# University of Waterloo
# Technical Report CS-2007-09

# The FireCollaborator: a Collaborative Approach for Attack Detection

Jérôme François[1], Adel El-Atawy[2], Ehab Al-Shaer[3], and Raouf Boutaba[4]

[1] This project was done as an internship at University of Waterloo
His permanent affiliation: MADYNES - INRIA Lorraine, CNRS, Nancy, France
jerome.francois@loria.fr
[2] School of Computer Science, DePaul University, Chicago, IL 60604, USA
aelatawy@cs.depaul.edu
[3] This project was done as part of his work as Adjunct Professor at University of Waterloo
His permanent affiliation: School of Computer Science, DePaul University, Chicago, IL 60604, USA
ehab@cs.depaul.edu
[4] David R. Cheriton School of Computer Science, University of Waterloo, Waterloo, Ontario, N2L 3G1, Canada
rboutaba@bbcr.uwaterloo.ca

**Abstract.** The distributed denial of service attacks are a major threat on Internet and detecting this kind of attack as far as possible from the victim in order to save resources is a real challenge. We propose a new framework to deal with this problem which is based on Intrusion Prevention System (IPS) deployed on a Internet Service Provider (ISP) level. The key point is to use compressed metrics based on the routing rules in order to extract suspect traffics thanks to a collaboration between routers on the same path and finally confirm an attack by using the IPS on different path but on the same level (number of hops before the potential victim). The main metric we use is the frequency of a rule but we decide to use the entropy too. We are able to detect a change in the traffic and define what rules changes a lot and have a too higher frequency. After the share of the different belief of each IPS is needed to determine the very potential attack before confirm it by computing the packets rate.
There are three main advantages of our proposition. The first is that because you select the rules, you can analyze precisely these rules and be sure that there is an attack. The second is that we save a lot of resources (network, CPU and storage) thanks to the choice of the metrics and the selection of rule. Finally because we determine the attack as far as possible the final host can save resources too because the attack traffic is blocked early and congestion can be avoided.

## 1 Introduction

The number of network threats increase a lot each year. That's why the number of security tools increased so much last years like firewalls or security patches. However at each time a solution is proposed for a threat, the attackers find another way to do malicious activities. In fact the attacks are more and more various and smart. Now the attackers can generate packets which seem to be legitimate traffic using a lot of parameters and different mechanisms like IP spoofing. However a large problem is the use of an army of zombies, the attackers attack very vulnerable machines and control them. After they can do anything they want with them, they can send a lot of Spam emails or do a denial of service attack. This last case is a huge problem because thanks to this distributed scheme (Distributed denial of service attack), the attacker can attack and crash protected servers easier because a lot of current solutions don't detect it. There are different propositions but some use content filtering and others are have to be applied directly by the victim. In theses cases, the victim have to use its resources for the detection and the attack traffic used a lot of bandwidth along the path.

The challenge is to detect the attack as far as possible from the victim and so counter the attack before it reaches the victim. Actually it's a proactive security. In this case the victim saves resources and the bandwidth along the path is saved. To deal with this problem, our goal is to detect the attack at the provider level thanks to Intrusion Prevention System (IPS). The main idea of the FireCollaborator is that at this level an IPS cannot easily detect an attack because it see only a little part of the traffic, so a collaboration is needed between the IPS. The decision is taken thanks to a collaboration between them. Moreover we have excluded content filtering solutions because at the ISP level, the flows are very diversified and the traffic is very high and content filtering need to analyze the payload of the packets and maybe aggregate the payloads before which need too much resources.

In the next section, the overall architecture will be presented. After the section 2 is a summary about the entropy definition. A full description of the different components and what they do is given in the section 3. A simple simulator implementation and our experiments are described in the section 4. The 5th section is about the related works. Finally we will conclude and introduce our future work.

## 2 Overall Architecture

Our goal is to detect the DoS attack as far from the victim as possible. Thus we aim to detect and counter the attack at the Internet Service Provider Level. Actually this level is splitted into a lot of levels or ring. Before reaching an host, a packet pass through different IPS. 2 IPS which are on the same path are qualified as sequential contrary to IPS which are parralels and which deal with different packets. The figure 1 is an example with few rings and the arrows represent the

attack traffic. Intuitively and as you can see more you are near the source more the traffic is concentrated. Then it's clear that it's easier to detect the attack at a low level (at a ring near the source) but our goal is to detect is at far as possible and so our approach use a detection process going from outer to inner ring.

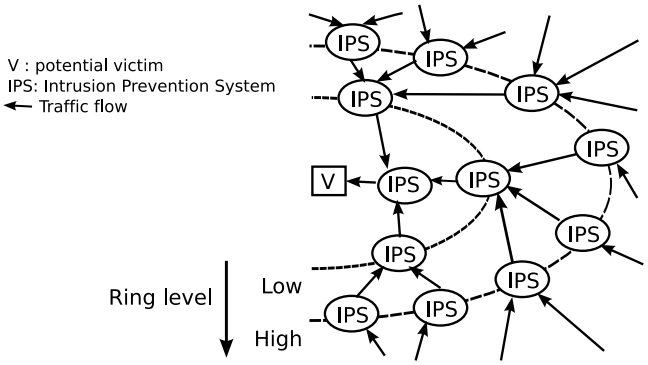


**Fig. 1.** A simple example of the IPS ring topology

An IPS have only rules (routing rules) but we assume that each rule matches packet with the same destination host which can be a computer, a network or a link. If it's not the case we can easily transform the rules to obtain that. The frequency of a rule aims to estimate if the rule is frequently matched. The frequency is used in [1] to optimize the rule ordering of a firewall. Assume that $f_i$ is the number of packets matching the rule $R_i$, the frequency of the rule is:

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

A frequency can be an indicator of a possible DoS attack which generates a lot of packets sent to the same hosts and so the frequency of the corresponding rules would be high. But we cannot only use directly the frequency because a service or an host can be popular and so have a high frequency. Moreover when an attack appears the behavior changes and it's not a high frequency we have to detect but a increase of the frequency. This variation of the frequency seems to be a good metric but in fact the variation of the frequency of one rule can be due to the fact that the traffic is more and more concentrated or due to other traffic which disappear. So we have to evaluate correctly the risk that there is an attack but in all cases if we want to confirm an attack we have to determine the packets rate of a traffic and the corresponding capacity of the server in order to see if there is an attack or not. We consider that a number of packets per second above the capacity of the server is an attack but in reality it can be due to a bad modeling and in all cases the final host could crash.

So there are different levels of ring. The general scheme is:

- an IPS detects an abnormal value using the frequency and the entropy (we discuss later the details)
- the IPS cannot conclude directly and that's why it send this information to next IPS on the path (vertical communication)
- at a lower level, an IPS receives different informations and can detect that a traffic is suspect on several upstram IPS and maybe at its own level. An horizontal communication is initiated in order to determine the general packets rate of this traffic and compare with the theoric capacity of the destination host.

Of course after the detection process, you have to provide a response to counter the attack. This is simple because in fact you detect an attack about an 'accept rule'. So you create another rule which is the same with 'deny' and you propagate from the IPS which detects the attack to the others on the same ring (horizontal communication) and to the upstream IPS (vertical communication), the new rule is propagated on the same manner on each higher level. Of course, you have to define a time limit which can be calculated from the aggressiveness of the attack. However our main goal is to provide a solution for the detection and we focus only on this point.

## 3 Entropy and relative entropy

The entropy for a set of rules is defined as.

$$H = -E[log f_i] = -\sum_{i=1}^{n} f(i) log_n(f_i) \quad \textit{where } f_i \textit{ is the frequency of } R_i \tag{2}$$

The entropy give an idea of the distribution of traffic. For example if all frequencies of the rules of an IPS are equal, the entropy is 1. More various the frequencies are various more the entropy decreases.

The relative entropy is another metric which is very useful because the aim is to compare the distance between two distributions. It is also called the Kullback-Leibler distance which is defined as:

$$K(f, f') = \sum_{i=1}^{n} f_i \ \psi_i \textit{ where } \psi_i = log\frac{f_i}{f'_i} \quad \begin{array}{r} f_i \textit{ is the frequency of } R_i \textit{ in the second distribution} \\ f'_i \textit{ is the frequency of } R_i \textit{ in the first distribution} \end{array} \tag{3}$$

If the distribution are equivalent, the entropy is 0. More different the distribution are more the value increases. Actually, $\psi_i$ represents the drop in the information content of this individual flow.

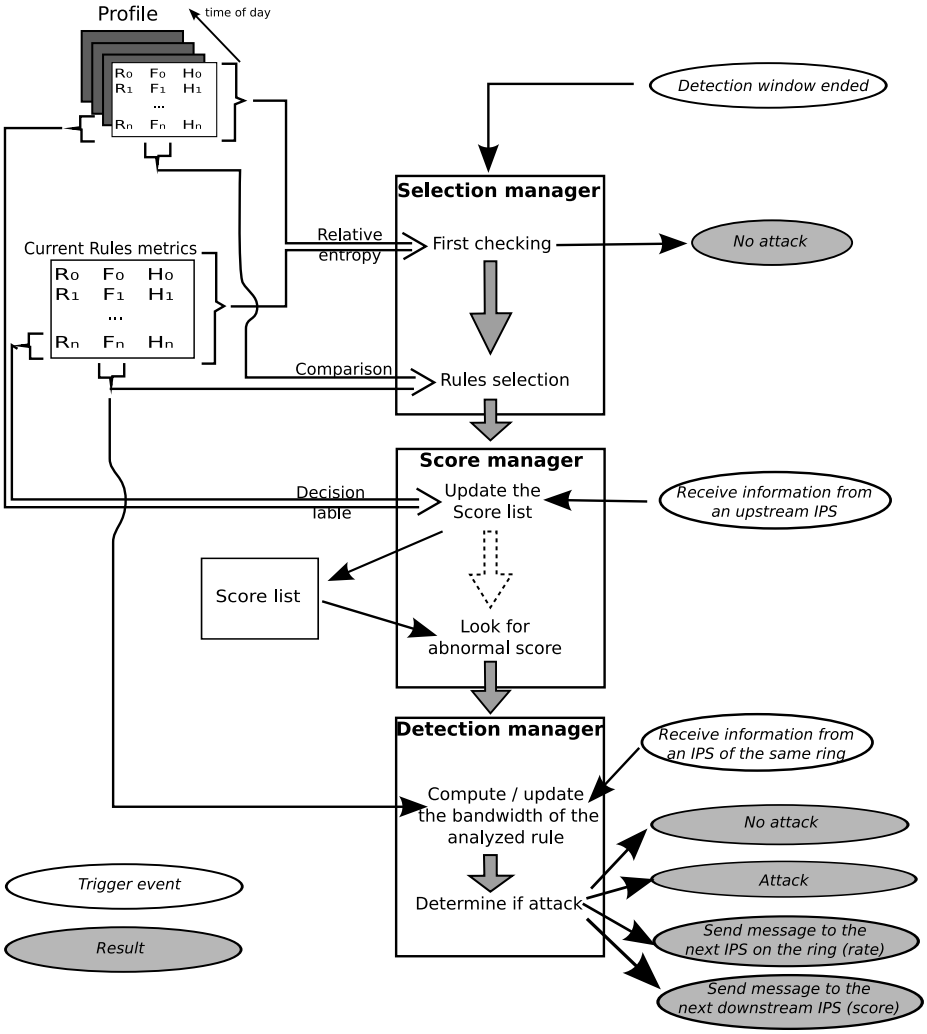


**Fig. 2.** The different components and the detection process

# 4 Components

## 4.1 Profile the traffic

As we will see we cannot try to determine an attack for each rule and a selection have to be done. A profile is very useful to detect abnormal values and so for the selection of the rules. In fact we have to remember the values of the rules frequency and the corresponding entropy. You can use different profile like daily or weekly for example or more simple the previous values can be the profile.

## 4.2 Selection manager

Due to scalability and real time constraints, we cannot analyze all rules of each IPS. So an IPS have to select the rules to analyze. Of course, it's not a random selection but a selection based on the attack belief. So the selection manager have to determine the rules for which something strange can be observed.

It can use two metrics which are the frequency and the entropy. In fact the detection begin at the end of a time window, the IPS will use the profile to determine if the current distribution is different from it. Thus, the trigger of a possible attack is the relative entropy defined in 3. In fact if there are significative variation this value must be above a threshold. So if the relative entropy is too low no rule will be selected because the traffic have not changed. This is the 'First checking' step on the figure 2 and so you continue to the next step if:

$$K(f, f') > \omega \text{ with } f \text{ the current distribution} \\ \text{and } f' \text{ the profile distribution} \tag{4}$$

This first checking is very useful in order to save resources if the traffic is the same than the normal profile. This trigger is a global trigger and means that there are changes and now the goal is to detect what are the changes and above all the changes which are potential attack. To have an objective view, you have to know the 'nature' of the traffic and so the entropy can help you. Of course the frequency is also a useful metric and that's why these 2 metrics can be used. In order to be more simple and clear, these metrics will be discretised. Each metric can be low or high depending on a threshold $\alpha$ for the entropy and $\beta$ for the frequency.

We have several case which are summarized in the table 1 but we cannot apply to all rules in order to save resources. The selection process is simple and use the profile. In fact all rules for which the frequency is different from the profile above a threshold are used:

$$R_i \text{ is selected at time } t \text{ if } \frac{f_i(t)}{f_i(profile)} > 1 + \gamma,\ 0 \leq \gamma \leq 1 \tag{5}$$

However, rules can have very high increases due to a little frequency like at the beginning of a communication. Thus only rules with a significant frequency will be selected. So before to use the previous statement we have to check:

$$R_i \text{ can be selected at time } t \text{ if } f_i(t) > \epsilon \tag{6}$$

Moreover, as you can see, we use the value $\frac{f_i(t)}{f_i(profile)}$ in the equations 5 and 3. Thus we can store this value in the table of the current metric, the frequency is updated when a packet arrive and at the end of the detection window all $\frac{f_i(t)}{f_i(profile)}$ can be computed before begin the detection process. Furthermore, we can deduce a relation between $\psi_i$ and $\gamma$:

$$\begin{aligned} \frac{f_i(t)}{f_i(profile)} &> 1 + \gamma \\ \psi_i &> log(\gamma + 1) \end{aligned} \tag{7}$$

### 4.3 Score manager

After this first selection, we have now to detect the potential attack. Actually, it's like a second selection in order to save resources like the first time because don't forget after you detect a potential attack you have to confirm the attack or not by computing the corresponding packets rate and so communicate with other IPS. So the selection manager gives the list of selected rules to the score manager which have to determine the agressiveness by using the table 1. If the entropy is high that means that the traffic is well distributed, thus there are two cases:

- the frequency is high (case 1): in this case an attack is potential because the traffic is well distributed and so all the frequencies are about the same but there are a lot of rules and so they cannot be all high, that's why this rule is very different than others and is a potential attack
- the frequency is low(case 2): so all the frequencies are about the same and it seems than is not an actual threat but maybe later because the frequency increased (first selection) and so can be higher the other frequency later

Now if we consider the entropy is low, there are also two cases but the conclusions are not so evident because in this case the frequencies are different:

- the frequency is high(case 3): in this case, the frequency increases and now is high but the problem is that the entropy is low and so the frequency can be high because there are a lot of different frequency which can be lower. We think in this case the attack is potential but not as much as when the entropy is high.
- the frequency is low(case 4): the frequency are very different and so probably there are high frequency and thus the current frequency is not a direct threat

As you can see only the first three cases seems to be used for the detection. Moreover it's clear that all this cases are not the same attack probability and an order from the most potential to the less potential could be: case 1 > case 3 > case 2.

| Case number | Entropy | Frequency | Conclusion |
|---|---|---|---|
| 1 | High | High | Potential |
| 2 | High | Low | Potential later |
| 3 | Low | High | Medium potential |
| 4 | Low | Low | Not potential |

**Table 1.** The decision table

Because the IPS can exchange its belief of an attack, this belief must be conditionned with the case. So 3 factors $b_1$, $b_2$ and $b_3$ are needed to be defined for respectively case 1, case 2 and case 3. Then you use this factor with the frequency to obtain a score:

$$\begin{aligned} S_i = f_i * b_j \ &where \ b_j \ is \ the \ corresponding \ factor \\ &after \ match \ f_i \ and \ the \ entropy \ in \ the \ table \ 1 \end{aligned} \tag{8}$$

As we said before, we are able to determine a score of a possible attack but now we have to confirm the attack or not. To do that each IPS have to maintain a score list of threats ie a list with tuples like $< Rule,\ Score,\ Last\_update >$.

This list is updated at the end of the detection window:

1. all scores are reduced by a factor $score_{factor}$ for example 0.5 and a score is removed if it is too low ie below a threshold $\upsilon$
2. the score indicates a belief and the IPS has the current scores list, the new scores list (previously calculated) and a list of the scores from other IPS. So thanks to a belief combination we can compute the new current scores which will be used for the next step. There are different belief combination functions but we have decided to use for our first tests the dempster's rule. In the future, we'll probably test other functions. For a clear representation of the demspter's rule, please see [2]. So the belief combination for Z of n scores is:

$$\begin{aligned} m_{1,2,...n}(Z) &= \frac{1}{1-K} \sum_{A \cap B ... \cap X = Z} m_1(A) m_2(B) ... m_n(X) \\ K &= \sum_{A \cap B \cap X = \emptyset} m_1(A) m_2(B) ... m_n(X) \end{aligned} \tag{9}$$

Actually, even if you have no score computed for a rule, the past value of it can be use but of course the past value is less important thanks to the division by a factor.

As you can see on figure 2 the next step is the detection of abnormal score. Thus at this step, the IPS have to find all rules in the scores list for which $Score > \tau$ where $\tau$ is a simple threshold. Now, these rules are considered as very potential attacks and the scores which are too old (using the $Last_{update}$ value) are removed (in fact it can be done by the previous step). Otherwise, the IPS

sends the scores to the next downstream IPS when $Score \leq \tau$, it's a vertical communication illustrated in figure 3

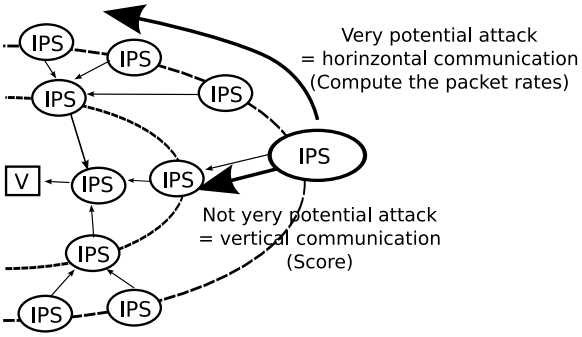


**Fig. 3.** Two kinds of communication

### 4.4 Detection manager

There is a final step which consists to determine the packets rate in packets per second. To be clear we present the method only for a specific rule but of course the IPS do that for each very potential attack. Firstly, the IPS calculate the packets rate thanks to the frequency of the rule and the general bandwidth (BW) like in the algorithm 1. If the rate is more than the capacity of server, the attack is directly detected. If the attack is not evident, the collaboration of other IPS is needed to determine the packets rate on the entire ring thanks to an horizontal communication (on the same ring). The algorithm 1 introduces it, the IPS send a message to the next sibling with its Id, the kind of the message which is 'Rate' and the rate it computes. As you can see, we consider a capacity in packets per second but this capacity can be given in bytes per second. In this case, additionally to the frequency we have to sum the size of the packets for each rule.

So when an IPS receives a request to calculate the packets rate. First it check if it is not the initiator of the request and thus in this case there is no attack because normally the previous sibling should detect it. Otherwise, it calculates the new rate by adding its own rate and try to determine if the capacity is reached. In this case the attack is detected, otherwise the message is sent to the next IPS with the new rate and so one. The algorithm 2 shows the details.

**Algorithm 1** Compare the rate with the capacity of the host

$Rate(R_i) \leftarrow F_i * BW$
$cap$ is the capacity of the targeted host in packets per second
**if** $Rate(R_i) > cap$ **then**
  Attack detected
**else**
  $msg \leftarrow \{IPS\_id,' Rate', R_i, rate(R_i)\}$
  send $msg$ to the next IPS on the ring
**end if**
Remove the rule from the very potential attack list

**Algorithm 2** How to deal with a request?

$cap$ is the capacity of the host in packets per second
$msg = \{initiator,' Rate', globa\_rate\}$
**if** $initiator = IPS\_id$ **then**
  no attack
**else**
  **if** $global\_rate + rate(R_i) > cap$ **then**
    attack detected
  **else**
    $msg2 \leftarrow \{IPS\_id,' Rate', R_i, global\_rate + rate(R_i)\}$
    send msg2 to the next IPS on the ring
  **end if**
**end if**

## 5 Experiments

This section describes our first simulations. The goal is to show if our proposition can really detect attacks. So we implement our own simulator without real packets and with a simple configuration.

### 5.1 Configuration

The topology of the network is illustrated in the figure 4. As you can see there are 5 hosts to defend and so 5 rules used by each router. So each router can reach each hosts. There are 3 rings because the IPS are located on the routers. The delay for the transmission on a link is 1. the detection process is triggered every 1 step. All hosts have the same capacity of 100 packets/step.

Our simulator is implemented in Java. There are a major class Router whose the goal is to receive traffic packets which are only represented by one rule, deal with this packets (update the frequency) and route the packets to another router or the final host. Some packets can be score packets that means that they have score informations from another router and in this case the router save the information until the end of the window. A manager deals with the event like the packet transmission because in fact a router don't send directly a packet to another but add an event representing this transmission in an events list. There

| | |
|---|---|
| $\gamma$ | 0.4 |
| $\omega$ | 0.05 |
| High entropy | 0.8 |
| High frequency | 0.4 |
| $b_1$ | 1 |
| $b_2$ | 0.65 |
| $b_3$ | 0.8 |
| $score_{factor}$ | 0.5 |
| $\epsilon$ | 0.01 |
| $\upsilon$ | 0.05 |
| $\tau$ | 0.5 (or 0.7) |

**Table 2.** The values of the different parameters

is another important kind of event which is the end of the detection window of a router and in this case the detection process is triggered. We think that this description is enough to understand the global functioning of our simulator because too much details could be boring.

For our first tests we have only generated normal traffic by using a uniform distribution on each outside router. We try to determine a good value of the parameters and finally we have fixed some parameters but a study about the parameters will be necessary in the future. The table 2 introduces the values of the parameters.

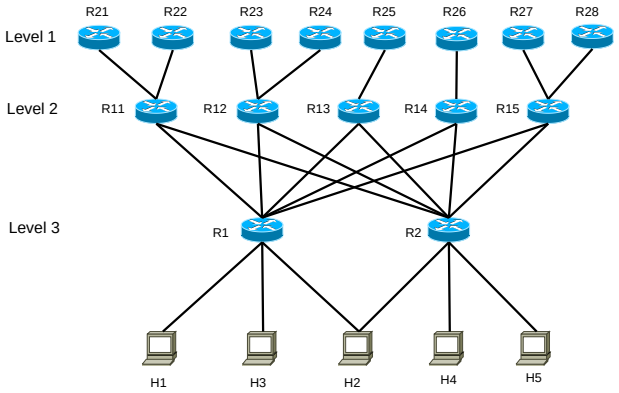


**Fig. 4.** The network topology

For the profile, because we don't generate periodic traffic, we decide to use the previous values. So we use an exponential moving average which is defined by:

$$f_i(new_profile) = a.f_i + f_i(profile).(1 - a)$$

The parameter $a$ can be tuned and for our tests the value is 0.5.

### 5.2 Evaluation criteria

In our simulations, a router detects an attack if the score is abnormal. The last step which is the calculation of the rate is not important for our simulation because our goal is to see if the selection process is good ie the rate calculation is not triggered when there is no attack. So we consider this case as a false positive. Furthermore, we will evaluate where the attack is detected (the ring level) because our goal is to detect the attack as far as possible from the targeted host. Finally we evaluate when the attack is detected because faster you detect it faster you can counter it. In our simulation, we don't take preventive measures after the detection in order to check if there is a real attack or not. Moreover we will determine the percentage of routers that detect the attack.

### 5.3 Results

Firstly, we generate a normal traffic on the outside routers. Actually, 40 packets are generated on the routers from 21 to 25 and 50 on the routers from 26 to 28. So the traffic generated shouldn't trigger an attack on the hosts. After we generate an attack traffic on different outside routers. So an host indicated when it is attacked.

All our experiments are done 5 times. The simulator will be improved in order to multiply the number of simulations and automatic analysis of the results. The simulation duration is 100 and the attack are triggered at step 50 with a duration of 10 steps. Because the profile is based on the previous values, the attacks detected on the first 10 steps are discarded from our analysis because the profile have to be established before.

**No attack** The aim here is to see if the number of false positive is not too high. So there is no attack but the traffic is very high. Firstly we use a value of 0.5 for $\tau$, the results are summarized in the table 3 and on the figure 5 representing the average of the different values. We name routers the percentage of routers on a level which detect the attack among the routers which can detect the attacks. The time represents the number of steps after the first time the detection was possible. So a time of 0 means that the attack is detected when the first packets are routed. As you can see the number of false positives is 10 on all levels. There are 15 routers which trigger the detection process during 90 steps (100-10). So we can consider that this number of false positive is reasonable.

If $\tau = 0.7$, the number of false positives is 1 which is very low. Of course this parameter is very important and that's why the next analysis will be helpful to determine a right value for it.

| | Level 1 | | | Level 2 | | | Level 3 | | | detection % |
|---|---|---|---|---|---|---|---|---|---|---|
| | routers % | time | fp | routers % | time | fp | routers % | time | fp | |
| No attack | 0 | 0 | 0 | 0 | 0 | 6,8 | 0 | 0 | 1,2 | |
| type 1.1 | 20 | 0,2 | 0 | 40 | 0,2 | 4,8 | 0 | 0 | 0,4 | 80 |
| type 1.2 | 20,12 | 0,8 | 0 | 60 | 0,2 | 3,6 | 40 | 0,6 | 1,6 | 100 |
| type 2.1 | 0 | 0 | 0 | 32 | 0,6 | 2 | 10 | 0,2 | 0 | 100 |
| type 2.2 | 12,5 | 0,2 | 0 | 36 | 0,4 | 1,2 | 60 | 0,8 | 0 | 100 |
| type 3.1 | 0 | 0 | 0 | 16 | 2,2 | 2,6 | 0 | 0 | 0 | 80 |
| type 3.2 | 0 | 0 | 0 | 44 | 0,2 | 1,6 | 30 | 0,8 | 0 | 100 |
| type 4 | 100 | 0,8 | 0 | 40 | 1,4 | 0,2 | 0 | 0 | 0 | 100 |

**Table 3.** The results with $\tau = 0.5$, fp means false positive

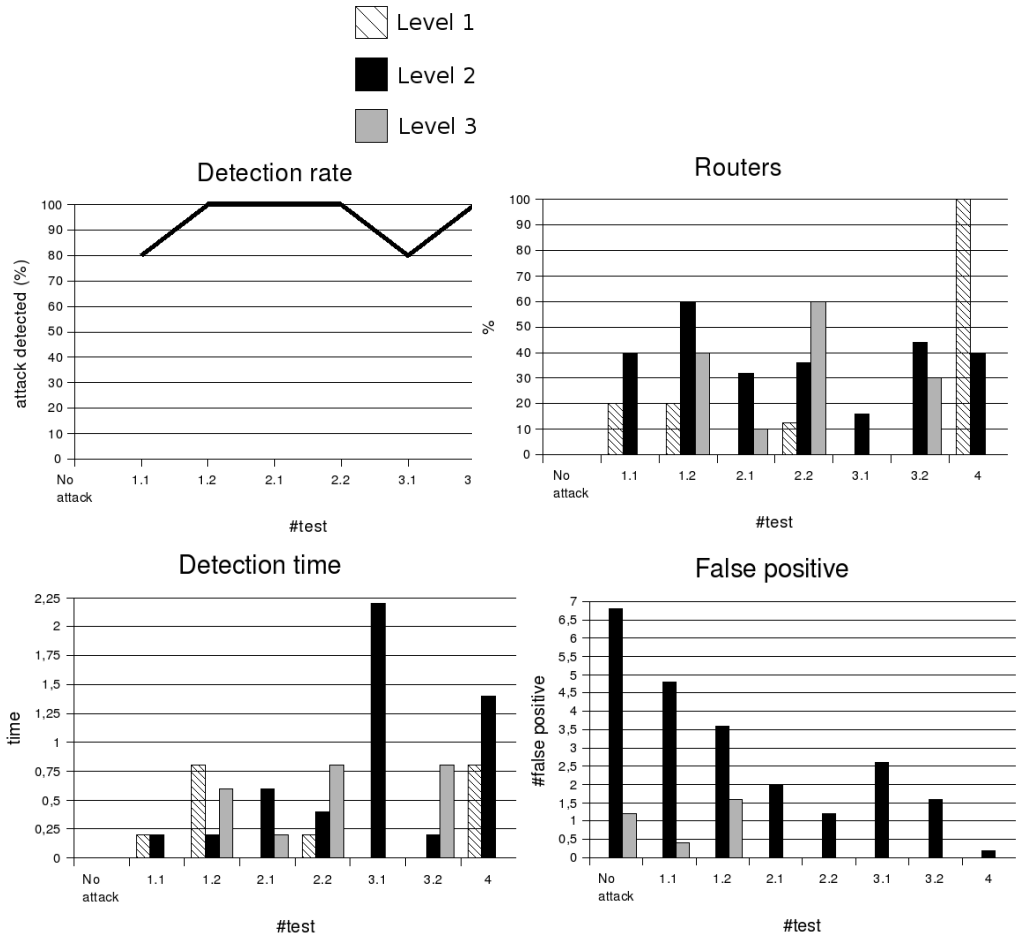


**Fig. 5.** The different average values regarding the ring level

**With an attack** Now, an attack is generated against the host 1 from the routers 21, 22, 23, 24 and 25. 5 packets are generated at each step at each router. Even if this number is low, it's sufficient to attack the host because the normal traffic is very high. The row of attack type 1.1 in the table 3 introduces the results and you can see the same results on the figure 5. The last column shows that in 5 simulations, the attack was not detected one time only. This not a bad result because the attack is very low.

The attack type 1.2 is the same with 10 generated packets per step. Fortunately, the results are better and all the attacks are detected. In the both cases we can see that the attack is well detected at level 2. So the method looks good because the false positive are low too. it can be amazing to see that the third level when the traffic is concentrated, the detection don't work well but in fact when a level 2 IPS detects an attacks, no scores are sent to the level 1 and so the IPS of this level have not enough information to conclude that there is an attack. As the same manner the level 1 IPS have not enough information to detect the attack. So it means that conclusion cannot be done without information from other IPS which is the key point of our proposition.

When we do the same attacks with $\tau = 0.7$, the results were very bad with for example only 2 attacks detected in the case 1.2. That's why we decide to use only $\tau = 0.5$ for our next simulations. Moreover in the next section we will see that it looks to be a good choice.

The attack type 2.1 is an attack against H2 with 5 packets generated from the router 26, 27 and 28. The number of generated is 10 in the case 2.2. So the attack is similar to the previous case but in fact these routers generates more normal packets so the attack packets are in a less proportion. In this case, all the attacks are detected and the conclusion are the same ie the level 2 is the best level to detect the attack.

The type 3.1 and 3.2 generate an attack on all the outside routers to host h1. So the results are about the same because the attack is detected as level 2 and so it's the same as the previous attacks. The only difference in the traffic can be visible at the router 1 but it doesn't appear in the results because the attacks are detected before.

Finally, the case 4 is a basic attack with a lot of generated packets (30 on one router) and logically the first router on the path detects the attacks directly. The next routers on level 2 don't detect always the attacks due to the same reason as before between level 2 and level 1.

### 5.4 Impact of the different parameters

We will discuss now the relationship between the performances of our system and the value of different parameters of the table 2. 5 parameters will be tested: $\gamma$, $\tau$, $\omega$ and the high entropy threshold $\alpha$. In fact, for each parameter, we repeat the same type 1.1 attack with different values of the parameters. We do five runs and compute the average value of the number of false positives and the detection rate (if only one router detect the attack, it's considered as detected). The different results are shown in the figure 6 and only for $\tau$ we have to separate

the graphics due to a too much difference of the scale between the false positives value and the detection rate. The X axis is the value of the parameter.

For the parameter $\tau$, the number of false positives decreases a lot between 0.3 and 0.5, the detection rate decreases a lot after 0.5. That's why the value 0.5 seems to a be a good choice. Of course, these variations are normal because more $\tau$ is low more you consider that a rule is under attack.

Consider now $\omega$ which is used for the first check. So if you have a too high value, the traffic will be considered as invariant and so no detection process will be triggered. That's why for the value of 0.1 the detection rate is 0. Once again, the number of false positives is very low. As you can see you can use a value of 0.01 which provide in this test a detection rate of 100% with not so more false positives than for 0.05.

$\gamma$ helps to select the rules which are responsible of the variation of the traffic. So the conclusions should be the same than previous. Of course, it's normal than false positive and detection rate variate in the same way. But in this case you can see that the detection rate is proportional to $\gamma$ and the number of false positives decrease but not so regularly. This can be explained by the fact that even if there are less rules which are selected for 0.4 than 0.3, the detection process don't provide the same results because the belief combination have not the same input.

Finally as we can see for the high entropy threshold, the detection rate is always the same and the false positive are about the same except for the first time. So we can conclude that this kind of attack implies that the rule have a very big entropy value.

Of course, this study is a preliminary study because we have not tested all the parameters, we only change one parameter and keep others invariant. Moreover we only do the tests with one type of attack. But this study validate our systems by demonstrating that the comportment of detection rate and false positives curves is normal considering the role of the different parameters.

### 5.5 The number of selected rules

To evaluate the efficiency of the solution, we can consider the number of rules which are selected ie for which a router send an information to another because we want to avoid too much information exchange which consume a lot of ressources. So the figure 7 shows the evolution of the average number of selected rules near the time 50 when an attack of type 1.1 is triggered. As we can see, there is a peak for the level 1 routers which represents an increase of the information exchange and which can be due the attack. However, we can see that we have similar values before the attack, so even if there is not a direct relation with the attack, we can show that the number seems to be variable around a value of 3. So 40% of rules are discarded automatically by our solution.

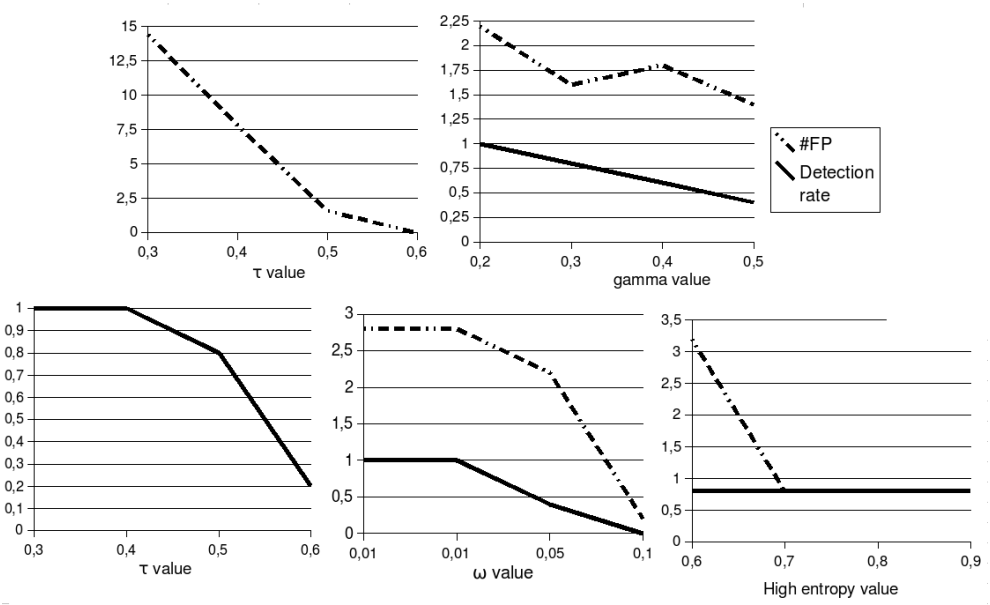


**Fig. 6.** The attack detection rate and the number of false positives (FP) regarding the values of different parameters

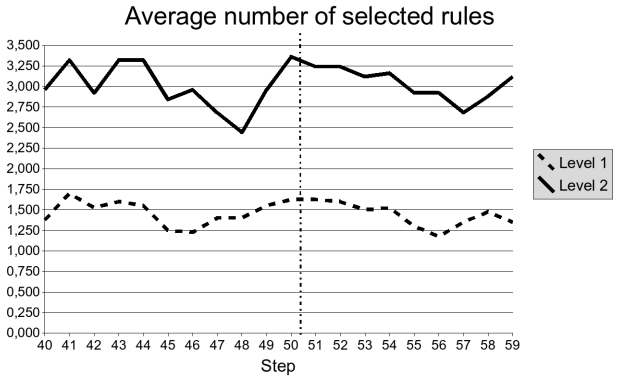


**Fig. 7.** The average number of rules selected at the levels 1 and 2 with an attack 1.1 at time 50

## 6 Related works

The collaboration between IPS or firewalls is a topic for which you can find a lot of information. In [3] the author proposes to use the distributed firewalls to counter efficiently the attacks like DoS attacks. The author highlights the communication security problems between firewalls and proposes a solution. The author collaborates with other to propose an implementation in [4] but in fact only the rules to enforce are exchanged. So each firewall have to detect the attacks alone, the collaboration's goal is to counter the attack at each firewall to avoid traffic congestion. Our work goal is to use the collaboration to detect the attack. The authors of [5] proposes a similar solution where a LAN can request a Gateway to block an attack traffic and each gateway send the request to others and so on, like an epidemic propagation. Moreover these solutions detects the attack after it occurs.

In [6], the author introduce clearly the differences between the traditional forensics and proactive forensic and show that proactivity can be done by analyzing data before the attack. She introduces some examples like for example an user who try to access an unauthorized file before doing malicious activities. The conclusion is that you have to collect data but you have to select it before because you cannot store all.

In [7] and [8], an intelligent firewall is described in order to detect attack proactively. In fact the authors try to determine if an attachment of an email is a virus but by analyzing network traces and so not at a client level. The idea is to use the different characteristics of the email but the problem is that you have to reconstruct the attachment and inspect it. In fact it's filetring content which need a lot of resources if you want to do that for each application. So at an ISP level, it need too much resources and it is too complicated contrary to our framework which use only the frequency which is easy to calculate. In [9], a solution is proposed to accelerate a firewall but thanks to antother component. So this solution can be use but at a ISP client level.

In [10] a peer-to-peer approach is introduced and in [11] a mobile-agent solution is proposed but in these 2 cases, the hosts communicate to exchange the new detected threats. Furthermore, they propose to detect another attack by using what happen on different hosts. The difference with our proposition is that the score is a compressed metric contrary to the previous solutions which need to inspect what a user do on another host or what are the files used for example.

To be brief, our system try to determine an attack using only the frequency of the rules which can be easily obtained contrary to several other solutions which are IDS and gather a lot of various data.

Other authors propose to use simple statistics but the metrics used are not distributed over several IPS or firewalls. In [12], the metric is a packet counter by flow. [13] proposes statistical approaches like the entropy to have more valuable information. The authors in [14] use the conditional legitimate probability to determine the deviation toward a profile.

## 7 Conclusion and future works

In this paper, we propose a new framework to improve the security at a higher level than a lot of other existing solutions. The key point is that the IPS can communicate to each other in order to exchange valuable information but not too much in order to avoid an attack whose the goal would be to overload this system. That's why we propose to select the rules for which an attack is very potential. The collaboration thanks to vertical communication help to do this choice and thanks to the horizontal communication an IPS can check if there is an attack by calculating the packet rates. Thanks to this next step you will counter an attack only if there is a risk for the host and so you should have no false positives.

Finally, we did preliminary experiments to validate our ideas and highlight the difficulties of the future real implementation like the number of parameters. The conclusion of our preliminary study is that the combination of the different information from different IPS is a good way to detect a denial of service attack when it is undetected by only one. The detection rate is significative high when the false positive is low and above all at the second level routers which is teh proof that the communication between IPS improve really the attack detection. Moreover thanks to our solution, we see that the attack is detected before the host. Indeed, the bandwidth is saved and the memory and cpu of the final host too because it don't have to deal with the malicious packets. Finnaly we showed that the selection process of rules to analyse is efficient because it discards a lot of rules.

Of course, our simulation use only 3 rings and a simple normal traffic and that's why we will plan to use other traffic distributions and more rings. As we said before, there are a lot of parameters and we saw that their values may have an important impact on the results. However we have to do more investigation about the right values for all the parameters.

Moreover, it's only the beginning of the FireCollaborator project and a validation would be necessary with real traces and we have to deal with other problems like:

- the rules can be different on each IPS: this problem can be handle by using set theory and in this cas our approach can manage other threats like worm propagation for which a rule about a service can be used. So our proposition is not limited to denial of service attacks.
- security for the IPS communication
- the different behaviors of the routers: no participation, liers...

## References

1. H. Hamed and E. Al-Shaer, "Dynamic rule-ordering optimization for high-speed firewall filtering," in *ASIACCS '06: Proceedings of the 2006 ACM Symposium on Information, computer and communications security.* New York, NY, USA: ACM Press, 2006, pp. 332–342.

2. A. Jsang, S. Pope, J. Diaz, and B. Bouchon-Meunier, "Dempster's rule as seen by little coloured balls," 2005.
3. S. M. Bellovin, "Distributed firewalls," *;login:*, vol. 24, no. Security, November 1999. [Online]. Available: citeseer.ifi.unizh.ch/bellovin99distributed.html
4. S. Ioannidis, A. D. Keromytis, S. M. Bellovin, and J. M. Smith, "Implementing a distributed firewall," in *CCS '00: Proceedings of the 7th ACM conference on Computer and communications security.* New York, NY, USA: ACM Press, 2000, pp. 190–199.
5. R. N. Smith and S. Bhattacharya, "A protocol and simulation for distributed communicating firewalls," in *Computer Software and Applications Conference, 1999. COMPSAC '99. Proceedings*, 1999, pp. 74–79.
6. A. Orebaugh, "Proactive forensics," *Journal of Digital Forensic Practic*, vol. 1, pp. 37–41, 2006.
7. U. Ultes-Nitsche and I. Yoo, "Steps toward and intelligent firewall - a basic model." Proc. Conference on Information Security for South Africa (ISSA2003), July 2003.
8. I. Yoo and U. Ultes-Nitsche, "Adaptive detection of worms/viruses in firewalls." Proc. of International Conference on Communication, Network, and Information Security (CNIS 2003), December 2003.
9. J. Moon, J. Park, G. Jung, P. Choi, Y. Kang, K. Choi, and S. Noh, "Accelerating firewall through intelligent self-learning," in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, 2003, pp. 3524–3529.
10. R. Janakiraman, M. Waldvogel, and Q. Zhang, "Indra: A peer-to-peer approach to network intrusion detection and prevention," in *Proceedings of IEEE WETICE 2003*, June 2003.
11. 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.
12. K. Hwang, S. Tanachaiwiwat, and P. Dave, "Proactive intrusion defense against ddos flooding attacks." nternational Conference on Advances in Internet, Processing, Systems, and Interdisciplinary Research, 2003.
13. L. Feinstein, D. Schnackenberg, R. Balupari, and D. Kindred, "Statistical approaches to ddos attack detection and response." in *DISCEX (1)*, 2003, pp. 303–.
14. 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*, vol. 3, no. 2, pp. 141–155, 2006.