sensor s_1. This solution results in a utilization of 0.80 and 0.53 for sensors s_0 and s_1, respectively. Thus $\alpha = 0.80$.

Example 1 shows how load balancing can be achieved when the aggregate inspection capacity exceeds the aggregate traffic rate. The subsequent dynamic scenarios will show how LP-IPS discriminates traffic when aggregate inspection capacity is less that aggregate traffic rate.

B. Illustrative Example 2: Dynamic Scenarios

The second illustrative example presents two dynamic scenarios where 1000 flows arrive in random sequential manner. Here the sensors must discriminate which flows are more important to inspect, since the aggregate inspection capacity is less than the aggregate traffic rate. All flows are inspected by 5 sensors with 100 traffic units of capacity each. Flow inter-arrival times are uniformly distributed between (1-60) time units and their durations are uniformly distributed from (1-15) time units. Alarm rates are also uniformly distributed between (0.0001-0.5). Two dynamic scenarios are tested. In scenario 1, the traffic rate is uniformly distributed between (1-50) traffic units (expected value of $r_n, n \in N$, is $E[r_n] = 25.5$). In Scenario 2, the traffic rate is uniformly distributed between (1-10) traffic units ($E[r_n] = 5.5$). Since the ILP-IPS is NP-hard, only the LP-IPS solution is shown.

TABLE I
Solution for Example 1

Scheme	Solution	α
LP-IPS	$x_{0,0} = 0.0, x_{0,1} = 1.0, x_{1,0} = 0.8, x_{1,1} = 0.2$	0.64
ILP-IPS	$x_{0,0} = 1.0, x_{0,1} = 0.0, x_{1,0} = 0.0, x_{1,1} = 1.0$	0.80

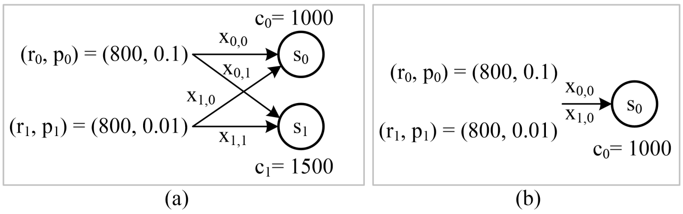

Fig. 2. (a) Dual-sensor scenario, (b) single-sensor scenario.

To highlight the benefits of incorporating reputation into LP-IPS, the dynamic scenarios are solved using two different approaches. In the first approach, denoted by LP equal alarm rate (LP-EAR), the value of p_n is the same for all flows $n \in N$. In the second approach (simply LP-IPS), flow reputation using p_n is incorporated. Additionally, α is not reported, as the aggregate traffic rate is higher than the aggregate inspection capacity. Figs. 3(a) and 3(b) show the results for dynamic scenario 1. The normalized EST of Fig. 3(a) is defined as:

$$\text{Normalized } EST = \frac{EST_{LP-IPS} - EST_{LP-EAR}}{EST_{LP-EAR}}, \quad (13)$$

where EST_{LP-IPS} and EST_{LP-EAR} are computed according to (7). Note that EST_{LP-IPS} fluctuates between 60% and 160% above that of LP-EAR during most of the timeline. This indicates that discriminating flows based on alarm rates allows LP-IPS to improve performance with respect to LP-EAR by up to 160%. Fig. 3(b) shows the percentage of flows integrally inspected by a single sensor using LP-IPS, which includes any flow $n \in N$ such that there is a single sensor $s \in S$ for which $x_{n,s} = 1$. The findings show that at any time other than the transient starting and ending times of simulation, approximately 80% of all flows are inspected by a single sensor. Consider simulation times between $t = 10$ and $t = 60$ units where the aggregate traffic rate is higher than the aggregate inspection capacity. From the flows inspected during this period, on average 80.07% are integrally inspected by a single sensor. The coefficient of variation for this percentage is 5.33%. Figs. 3(c) and 3(d) show the results for dynamic Scenario 2, where traffic rates vary between (1-10). Fig. 3(c) shows that EST_{LP-IPS} fluctuates between 20% and 50% above that of LP-EAR. Between $t = 0$ and $t = 8$, and $t = 64$ and $t = 70$, the performance of LP-IPS and LP-EAR are the same because the aggregate traffic rate is less than the aggregate inspection capacity. Finally, in the interval between $t = 10$ and $t = 60$ units, when the aggregate traffic rate is greater than the aggregate inspection capacity, Fig. 3(d) shows

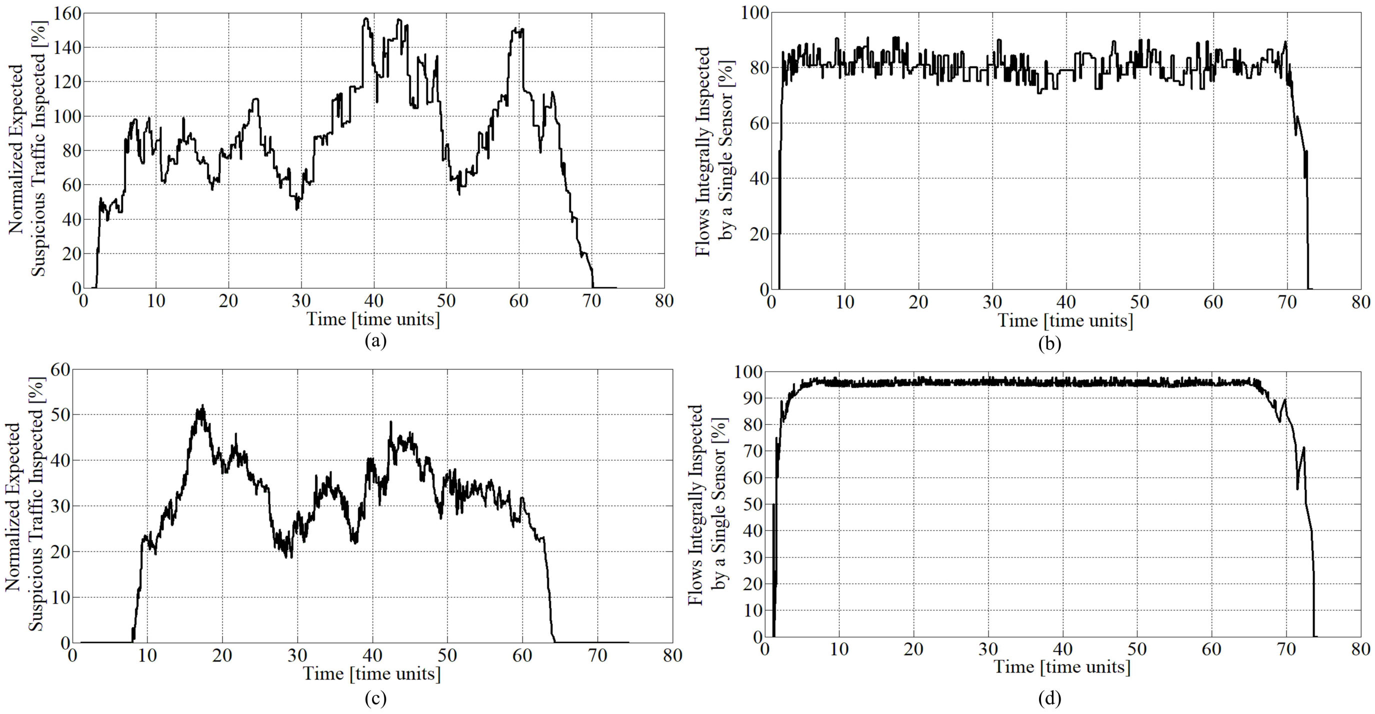

Fig. 3. (a), (b): Results for the dynamic scenario 1, for r_n uniformly distributed between (1-50), $n \in N$. (a) Normalized expected suspicious traffic inspected by sensors, computed according to Eq. (13), (b) Percentage of flows integrally inspected by a single sensor using LP-IPS, (c), (d): Results for the dynamic Scenario 2, for r_n uniformly distributed between (1-10).

that an average of 95.5% of inspected flows are integrally inspected by a single sensor. The coefficient of variation for this percentage is 0.87%.

The results in Fig. 3(b) and Fig. 3(d) indicate that the percentage of flows integrally inspected by a single sensor is sensitive to the size of the flow rates with respect to the IPS inspection capacities. Larger numbers of flows with smaller flow rates (r_n) are inspected by a single sensor, $n \in N$. This is confirmed in Section V-B, as presented next.

V. Preservation of State Information

This section presents an analysis of how Simplex solves LP-IPS. Section V-A shows that the solutions by Simplex permit the collection of state information. Section V-B provides an approximation of the percentage of traffic flows for which state information can be collected.

A. Simplex Solution for LP-IPS

As observed in Figs. 3(b) and 3(d), LP-IPS allows for state preservation without imposing integral constraints. This is a key result to efficiently solve the scheduling problem. Consider Fig. 2(b), where two flows are inspected by a single sensor. Assume that the only objective is the maximization of the EST, i.e. $w_1 = 1$, $w_2 = 0$, and $\alpha = 1$. The corresponding LP, in canonical form, is defined by (14)-(17).

$$
\begin{aligned}
80x_{0,0} + 8x_{1,0} & = F, && (14)\\
x_{0,0} + \mathbf{x'_{0,0}} & = 1, && (15)\\
x_{1,0} + \mathbf{x'_{1,0}} & = 1, && (16)\\
800x_{0,0} + 800x_{1,0} + \mathbf{x_{s_0}} & = 1000. && (17)
\end{aligned}
$$

The basic variables $x'_{0,0} = 1$, $x'_{1,0} = 1$, and $x_{s_0} = 1000$ are shown in bold. The variables $x'_{0,0}$, $x'_{1,0}$, and x_{s_0} are slack variables used to drive the problem into canonical form. The current basic feasible solution is given by $(x_{0,0}; x_{1,0}; x'_{0,0}; x'_{1,0}; x_{s_0}) = (0; 0; 1; 1; 1000)$. The basic variables are also the slack variables, and the objective value is $F0$. In Eq. (14), 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 (15) and (17) as:

$$x'_{0,0} = 1 - x_{0,0} \geq 0, \qquad x_{s_0} = 1000 - 800x_{0,0} \geq 0. \quad (18)$$

The maximum value that satisfies both constraints is:

$$x_{0,0} = \min\left\{1, \frac{1000}{800}\right\} = 1. \quad (19)$$

The leaving basic variable is $x'_{0,0}$, i.e., by setting $x_{0,0}$=1, Simplex schedules flow 0 integrally. Note that this would still be the case in a multi-sensor scenario. The ratio $\frac{1000}{800}$ in Eq. (19) is the residual capacity of sensor s_0 to the flow rate r_0 being scheduled. The revised linear program with the objective function expressed in term of non-basic variables is defined by Eqs. (20)-(23).

$$
\begin{aligned}
8x_{1,0} - 80x'_{0,0} & = -80 + F, && (20)\\
\mathbf{x_{0,0}} + x'_{0,0} & = 1, && (21)\\
x_{1,0} + \mathbf{x'_{1,0}} & = 1, && (22)\\
800x_{1,0} - 800x'_{0,0} + \mathbf{x_{s_0}} & = 200. && (23)
\end{aligned}
$$

The current basic feasible solution is given by $(x_{0,0}; x_{1,0}; x'_{0,0}; x'_{1,0}; x_{s_0}) = (1; 0; 0; 1; 200)$. The basic variables are $x_{0,0}$, $x'_{1,0}$, and x_{s_0}, and the objective function is $F = 80$. Note how Simplex integrally schedules flow 0 rather than fractional values of flows 0 and 1, permitting sensor s_0 to maintain state information for flow 0. In the next iteration, $x_{1,0}$ becomes the new basic variable because its coefficient in (20) is positive, and the leaving basic variable is determined by Constraints (22) and (23):

$$x'_{1,0} = 1 - x_{1,0} \geq 0, \qquad x_{s_0} = 200 - 800x_{1,0} \geq 0. \tag{24}$$

The maximum value that satisfies both constraints is:

$$x_{1,0} = \min\left\{1, \frac{200}{800}\right\} = \frac{1}{4}. \tag{25}$$

The leaving basic variable is x_{s_0}, which indicates that sensor s_0 will not have any residual capacity in the next iteration. Simplex maximizes the entering basic variable $x_{1,0}$ to $\frac{200}{800} = \frac{1}{4}$, which is the ratio of residual sensor capacity to the flow rate r_1. Eqs. (26)–(29) are the last iteration of Simplex.

$$\begin{aligned}
& & -72x'_{0,0} & & -\frac{1}{100}x_{s_0} &= -82 + F, && (26)\\
\mathbf{x_{0,0}} & + & x'_{0,0} & & &= \quad 1, && (27)\\
& & x'_{0,0} + \mathbf{x'_{1,0}} & & -\frac{1}{800}x_{s_0} &= \frac{3}{4}, && (28)\\
\mathbf{x_{1,0}} & - & x'_{0,0} & & +\frac{1}{800}x_{s_0} &= \frac{1}{4}. && (29)
\end{aligned}$$

The coefficients of the non-basic variables $x'_{0,0}$ and x_{s_0} in Eq. (26) are negative. Since the linear program is in canonical form and any feasible solution to the constraints has non-negative coordinates, the largest possible value for F has been reached ($F = 82$). This value is assumed at $(x_{0,0}; x_{1,0}; x'_{0,0}; x'_{1,0}; x_{s_0}) = (1; \frac{1}{4}; 0; \frac{3}{4}; 0)$. The variables of interest with physical representation are $x_{0,0} = 1$ and $x_{1,0} = \frac{1}{4}$, which indicate that 100% and 25% of flows 0 and 1 respectively will be inspected.

A key observation from the above is that the new entering basic variable $x_{e,0}$ at each iteration, $e \in N$, is the flow to be scheduled by Simplex, and is given by:

$$x_{e,0} = \min\left\{1, \frac{c^{res}_{s_0}}{r_e}\right\}, \tag{30}$$

where $c^{res}_{s_0}$ is the residual capacity of sensor s_0. During the initial iteration, flow 0 is scheduled integrally, see (19). The indicator that flow 0 is integrally scheduled is determined by setting $x_{e,0} = 1$, which is the general case, provided the residual capacity $c^{res}_{s_0}$ is greater than the traffic rate r_e.

B. State Information Collection: A Simple Approximation

Note that Figs. 3(b) and 3(d) indicate that the number of traffic flows integrally inspected by a single sensor depends on sensor capacity and expected traffic rate. Let $E[r_n]$ be the expected value of the traffic rate for flow $n \in N$ and assume that all sensors have the same inspection capacity, i.e. c_s=c for all $s \in S$. Assume that $c > E[r_n]$, which is the case for enterprise IPS sensors, and define the ratio of sensor capacity to expected traffic as:

$$Q = \text{round}\left(\frac{c}{E[r_n]}\right). \tag{31}$$

On average one can assume that half of the $|S|$ sensors will inspect Q flows integrally and the other half will inspect $Q-1$ flows integrally. The approximate number of flows integrally inspected is:

$$I \approx \frac{|S|}{2}Q + \frac{|S|}{2}(Q-1) = \frac{|S|}{2}(2Q-1). \tag{32}$$

Sensors use their residual capacity to inspect fractions of flows, see (30). Thus, after inspecting integral flows, a sensor would only inspect one fractional flow, i.e. Simplex attempts to fully schedule one flow before scheduling another. On average there would be $|S|$ fractional flows, i.e. one flow per sensor. The ratio of the total number of flows integrally inspected by a single sensor to the total number of flows inspected is then approximated as:

$$\text{ratio} \approx \frac{I}{I + |S|} = \frac{2Q-1}{2Q+1}. \tag{33}$$

Clearly, the percentage of traffic flows for which state information can be collected depends on Q. Fig. 4 shows 5 dynamic scenarios where the input parameters are the same as those in Example 2, but with different traffic rate. $E[r_n]$ is denoted by $E[r]$, as the traffic rate distribution is the same for all $n \in N$. "Avg" indicates the ratio of flows integrally inspected by a single sensor to the total number of flows inspected, in the $[10 - 60]$ interval (percentage). "A-Avg" indicates the corresponding approximate value computed using (33) (percentage). Additionally, the error between the two is provided. The results show that as traffic rates increase with respect to the capacity of sensors, i.e. Q decreases, the number of flows for which sensors can collect state information decreases. However in most current networks the capacity of sensors is still few orders of magnitude larger than traffic rates, allowing the preservation of most state information (see green curve in Fig. 4). Even when $Q = 20$, more than 80% of inspected flows are integrally inspected.

VI. Practical Observations of LP-IPS

This section presents few practical recommendations regarding the computation of flow reputation based on the alarm rate

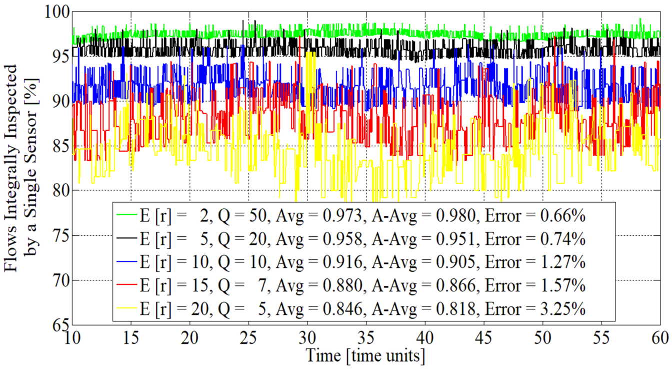

Fig. 4. Simulation results of 5 dynamic scenarios using similar parameters as Example 2: $|S| = 5, c_s = 100, |N| = 1000, p_n$ uniformly distributed between (0.0001-0.5). Arrival time and flow duration uniformly distributed between (1-60) and (1-15). Traffic rates uniformly distributed between (1-3) (green), (1-9) (black), (1-19) (blue), (1-29) (red), (1-39) (yellow) units.

metric (Section VI-A) and the implementation of IPS jointly with access-control lists (Section VI-B).

A. Flow Sampling and Reputation

In an ideal scenario all traffic entering a network should be inspected. However, as the amount of traffic increases, an improved discrimination mechanism based on flow reputation is needed. Information about flows (such as source address, destination address, source port, destination port, and traffic rate) can be dynamically obtained from devices such as routers and switches. Many modern routers and switches include flow management applications such as Netflow [15].

The LP-IPS model does not include any requirement to force at least a minimum sampling of flows to continuously update the respective alarm rates. Instead this requirement can be met independently of LP-IPS, or alternatively, it can also be incorporated as follows: $\sum_{s\in S} r_n x_{n,s} \geq \delta_n, n \in N$. This constraint requires sensors to inspect a minimum amount of traffic δ_n and to report all alarms A_n raised during inspection. This information can be used to calculate the alarm rate p_n. Note that p_n can be considered as a point estimate of a signature inspection *match probability*. Assuming that p_n follows a binomial distribution, a minimum sample size δ_n for estimating the match probability can be computed given a maximal margin of error E for a confidence level L: $\delta_n = p_n(1-p_n)\left(\frac{z_L}{E}\right)^2$. The parameter z_L is the critical value from the normal distribution for the confidence level L [16].

In order to capture dynamic conditions where intrusion attempts may occur, it may be desirable to continuously update the alarm rate p_n. One approach is to use an exponential average of the previous alarm rates. Eq. (1) can include a time dimension $p_n(t) = \frac{A_n}{r_n}(t)$, which measures the alarm rate during the interval between t and $t+1$. For $0 \leq \lambda \leq 1$, the predicted alarm rate at $t+1$ is given by: $\hat{p}_n(t+1) = \lambda p_n(t) + (1-\lambda)\hat{p}_n(t)$. $p_n(t)$ stores the most recent information, whereas $\hat{p}_n(t)$ tracks past history. The parameter λ controls the relative weight of recent and past history. If $\lambda = 1$, then $\hat{p}_n(t+1) = p_n(t)$, i.e. only the recent alarm rate matters and history is irrelevant. Note that $\hat{p}_n(0)$ can be defined as a constant. For example, flows inspected for the first time for which there are no alarm records would have a large $\hat{p}_n(0)$ value. According to the objective function (2), LP-IPS would then maximize the inspected traffic from this flow. Over time, $\hat{p}_n(t)$ will be adjusted according to the above exponential average expression.

B. Access Control Lists

Commercial IPS sensors have the capability to block traffic from a particular flow, commonly specified in an access control list (ACL). An ACL is a sequential list of permit or deny statements that apply to IP addresses and upper-layer protocol features, such as ports. A black ACL consists of flows that are blocked (from attacking systems) and do not consume any IPS resources. One advantage of this blocking action is that a single IPS sensor can stop traffic at multiple locations throughout the network, regardless of its location. For example, in a multi-homed network that has more than one connection to external networks, several independent IPS sensors can be deployed. If a set of sensors detects a high level of matching signatures, that set can add the corresponding flow to a black ACL and apply the ACL to itself and other sensors. A white ACL consists of flows that are completely trusted and do not require inspection. Both white and black ACLs release computational resources as their listed flows are not inspected by IPS sensors. Since ACLs are implemented in the forwarding hardware of a device, they do not compromise performance. Such mechanisms are preferred to secure a SDMZ [4].

VII. Conclusion

This paper presents an optimization scheme to maximize the amount of suspicious traffic inspected by IPS sensors. The scheme uses flow reputation to prioritize the inspection of flows with high alarm rates. An additional feature of the scheme is the load balancing by which traffic flows are scheduled according to the capacity of sensors.

An analysis of how Simplex solves the LP-IPS model demonstrates that state information can be preserved without imposing integral constraints (i.e., correlating data across multiple packets with specific composite signatures is achievable). Results show that the number flows for which state information is collected depends on the ratio IPS sensor capacity to traffic flow rates (size). In simulated scenarios, when this ratio is 50 (IPS sensor capacity is 50 times that of flow rates), the percentage of flows for which sensors can correlate data across multiple packets is above 95%. Even when this ratio is only 20, the percentage of flows for which sensors can correlate data across multiple packets is above 80%.

Since LP-IPS is not constrained to integer variables, the scheme can be solved and implemented efficiently. As the above results indicate, the scheme is effective for protecting networks from attacks characterized by both atomic and composite signatures. The paper concludes with practical observation for the implementation of the proposed scheme.

References

[1] V. Sekar, R. Krishnaswamy, A. Gupta, M. Reiter, “Network-Wide Deployment of Intrusion Detection and Prevention Systems,” *ACM International Conference on Emerging Networking Experiments and Technologies (CoNEXT)*, Philadelphia, PA, Nov. 2010.

[2] A. Le, E. Al-Shaer, R. Boutaba, “On Optimizing Load Balancing of Intrusion Detection and Prevention Systems,” *IEEE International Conference on Computer Communications (INFOCOM)*, Phoenix, AZ, Apr. 2008.

[3] Cisco Systems, [Online]. Available: http://www.cisco.com

[4] ESnet, [Online]. Available: http://fasterdata.es.net/science-dmz

[5] J. Crichigno, N. Ghani, “A Linear Programming Scheme for IPS Traffic Scheduling,” *IEEE International Conference on Telecommunications and Signal Processing (TSP)*, Prague, Czech Republic, July 2015.

[6] W. Stallings, “Network Security Essentials,” 5th Edition, Prentice Hall, 2013.

[7] H. Jiang, G. Zhang, G. Xie, K. Salamatian, L. Mathy, “Scalable High-Performance Parallel Design for Network Intrusion Detection Systems on Many-Core Processors,” *ACM/IEEE Symposium on Architectures for Networking and Communications Systems*, San Jose, CA, Oct. 2013.

[8] L. Foschini, A. Thapliyal, L. Cavallaro, C. Kruegel, G. Vigna, “A Parallel Architecture for Stateful, High-Speed Intrusion Detection,”, *International Conference on Information Systems Security*, Hyderabad, India, Dec. 2008.

[9] A. Le, E. Al-Shaer, R. Batouba, "Correlation-Based Load Balancing for Intrusion Detection and Prevention Systems," *International Conference on Security and Privacy in Communication Networks*, Istanbul, Turkey, Sep. 2008.

[10] L. Zhang, G. Shou, Y. Hu, Z. Guo, "Deployment of Intrusion Prevention System Based on Software Defined Networking," *IEEE International Conference on Communication Technology (ICCT)*, Guilin, China, Nov. 2013.

[11] S. Laniepce, M. Lacoste, M. Kassi-Lahlou, F. Bignon, K. Lazri, A. Wailly, "Engineering Intrusion Prevention Services for IaaS Clouds: The Way of the Hypervisor," *IEEE International Symposium on Service Oriented System Engineering (SOSE)*, San Francisco, CA, Mar. 2013.

[12] T. Xing, D. Huang, L. Xu, C. Chung, P. Khatkar, "SnortFlow: A OpenFlow-Based Intrusion Prevention System in Cloud Environment," *GENI Research and Educational Experiment Workshop (GREE2013)*, Salt Lake City, UT, Mar. 2013.

[13] J. Crichigno, N. Ghani, J. Khoury, W. Shu, M. Wu, "Dynamic Routing Optimization in WDM Networks," *IEEE Global Communications Conference (GLOBECOM)*, Miami, FL, Dec. 2010.

[14] S. Boyd, L. Vandenberghe, "Convex Optimization," *Cambridge University Press*, 2004.

[15] Netflow, [Online]. Available: http://www.cisco.com/c/en/us/products/ios-nx-os-software/ios-netflow

[16] C. Brase, C. Brase, "Understanding Basic Statistics," 6th Edition, Cengage Learning, 2012.