FireCol: A Collaborative Protection Network for the Detection of Flooding DDoS Attacks

Jérôme François, Issam Aib, Raouf Boutaba

FireCol: A Collaborative Protection Network for the Detection of Flooding DDoS Attacks

Jérôme François, Issam Aib, *Member, IEEE,* and Raouf Boutaba, *Fellow, IEEE*

***Abstract*—Distributed Denial of Service (DDoS) attacks remain a major security problem the mitigation of which is very hard especially when it comes to highly distributed botnet-based attacks. The early discovery of these attacks, although challenging, is necessary to protect end users as well as the expensive network infrastructure resources.**

In this paper, we address the problem of DDoS attacks and present the theoretical foundation, architecture and algorithms of *FireCol*. The core of *FireCol* is composed of Intrusion Prevention Systems (IPSs) located at the Internet Service Providers (ISPs) level. The IPSs form virtual protection rings around the hosts to defend and collaborate by exchanging selected traffic information. The evaluation of *FireCol* using extensive simulations and a real dataset is presented, showing *FireCol* effectiveness and low overhead, as well as its support for incremental deployment in real networks.

***Index Terms*—distributed denial-of-service, flooding, detection, collaboration, network security**

I. Introduction

Distributed Denial of Service (DDoS) attacks still constitute a major concern [1] even though many works have tried to address this issue in the past (ref. survey in [2]). As they evolved from relatively humble megabit beginnings in 2000, the largest DDoS attacks have now grown a hundredfold to break the 100 Gbps, for which the majority of ISPs today lack an appropriate infrastructure to mitigate them [1].

Most recent works aim at countering DDoS attacks by fighting the underlying vector which is usually the use of botnets [3]. A botnet is a large network of compromised machines (bots) controlled by one entity (the master). The master can launch synchronized attacks, such as DDoS, by sending orders to the bots via a Command & Control channel. Unfortunately, detecting a botnet is also hard and efficient solutions may require to participate actively to the botnet itself [4], which raises important ethical issues, or to firstly detect botnet-related malicious activities (attacks, infections, etc), which may delay the mitigation.

To avoid these issues, this paper focuses on the detection of DDoS attacks and per se not their underlying vectors. Although non distributed denial of service attacks usually exploit a vulnerability by sending few carefully forged packets to disrupt a service, DDoS attacks are mainly used for flooding a particular victim with massive traffic as highlighted in [1]. In fact, the popularity of these attacks is due to their high effectiveness against any kind of service since there is no need to identify and exploit any particular service-specific flaw in the victim. Hence, this paper focuses exclusively on flooding DDoS attacks [1].

A single IPS (Intrusion Prevention System) or IDS (Intrusion Detection System) can hardly detect such DDoS attacks, unless they are located very close to the victim. However, even in that latter case, the IDS/IPS may crash because it needs to deal with an overwhelming volume of packets (some flooding attacks reach 10-100Gbps). In addition, allowing such huge traffic to transit through the Internet and only detect/block it at the host IDS/IPS may severely strain Internet resources.

This paper presents *FireCol*, a new collaborative system that detects flooding DDoS attacks as far as possible from the victim host and as close as possible to the attack source(s) at the Internet Service Provider (ISP) level. *FireCol* relies on a distributed architecture composed of multiple IPSs forming overlay networks of protection rings around subscribed customers.

FireCol is designed in a way that makes it a service customers can subscribe to. Participating IPSs along the path to a subscribed customer collaborate (vertical communication) by computing and exchanging *belief scores* on potential attacks. The IPSs form virtual protection rings around the host they protect. The virtual rings use horizontal communication when the degree of a potential attack is high. In this way, the threat is measured based on the overall traffic bandwidth directed to the customer compared to the maximum bandwidth it supports. In addition to detecting flooding DDoS attacks, *FireCol* also helps in detecting other flooding scenarios, such as flash crowds, and for botnet-based DDoS attacks.

The paper proceeds as follows. Section II describes the architecture and the global operation of *FireCol*. The different leveraged metrics and components of the system are presented in III. Section IV presents *FireCol* attack detection algorithms. Section V explains the mitigation technique used once an attack has been detected. Section VI presents the simulations

This work was supported in part by the Natural Science and Engineering Council of Canada under its discovery program and in part by the World Class University program through the Korea Science and Engineering Foundation funded by the Ministry of Education, Science, and Technology under Project R31-2008-000-10100-0.

J. François is with SnT, University of Luxembourg, Luxembourg, this work was partially done when J. François was with David R. Cheriton School of Computer Science, University of Waterloo, Canada, email: jerome.francois@uni.lu.

I. Aib is with the Ontario Public Service, Community Services I&IT Cluster, Canada. This work was partially conducted while Issam was with D.R School of Computer Science, University of Waterloo, Canada, email: issam.aib@gmail.com.

R. Boutaba is with the David R. Cheriton School of Computer Science, University of Waterloo, Waterloo, ON N2L 3G1, Canada, and also with the Division of IT Convergence Engineering, Pohang University of Science and Technology, Gyungbuk 790-784, Korea, email: rboutaba@cs.uwaterloo.ca.

[1]This paper substantially extends our previous work in [5]

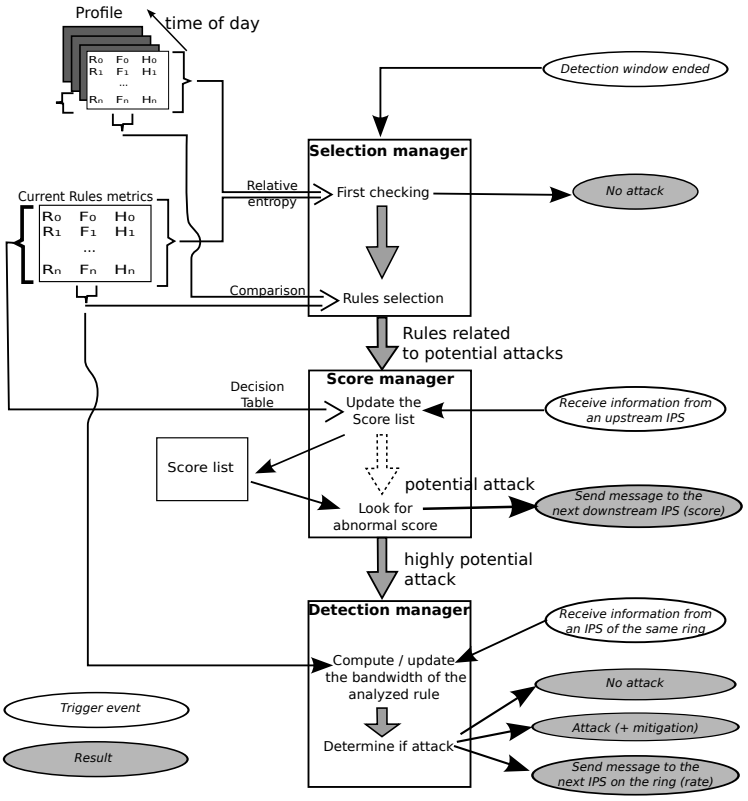


Fig. 1. *FireCol* architecture

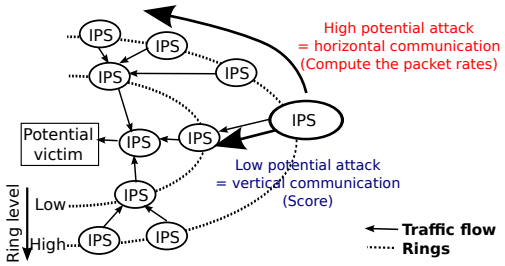


Fig. 2. Horizontal and Vertical communication in *FireCol*

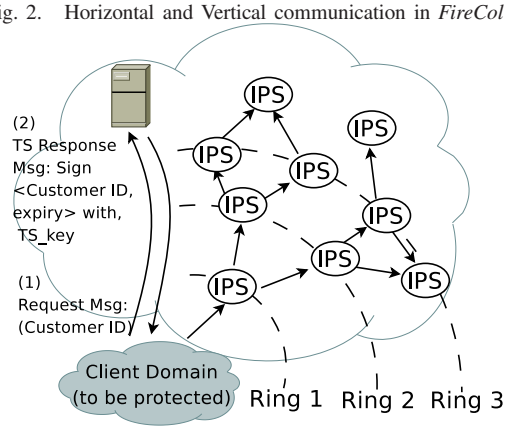


Fig. 3. *FireCol* subscription protocol

we conducted in order to evaluate *FireCol*. The complexity of *FireCol* is analyzed in section VII. Section VIII summarizes related work. Finally, section IX concludes the paper and outlines future research directions.

II. The *FireCol* architecture

A. Ring-based Overlay Protection

The *FireCol* system (Fig. 1) maintains virtual, rings or shields of protection, around registered customers. A ring is composed of a set of IPSs that are at the same distance (number of hops) from the customer (Fig. 2). As depicted in Fig. 1, each *FireCol* IPS instance analyzes aggregated traffic within a configurable *detection window*. The *metrics manager* computes the frequencies and the entropies of each rule (section III-A). A rule describes a specific traffic instance to monitor and is essentially a traffic filter, which can be based on IP addresses or ports.

Following each detection window, the *selection manager* measures the deviation of the current traffic profile from the stored ones, selects out of profile rules, then forwards them to the *score manager*. Using a decision table, the *score manager* assigns a score to each selected rule based on the frequencies, the entropies, and the scores received from upstream IPSs (vertical collaboration/communication). Using a threshold, a quite low score is marked as a *low potential attack* and is communicated to the downstream IPS which will use to compute its own score. A quite high score on the other hand is marked as *high potential attack* and triggers ring-level (horizontal) communication (Fig. 2) in order to confirm or dismiss the attack based on the computation of the *actual packet rate* crossing the ring surpasses the known, or evaluated, customer capacity (section II-B). As can be noticed, this detection mechanism inherently generates no false positives since each potential attack is checked. However, since the entire traffic cannot be possibly monitored, we promote the usage of multiple levels and collaborative filtering described previously for an efficient selection of rules, and so traffic, along the process. In brief, to save resources, the *collaboration manager* is only invoked for the few selected candidate rules based on resource-friendly metrics.

B. Subscription protocol

FireCol protects subscribers (*i.e.*, potential victims), based on defined rules. A *FireCol* rule matches a pattern of IP packets. Generally, this corresponds to an IP subnetwork or a single IP address. However, the rule definition can include any other monitorable information which can be monitored, such as the protocols or the ports used.

FireCol is an added value service to which customers subscribe using the protocol depicted in Fig. 3. The protocol uses a trusted server of the ISP which issues tokens. When a customer subscribes for the *FireCol* protection service, the trusted server adds an entry with the subscribing rule along with its subscription period (TTL) and the supported capacity. The server then issues periodically a corresponding token to the customer with a TTL and a unique ID signed using its private key. All communications between subscribers and the server are secured a using private/public key encryption scheme.

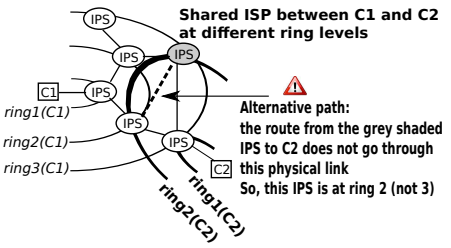


Fig. 4. *FireCol* with two cusomers: C1 and C2

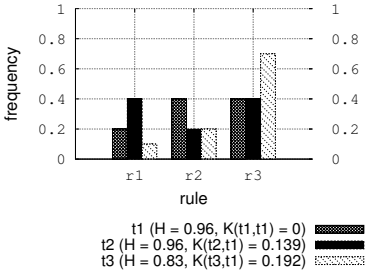


Fig. 5. Entropy Example

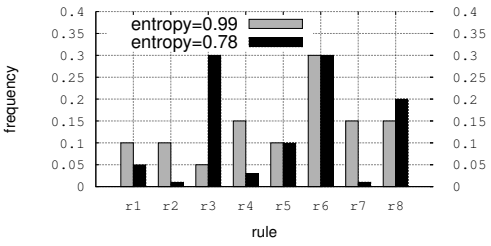


Fig. 6. Examples of score rule frequencies

The ring level of a *FireCol*-enabled router (IPS) is regularly updated based on the degree of stability of IP routing. This is done using a two phase process. First, the router sends a message *RMsg* to the protected customer containing a counter initialized to 0. The counter is incremented each time it passes through a *FireCol*-enabled router. The customer (or first-level *FireCol* router) then replies to the initiating router with the value of its ring level. This procedure is optimized through aggregation when several routers are requesting a ring-level update.

In practice, the ring level value is network dependent. However, routing stability has been well investigated and enhanced [6], [7]. The study done in [8] shows that most routes are usually stable within the order of several days while flooding attacks generally operate within the order of minutes in order to have a high impact. For further analysis, section VI-I quantifies the impact of routers not assigned to the right level. It shows that updating the ring topology at regular intervals is sufficient even if some IPSs are not well configured with respect to the ring they belong to. A more sophisticated mechanism could monitor route changes to force ring updates.

In *FireCol*, a capacity is associated to each rule. Rule capacities can be provided either by customers or the ISP (for overall capacity rules). For sensitive services, customers can specify the capacity. IT services of large companies should be able to provide such information regarding their infrastructure. For smaller customers, statistical or learning algorithms, running at customer premises or first hop IPS, might be leveraged to profile traffic throughput [9]. Similar to [10], the threshold can be tuned to keep a small proportion (*i.e.*, 5%) for analysis. Finally, for very small customers, such as a household, a single rule related to the capacity of the connection can be used. The maximum capacity, or throughput quota, is generally readily available to the ISP based on the customer SLA [11], [12].

C. Multiple customers

Because of their inherent complete independence, *FireCol* allows the coexistence of multiple virtual protection rings for multiple customers across the same set of IPSs. Therefore, a single IPS may act at different levels with respect the customers it protects as depicted in Fig. 4. Although most of the figures in this paper represent overlay networks with a single route, from an ISP to a customer, this figure highlights that alternative paths are possible. However, as discussed in the previous section, the rings are dependent of the routing at a certain time, which is quite stable compared to the typical duration of flooding attacks, and so only the current route is considered for building the rings.

III. The *FireCol* system

A. FireCol *metrics*

With set of rules $R = \{r_i | i \in [0, n]\}$, *FireCol* maintains the following frequency and entropy-based metrics:

1) Frequency: The frequency, f_i, is the proportion of packets matching rule r_i within a detection window.

$$f_i = \frac{F_i}{\sum_{j=1}^{n} F_j} \tag{1}$$

where F_i is the number of packets matched by rule r_i during the detection window. Note that every customer rule set $R = \{r_i | i \in [0, n]\}$ is complete, in the sense that every packet must match at least one rule. This is ensured by always having a default rule matching all traffic not covered by the supplied rules.

The frequency distribution is then defined as $f = \{f_1, \ldots, f_n\}$.

2) Entropy: The entropy H (Eq. (2)) measures the uniformity of distribution of rule frequencies.

If all frequencies are equal (uniform distribution), the entropy is maximal, and the more skewed the frequencies are, the lower the entropy is. Fig. 5 shows the frequencies of three rules r_1, r_2, r_3 from three distributions representing different detection windows (t_1, t_2, t_3) and values for entropies and relative entropies.

$$H = -E[log_n f_i] = -\sum_{i=1}^{n} f_i log_n(f_i) \tag{2}$$

3) Relative entropy: The relative entropy metric $K(f, f')$ (Eq. (4)) (the Kullback-Leibler distance) measures the dissimilarity between two distributions (f and f'). If the distributions are equivalent, the relative entropy is zero, and the more deviant the distributions are, the higher it becomes.

$$\psi_i = log \frac{f_i}{f'_i} \tag{3}$$

$$K(f, f') = \sum_{i=1}^{n} f_i \ \psi_i \tag{4}$$

The example in Fig. 5 shows that the t_2's frequencies are more similar with t_1 than are t_3's with t_1, hence $K(t_2, t_1) = 0.139 < 0.192 = K(t_3, t_1)$. The relative entropy metric is necessary because even if two distributions were different, they still can have the same simple entropy (*e.g.*, entropy is preserved by permutations).

B. FireCol *components*

The *FireCol* system is composed of several collaborating IPSs each enriched with the following components (Fig. 1 in section II):

1) Packet processor: Examines traffic and updates elementary metrics (counters and frequencies) whenever a rule is matched.

2) Metrics manager: Computes entropies (Eq. (2)) and relative entropies (Eq. (4)).

3) Selection manager: The *detection_window_ended* event (Fig. 1) is processed by the *selection manager*, which checks whether the traffic during the elapsed detection window was within profile. It does so by checking whether the traffic distribution represented by frequencies follows he profile. This corresponds to check if $K(f, f') \leq \omega$ (Eq. (4)), where f is the current distribution of frequencies, f' is the stored distribution of the traffic profile, and ω the maximum admitted deviation from it.

If $K(f, f') > \omega$, the traffic is marked as abnormal and requires further investigation. If there is a flooding DDoS attack, the traffic volume increases and so does the frequency of some rules. Thus, a rule r_i with a frequency higher than a certain threshold and a certain deviation from the profile will be selected as a potential attack at time t iff:

$$\frac{f_i}{f'_i} > 1 + \gamma,\ 0 \leq \gamma \leq 1 \tag{5}$$

$$f_i(t) > \epsilon \tag{6}$$

In our implementation, the traffic profile is based on a weighted moving average updated as follows:

$$f'_i \leftarrow a \times f_i + f'_i \times (1 - a) \tag{7}$$

a is fixed to 0.5 to give an equivalent weight to the current and past traffic activities.

4) Score manager: The *score manager* assigns a score to each of the selected rules depending on their frequencies and the entropy. The entropy and the frequency are considered high if they are respectively greater than a threshold α and β. The different cases are presented in Table I:

1) *High Entropy and High rule frequency*:
In this case, the traffic is well distributed meaning that most rules have about the same frequency (they cannot be all high as the sum is one). Hence, having one rule that is quite different from the others is a good sign that it is a potential attack. In Fig. 6, this is the case for rule r6 of the grey distribution.
2) *Low Entropy and High rule frequency*:
In this case, the attack is only potential but not as much as when the entropy is high. In Fig. 6, the black distribution has several high and low frequencies, and it is not clear if the high frequencies represent direct threats as they can be only due to the low values of other frequencies.
3) *High Entropy and Low rule frequency*:
This case represents a potential threat. Here all frequencies are about the same making it not a threat as the frequency is low. However, since it is increasing and deviates from the profile (first selection by the *selection manager*) (Eq. (5) and Eq. (6)) it may surpass other frequencies later on in time.
4) *Low Entropy and Low frequency*:
This case includes both high and low frequencies because of the low entropy. Thus, it is not possible to conclude about any threat.

TABLE I
THE DECISION TABLE

Case	Entropy	Frequency	Conclusion	Score
1	High ($> \alpha$)	High ($> \beta$)	Potential	b_1
2	Low ($\leq \alpha$)	High ($> \beta$)	Medium threat	b_2
3	High ($> \alpha$)	Low ($\leq \beta$)	Potential later	b_3
4	Low ($\leq \alpha$)	Low ($\leq \beta$)	No threat	$b_4 = 0$

Algorithm 1 checkRule (IPS_id, i, $rate_i$, cap_i)

```
1: if b_i ∧ (IPS_id ≠ null) then
2:    if IPS_id == myID then
3:       b_i = false;
4:       return
5:    else
6:       rate_i ← rate_i + F_i
7:       if rate_i > cap_i then
8:          b_i = false;
9:          raise DDOS alert;
10:         return
11:      else
12:         nextIPS.checkRule(IPS_id, i, rate, cap_i)
13:      end if
14:   end if
15: else
16:   b_i = true;
17:   nextIPS.checkRule(myID, i, 0, cap_i)
18: end if
```

Each of the above cases is associated with a score factor b_j indicating the aggressiveness of the attack where $b_1 > b_2 > b_3 > b_4$ (table I). The score S_i of rule i is then obtained as follows:

$$S_i = f_i \times b_j \tag{8}$$

Using a Dempster-Shafer belief combination function ([13], [14]), the scores are updated at the end of every detection window based on the current score, previous score, and those provided by upstream IPSs (higher ring).

Afterwards, the rules, which the score is lower than a small threshold υ, are automatically discarded as they do no more represent potential attacks. If the rule score is greater than parameter $\tau >> \upsilon$, the attack is considered highly potential and this alert is forwarded to the collaboration manager for aggressiveness checks. Otherwise ($\tau \leq S_i < \upsilon$), the decision

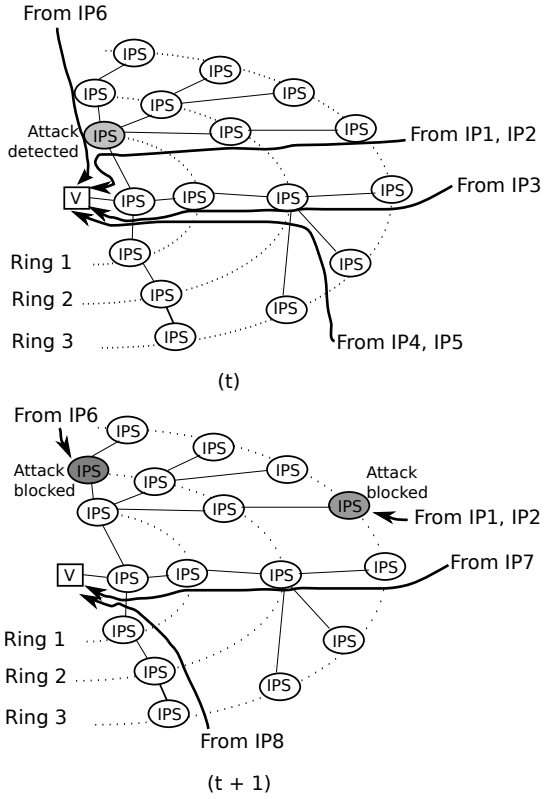


Fig. 7. At the end of time t, an attack against host V is detected. At time t+1, the traffic from attack sources is blocked.

is delegated to a downstream IPS on the path to the victim. This process of vertical communication is illustrated in Fig. 2.

Finally, scores are also affected by an aging factor λ_{age} as follows:

$$S_i\{t+1\} = \lambda_{age} \times S_i\{t\} \tag{9}$$

From a practical point of view, scores sent by the same IPS to the same downstream IPS are combined in one message to reduce the overhead.

5) Collaboration manager: The *collaboration manager* is the last component in charge of confirming potential attacks. We claim that detecting a flooding attack can be confirmed only if the traffic it generates is higher than the customer's capacity. Hence, the IPS where the alert is triggered has to initiate a ring-level communication to calculate the average traffic throughput for subsequent comparison with the subscribers capacity. This is detailed in the next section.

IV. *FireCol* ATTACK DETECTION ALGORITHMS

For each selected r_i, the collaboration manager computes the corresponding packet rate using rule frequencies and the overall bandwidth (bw_m) consumed during the last detection window. If the rate is higher than the rule capacity cap_i, an alert is raised. Otherwise, the computed rate is sent to the next IPS on the ring (Alg. 1).

When an IPS receives a request to calculate the aggregate packet rate for a given rule, it first checks if it was the initiator. In this case, it deduces that the request has already made the round of the ring and hence there is no potential attack. Otherwise, it calculates the new rate by adding in its own rate and checking if the maximum capacity is reached, in which case an alert is raised. Otherwise, the investigation is delegated to the next horizontal IPS on the ring.

Alg. 1 shows the details of this procedure. It is initially called with an empty IPS_{id}. The first IPS fills it and sets the boolean b_i to true (line 16). b_i is reset after the computation finishes, *i.e.*, when the request has made the round of the ring or when the alert is triggered. With simple adjustments, ring traversal overhead can further be reduced if several suspect rules are investigated in one pass.

Rate computation can be performed based on the number of packets per second (pps) or bytes per second (bps). The first method is more suitable for detecting flooding DDoS attacks having a small packet pattern, such as SYN floods. Bytes-based method is better for detecting flooding attacks with large packet payloads. *FireCol* customers can subscribe to either or both protection types.

V. MITIGATION

A. Mitigation shields

When an attack is detected, *FireCol* rings form protection shields around the victim. In order to block the attack as close as possible to its source(s), the IPS that detects the attack informs its upper ring IPSs (upstream IPSs), which in turn apply the vertical communication process and enforce the protection at their ring level (Alg. 2). To extend the mitigation, the IPS that detects the attack inform also its peer IPSs on the same ring to block traffic related to the corresponding rule. This is done by forwarding the information as the same manner as done by the *collaboration manager* (Alg. 1). Only traffic from suspected sources (*i.e.*, triggered some rule r_i) is blocked as shown in Fig. 7. This is performed by the *block_IPs* function in Alg. 2 line 5.

This process entails the potential blocking of benign addresses. However, this is a temporary cost that is difficult to avoid if a flooding attack is to be stopped. Potential alternatives are describes in the next section.

It may be impossible to determine all attack sources during a single detection window due to inherent network delays and/or resource limitations. The attacker can also invoke an attack scenario from different machines at different times to reduce the risk of detection.

For this, after the detection and mitigation of an attack against some host h, *FireCol* continues the detection process

Algorithm 2 mitigate (r_i,firstRing)

```
1: for all ips ∈ upstreamIPSs do
2:     ips.mitigate(r_i, False)
3: end for
4: for all a ∈ getAddr(r_i) do
5:     block_IPs(a)
6: end for
7: if firstRing = True then
8:     nextIPS.mitigate(r_i, True)
9: end if
10: setCautiousMode(r_i)
```

looking for some additional attack sources. Furthermore, in order to limit the effect of potentially additional attack sources, after the blocking period elapses, the IPS may activate a *cautious mode* phase wherein a rate limitation of packets corresponding to the triggered rule is applied.

The actual duration of the blocking and caution period depends on the aggressiveness of the attack, *i.e.*, on the difference between the observed packet rate $rate_i$ and the host capacity cap_i.

B. Careful mitigation

This section gives an overview of common techniques to improve attack mitigation by blocking only attacks-related IP sources.Only those associated to high packet rates or which have open most of the sessions recently might be blocked like in [15]. Moreover, identifying not yet seen IP addresses is another way to detect the potential spoofed addresses or zombies used to perform a DDoS attack [16]. The authors in [17] propose other heuristics based on the difference between incoming and outgoing traffic. A solution could be to capture all traffic associated with a triggered alert by the *score manager* and use signatures to clearly identify an attack. Furthermore, a general blacklist can be imported from external databases, like SpamHaus [18] which stores IP addresses related to Spam meaning that they are probably zombie computers. Non-assigned IP addresses or abnormal source IP addresses (multicast, private addresses...) [19] could be also a starting point of such blacklisting.

VI. Evaluation

The objective of the experiments is to evaluate the accuracy of *FireCol* in different configurations. Furthermore, the robustness of *FireCol* is evaluated in abnormal situations such as the existence of non-cooperative routers or configuration errors.

A. Simulations

Although obtaining real router traces is possible, getting synchronized traffic and host states of a real network along with its detailed topology is quite difficult for security, privacy, and legal reasons. Thus, we mainly used a simulation-based approach for the evaluation of the *FireCol* system.

We tested different topologies with a variable number of rings. Fig. 8 shows a sample topology of five customers with a specific rule for each. The lowest ring (closest to hosts) is composed of two IPSs. The *fan-out* effect (increase in connectivity) is taken into consideration with the number of IPSs between rings i and $i+1$ multiplied by factor $fan = 1.5$. This fan-out effect generates enough routers for highlighting the collaboration. Varying it does not significantly impact the results, except a little delay in the time needed to detect an attack due to a larger number of collaborating routers. In fact, only extreme cases, as described in VI-K, have a significant impact.

Besides, a router at level i is connected to a router at level $i-1$ with a probability $1/i$. Each simulation lasts for 100 detection windows. Table II shows the values used for the parameters. All hosts have been given the same capacity. Flow sizes representing background traffic are distributed according to a power law formula to follow the behavior of flow sizes and topology properties in the Internet [20], [21]. The main property of power law formulae is scale invariance.

This property is also preserved by the exponential law. Therefore, we define the relative traffic flow size to host i as:

$$b \times e^{-c \times i} \tag{10}$$

where $c = 0.3$ is the *skewness parameter* (worst case for assessing *FireCol* as highlighted in VI-J), and b is chosen so that the sum of relative sizes equals one. Each experiment is run 25 times (except otherwise mentioned) in order to generate different topologies and background traffic.

One specific benign traffic and two malicious ones are generated between time windows 10 and 20. To strengthen the evaluation, the benign one is heavy and close to a flooding attack in terms of packet rate. The first malicious traffic simulates a stealthy attack on H1 (Fig. 8) with a frequency $\leq 10\%$. The second is targeted against host 3 and simulates a more aggressive attack with a frequency $\geq 30\%$. Both types of malicious traffic are generated at the outer virtual ring on about half of the routers.

The stealthy attack targets the first host (H1 on Fig. 8) where the normal traffic flow is the heaviest due to the formula we used for flow generation (Eq. (10)), hence making it stealthier and more difficult to detect. In this way, including more customers during the simulations is not useful since this would split the normal traffic among more hosts and so the attack traffic would be more distinguishable. However, experiments with real data, in section VI-L, involve more customers.

B. Metrics

The True Positive Rate (TPR) measures the proportion of rightly detected attacks. The False Positives (FP) counter represents the amount of benign traffic wrongly flagged as malicious. As previously described, horizontal communication discards all of them by computing the real packet rates. However, the number of rules to analyze the traffic has to be as low as possible and so we will consider the mis-selected rules as false positives. From a practical point, this corresponds to taking the output of the *score manager* (section III) as the final result.

In *FireCol*, an alert pertains to rules and may only be generated following the elapse of a detection window. Thus both the TPR (in proportion) and the FPs (absolute value) are computed on a time-window basis.

Because *FireCol* works in a time-window and per rule basis, an alert may be generated (true or false positive) or not (true or false negative), for each rule, at each IPS at the end of each detection window. Due to that, evaluating false positives as a ratio is irrelevant. Since, benign traffic is in majority, the false positive ratio does not vary significantly because calculated regarding a large number of true negatives. For example, when 70 FP are observed in next experiments, it may only represents 4% with a 5 rings topology or less than

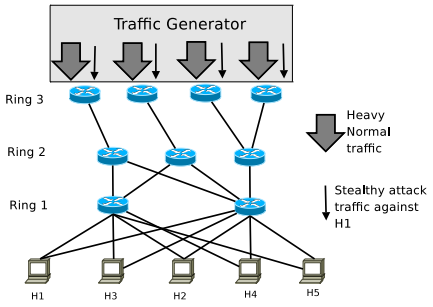


Fig. 8. Sample simulation topology

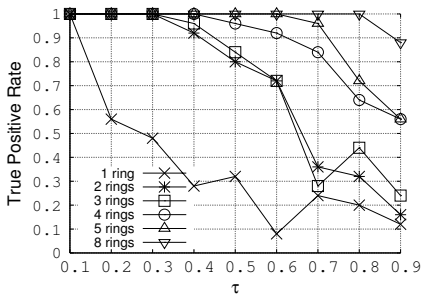


Fig. 9. Effect of the Score threshold τ on the TPR

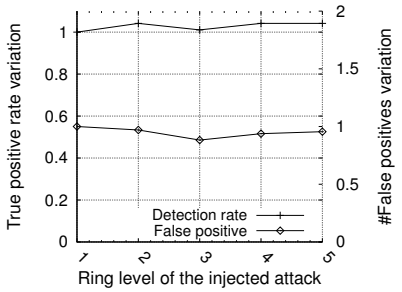


Fig. 10. Insignificant impact of the attack injection location (using an attack injected at the first ring as reference value) – 5 rings configuration

1% with 12 rings. Hence, using the absolute value of FPs is more suitable and helps to evaluate the efficiency of *FireCol*, which has to discard, as much as possible, candidate rules along the selection process.

Last is the *detection time*, *i.e.*, the delay between the attack occurs and when it is detected. In the evaluation, we focused on the detection phase and not the counter measures.

C. Impact of the Score threshold (τ)

Fig. 9 reports the effect of the score threshold τ on the TPR where each point represents an average of the 25 simulation runs. When τ increases, fewer rules are suspected of being related to highly potential attacks. This reduces the number of raised alerts and thus the number of false positives and the TPR. Simulations helped to determine the optimal value for τ depending on the input topology. We found, for example, that $\tau = 0.7$ is best for a five rings topology (TPR close to 100%). The average number of false positives is about 10 in this case, which is only 2% of the maximal number of false positives.

For a five rings topology, there are 24 IPSs, thus the average number of false positives per IPS is 0.42. Assuming a TPR objective of at least 90%, the five rings topology is found to be the most suitable. This explains why this configuration is used in most of our evaluations. In addition, the detection time is relatively low and is less than one detection window in most cases with the highest observed value being 2.32 windows.

As can be noticed from Fig. 9, a single ring topology reveals poor performance unless a small score threshold is used (in which case 7 times more false positives are generated). Since a single ring topology implies no vertical score exchange, figure demonstrates the benefit of collaboration. Thus, the *FireCol* rule selection process is not fitted for a single IPS.

TABLE II
VALUES OF MAIN PARAMETERS

γ	0.4	ω	0.05
α	0.8	β	0.4
b_1	1	b_2	0.65
b_3	0.8	τ	0.5
ϵ	0.01	υ	0.05

D. Ring levels of the attack

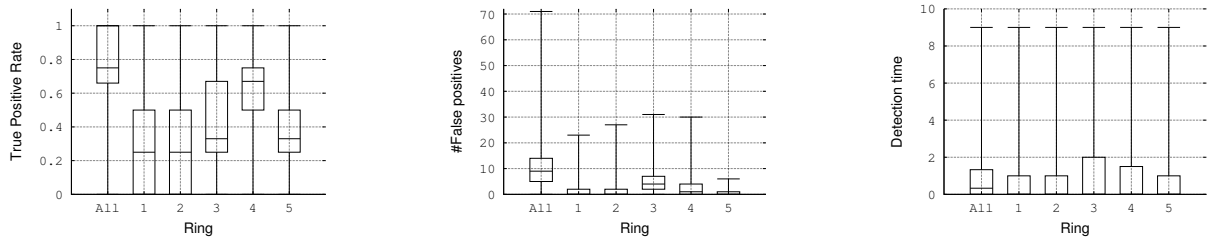
The previous experiment assumes attacks come from beyond outer rings. A skilled attacker however might launch an attack from within the vicinity of the victim, hence avoiding high order rings. The extreme case corresponds to a single ring. However, this rare case implies that the attack is no more distributed and can be detected without collaboration since its traffic is more concentrated and distinguishable. The previous experiment of section VI-C (Fig. 9) shows that deployments with a few rings are not efficient. τ has to be decreased for detecting attacks at the lower level rings also leading to higher false positives. However, this does not mean that *FireCol* cannot detect attacks injected at the lowest rings. For instance, using only one or two rings is not efficient because all traffic, including benign one, is also analyzed by only these rings and so not really distinguishable from attack traffic. However, by using a five rings topology with attacks injected at the first or the second rings, the benign traffic is also analyzed by the upper rings, which helps in distinguishing it from the malicious ones. Hence, section VI-C shows the interest in having five ring topologies. Moreover, Fig. 10 highlights that such a five rings topology is also suited to detect attacks emanating from lower order rings. Figure depicts both the number of false positives (right vertical axis) and true positives, *i.e.*, the TPR (left vertical axis) as a ratio comparing with an attack launched at the first ring. This proves that there is no significant impact on accuracy when attacks are launched from the lowest rings, *i.e.*, in the vicinity of the victims.

E. Impact of the entropy threshold α and profile parameter γ

In this phase, τ is fixed so that the TPR $\geq$ 90%. Table III shows that the TPR can vary 10 points when the high entropy threshold γ varies from 0.6 to 0.8. The number of false positives increases in the same way. In the considered case, false positives are multiplied by 1.5.

TABLE III
EFFECT OF α ON A 5 VIRTUAL RINGS TOPOLOGY

High entropy	0.600	0.700	0.800	0.900
TPR	0.905	0.843	0.787	0.810
False positives	10.320	9.400	6.840	9.720



(a) Detection accuracy (b) False positives (c) Detection delay

Fig. 11. Results of a 5 rings topology with a mix of attacks

γ determines if a rule frequency is out of the profile. The TPR is improved by about 10 points from 0.830 to 0.938 when γ varies from 0.4 to 0.2 (Table IV). However, the number of false positives for $\gamma = 0.2$ is more than twice that for $\gamma = 0.4$. Therefore, it is better to improve the accuracy by adjusting the high entropy threshold α rather than by adjusting γ. This is because the accuracy is improved in a similar manner but the variation in false positives is worse when γ is adjusted.

F. Ring efficiency

In this experiment, four attacks are generated on a 5 rings topology with $\tau = 0.7$: two stealthy (frequency $< 10\%$) at times 40 and 50, and two aggressive (frequency $> 50\%$) at times 50 and 60. The 20th, 50th (median) 80th percentiles, minimum and maximum values of 250 simulation runs are computed. The TPR is detailed for each ring with the best ring being number 4 followed by ring 3 as shown in Fig. 11(a). In fact, 60% of the computed TPRs are within the 20th and 80th percentiles which means that 60% of TPRs are between 0.5 and 0.75 for the ring 4. The 5^{th} ring has a relatively low TPR close to 0.33 for 60% of simulations because it receives no information from upstream routers. This proves that the vertical exchange of scores between rings improves the accuracy.

The TPRs of rings 1 and 2 are very low because the upper rings have already detected most attacks and hence no vertical communication is performed. A similar argument also explains why rings 1 and 2 have less false positives (Fig. 11(b)). The Fig. 11(c) shows the minimum, the 20th, 50th (median), 80th percentile and the maximum detection delay. The median value is always 0 for all rings and 0.33 by considering all of them. This means that the attacks are generally detected in the same window where they occur. The detection delay is generally very low and the worst case corresponds to the ring 3 where 80% of attacks are detected after 2 detection windows at most.

Thus, it can also be observed that the core of the prevention system is located at rings 3, 4 and 5 due to an efficient detection accuracy for a fast detection of attacks. This information is useful for a real deployment because it identifies routers which are the best candidates for supporting *FireCol*. It also shows that the attack is promptly detected and early before reaching the final host.

TABLE IV
EFFECT OF γ ON A 5 VIRTUAL RINGS TOPOLOGY

γ	0.200	0.300	0.400	0.500
TPR	0.938	0.875	0.830	0.728
False positives	9.960	6.660	4.600	4.680

G. Efficiency of the multi-level approach

Fig. 12 plots the relative number of FP compared with the value if no system is used. The first value represents the results when both the selection and score managers are enabled. The second value is when only the selection manager is enabled. τ is fixed to have a detection rate higher than 0.9.

The *selection manager* reduces the number of FP by more than 50%, whereas the *score manager* is generally less efficient. However, it can be noticed that 49 FP are avoided when a 5 rings shield is used. The reduction of false alerts is more important for simulations with a lower number of virtual rings.

H. Percentage of collaborative routers

FireCol effectiveness relies on the collaboration between different IPSs. Since a real deployment of such a system is expected to be incremental, we provide in here a way to check its performance when only few routers support it. A router which does not support *FireCol* is referred to as *non collaborative*. We study two types of non collaborative routers. The first are routers that cannot perform detection but can forward score packets to downstream routers. An operator could use this type of routers to test *FireCol* on only few routers while still ensuring the IPS collaboration. Second type routers act as black holes and do not forward score packets. This can be due to software or hardware limitations or the fact

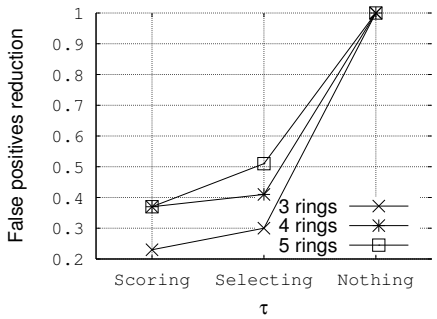
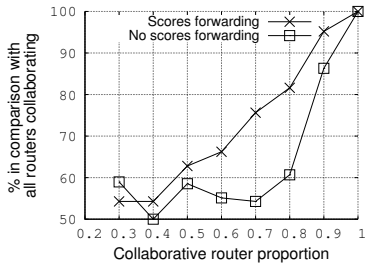
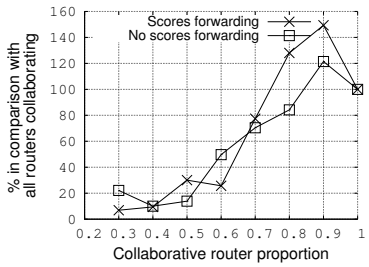


Fig. 12. False positives reduction according to manager activity



(a) TPR



(b) False positives

Fig. 13. Effect of the percentage of collaborative routers

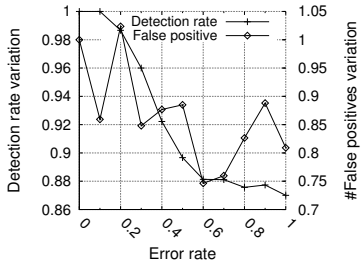


Fig. 14. Impact of *FireCol* configuration errors

that the routers have been compromised in preparation for a future attack.

A four rings topology is used in Fig. 13(a). The x axis represents the proportion of collaborative routers and 200 simulations are conducted for each case. The TPR is plotted against the case when all routers collaborate. Even if *FireCol* cannot be enabled on all routers, forwarding score packets without processing still provides a gain in attack detection.

It can be noticed that non collaborative routers do not have a high impact on the number of false positives as depicted in Fig. 13(b). In addition, for a high percentage of collaborative routers, such as 0.8 or 0.9, the number of false positives is higher than the case of 100% collaborating routers. This is due to a lack of shared information which leads to additional false positives. This occurs for instance when a router does not have the low score from an upstream router of a rule, which would decrease its combined score. However, this value decreases when the percentage of participating routers is less than 80%. During this stage, the IPS does not have enough information to conclude, resulting in few false positives. However, this is also due to a reduced number of participating routers. For example, with 30% collaborating routers, 20 false positives correspond proportionally to 67 false positives for a complete 100% collaboration ($\simeq 20/0.3$).

Finally, when very few routers are deployed, they have various locations regarding the different simulations leading to a high instability in the information exchanged as well as for the TPR and FPs in figure 13. Based on the previous experiments, protecting a new customer with a precision equivalent to 80% of a full deployment requires at least 80% of configured IPSs with a 4 rings-based topology.

I. Configurations errors

Section II mentions issues related to routing instabilities where an IPS might be assigned to the wrong ring. This is referred as a configuration error in this section. During a configuration error an IPS may receive information not sent by a real upstream one. This configuration error may be deliberately inputed by an attacker. In Fig. 14, the ratio of IPSs concerned by such errors are referred as the error rate and varies from 0 to 100%. Figure plots the TPR and number of false positives (FPs) as ratios comparing with the reference value when there is no error. The TPR is never affected by more than 14% since a misconfigured IPS still continues to send information to another randomly selected IPS. Hence, the collaboration is not totally disrupted but is only perturbed. For instance, ring level 5 may directly send score information to the second one. The variation of FP is more chaotic however quite limited. This concludes that *FireCol* exhibits good robustness against configuration errors.

J. Impact of the skewness parameter

The distribution of traffic flows at the routers is defined by the power law formula (Eq. (10)), where c is the *skewness parameter*. The TPR is plotted in Fig. 15 where c varies. As can be noticed, there are limited variations and the TPR is always higher than 0.7. This shows that the skewness parameter has a limited impact. Moreover, this proves that our system also works with different types of background traffic. The worst results are observed for $c \approx 0.3$, which is the value we selected for the other simulations in order to test *FireCol* in worst scenarios.

K. Validation with real Internet topologies

In this experiment, Ark's publicly available *router adjacency* dataset [22] is used to assess *FireCol* against real topologies. Since knowledge about most end-hosts in this data is not provided, nodes in the undirected adjacency graph with a single link are considered final hosts (assuming they are close to the actual end-hosts). Attacks are simulated as before. For each considered end-host, a five rings overlay of IPSs around it is extracted. Figure 16 plots the TPR is plotted against the FPs using 400 topologies (out of about 42,000). The average

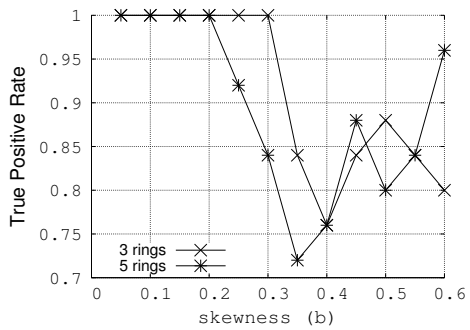


Fig. 15. Skewness impact on the TPR

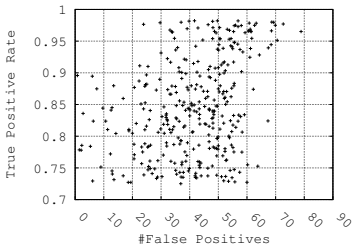


Fig. 16. Accuracy assessment with real topologies (each dot represents one tested topology)

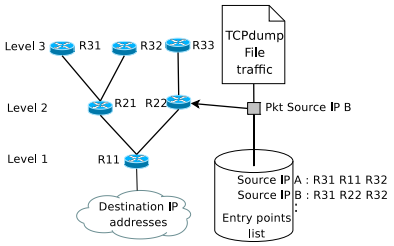


Fig. 17. Dataset injection on a 3 rings configuration with ep = 3. Example with one packet from B which is randomly assigned to R2

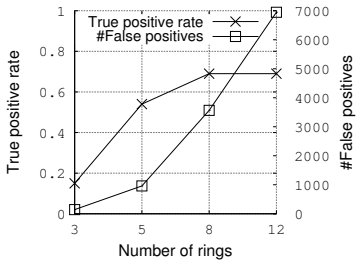


Fig. 18. DARPA'99: TPR and FP

TPR is around 0.87 with about 41 FP. These results are similar to those observed with the generated topologies. However, the relationship between TPR and FP is more unclear since the meantime increase of TPR and FP is not easily distinguishable. When looking into cases with a low TPR (< 0.75), it appears they correspond to topologies with a large fan-out effect. One such topology has 11 IPSs in the second ring and around 4000 in the fifth. This looks abnormal but, as mentioned before, may be due to the non availability of data about the actual end-hosts.

L. Validation with the DARPA'99 Dataset

1) Description: In this experiment, the effectiveness of *FireCol* is tested using traces from the DARPA'99 dataset [23]. Table V gives an overview of this dataset. Even if considered as outdated nowadays, we still tested *FireCol* with it because it is publicly available and heavily used in related work. The fourth week was chosen because it contains real attacks. There are only 4 days with different DoS attack types as detailed in table Table VI. Since there is no available dataset which provides simultaneous parallel traffic traces on different routers, we simulate this by distributing the dataset traffic over the simulation network. The topology is constructed as before and all destination IP addresses are connected to the rings through a unique first router.

The dataset is run with different topologies of varying ring numbers. All the 52 internal IP addresses are considered as customers, resulting in exactly one rule per destination address. To simulate DDoS attacks, the entry points of packets varies. Since the totally random selection is not realistic, we defined for each single source ep different routers as entry points over which packets are uniformly distributed.

Fig. 17 shows an example with $ep = 3$. In our experiments, ep is fixed to 5.

In addition, because the prior knowledge of the capacity of the potential victim is unavailable, *FireCol* relies on a confidence level to confirm potential attacks. This level is computed from the difference between the score and the high potential attack threshold (the denominator normalizes the value between 0 and 1):

$$l_i = \frac{S_i - \tau}{b_1 - \tau(1 - \lambda_{age})} \tag{11}$$

An attack is confirmed if this level is higher than 0.2. Finally, the detection window dw was set to 120 seconds.

2) Results: In the evaluation, *FireCol* detects a DoS attack only if it does so before the attack ends. The output shown in Fig. 18 confirms the results of the previous simulations, *i.e.*, the TPR still proportionally follows the number of protection rings. However, the maximal value it can reach is 0.7 regardless of how many rings are added.

Some attacks, listed in Table VII, always fail to be detected. All but the last one are in fact of application level because their goal is to send few specific messages to exploit a flaw in the protocol or the application. By design, *FireCol* detects *flooding attacks* and cannot logically detect other kinds of attacks. The last attack is a mail bomb, which is an application-level DoS attack whose goal is to saturate the queue of the mail server. In the DARPA dataset, the mail bomb generates about 5 packets per second, which is not a flooding attack at the network capacity level. The mail bomb attack can be detected only if the capacity of the mail server is known.

The number of false positives is also plotted in Fig. 18. It can be noticed that it increases proportionally to the number of protection rings. However, considering the temporal aspect and the different IPSs, the number of false positives is relatively

TABLE V
DARPA'99 DATASET STATISTICS

	#bytes	#packets	#IP addr	duration
Monday	247 MB	1,647,573	860	21:59:55
Wednesday	354 MB	1,766,074	1121	21:59:48
Thursday	459 MB	2,356,503	955	21:59:48
Friday	321 MB	1,945,538	1018	21:59:52

TABLE VI
DOS ATTACKS IN THE DARPA'99 DATASET (FOURTH WEEK)

Monday	Crashii, Smurf
Wednesday	ProcessTable, ArpPoison, Smurf, Mailbomb
Thursday	DosNuke, SshProcesTable, Mailbomb, TearDrop
Friday	Smurf, ArpPoison, Mailbomb

TABLE VII
SUCCESSFUL DDOS ATTACKS IN THE DARPA'99 DATASET

Attack	Description
Crashiis	Malformed request sent to an NT IIS web server
TearDrop	Exploit a flaw in old TCP/IP stack implementations
ArpPoison	Responds to ARP requests with a false address so as to re-route traffic destined to the victim
MailBomb	Sends a burst of messages to a mail server

low as shown in Table VIII. We can deduce that there is an optimal number of rings to be determined. In our simulation the 12 rings architecture generates more false positives than the 8 rings one without improving the TPR. Moreover, attacks are better detected on the 3 highest rings as shown in Fig. 11(a) (Section VI-F). By discarding the alerts of lowest rings ($\notin$ three highest), it can be observed that the number of false positives is divided by 1.82, which shows that it is better to focus the detection on the three highest rings.

VII. Complexity Analysis

A. Communication requirements

To evaluate the scalability of *FireCol*, we study the number of exchanged messages. This requires knowing the number of IPSs composing a ring at a certain level. Because of the fan-out effect (fan) and that each client is directly connected to one single *FireCol* IPS, the number of IPSs n_l at ring level l is given by:

$$n_1 = \begin{cases} 1 & \text{if } l = 1 \\ \lceil fan \times n_{l-1} \rceil & \text{otherwise} \end{cases} \quad (12)$$

An IPS of level l is connected to one IPS of level $l-1$ with probability $1/l$. Hence, the average number of connections between rings l and $l-1$ is equal to:

$$\frac{n_l \times n_{l-1}}{l} \quad (13)$$

Since attacks are blocked at the highest virtual rings, we simulate the case where messages are exchanged between the highest 2, 3, and 4 rings. Fig. 19 shows the maximal number of messages per number of rings by considering the average number of connections between two rings and only one customer targeted with one attack. The scalability is closely dependent on the number of rings. The number of messages is less than 500 for less than 6 rings topology. Considering good configurations highlighted in previous sections, 5 rings topology with the 3 highest rings participating, the average number of messages is only 13.75. In this case, an attack or a false positive generates an overhead of about 17 messages in the network. If we consider an 8 rings topology (best case with the DARPA dataset experiment) the value is about 90, which means that in every dw of 120 seconds 90 messages are exchanged, which is still reasonable. Moreover, it is the maximum number of messages and so it does not always reflect the reality because multiple alerts may be aggregated within a single message.

Fig. 19 does not consider messages exchanged for computing packet rates. However, as this is computed on a single ring, this value is always very low. For instance, computing the packet rate at ring 3 requires at most 4 messages because there are 4 IPSs. To avoid that, different rings separately compute the same packet rate, only one ring can be dedicated to that. For example, if ring 4 detects a highly potential attack, it may request ring 3 to compute the rate. Ultimately, computing the rate at ring 1 is faster as performed by only one IPS. However, it is better to keep some distance from lower lever rings because they are more vulnerable to flooding.

TABLE VIII
False positives for the DARPA99 dataset

# False Positives	3 rings	5 rings	8 rings	12 rings
Per router	23.83	60	62.56	24.46
Per hour	6.5	43.64	162.09	315.77
Per router and per hour	1.08	2.73	2.84	1.11
Per dw	0.22	1.45	5.4	10.53
Per router and per dw	0.04	0.09	0.09	0.04

B. Main metrics

At the end of each detection window, *FireCol* computes different metrics. Assuming that it has incremented the F_i counters (Eq. (1)) on the fly for the n rules, n divisions are required to compute the frequencies (Eq. (1)), $2n$ operations for the entropy (Eq. (2)), $2n$ operations and 1 comparison for the relative entropy (Eq. (4)), n operations and n comparisons to extract $n' \leq n$ suspect rules (Eq. (5)), and $n'+1$ comparisons to examine these rules (Table I). For score factors b_i, when $i \in 1, 2, 3$, $n'' \leq n'$ rules are selected requiring n'' operations for score computation (Eq. (8)). It results that the complexity for suspect rule selection is linear ($O(n)$) both in the required computations and storage.

The score exchange phase (vertical communication) is expected to occur less frequently and for only a small number of rules. Although in the general case the Dempster-Schaffer belief combination can be exponential, it is almost linear for our case because we only investigate one rule at a time [24]. Hence, the linearity property is still respected.

C. Case of multiple customers

In practice, the *FireCol* system is expected to simultaneously protect multiple customers. Assuming N IPSs and C customers to protect, the average number of IPSs at a certain ring level, l, is computed. For this, we first compute the probability $P_N^C(x, l)$ to have x different IPSs at level l.

At level l, there is at least n_l different IPSs corresponding to the ring of a single customer (Eq. (12)). Hence, the maximal number of IPSs for C customers is:

$$m_l = min(N, n_l \times C) \quad (14)$$

We then have:

$$n_l \leq x \leq m_l \quad (15)$$

The number of IPSs at level l, x, is hence between n_l and $min(N, n_l \times C)$. Let $s_N^C(x, l)$ denote the number of ways to define the $n_l \times C$ customer-IPS relationships of the C customers with at most x different IPSs at level l. Since for each customer, n_l IPSs from among the x are assigned, we have:

$$s_N^C(x, l) = \begin{pmatrix} x \\ n_l \end{pmatrix}^C \quad (16)$$

Let $q_N^C(x, l)$ be the number of ways to choose the x different IPSs. The total number of different IPSs has to be x (and not $< x$):

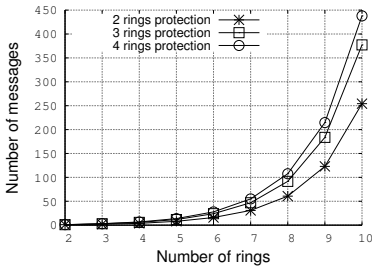


Fig. 19. Messages per number of virtual rings

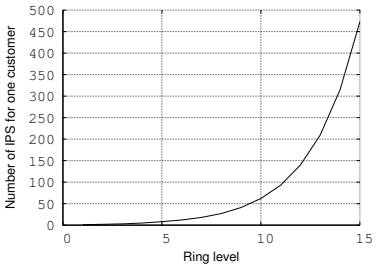


Fig. 20. Number of IPSs per level l (Eq. (12))

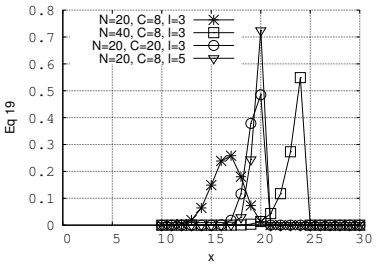
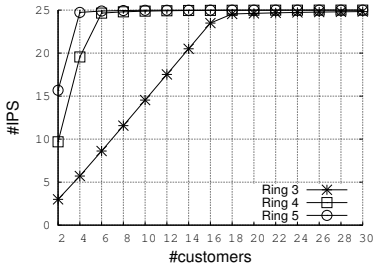
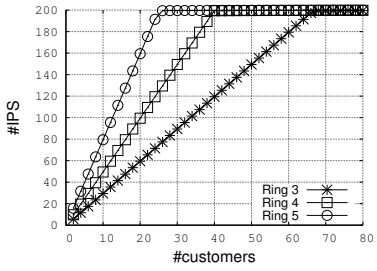


Fig. 21. $P_N^C(x,l)$ probability function (Eq. (18))



(a) N = 25



(b) N = 200

Fig. 22. Average Number of IPSs per virtual ring (Eq. (18))

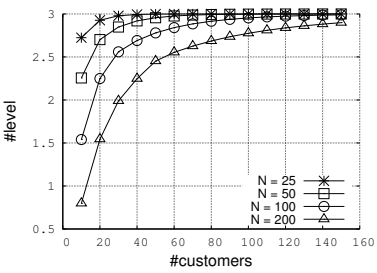


Fig. 23. Average number of different ring levels for a single IPS

$$q_N^C(x,l) = \begin{cases} \binom{N}{x} & \text{for } x = n_l \\ \binom{N}{x}\left(\binom{x}{n_l}^C - \binom{x-1}{n_l}^C\right) & \text{for } n_l < x \leq m_l \end{cases} \tag{17}$$

Therefore, the definition of $P_N^C(x,l)$ for $x \in [n_l, m_l]$ is:

$$P_N^C(x,l) = \frac{q_N^C(x,l)}{\sum_{y=n_l}^{y=m_l} q_N^C(y,l)} \tag{18}$$

Fig. 21 plots the probability function of $P_N^C(x,l)$. When the number of customers increase, the observed peak is thinner, meaning that most IPSs act at the considered level because the load is shared. The peak highlights the most probable number of IPSs with the corresponding configuration. The same effect (for the same reasons) can be observed when the ring level increases because more IPSs are needed to provide protection to all clients. Finally, when the number of IPSs increases, the curve is shifted because more IPSs are available.

Fig. 22(a) and 22(b) highlight the number of IPSs at a certain level with a fan-out effect of 1.5 and a 3 rings configuration (from 3 to 5). Logically the curves tend to the total number of IPSs in the system, where each IPS act at most at each level. Moreover, the more IPSs there are, the less they participate into the rings because the responsibility of the protection of the different hosts is distributed among all IPSs, as illustrated in Fig. 23. This proves that the detection has to be distributed. Furthermore, Fig. 22(a) and 22(b) show the worst case, *i.e.*, the maximal number of IPSs, equivalent to having the maximal number of disjoint routes among customers. If they share more paths, the system can be better optimized by having more IPSs shared between multiple customers.

VIII. Related work

Our previous paper [5] describes a preliminary architecture of *FireCol* with initial simulations. In this paper, these are substantially extended by enhancing and detailing the communication algorithms. A mitigation technique is provided as well as a detailed investigation of *FireCol* configuration. Experimentation with a real dataset and different traffic patterns was also performed as well as an analytical analysis of the complexity.

Even though a publicly available dataset was used, this does not ease the quantitative comparison with related work. Unlike packet-based methods, false and true positives are computed globally taking into account each router and each time window. This is why the focus of the comparison needs to be on qualitative aspects.

Bellovin proposes in [25] the use of distributed firewalls, which is implemented in [26]. However, only firewall rules are exchanged and each firewall must detect the attacks on its own. The authors of [27] propose a similar solution where a Gateway is requested to block the traffic of an attack. In [28], [29], [30], only the DDoS mitigation of the attacks is distributed but the detection is located very close to the victim. Unlike *FireCol*, all previously mentioned solutions do not exploit effective use of collaboration.

In [31], the approach is based on content-filtering. In [32], a peer-to-peer approach is introduced and in [33] mobile-agents are leveraged to exchange newly detected threats. *FireCol* provides a simpler solution in the sense that it uses simple metrics while the former approaches can be costly in terms of resource consumption. Other approaches promoting the use

of simple statistics are not distributed. [34] uses a packet counter per flow, while [35] proposes entropy for a better expressiveness. Authors in [36] use the conditional legitimate probability to determine the deviation from a defined profile.

Mahajan et al. introduce in [37] a technique for detecting overloaded links based on traffic aggregation. Belief functions are also used by Peng et al. in [38] to detect DDoS attacks based counting new IP addresses. These works are close but differ from *FireCol* which detection is focused on the potential victim. Authors in [39] dealt with DoS related overload issues by a cluster architecture to analyze firewall observations.

In [40], a DoS resistant communication mechanism is proposed for end hosts by using acknowledgments. Another solution [41] relies on tokens delivered to each new TCP flow. In [42], each router between the source and the destination marks the path to detect spoofed addresses. Detection of specific SYN flooding attacks at the router level is investigated in [43]. Authors in [44] also analyzed the correlation between the requests and replies to detect flooding attacks to limit overhead. The observation of past attacks or legitimate traffic in order to create a community-of-interest is another alternative [45]. Information sharing about DDoS attacks is also addressed in [46] but from a high-level perspective where a trusted network of partners (networks) is built. Detecting DDoS attacks by detecting IP spoofing is addressed in [47]. [48], [49], [50] are related to our work as the goal is to speed up and limit the costs of packet filtering especially in the case of DoS attack in [48]. Besides, statistics on the network traffic are used like the entropy in [49], [50]. There are also DDoS countering techniques dedicated to specific applications such as web-servers [51] or clouds [52].

Detecting the DDoS attacks at the ISP level was also studied in [53], [54] but these approaches analyze all traffic unlike *FireCol* which is based on a local mechanism enhanced by the collaboration when needed. Although [55] shares information between different network nodes to mitigate efficiently flooding attacks, *FireCol* leverages ring semantic in order to enhance the analysis of shared information.

IX. Conclusion and future works

This paper proposed *FireCol*, a scalable solution for the early detection of flooding DDoS attacks. Belief scores are shared within a ring-based overlay network of IPSs. It is performed as close to attack sources as possible, providing a protection to subscribed customers and saving valuable network resources. Experiments showed good performance and robustness of *FireCol*,and highlighted good practices for its configuration. Also the analysis of FireCol demonstrated its light computational as well as communication overhead.

Being offered as an added value service to customers, the accounting for FireCol is therefore facilitated, which represents a good incentive for its deployment by ISPs.

As a future work, we plan to extend FireCol to support different IPS rule structures.

References

[1] A. Networks, "Worldwide ISP security report," Arbor, Lexington, MA, USA, Tech. Rep., 2010.

[2] T. Peng, C. Leckie, and K. Ramamohanarao, "Survey of network-based defense mechanisms countering the DoS and DDoS problems," *Comput. Surv.*, vol. 39, April 2007.

[3] E. Cooke, F. Jahanian, and D. Mcpherson, "The zombie roundup: Understanding, detecting, and disrupting botnets," in *Workshop on Steps to Reducing Unwanted Traffic on the Internet (SRUTI)*, June 2005.

[4] T. Holz, M. Steiner, F. Dahl, E. Biersack, and F. Freiling, "Measurements and mitigation of peer-to-peer-based botnets: a case study on storm worm," in *Workshop on Large-Scale Exploits and Emergent Threats (LEET)*. USENIX, 2008.

[5] J. François, A. El Atawy, E. Al Shaer, and R. Boutaba, "A collaborative approach for proactive detection of distributed denial of service attacks," in *IEEE Workshop on Monitoring, Attack Detection and Mitigation - MonAM2007*, Toulouse France, 11 2007.

[6] A. Feldmann, O. Maennel, Z. M. Mao, A. Berger, and B. Maggs, "Locating internet routing instabilities," *SIGCOMM Comput. Commun. Rev.*, vol. 34, no. 4, pp. 205–218, 2004.

[7] A. Basu and J. Riecke, "Stability issues in ospf routing," in *SIGCOMM: Proceedings of the conference on Applications, technologies, architectures, and protocols for computer communications*. New York, NY, USA: ACM, 2001, pp. 225–236.

[8] V. Paxson, "End-to-end routing behavior in the internet," *IEEE/ACM Trans. Netw.*, vol. 5, no. 5, pp. 601–615, 1997.

[9] K. Xu, Z.-L. Zhang, and S. Bhattacharyya, "Internet traffic behavior profiling for network security monitoring," *IEEE/ACM Transactions on Networking*, vol. 16, December 2008.

[10] Z. Zhang, M. Zhang, A. Greenberg, Y. C. Hu, R. Mahajan, and B. Christian, "Optimizing cost and performance in online service provider networks," in *USENIX conference on Networked systems design and implementation (NSDI)*, 2010.

[11] M. Dischinger, A. Mislove, A. Haeberlen, and K. P. Gummadi, "Detecting bittorrent blocking," in *SIGCOMM conference on Internet measurement*. ACM, 2008.

[12] Y. Zhang, Z. M. Mao, and M. Zhang, "Detecting traffic differentiation in backbone ISPs with netpolice," in *SIGCOMM conference on Internet measurement conference*. ACM, 2009.

[13] A. Jsang, S. Pope, J. Diaz, and B. Bouchon-Meunier, "Dempster's rule as seen by little coloured balls," *Information Fusion Journal*, 2005.

[14] G. Shafer, *A Mathematical Theory of Evidence*. Princeton University Press, 1976.

[15] T. M. Gil and M. Poletto, "Multops: a data-structure for bandwidth attack detection," in *Proceedings of 10th Usenix Security Symposium*, 2001, pp. 23–38.

[16] T. Peng, C. Leckie, and K. Ramamohanarao, "Protection from distributed denial of service attacks using history-based ip filtering," in *IEEE International Conference on Communications (ICC)*, vol. 1, May 2003, pp. 482–486 vol.1.

[17] C. Siaterlis and B. Maglaris, "Detecting DDoS attacks with passive measurement based heuristics," in *International Symposium on Computers and Communications*, 2004.

[18] "The spamhaus block list," http://www.spamhaus.org/sbl/.

[19] J. François, R. State, and O. Festor, "Activity monitoring for large honeynets and network telescopes," *International Journal on Advances in Systems and Measurements*, vol. 1, no. 1, pp. 1–13, 2008.

[20] N. Brownlee and K. Claffy, "Understanding internet traffic streams: Dragonflies and tortoises," *IEEE Communications Magazine*, vol. 40, no. 10, pp. 110–117, 2002.

[21] M. Faloutsos, P. Faloutsos, and C. Faloutsos, "On power-law relationships of the internet topology," in *SIGCOMM*, 1999, pp. 251–262.

[22] "The cooperative association for internet data analysis," http://www.caida.org/projects/ark/.

[23] R. Lippmann, J. W. Haines, D. J. Fried, J. Korba, and K. Das, "The 1999 DARPA off-line intrusion detection evaluation," *Comput. Netw.*, vol. 34, no. 4, pp. 579–595, 2000.

[24] J. A. Barnett, "Computational methods for a mathematical theory of evidence," in *In Proceedings, 7th Int. Joint Conf. Artificial Intelligence*, 1981, pp. 868–875.

[25] S. M. Bellovin, "Distributed firewalls," *Login magazine, special issue on security*, vol. 24, no. 5, pp. 37–39, Nov 1999.

[26] S. Ioannidis, A. D. Keromytis, S. M. Bellovin, and J. M. Smith, "Implementing a distributed firewall," in *Proceedings of the 7th ACM conference on Computer and communications security (CCS)*. New York, NY, USA: ACM Press, 2000, pp. 190–199.

[27] R. N. Smith and S. Bhattacharya, "A protocol and simulation for distributed communicating firewalls," in *Proceedings of Computer Software and Applications Conference (COMPSAC)*, 1999, pp. 74–79.

[28] Y. You, M. Zulkernine, and A. Haque, "A distributed defense framework for flooding-based ddos attacks," in *Third International Conference on Availability, Reliability and Security (ARES)*, March 2008, pp. 245–252.
[29] X. Bi, W. Tan, and R. Xiao, "A ddos-oriented distributed defense framework based on edge router feedbacks in autonomous systems," in *International Multisymposiums on Computer and Computational Sciences*, Oct. 2008, pp. 132–135.
[30] S. H. Khor and A. Nakao, "Overfort: Combating ddos with peer-to-peer ddos puzzle," in *IEEE International Symposium on Parallel and Distributed Processing (IPDPS) 2008.*, April 2008, pp. 1–8.
[31] I. Yoo and U. Ultes-Nitsche, "Adaptive detection of worms/viruses in firewalls," in *International Conference on Communication, Network, and Information Security (CNIS)*, December 2003, pp. 10–12.
[32] R. Janakiraman, M. Waldvogel, and Q. Zhang, "Indra: A peer-to-peer approach to network intrusion detection and prevention," in *Proceedings of IEEE WETICE*, Jun. 2003, pp. 226–231.
[33] K. Deeter, K. Singh, S. Wilson, L. Filipozzi, and S. T. Vuong, "Aphids: A mobile agent-based programmable hybrid intrusion detection system." in *MATA*, 2004, pp. 244–253.
[34] K. Hwang, S. Tanachaiwiwat, and P. Dave, "Proactive intrusion defense against ddos flooding attacks," in *International Conference on Advances in Internet, Processing, Systems, and Interdisciplinary Research*, 2003.
[35] L. Feinstein, D. Schnackenberg, R. Balupari, and D. Kindred, "Statistical approaches to DDoS attack detection and response." in *DARPA Information Survivability Conference & Exposition*, 2003, pp. 303–314.
[36] Y. Kim, W. C. Lau, M. C. Chuah, and H. J. Chao, "Packetscore: A statistics-based packet filtering scheme against distributed denial-of-service attacks," *IEEE Transactions on Dependable and Secure Computing (TDSC)*, vol. 3, no. 2, pp. 141–155, 2006.
[37] R. Mahajan, S. M. Bellovin, S. Floyd, J. Ioannidis, V. Paxson, and S. Shenker, "Controlling high bandwidth aggregates in the network," *SIGCOMM Comput. Commun. Rev.*, vol. 32, no. 3, pp. 62–73, 2002.
[38] T. Peng, C. Leckie, and K. Ramamohanarao, "Detecting distributed denial of service attacks by sharing distributed beliefs," in *Proceedings of 8th Australasian Conference on Information Security and Privacy (ACISP)*, Wollongong, Australia, July 2003, pp. 214–225.
[39] M. Vallentin, R. Sommer, J. Lee, C. Leres, V. Paxson, and B. Tierney, "The NIDS cluster: Scalable, stateful network intrusion detection on commodity hardware," in *10th International Symposium on Recent Advances in Intrusion Detection (RAID)*, Sept 2007, pp. 107–126.
[40] G. Badishi, A. Herzberg, and I. Keidar, "Keeping denial-of-service attackers in the dark," *IEEE Transactions on Dependable and Secure Computing (TDSC)*, vol. 4, no. 3, pp. 191–204, 2007.
[41] H. Farhat, "Protecting tcp services from denial of service attacks," in *Proceedings of the SIGCOMM workshop on Large-scale attack defense (LSAD)*. New York, NY, USA: ACM, 2006, pp. 155–160.
[42] A. Yaar, A. Perrig, and D. Song, "Siff: a stateless internet flow filter to mitigate ddos flooding attacks," *IEEE Symposium on Security and Privacy*, pp. 130–143, May 2004.
[43] D. Nashat, X. Jiang, and S. Horiguchi, "Router based detection for low-rate agents of ddos attack," in *High Performance Switching and Routing, 2008. HSPR 2008. International Conference on*, May 2008, pp. 177–182.
[44] H. Wang, D. Zhang, and K. Shin, "Change-point monitoring for the detection of dos attacks," *IEEE Transactions on Dependable and Secure Computing (TDSC)*, vol. 1, no. 4, pp. 193–208, Oct.-Dec. 2004.
[45] P. Verkaik, O. Spatscheck, J. Van der Merwe, and A. C. Snoeren, "Primed: community-of-interest-based ddos mitigation," in *Proceedings of the SIGCOMM workshop on Large-scale attack defense (LSAD)*. New York, NY, USA: ACM, 2006, pp. 147–154.
[46] G. Koutepas, F. Stamatelopoulos, and B. Maglaris, "Distributed management architecture for cooperative detection and reaction to ddos attacks," *J. Netw. Syst. Manage.*, vol. 12, March 2004.
[47] I. B. Mopari, S. G. Pukale, and M. L. Dhore, "Detection of ddos attack and defense against ip spoofing," in *Proceedings of the International Conference on Advances in Computing, Communication and Control (ICAC3)*. New York, NY, USA: ACM, 2009, pp. 489–493.
[48] A. El-Atawy, E. Al-Shaer, T. Tran, and R. Boutaba, "Adaptive statistical optimization techniques for firewall packet filtering," in *IEEE International Conference on Computer Communications (INFOCOM)*, April 2009.
[49] H. Hamed, A. El-Atawy, and E. Al-Shaer, "Adaptive statistical optimization techniques for firewall packet filtering," in *IEEE International Conference on Computer Communications (INFOCOM)*, April 2006, pp. 1–12.
[50] A. El-Atawy, T. Samak, E. Al-Shaer, and H. Li, "Using online traffic statistical matching for optimizing packet filtering performance," in *IEEE International Conference on Computer Communications (INFOCOM)*, May 2007, pp. 866–874.
[51] D. Das, U. Sharma, and D. K. Bhattacharyya, "Detection of http flooding attacks in multiple scenarios," in *International Conference on Communication, Computing & Security*. ACM, 2011.
[52] H. Liu, "A new form of dos attack in a cloud and its avoidance mechanism," in *Workshop on Cloud computing security workshop*. ACM, 2010.
[53] A. Sardana, R. Joshi, and T. hoon Kim, "Deciding optimal entropic thresholds to calibrate the detection mechanism for variable rate ddos attacks in isp domain," in *Information Security and Assurance, 2008. ISA 2008. International Conference on*, April 2008, pp. 270–275.
[54] B. Gupta, M. Misra, and R. Joshi, "Fvba: A combined statistical approach for low rate degrading and high bandwidth disruptive ddos attacks detection in isp domain," in *Networks, 2008. ICON 2008. 16th IEEE International Conference on*, Dec. 2008, pp. 1–4.
[55] J. L. Berral, N. Poggi, J. Alonso, R. Gavaldà, J. Torres, and M. Parashar, "Adaptive distributed mechanism against flooding network attacks based on machine learning," in *Workshop on Artificial Intelligence and Security*. ACM, 2008.

Jérôme François is a research associate at the Interdisciplinary Centre for Security, Reliability and Trust of University of Luxembourg. He studied at ESIAL, a french leading school in computer science. His Ph.D. was supervised by Olivier Festor and Radu State in INRIA Lorraine - Nancy Grand Est, France. He received his Ph.D. on robustness and identification of communicating applications from the University Henri Poincaré in Nancy, France, in December 2009. He published in major conferences on topics related to security and network management. His main research interests are related to security and privacy including large scale anomaly detection, fingerprinting, security in vehicular networks and digital forensics.

Issam Aib received a DEA (MMath) in Computer Science from the University of Pierre & Marie Curie, Paris, France, in 2002. He completed his PhD in 2007 under the supervision of Pr. Raouf Boutaba, University of Waterloo, Canada, where he continued as a research associate until 2010. He is now serving as an Application Services Manager at the Ontario Public Service, I&IT Community Services Cluster. His research covered topics in policy-based and business-driven management, trust management, cloud computing, and network security. Issam was twice the recipient of the best paper award of the IFIP/IEEE International Symposium on Integrated Network Management (IM) in 2007 and 2009, respectively.

Raouf Boutaba received the M.Sc. and Ph.D. degrees 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 and a distinguished visiting professor at the division of IT convergence engineering at POSTECH. His research interests include network, resource and service management in wired and wireless networks. He is the founding editor in chief of the IEEE Transactions on Network and Service Management (2007-2010) and on the editorial boards of 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, and the Joe LociCero and the Dan Stokesbury awards in 2009. He is a fellow of the IEEE.