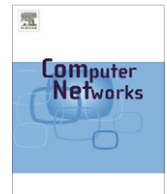




ELSEVIER

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

REFACING: An autonomic approach to network security based on multidimensional trustworthiness

F. Oliviero, L. Peluso, S.P. Romano *

University of Napoli "Federico II", Via Claudio, 21, 80125 Napoli, Italy

ARTICLE INFO

Article history:

Received 28 May 2007

Received in revised form 31 January 2008

Accepted 20 April 2008

Available online 5 June 2008

Responsible Editor: Christos Douligeris

Keywords:

Trustworthiness

Network security

Autonomic communication

Information fusion

Dempster

Shafer theory

ABSTRACT

Several research efforts have recently focused on achieving *distributed* anomaly detection in an effective way. As a result, new *information fusion* algorithms and models have been defined and applied in order to correlate information from multiple intrusion detection sensors distributed inside the network. In this field, an approach which is gaining momentum in the international research community relies on the exploitation of the *Dempster–Shafer* (D–S) theory. Dempster and Shafer have conceived a mathematical theory of evidence based on belief functions and plausible reasoning, which is used to combine separate pieces of information (*evidence*) to compute the probability of an event.

However, the adoption of the D–S theory to improve distributed anomaly detection efficiency generally involves facing some important issues. The most important challenge definitely consists in sorting the uncertainties in the problem into a priori independent items of evidence. We believe that this can be effectively carried out by looking at some of the principles of autonomic computing in a *self-adaptive* fashion, i.e. by introducing support for *self-management*, *self-configuration* and *self-optimization* functionality.

In this paper, we intend to tackle some of the above mentioned issues by proposing the application of the D–S theory to network information fusion. This will be done by proposing a model for a self-management supervising layer exploiting the innovative concept of *multidimensional reputation*, which we have called *REFACING* (*REL*ationship–*F*amiliarity–*C*onfidence–*I*nteGrity).

© 2008 Elsevier B.V. All rights reserved.

1. Introduction

As computer attacks become more and more sophisticated, the need to provide effective intrusion detection methods increases. Current best practices for protecting networks from malicious attacks rely on the deployment of an infrastructure that includes network intrusion detection systems. However, most such practices suffer from several deficiencies, like the inability to detect distributed or coordinated attacks and the high false alarm rates. Indeed, detecting intrusions becomes a hard task in any networked environment, since a network naturally lends itself

to a distributed exploitation of its resources. In such a scenario, the identification of a potential attack requires that information is gathered from many different sources and in many different places, since no *locality principle* (neither *spatial* nor *temporal*) can be fruitfully applied in the most general case.

The classical approaches to distributed protection of a network rely on the effective dissemination of probes and classifiers/analyzers across the infrastructure.

We claim that the current solutions to the above mentioned issues lack two fundamental features, namely *variability* and *trustworthiness*. Indeed, in our view a network should be capable to self-protect against attacks by means of an autonomic approach which highly depends on the effective exploitation, in each node, of on-line information coming both from local analysis of traffic and from

* Corresponding author. Tel.: +39 0817683823; fax: +39 0817683816.
E-mail addresses: folivier@unina.it (F. Oliviero), lorenzo.peluso@unina.it (L. Peluso), spromano@unina.it (S.P. Romano).

synthetic information delivered by neighboring nodes. Self-organization demands for an un-coordinated capability to appropriately orchestrate the behavior of a number of distributed components. Besides this, the second challenge we identify resides in the need for having an agreed-upon means of deciding whether or not information coming from the outside world can be assumed to be reliable.

In this paper, we discuss the main issues related to improving network security through manipulating and combining data coming from multiple sources. To this regard, we start in Section 2 with an analysis of the state of the art in the field of information fusion with a special focus on distributed intrusion detection. In the same section, we devote particular attention to the Dempster and Shafer's approach [1,2], which is quite well-known in the international research community thanks to the so-called *theory of evidence*. In Section 3, we introduce the basic principles on which lays the *autonomic communication* paradigm; this is preparatory to the heart of the paper, whose main contribution comes in the subsequent section. More precisely, we discuss in Section 4 a model for a self-management supervising layer exploiting the innovative concept of *multidimensional reputation*. A thorough performance evaluation of the proposed model is conducted in Section 5. Section 6 proposes a survey of works which have some points in common with our approach, since they exploit the two main features of our solution, namely cooperation and reputation-based information sharing. Conclusions are provided in Section 7.

2. Detection from multiple sources

As soon as one starts spreading detection components across a network, the issue arises to appropriately orchestrate their operation. In fact, information retrieved from a single sensor is usually limited and sometimes provides for low accuracy. The use of multiple sensors definitely represents a valid alternative to infer additional information about the environment in which the sensors operate [3–8]. To this aim, many research efforts have so far been conducted with the goal of defining effective approaches for the combination of information coming from multiple sources [9]. Data fusion deals with the combination of information produced by different sensors, with the final aim of improving both the accuracy of the classification process and the reliability of the decision-making process.

Evidently, any approach relying on information fusion brings in some contrasting points. In fact, if on one hand the data fusion process can highly improve reliability of the detection, on the other hand it also makes a strong hypothesis on the reliability of the information which is subject to the analysis. Stated in different terms, as soon as we start relying on data coming from the outside world, we have to ensure that such data can be considered as reliable as our local information, in order to avoid that the fusion process becomes even worse than it used to be in the absence of cooperation. This adds a further level of complexity to the overall intrusion detection system. The ideal situation foresees the possibility to associate local and for-

eign decisions with a corresponding weight, which actually represents the *current* level of trustworthiness assigned to the corresponding originating source. In the depicted scenario, each decision in a single node would be taken by appropriately measuring a weighted combination of local and foreign data, with the weights which should vary in time as a function of the reliability of all participating nodes all along their past history.

While simple in its formulation, the above ideal scenario definitely looks *ideal*, in the sense that it is not at all easy to dynamically set weights in an ever-changing environment such as a network crossed by a variegated portfolio of potential traffic profiles (with each such profile subject to unpredictable changes in space and time). Hence, the contribution of our research aims to bring some insights specifically suited to tackle the above mentioned issue. To this aim, we propose to exploit the concept of weighted information fusion in a highly dynamic fashion. The key issue we are addressing is that of dynamically changing the values of the weights assigned to information sources in such a way as to let them concretely follow the current level of reliability of the sources themselves. The system we devise can be compared to a dynamic controller which appropriately sets the values of the parameters of a control function in which the variables to be tuned represent the decisions taken at different points of the network.

By summarizing the above considerations, we can easily identify one major challenge, concerning the need to effectively measure the level of trustworthiness to be assigned to both local and foreign decisions.

In the following of this paper we will touch upon the above issue. We will introduce a new model for determining the degree of *fairness* of a node based on a multidimensional framework (REFACING – *Relationship–FAmiliarity–Confidence–INteGrity*) envisaging the thorough analysis of the relations between each pair of interacting nodes.

Before delving into the details of the REFACING model, though, we have to introduce some fundamental background concepts related to, respectively, the Dempster–Shafer Theory of evidence and the autonomic communication paradigm.

2.1. D–S theory

The *Dempster–Shafer* (D–S) theory provides an interesting alternative to traditional bayesian models for the mathematical representation of uncertainty [10–12]. The basic principle which this theory is based upon consists in the fact that no assumption about the probability of the constituents of a generic set of events is needed. From this perspective it can be interpreted as a generalization of the classical probability theory where probabilities are assigned to sets or intervals rather than to mutually exclusive singletons. Moreover, the most innovative and interesting aspect of the D–S theory is the effective rule which allows to combine evidence coming from multiple sources and to model conflicts among them.

There are three important functions in the D–S theory: the *basic probability assignment* function (*bpa*), the *Belief* function, and the *Plausibility* function. The basic probability assignment is a primitive of evidence theory and defines a

mapping of the power set to the interval between 0 and 1. More exactly, the value $m(A)$ of the *bpa* for a given set A expresses the proportion of all relevant and available evidence that supports the claim that a particular element of the considered global set belongs to the set A but to no particular subset of A . Formally, this description of *bpa* can be represented with the following equations:

$$m(\emptyset) = 0, \quad (1)$$

$$\sum_{A \in P(X)} m(A) = 1, \quad (2)$$

where $P(X)$ represents the power set of X , \emptyset is the null set, and A is a set in the power set $A \in P(X)$. From the

$$m : P(X) \rightarrow [0, 1], \quad (3)$$

basic probability assignment, the upper and lower bounds of an interval can be defined. This interval contains the precise probability of a set of interest (in the classical sense) and is bounded by two nonadditive continuous measures called, respectively, *Belief* and *Plausibility*. The lower bound, *Belief*, for a set A is defined as the sum of all the basic probability assignments of the proper subsets B of the set of interest A ($B \subseteq A$). The upper bound, *Plausibility*, is the sum of all the basic probability assignments of the sets B that intersect the set of interest A ($B \cap A \neq \emptyset$). Formally, for all sets A that are elements of the power set ($A \in P(X)$)

$$\text{Belief}(A) = \sum_{B|B \subseteq A} m(B), \quad (4)$$

$$\text{Plausibility}(A) = \sum_{B|B \cap A \neq \emptyset} m(B). \quad (5)$$

The D–S theory provides a rule to combine evidences from independent observers O_1 and O_2 into a single and more informative hint:

$$m_{12}(A) = \frac{\sum_{B \cap C = A} m_1(B)m_2(C)}{\sum_{B \cap C \neq \emptyset} m_1(B)m_2(C)}. \quad (6)$$

The depicted D–S combining rule implies that we equally trust all observers. This assumption normally does not hold in several environments, such as a distributed intrusion detection system, whose sensors span multiple domains [11]. There are several reasons for this. First, information coming from different remote sources is considered less trustworthy than that produced by local sources. Second, multiple sources are located in different places; hence, they may capture different traffic profiles. Third, multiple sources usually perform differently in detecting identical events.

To address the above mentioned issue, many research activities have so far been conducted with the intent of defining efficient approaches and algorithms to address the fact that we cannot trust all sensors equally and that a given observer might have different effectiveness in detecting individual misuse types. The most important outcomes have been obtained by developing an extended D–S theory which considers a “conditioned” view of evidence and proposes a modified combining rule able to take into account the above mentioned weights [11]. In the ci-

ted research work, the authors have proposed the following combining rule:

$$m_{12}(A) = \frac{\sum_{B \cap C = A} [m_1(B)]^{w_1} [m_2(C)]^{w_2}}{\sum_{B \cap C \neq \emptyset} [m_1(B)]^{w_1} [m_2(C)]^{w_2}}. \quad (7)$$

Although this approach is more general than the basic one and can be applied to several kinds of scenarios, it still leaves some important open issues. This is mainly due to the fact that information fusion approaches are mostly focused on the development of effective analytic methods to evaluate combined evidence. However, we have to take into account the actual deployment of these approaches in real-world scenarios in which the inherent variability enforces both evidences and weights to change in order to adapt combining rule’s results to the *situational-context*.

The above consideration has been the main source of inspiration for our recent work, which mainly deals with the practical application of the mentioned theoretical results to the effective management of computer networks. We have soon realized that one major issue arises as soon as one tries to cope with variability. This issue concerns the procedure for adjusting the weights that have to be adopted while evaluating the enhanced D–S combining rule and it can be addressed by equipping a system with self-management functionality.

In summary, we claim that in the above mentioned scenarios (and especially in those related to network security) a broader vision should be considered in order to make the deployment of the information fusion process become more concrete. To this aim, in this paper we propose a novel approach to network security in which information fusion techniques are seen from a wider perspective, related to the *autonomic computing* paradigm, and combined with *self-management* functionality. Hence, in the next section we will briefly introduce some autonomic computing ideas which will help us set the ground to the presentation of our proposal for a self-management approach to network protection. As it will come out from the following of the paper, we are going to propose an interpretation of Information Fusion as a *situational-aware* and *automatically adaptive* decision-making process.

3. Autonomic communications

In the recent years, we have been witnessing many radical changes in thinking computer networks. The on-going convergence of networked infrastructures and services, in fact, has changed the traditional view of the network from the simple wired interconnection of few manually administered homogeneous nodes, to a complex infrastructure encompassing a multitude of different technologies, heterogeneous nodes, and diverse services. This situation has put a challenge for the research community to engineer systems and architectures that will increase the robustness of the current and future internetwork whilst alleviating both management costs and operational complexity. The autonomic communications research community has been formed to respond to this challenge.

From this perspective, *autonomic communication* (AC) represents a new emergent paradigm for today’s

networked cooperation. Many efforts have been devoted to proposing its most appropriate definition and application in different actual scenarios. Based on interdisciplinary grounds, AC tries to tackle the problem by developing architectures and models of networked systems that can manage themselves in a reliable way always fulfilling their service mission. In fact, the essence of autonomic computing systems consists in the *self-management* requirements, the intent of which is to free system administrators from the details of system operation and maintenance and to allow systems managing themselves given high-level objectives.

Independently from networked systems' behaviors and purposes, the following properties should be exhibited by any autonomic computing system in order to fulfill self-management needs [13]:

- *Automatic*: this essentially means being able to self-control its internal functions and operations. As such, an autonomic system must be self-contained and able to operate without any external intervention;
- *Adaptive*: an autonomic system must be able to change its operation. This will allow the system to cope with temporal and spatial changes in its operational context either long term (environment customization/optimization) or short term (exceptional conditions such as faults, attacks);
- *Aware*: an autonomic system must be able to monitor its operational context as well as its internal state in order to be able to assess if its current operation serves its purpose. Awareness will control adaptation of its operational behavior in response to situation or state changes.

The sequence of the above mentioned properties highlights the basic principle of the autonomic computing paradigm. Any autonomic system must have a *sensing capability* in order to enable the overall system to observe its external operational context and to *self-adapt* its behavior to fit any environment changes.

3.1. Applying AC principles to reputation assessment

Autonomic communication systems support dynamic coalitions of users or entities sharing common interests. In this context, self-management approaches become fundamental to enforce “law and order” through distributed an loosely coupled schemes based on democratic rules, therefore avoiding the complexity and rigidity of centralized control at an extreme, and the complete anarchy leading to irrelevant information, malicious or free behavior at the other extreme. Therefore, the need arises to reach the following objectives: (i) to distribute community control into the community itself in order to allow self-management; (ii) to detect, remove and isolate malicious and malfunctioning components; (iii) to identify components that are overloaded or prone to failure or simply have lower capabilities.

As an example of the above considerations, the ROCQ (*Reputability–Opinion–Credibility–Quality*) self-management approach has been proposed in [14]. In their work, the authors present a distributed reputation-based trust

management system that computes the trustworthiness of peers on the basis of transaction-based feedback. It allows the construction and dissemination of a system-wide consensus view of a node's behavior. The following represent some useful metrics to characterize a node's behavior: honesty, quality of the provided service, reliability, level of cooperation, etc.

A first comment can be made about the above characterization metrics. Some of them, like quality of service and reliability, deal with intrinsic capabilities of the community nodes. Some others, like honesty and cooperation level, are instead in strict relation to the overall behavior of a system made of cooperating entities sharing common objectives. With respect to these last parameters, in an ideal world the entities belonging to the same community should frankly “report” to the system information about their own behavior. In reality, this naturally entails the possibility that some node in the community starts lying, or at least behaving in an “egoistic” fashion. Thus, it looks evident that any autonomic system should be equipped with a coherent self-management mechanism based on explicit feedback about a node's behavior provided by the peering entities which such a node interacts with, rather than by the node itself.

According to the previous analysis, the ROCQ model combines the following four parameters: *Reputation*, or a peer's global trust rating; *Opinion*, formed by a peer's first-hand interactions; *Credibility* of a reporting peer; *Quality*, or the confidence a reporting peer puts on the feedback it provides.

Basically, ROCQ represents a model for a management system based on the concept of reputation, whereby each participant shares with the other system entities behavioral feedback information. Such feedback definitely represents a participant's judgement about both himself and the others and is exploited at the management level in order to estimate the system-wide reputation of a generic member of the community.

Although this model seems very intuitive and its experimental results confirm the approach, we believe that the metrics on which it is based result too subjective to be monitored and applied. The ROCQ approach is based on the concept of opinion which consists in an *a priori* feedback formed by a peer's interactions. In some scenarios, such as distributed intrusion detection systems, a generic node cannot take decisions or make considerations about other nodes before any “global” decision is taken at the system level. In fact, in a distributed attack scenario, only after a final, agreed-upon decision has been taken by the entire community, each node can form an opinion about the interactions it carried out with the other nodes. Besides, a generic network element might lie about its opinion related to other nodes, thus making it harder (not to say, unpredictable) the decision-making process.

4. REFACING: dynamically renewing network nodes' reputation

The model we propose to assess the reputation of network components taking part to the distributed detection

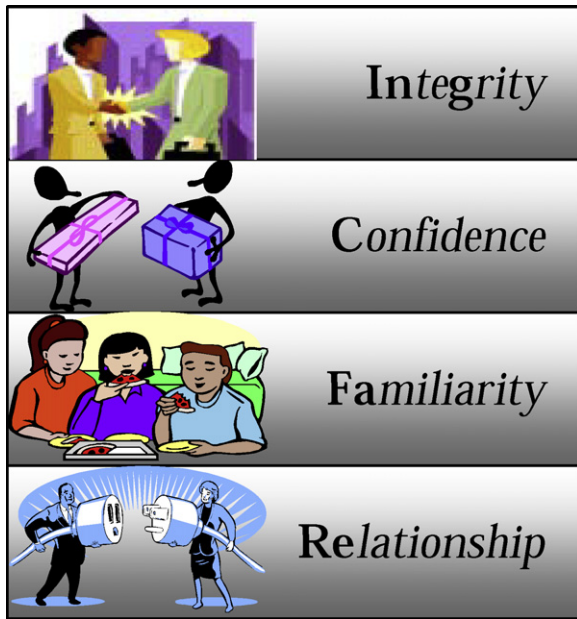


Fig. 1. The REFACING model.

process is called *REFACING* (*REL*ationship–*F*amiliarity–*C*onfidence–*I*nteGrity) and is based on a multi-layered approach, as depicted in Fig. 1.

The lowermost layer provides information about the existence of some form of *connection* among detection components (probes, detection engines, decision engines, etc.). The absence of connection indicates the actual impossibility of carrying out any form of *social relationship* with the other nodes of the network. Otherwise, the second layer in the stack can prove useful to *quantitatively*

measure the level of interaction existing between each pair of network nodes. The more we interact, the more familiar we result with respect to each other. Though, this does not necessarily imply that we trust each other: I can know you quite well, but (or even better, just because of this) I can hardly trust you if our past interactions showed me that you are not that reliable. This is the reason why we introduce the third layer of the trustworthiness stack, which deals with *confidence*. If I have relations with others, and if I am familiar with the others as well, I can much more objectively determine their level of trustworthiness with respect to our social interactions. This said, to further foster the capability of assessing someone else's *fairness* level related to his/her interactions in the network, one more dimension should be taken into account to somehow reflect the *variability* in the behavioral interaction patterns of each node. To make things clearer, the fact that some node has showed a balmeless behavior in one single interaction does not necessarily mean that such node shall be irreproachable also in its subsequent interactions. Some form of estimation of the *line of conduct* over time is definitely needed for all nodes: the more coherent my behavior has been in the past, the less probable it will be that I will behave badly in the near future. This is dealt with at the uppermost layer, which provides information about the level of *integrity* of network nodes.

We do believe that the adoption of such a multi-layered model helps add objectivity to the assessment of network nodes' reputation, since it takes into account a number of complementary, though highly correlated, facets.

In our view, the REFACING methodology is implemented at the level of management of the overall infrastructure, as depicted in Fig. 2. The management layer has a global view of the physical topology of the network and is thus capable to determine whether or not there

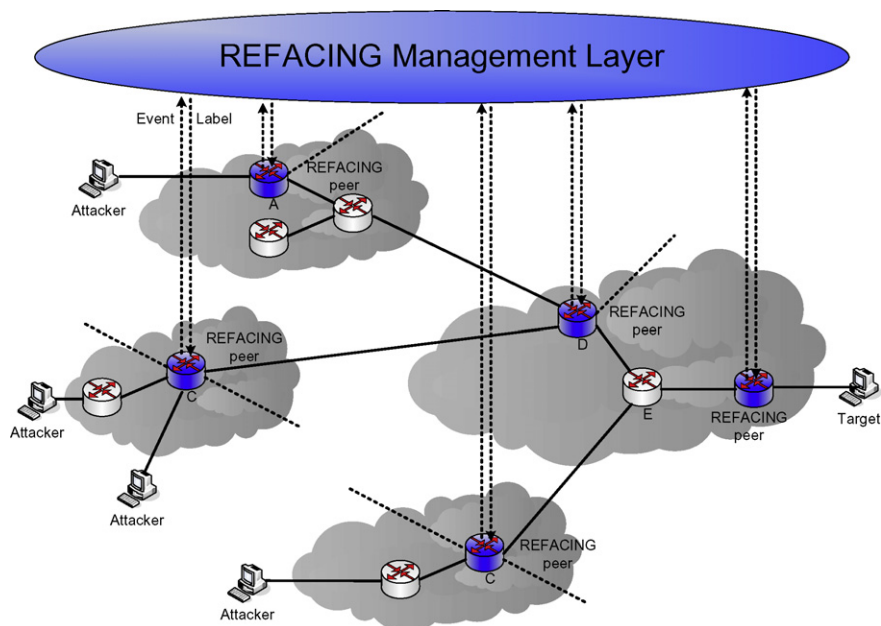


Fig. 2. The REFACING methodology.

exists some form of relationship (layer 1 in the trustworthiness stack) between the network nodes. Furthermore, thanks to monitoring it can also determine the frequency of the interactions among the network elements (layer 2 of the stack). Information pertaining to the third layer can be retrieved through a comparison between each evaluation provided by a single node and the global *opinion* of the system (e.g. my *confidence* level gets higher if my personal evaluation was found in accordance with the final decision taken by the distributed detection system after analyzing all single decisions coming from the network nodes). Finally, data at the fourth layer can be computed by statistically analyzing the information related to all past interactions among all underlying nodes (e.g. my *integrity* level gets higher if my *confidence* level has kept on growing over the past interactions).

After each evaluation turn, the management layer can compute a set of labels (one for each network node involved in the detection process), which are assigned to the nodes through, for example, a policy-based approach. The label computation process can be as general as possible and will normally be influenced by information belonging to all of the layers in the trustworthiness stack (in a simplistic scenario, it might for example be a simple weighted sum of the values computed at each of the four layers). The labels are then used by all nodes whenever they start a new interaction. Each label acts like a *business card* for the node involved in the interaction and can be used by the other nodes in order to assign a weight to the information they have received from their partners.

In order to better highlight the potential application of this novel self-management approach to improve the information fusion process, let us consider the weighted D–S combining rule expressed in (7). In this formula, w_i

is the weight for the generic observer O_i . As above claimed, depending on the situational-context, the management process can evaluate the level of trustworthiness of each node involved in a generic transaction by quantitatively measuring the relationship, familiarity, confidence and integrity metrics. It can then assign each observer its computed *trustworthiness label*. During each transaction, each node has to evaluate the i th weight to be employed in the D–S combining rule. To this aim, right after the label exchanging phase, it can apply a *utility function*, which we call *management function* (MF), to the previously measured values as expressed in the following expression:

$$w_i = \text{MF}(\text{relationship, familiarity, confidence, integrity}). \quad (8)$$

For administrative purposes the *management function* can be appropriately suited to meet the specific domain's high-level goals and/or requirements.

4.1. Implementation of the REFACING model

In this subsection, we describe a practical implementation of the REFACING model, exploiting both information fusion (through the weighted Dempster–Shafer theory) and the “trustworthiness” estimation mentioned above. The informal data flow diagram in Fig. 3 depicts the behavior of the system we realized.

We assume that we have a certain number of REFACING peers which are entitled to express their opinion about a specific situation. As an example, these peers might be intrusion detection systems used to detect potential attacks to a network infrastructure. At the occurrence of specific events (e.g. when a new packet is captured, when a

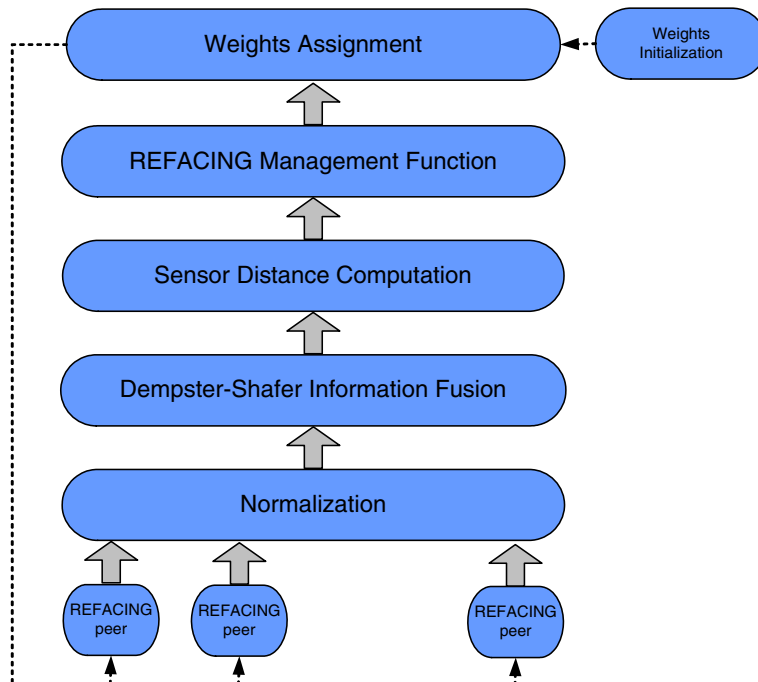


Fig. 3. REFACING system data flow.

timeout for exporting flow information expires, etc.) each such peer will emit his own verdict depending on the inner behavioral rules it is based upon.

Taking the example of a packet-based distributed intrusion detection process, we can assume that each IDS classifies network traffic by associating each packet with a specific “class”. For the sake of simplicity, we herein make the hypothesis that the classes involved are the following: *attack*, *normal*, *attack/normal*. This means that upon reception of a packet, each IDS classifies it depending on its own *bpa*, which is seen as a vector of three possible values, associated, respectively, with the probability that the packet in question is either an attack, or a normal piece of information, or something which the IDS is not able to classify as belonging to either of the above classes.

The output of the different peers is then normalized in order to have a uniform view of the various responses. The normalization process is performed by computing an appropriate *indicator*, X , which in our case assumes the following aspect:

$$X = \frac{m(\text{attack}) - m(\text{normal})}{1 - (m(\text{attack}/\text{normal}))}. \quad (9)$$

As explained in Section 2.1, the value $m(A)$ of the *bpa* for a given set A expresses the proportion of all relevant and available evidence that supports the claim that a particular element of the considered global set belongs to the set A but to no particular subset of A .

The above indicator has been conceived in such a way as to guarantee that the following properties are always respected:

- the sign of the function must depend just on the numerator;
- the variability range of the indicator must be $[-1, +1]$. The closer to 1, the more likely the classified packet belongs to an attack session; similarly, the closer to -1 , the more likely the packet contains normal data.
- The value of the denominator has no influence on the sign of the indicator. Though, the higher the value assigned to the composite hypothesis (i.e. *attack/normal*), the farther the indicator will be from both -1 and $+1$ limit values.

We remark that such indicator proves useful just in case we consider two alternative classification clusters (like the classes *attack* and *normal* in our example, plus the third class *attack/normal* which is just used to deal with uncertainty). In case more classes are present, a different indicator should be considered.

Given the above formulation, we can try to strike the balance between the number of rejected packets (i.e. those packets for which we are unable to issue a verdict) and the misclassification rate, by introducing an ad hoc threshold value τ (see Fig. 4). The classification process will be performed as follows:

- Attacks: all packets for which $X > \tau$.
- Normal: all packets for which $X < -\tau$.
- Rejected: all packets for which $-\tau \leq X \leq \tau$.

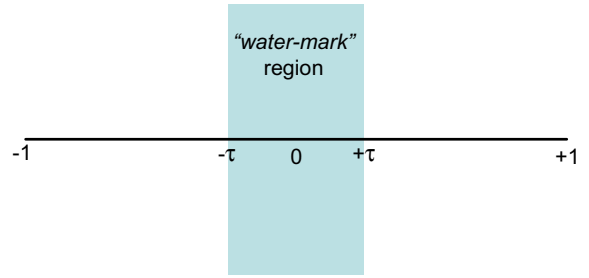


Fig. 4. The water-mark region.

Coming back to Fig. 3, we see that once the normalization process has been performed for all the peers, we can apply the Dempster–Shafer’s weighted combining rule reported in Eq. (7) to obtain the final verdict of the overall system. Such verdict also comes in the form of an indicator, which we will from now on call X_{REFACING} and which is used to converge on a common decision with respect to the presence of a potential attack associated with the analyzed packet. The following blocks in Fig. 3 are all related to the updating of the reputation weights assigned to the REFACING peers. More precisely, for each such peer the distance Δ is computed between the value $X_{\text{BeforeFusion}}$ of its own indicator before the fusion process and the value X_{REFACING} mentioned above:

$$\Delta_{\text{Fusion}} = |X_{\text{BeforeFusion}} - X_{\text{REFACING}}|. \quad (10)$$

The value of Δ_{Fusion} can be considered as a measure of the level of disagreement between the peer’s verdict and the verdict of the overall system. It is possible to determine whether or not a peer’s response was in accordance with the final response of the system by first analyzing the product ($X_{\text{BeforeFusion}} \cdot X_{\text{REFACING}}$) between the peer’s indicator and the system’s one. If such product is negative, there is disagreement in the responses; on the other hand, a positive result means that the specific peer and the system have converged on a common decision. The value of Δ_{Fusion} helps quantify the level of agreement/disagreement between the two entities and it is used in the REFACING management function to update the peer’s reputation weight, as it will be thoroughly explained in the following subsection.

4.1.1. The REFACING management function

For the example above, we have derived the following management function, used to perform the weight updating process.

Let us define:

- W_i : the value of the weight of a generic REFACING peer at the i th iteration;
- RD_i : the degree of correlation of the peer at the i th iteration;
- A_i : the mean value of the peer’s weight at the i th iteration;
- V_i : the variance of the peer’s weight at the i th iteration;
- SR_i : the number of transactions during which the peer has actively contributed (i.e. it has issued his own verdict) to the distributed detection process, at the i th iteration;
- NR : the total number of transactions.

First, we compute the instantaneous value of the weight at the i th transaction, based on the values it assumed in the past:

$$W_i = A_{i-1} + \Delta,$$

where Δ is computed as follows:

$$\Delta = \begin{cases} \text{In case of agreement :} \\ \Gamma * \frac{1-A_{i-1}}{2} * \left(1 - \frac{\Delta_{\text{Fusion}}}{2}\right) \\ \text{In case of disagreement :} \\ -\Gamma * \frac{A_{i-1}}{2} * \left(\frac{\Delta_{\text{Fusion}}}{2}\right) \end{cases} \quad (11)$$

In Eq. (11), the term Γ is computed as follows:

$$\Gamma = \alpha * A_{i-1} + \beta * RD_{i-1} + \chi * V_{i-1},$$

with

$$\alpha + \beta + \chi = 1.$$

In our case, the parameters above have been set by trial and error and assume the following values:

$$\alpha = 0.6, \quad \beta = 0.3, \quad \chi = 0.1.$$

At this point, we can update the old values, which will be used to compute the instantaneous weight at the occurrence of the next transaction:

$$RD_i = 1 - \frac{NR - SR_i}{NR},$$

$$A_i = \frac{SR_{i-1} * A_{i-1} + W_i}{SR_i},$$

$$A_i = \frac{SR_{i-1} * V_{i-1} + (W_i - A_{i-1})^2}{SR_i}.$$

5. Performance evaluation

In this section, we present a performance evaluation of our solution. We show the improvement achieved by our system with regards to previous solutions. The analysis is conducted through an extensive simulation-driven campaign. We developed a simulator, called *RefacingSimulator*,¹ allowing us to test the performance of the solution adopted in a number of different scenarios.

The tests we are going to present in the paper have been specifically conceived to assess the feasibility of our approach. We focus on heterogeneity aspects (i.e. on the applicability of our methodology in a number of different scenarios involving numerous behaviors of the network nodes), rather than on the capability to detect specific kinds of attack. We would like to clarify that our contribution is not related to improving detection of specific attack patterns, but rather to improving the overall detection capability of the framework in the presence of a sort of “weighted” (i.e. reputation-based) cooperation. The specific features of the cooperating nodes are out of the scope of this work. We take for granted in the paper that each single node is equipped with a

particular intrusion detection engine, which might exploit either signature-based or anomaly-based detection techniques. This entails that the simulations do not mimic any specific distributed attack, while implementing multiple “behaviors” of the nodes (*reliable, liar, variable, etc.*).

In the following subsections we first provide a description of the simulator, then we describe some interesting experimental results.

5.1. REFACING simulator

The *RefacingSimulator* is a software tool implemented in Java following the classical Object Oriented design patterns. Its logical structure, depicted in the class diagram of Fig. 5, is composed of the following elements:

- *Decision maker*. It is the main component of the architecture and is in charge both of the configuration of the simulation scenarios and of the appropriate orchestration of all available sensor instances;
- *Attacker*. It represents a simple event (e.g. Attack/Normal) generator, which can be configured through a parameter indicating the event generation probability;
- *Dempster–Shafer* class. It is responsible for the implementation of the D–S detection fusion technique, in both its formulations (i.e. *basic* and *weighted*);
- *Sensor* abstract class. It represents all potential sensors involved in a specific scenario. It just implements a single method, called *UpdateSensorStatistics()*, which realizes the process of updating both sensor weights and statistics, independently of the specific type of sensor instantiated. All implementations of this abstract class represent potential sensor categories, each characterized by its own realization of the *GeneraEvent()* method, which returns the *bpa* associated with the event detected by the sensor. This method requires an input parameter corresponding to the specific event produced by the event generator: depending on its implementation, a certain sensor can either agree or disagree with the input received, thus determining either a successful or a faulty detection.

The sequence diagram in Fig. 6 describes a typical operation scenario of the REFACING simulator. The diagram is self-explanatory and simply shows the decision maker which first instantiates and then orchestrates a certain number of instances of the above mentioned sensors.

5.1.1. The reliable, the liar and the shy

We implemented a number of sensor categories, which have been used for our experimental measurements. These categories are briefly described in the following:

- *Reliable sensor*. It represents the ideal class of sensors, since it provides a response which is always in accordance with the event generated by the *Attacker*;
- *Lying sensor*. It is the worst conceivable sensor, since it always disagrees with the event generated by the attacker, thus providing an error detection rate of 100%;

¹ The REFACING simulator is publicly available as open source software at the following site: <http://refacing.sourceforge.net>.

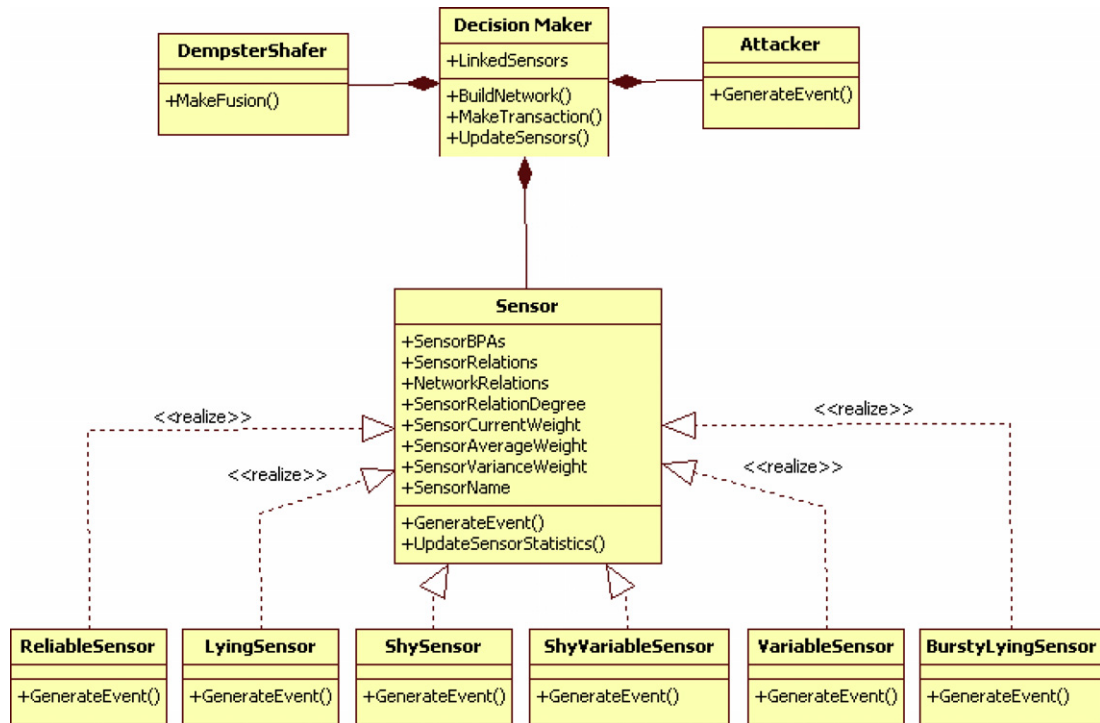


Fig. 5. The REFACING simulator.

- *Shy sensor*. It is shy because it not always participates in the detection process. Though, in case of participation, it provides a response which is always in accordance with the event generated by the *Attacker*;
- *Variable sensor*. It decides randomly, based on a configurable probability value, whether or not to provide a response which is in accordance with the event generated by the *Attacker*;
- *Shy variable sensor*. It implements a sensor showing a hybrid behavior, influenced by both the *shy* and the *variable* paradigms. More precisely, it does not always participate in the detection process, but in case of involvement it randomly decides whether or not to provide a response which is in accordance with the event generated by the *Attacker*;
- *Bursty lying sensor*. It represents a liar, as in the case of the simple *lying sensor*, with the difference that the lies it tells are concentrated in bursts whose duration can be a priori configured.

5.2. Experimental results

For all the scenarios we are going to describe, we performed 1000 transactions, by choosing an event generator (the *Attacker* in the class diagram of Fig. 5) having an attack generation probability of 0.5 (i.e. one out of two generated events represents, on average, an attack). For the sake of simplicity, the following *bpa* function has been adopted for all peers:

- $bpa(Normal) = (0.896, 0.0, 0.103)$;
- $bpa(Attack) = (0.0, 0.896, 0.103)$.

5.3. Scenario 1

Scenario 1 (Table 1) presents the following configuration, composed of a total number of 20 REFACING peers:

- 10 reliable sensors;
- 10 variable sensors, with a detection error probability of 0.3.

Once done with the simulations we observed the following results (Fig. 8):

1. *variable* peers present an average number of false negatives equal to 25% of the total number of transactions. This is in accordance with the detection error probability they had been assigned;
2. by applying the basic D–S formula (i.e. in the absence of REFACING), we observed, on average, a reduction in the number of detection errors equal to 97%;
3. by applying the weighted D–S formula (with the REFACING approach) we observed a further decrease in the number of detection errors, which are in this case completely eliminated.

From a more detailed analysis of the graphs (Fig. 7) we can also draw the following considerations:

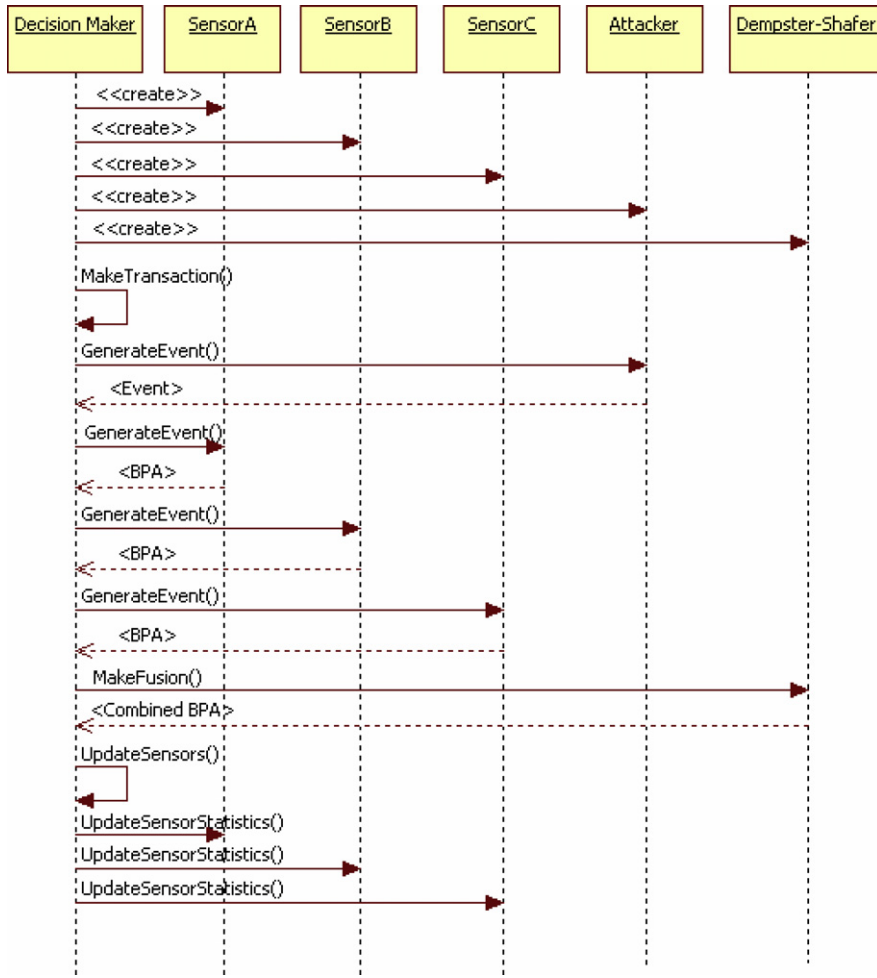


Fig. 6. A sequence diagram showing the operation of the REFACING simulator.

Table 1
Scenario 1

Scenario 1 (1000 transactions, attack probability = 0.5)	
Sensors	Detection errors (false negatives + false positives)
10 reliable sensors	0
10 variable sensors (Pe = 0.3)	250
No REFACING D-S	6
REFACING D-S	0

- the instantaneous value (and thus the average value) of the weight assigned to *reliable* sensors keeps on growing during the simulation. This means that the decisions taken by reliable peers become more and more important (i.e. they have more and more influence on the final verdict) as long as the system evolves. This result is justified by observing the average behavior of such peers, which present the maximum degree of correlation (equal to 1 all along the simulation), a very small variance, and a very high average weight;

- on the other hand, *variable* peers show an ever-decreasing trend with respect to the values of their weights (which are also rather low). This is due to the fact that their detections suffer from both errors and discontinuities;
- weight decreases in case of detection errors are practically the same (in module) as weight increases in case of correct detection. This is justified by the “constant” behavior of all peers in terms of both average values and variance of their weights;
- weight reductions obtained for variable sensors during the first transactions are due to the fact that the detection errors produced by such peers are highly biased around the first few simulation transactions.

5.4. Scenario 2

Scenario 2 (Table 2) presents the following configuration, composed of a total number of 20 REFACING peers:

- 10 reliable sensors;

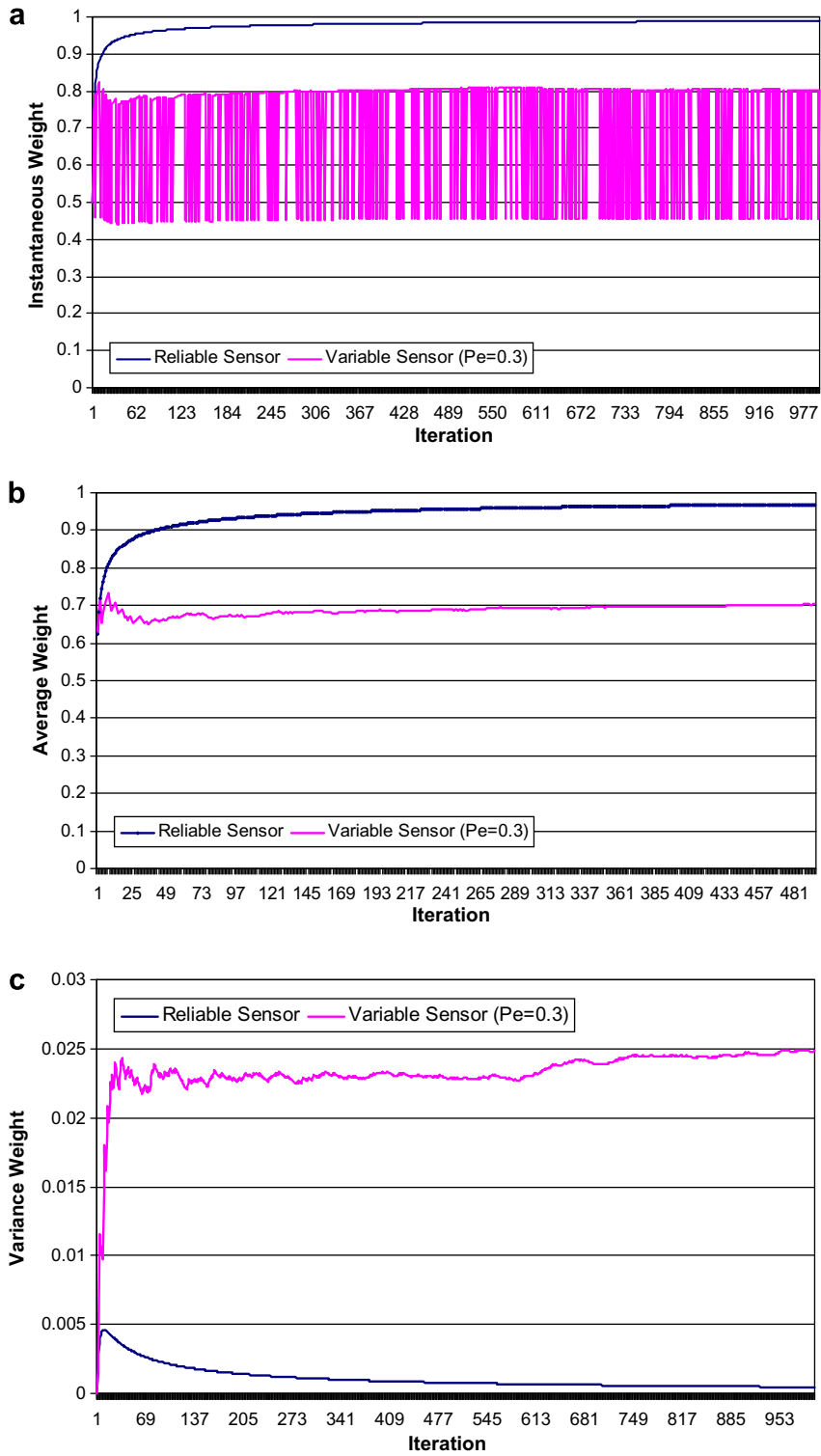


Fig. 7. Scenario 1. (a) Instantaneous weight. (b) Average weight. (c) Weight variance.

- 10 variable sensors, with a detection error probability of 0.7.

Once done with the simulations we observed the following results (Fig. 8):

Table 2
Scenario 2

Scenario 2 (1000 transactions, attack probability = 0.5)	
Sensors	Detection errors (false negatives + false positives)
10 reliable sensors	0
10 variable sensors (Pe = 0.7)	702
No REFACING D-S	29
REFACING D-S	0

1. *variable* peers present an average number of false negatives equal to 70% of the total number of transactions. This is in accordance with the detection error probability they had been assigned;
2. by applying the basic D-S formula (i.e. in the absence of REFACING), we observed, on average, a reduction in the number of detection errors equal to 97%;
3. by applying the weighted D-S formula (with the REFACING approach) we observed a further decrease in the number of detection errors, which are in this case completely eliminated.

From a more detailed analysis of the graphs we can draw almost the same considerations as in the previous scenario. Though, differently than before, we can now observe a greater decrease in the value of the weight assigned to variable sensors. This is clearly justified by the higher degree of uncertainty associated with detection errors. More precisely, instantaneous weight values (and hence average weight values), are much lower than in the previous case. This entails a reduced impact of unreliable peers, and definitely improves the system's detection capability (thanks to the decrease in the number of detection errors).

5.5. Scenario 3

Scenario 3 (Table 3) presents the following configuration, composed of a total number of 29 REFACING peers:

- 2 *reliable* peers;
- 12 *variable* peers, with a detection error probability of 0.3;
- 5 *lying* peers;
- 10 *bursty variable* peers, with a detection error probability of 0.3 and a burst of 50 detection errors all in the first transactions.

Once done with the simulations we observed the following results (Fig. 9):

Table 3
Scenario 3

Scenario 3 (1000 transactions, attack probability = 0.5)	
Sensors	Detection errors (false negatives + false positives)
2 reliable sensors	103
12 variable sensors (Pe = 0.3)	336
5 lying sensors	897
10 bursty variable sensors	134
No REFACING D-S	520
REFACING D-S	103

1. *reliable* peers present a false negative rate equal to 10, 3% of the total transactions;
2. *variable* peers present an average number of false negatives equal to 33, 6% of the total number of transactions. This is in accordance with the detection error probability they had been assigned;
3. *lying* peers present an average number of false negatives equal to 89, 7% of the total number of transactions;
4. *bursty variable* peers present an average number of false negatives equal to 13, 4% of the total number of transactions;
5. by applying the weighted D-S formula (with the REFACING approach) we observed a very sensitive decrease in the number of misdetections: the number of detection errors is, in fact, 75% less than in the case the basic D-S formula is adopted. This means that the application of the REFACING model to the weighted combination procedure allows for an optimization of the overall system's behavior: the number of detection errors (after the information fusion process) for this scenario is equal to the error detection rate of the involved *reliable* peers.

From a more detailed analysis of the graphs we can also draw the following considerations:

- due to the presence of detection errors generated by both *bursty variable* and *lying* peers (which do represent the majority of detection errors), during the first few transactions the values of the instantaneous weights associated with such sensors are higher than those of the other classes of peers. This situation brings to a high number of detection errors before the system reaches its steady state, both in the case of the basic D-S fusion and in the case of the weighted D-S fusion with the REFACING approach. Finally, this also explains why most of the overall system's detection errors belong to this transient phase.
- in the steady state, the instantaneous weight values of both *variable* and *reliable* peers become such that the information fusion process produces the correct response most of the times. This can be observed in the graph showing the values of the instantaneous weights, where it can be easily noticed how the weights of the mentioned classes of peers assume a value which is much higher than the value associated with both *bursty variable* and *lying* peers.

5.6. Scenario 4: not all that glitters is gold

Scenario 4 (Table 4) presents a simple configuration, composed of a total number of just 10 *variable* REFACING peers, with a detection error probability of 0.5.

Once done with the simulations we observed the following results (Fig. 10):

1. *variable* peers present an average number of false negatives equal to 37, 8% of the total number of transactions;

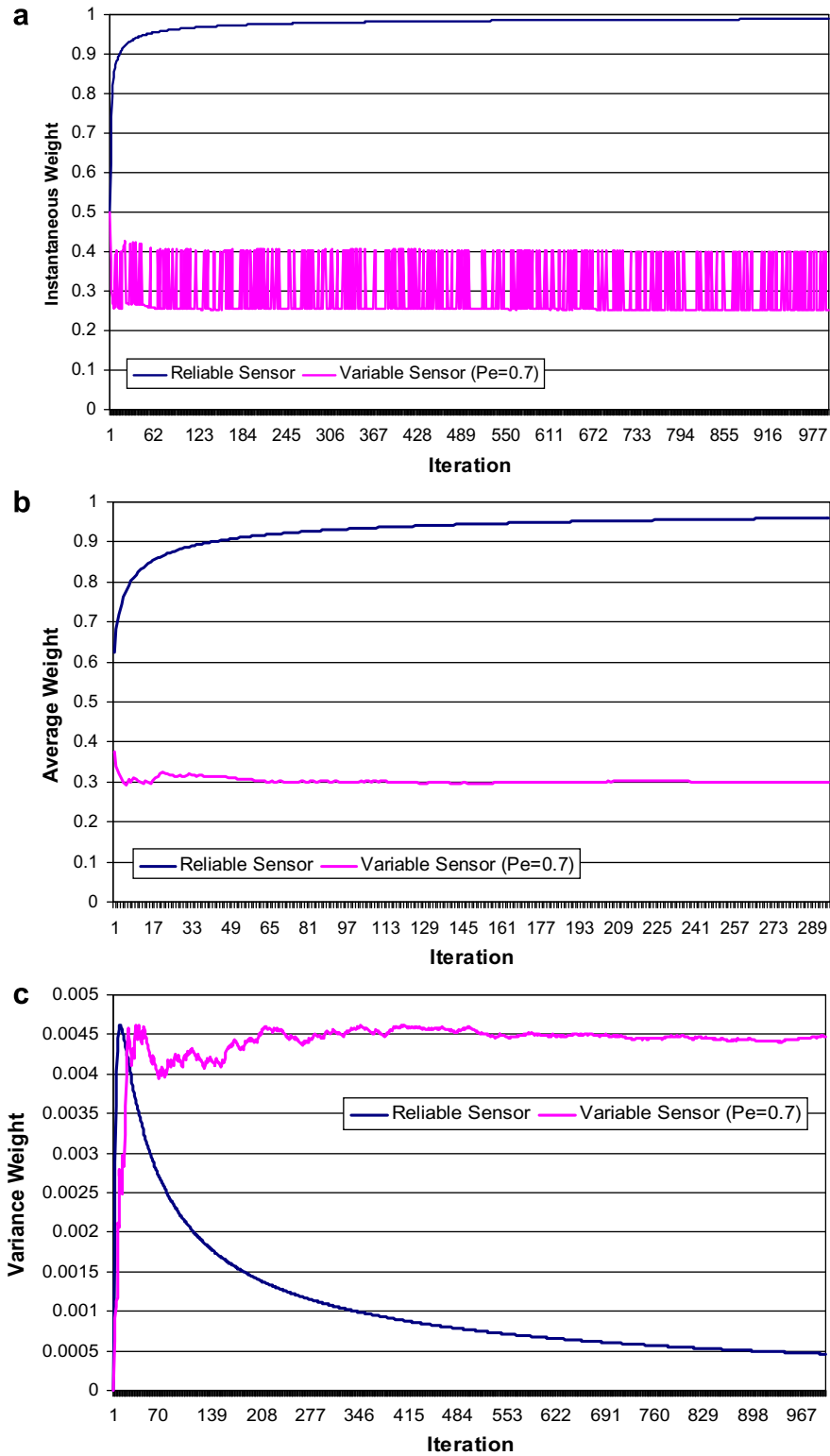


Fig. 8. Scenario 2. (a) Instantaneous weight. (b) Average weight. (c) Weight variance.

2. by applying the weighted D-S formula (with the REFACING approach) we observed an increase in the

number of misdetections with respect to the case when the basic D-S formula is applied.

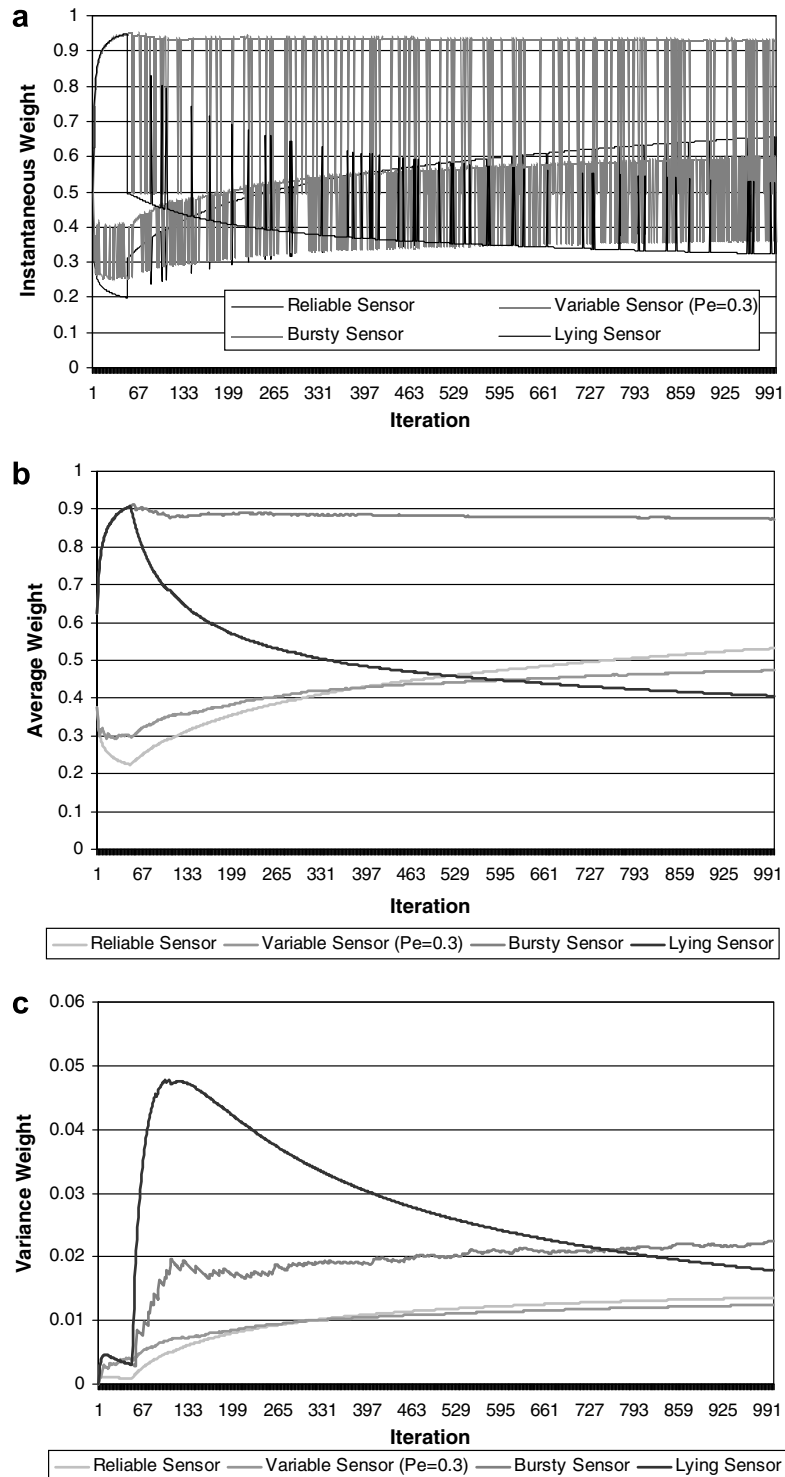


Fig. 9. Scenario 3. (a) Instantaneous weight. (b) Average weight. (c) Weight variance.

From a more detailed analysis of the graphs we can observe that the high false negative rate characterizing the *variable* peers (with no mitigation effect from the reliable

peers as it happened in the previous scenarios) brings to assigning high instantaneous weight values to these sensors. This has the effect of “inverting” in most cases the

Table 4
Scenario 4

Scenario 4 (1000 transactions, attack probability = 0.5)	
Sensors	Detection errors (false negatives + false positives)
10 variable sensors (Pe = 0.5)	378
No REFACING D–S	480
REFACING D–S	508

output of the fusion process, thus increasing the percentage of detection errors with respect to the case when the basic D–S formula is applied.

This scenario clearly represents an example of a potential case in which the very low reliability level of the detection peers causes an appreciable reduction of the otherwise positive effects of the weighted fusion process exploiting the REFACING approach.

5.7. Summary considerations

To summarize the considerations we made, we herein provide in tabular form the main performance figures characterizing each of the aforementioned scenarios.

Furthermore, in Fig. 11 we analyze the performance of the proposed Management Function. The figure shows the percentage of detection errors respectively in the presence and in the absence of the REFACING management function. It can be easily noticed that in all but the last scenario we achieve a considerable performance improvement. The best figures are obtained for both scenarios 1 and 2. Indeed, Fig. 11b highlights a performance improvement of 100% in both such cases. Good results are also attained with scenario 3, with an overall decrease in the error detection rate that is close to 80%. In all cases we observe that the adoption of the REFACING approach guarantees an error rate which is less than 10%. Again, with reference to scenario 4, the trend inverts and shows a performance decrease close to 6%. As already discussed, the reason behind such performance debacle can be ascribed to the presence of a high number of sensors characterized by very poor detection capabilities.

5.7.1. Some remarks about weight assignment strategies

The D–S theory natively provides some means to tackle the issue of appropriately assigning weights to the available sensors. We try to elaborate on this point in the following.

First, we observe that, given a specific event, two different detection engines might be given different trustworthiness levels depending on their intrinsic capabilities. Let's take the following example:

- Events: {Attack, Normal, Attack/Normal}.
- First IDS: Snort (signature-based). In case of an attack (A), we might assign the following bpa: $m(N) = 0; m(A) = 0.9; m(N, A) = 0.1$. This because a signature-based IDS (with a reliable signature database) is good at identifying well-known attack patterns. In the other case (i.e. non-attack event), we might end up with

something like: $m(N) = 0.6; m(A) = 0; m(N, A) = 0.4$. This because there is some chance that a signature-based IDS is not able to identify new attack patterns.

- Second IDS: SVM (Support Vector Machine). In case of an attack (A), we might assign the following bpa: $m(N) = 0; m(A) = 0.5; m(N, A) = 0.5$. This because an SVM has a lower confidence level with respect to attack identification.

In summary, the bpa brings in an intrinsic capability to represent how good an engine is at identifying a specific event.

Second, different events can be discriminated by the same classifier if we introduce a finer granularity level. This entails that the bpa vector can be easily extended in order to account for an increased number of attack classes. Stated in different terms, when compared to the above example, the so-called frame of discernment would now contain more elements: {Attack Class 1, Attack Class 2, ..., Attack Class N, Normal, Attack/Normal}. This enables us to discriminate among a broader number of events.

We observe that the reputation weight comes out from an evaluation of the behavior of a single element when considered as part of a community of elements. As such, the weight should be independent of the specific class of event that has been classified by the node, and should rather indicate how reliable that node has proved to be for a number of different events in a number of different situations. Under this perspective, a single reputation weight definitely represents a feasible solution. We try to better explain such claim with the following example. If an IDS is not good at identifying a specific class of attack, its low reliability (in that specific case) will have already been taken into account in the bpa assigned to it. It is likely that such IDS will provide a response that is not in accordance with the final response of the system when such an event occurs. In this case, the reputation level of the node in question will negatively suffer from the wrong verdict emitted. If the node in question keeps on being found in discordance with the overall verdict of the system, its reputation weight will keep on going down. This is exactly what should happen in a cooperating environment like the one we are proposing. Indeed, this means that the node has been an *unlucky* one, since it is analyzing events for which it has a low detection capability (e.g. a signature-based IDS in a network where most of the attacks are still unknown).

6. Related work

In this section, we present a brief survey of existing methodologies and architectures for cooperative network security. All of the mentioned works have some points in common with our approach, since they all exploit in some way the two main features of our solution, namely cooperation and reputation-based information sharing. We also share with the authors of the cited works the idea of defining architectures that are independent of the specific intrusion detection mechanism adopted. By exploiting existing solutions for local attack detection, we can then try to

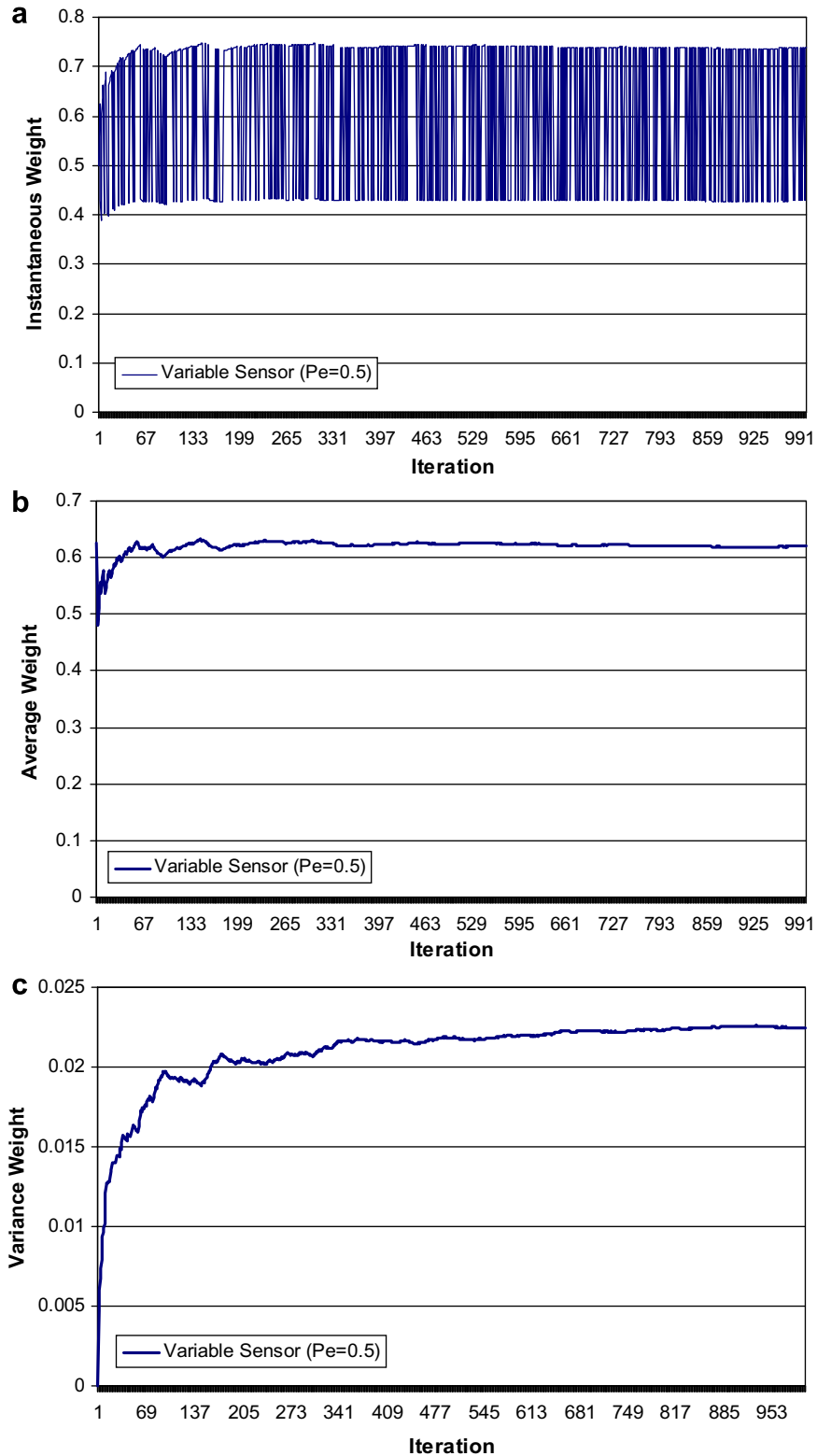


Fig. 10. Scenario 4. (a) Instantaneous weight. (b) Average weight. (c) Weight variance.

increase the effectiveness of the overall process based on the assumption that sharing evidences of attacks might

provide a clearer vision of the ongoing situation. This can improve the capability of correct detection, through a

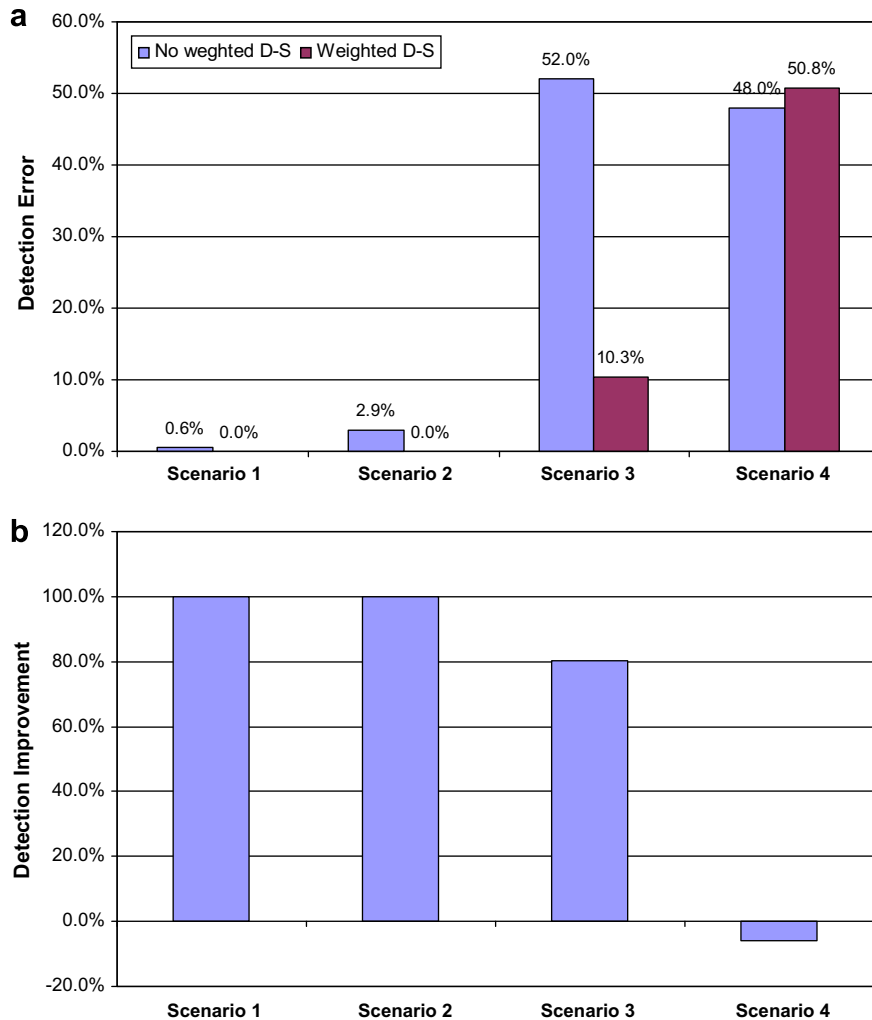


Fig. 11. REFACTING evaluation. (a) Management function evaluation. (b) REFACTING performance improvement.

decrease of both false positives and false negatives, thus bringing to a prompt response.

New distributed solutions for intrusion detection have been recently proposed, supported also by the consideration that by correlating alerts it is possible to reduce the false alarm rate [15].

The idea of appropriately combining information coming from heterogeneous sources can be found in an interesting work from Zhang et al. [16], who present an architecture composed essentially of two stages. At the first stage each system node detects the attack autonomously by exploiting a variety of existing IDS solutions. The victim is protected by adopting a local rate limit on traffic directed to it from a suspicious node, according to a local policy. Then, in the second stage each node adjusts dynamically its rate limit according to the information shared with other nodes in the infrastructure. The information exchange exploits gossip communication mechanisms. By adopting a set of metrics M_i each node locally profiles the traffic monitored. Based on these metrics the single element assigns to suspicious traffic a confidence

degree *conf*, weighting each metric with its reliability in terms of false positives or negatives. Thanks to such *conf* value, the node limits suspicious traffic. Moreover, by using gossip communication mechanisms, it provides its neighbor peers in the overlay network with information about the suspicious traffic profile, the metrics values related to it, and its confidence *conf*. Gossip communication, in particular, provides a “light” and reliable mechanism for sharing information; it exploits solutions based on well-known epidemic theory algorithms. The data exchange about DDoS attack evidence allows a more accurate process of traffic limiting by adopting algorithms that aggregate information.

Coming to the exploitation of reputation-based information, CONFIDANT (Cooperation Of Nodes: Fairness In Dynamic Ad-hoc NeTworks) [17] represents an interesting approach. It is based on misbehavior detection and subsequent reaction. More precisely, upon detection of a node’s malicious behavior, the system responds by blocking the forwarding process of packets coming from that node. A sort of *re-integration* possibility, after an “expiation”

period, is offered to misbehaving nodes which return to work correctly. Cooperation is the main contribution of the CONFIDANT protocol.

It provides two mechanisms to detect malicious nodes: (i) learning correctness of neighbor nodes behaviors from their direct observation; (ii) sharing information about malicious nodes with other components of the network. These two mechanisms allow the network to isolate nodes that don't have an "exemplary conduct". In CONFIDANT, the Reputation System is a distributed component responsible for management of nodes reputation. It is provided with a table where all the reputation ratings are stored. Anytime we collect enough evidences of a node's misbehavior, the Reputation System modifies the rating for that node. The update is computed by merging the different evidences of a misbehavior. Different weights are assigned to, respectively, direct experience and other nodes observations.

CORE (Collaborative REputation) [18] is another solution for improving routing security in wireless scenarios through a distributed reputation model. CORE defines two different kinds of reputation: (i) Subjective Reputation; (ii) Indirect Reputation. The former is the reputation observed locally by a node with regard to other nodes. By monitoring temporal evolution of a node's behavior, the subjective reputation is computed by giving more relevance to past observations than to recent ones. The Indirect Reputation is reputation provided to a node from other nodes. It does not come from direct observations but rather from observations of other entities. Subjective Reputation and Indirect Reputation are merged by means of a weighted combining formula in order to compute a final value of reputation.

Finally, with respect to the issue of merging information coming from multiple sources, the literature proposes numerous inspiring works.

As an example, in [11] the authors propose an approach that performs alert confidence fusion based on the weighted D–S theory. As already mentioned in this paper, this extended theory first determines weights based on the sources of gathered observations and then combines individual confidence scores using such weights. The weighting process comprises features such as the level of trust in specific observers, and the capacity of specific observers to make particular observations. For example, alerts from remote sites are not considered to be as trustworthy as alerts from local sites. As to the weights calibration logic, they experiment both the *Maximum Entropy* and the *Minimum Mean Square Error* (MMSE) based approaches. Although the proposed solution clearly shares some commonalities with the REFACING approach, the algorithm which governs the weights calibration process is based on a centralized logic which does not take into account levels of trust evaluated from distributed and cooperative nodes.

7. Conclusions and future work

In this paper, we presented a novel approach to distributed detection of network threats. The core of our contri-

bution resides in having designed a self-management layer exploiting the concept of trustworthiness in order to make the detection process more reliable.

The idea of dynamically tuning the currently estimated level of trust of each peer in the community proves fundamental during the information fusion process, which in our architecture is based on the application of an enhanced version of the well-known Dempster and Shafer's theory of evidence. Such enhanced version of the D–S formula proposes to appropriately weigh the various inputs to the information fusion process on the basis of their estimated impact on the final merged information.

This is dynamically carried out by looking at some of the principles of autonomic computing in a *self-adaptive* fashion.

The paper clearly shows, through extensive measurements based on simulation, that our solution helps dramatically improve the overall performance of the detection process in a number of real-world operational scenarios. On the other hand, it also helps set the limits of our approach when applied to situations envisaging the presence of a high number of unreliable sensors whose responses can negatively bias the output of the information fusion process towards a faulty decision.

As a final remark, we observe that the main goal of the presented work consisted in the introduction of a novel framework. We did not focus in this specific paper on the interesting aspects related to noisy training data sets. Though, the architecture has been conceived at the outset with such issues in mind. To the purpose, we envision the exploitation of some of the most well-known techniques available in the literature [19,20], related to the proper (either statistical or deterministic) filtering of the training data.

Acknowledgements

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007–2013) under Grant agreement no. 216585 (INTERSECTION Project). It has also been funded by the Italian "Ministero dell'Istruzione, dell'Università e della Ricerca" (MIUR), in the framework of the COSMIC project.

References

- [1] Jean Gordon, E. Edward, H. Shortliffe, Rule-Based Expert Systems, Chapter The Dempster–Shafer Theory of Evidence.
- [2] Glenn Shafer, A Mathematical Theory of Evidence, Princeton University Press, 1976.
- [3] Nong Ye, Mingning Xu, Information fusion for intrusion detection, in: Third International Conference on Information Fusion, vol. 2, July 2000, pp. 17–20.
- [4] F. Cuppens, A. Mieke, Alert correlation in a cooperative intrusion detection framework, in: Proceedings of the IEEE Symposium on Security and Privacy, 2002, pp. 202–215.
- [5] Julia Allen, Alan Christie, William Fithen, John McHugh, Jed Pickel, Ed Stoner, State of the practice of intrusion detection technologies, 1999.
- [6] James P. Anderson, Computer security threat monitoring and surveillance, Fort Washington, Pennsylvania, April 1980.
- [7] Daniel J. Burroughs, Linda F. Wilson, George V. Cybenko, Analysis of distributed intrusion detection systems using bayesian methods, in:

International Performance Computing and Communication Conference, April 2002.

- [8] Vincent Berk, Robert Gray, George Bakos, Using sensor networks and data fusion for early detection of active worms, in: SPIE Aerosense Conference, vol. 5071, April 2003, pp. 92–104.
- [9] T. Bass, Intrusion detection systems and multisensor data fusion, Communications of the ACM, 43, ACM Press, NY, USA, 2000. pp. 99–105, April.
- [10] Ian Ruthven, Mounia Lalmas, Using Dempster–Shafer's theory of evidence to combine aspects of information use, J. Int. Inf. Syst. 19 (3) (2002) 267–301.
- [11] D. Yu, D. Frincke, Alert confidence fusion in intrusion detection systems with extended Dempster–Shafer theory, in: Proceedings of the 43rd annual southeast regional conference, ACM Press, New York, NY, USA, March 2005, pp. 142–147.
- [12] H. Wu, M. Siegel, R. Stiefelwagen, J. Yang, Sensor fusion using Dempster–Shafer theory, in: Proceedings of IEEE Instrumentation and Measurement Technology Conference, Anchorage, AK, USA, 2002.
- [13] Jeffrey O. Kephart, David M. Chess, The vision of autonomic computing, Computer 36 (1) (2003) 41–50.
- [14] A. Garg, R. Battiti, R. Cascella, Reputation Management: Experiments on the Robustness of ROCQ, Technical Report DIT-05-087.
- [15] F. Valeur, G. Vigna, C. Kruegel, R. Kemmerer, A comprehensive approach to intrusion detection alert correlation, IEEE Trans Dependable Secure Comput 1 (3) (2004) 146–169.
- [16] Guangsen Zhang, Manish Parashar, Cooperative defence against ddos attacks, J Res Pract Inf Technol 38 (1) (2006) 69–84.
- [17] Sonja Buchegger, Jean-Yves, Le Boudec, Performance analysis of the confidant protocol (cooperation of nodes: fairness in dynamic ad-hoc networks), in: Proceedings of IEEE/ACM Symposium on Mobile Ad Hoc Networking and Computing, MobiHOC, June 2002.
- [18] Pietro Michiardi, Refik Molva, Core: a collaborative reputation mechanism to enforce node cooperation in mobile ad-hoc networks, in: Proceedings of Communications and Multimedia Security Conference, CMS, September 2002.
- [19] Sofie Verbaeten, Anneleen Van Assche, Ensemble methods for noise elimination in classification problems, Lecture Notes in Computer Science, vol. 2709, Springer-Verlag, 2003.
- [20] X. Zeng, T. Martinez. A noise filtering method using neural networks, in: Proceedings of the IEEE International Workshop on Soft Computing Techniques in Instrumentation, Measurement and Related Applications, May 2003, pp. 26–31.



Francesco Oliviero received his MS in Telecommunications Engineering and Ph.D. in Computer Engineering from Federico II University of Napoli, Italy, in 2004 and 2007, respectively. Currently, he is postdoc at the Department of Computer Engineering and Systems at Federico II University of Napoli. He is member of the COMICS (COMputers for Interaction and CommunicationS) research group led by Prof. Giorgio Ventre. His research interests are in the areas of network security systems and routing protocols for wireless

mesh networks. Francesco Oliviero is member of IEEE Computer Society.



Lorenzo Peluso is Ph.D. student in Networking at the Department of Computer Science of the University of Napoli Federico II. He received the M.S. degree in telecommunication engineering from the University of Napoli Federico II in 2001. From 2006 to 2008 he worked at FOKUS Fraunhofer Institute, Berlin, Germany, in the research group on Autonomic Communication technologies (NET). He was also involved in the ANA FP6 European project. His current research interests include the design of new communication paradigms

inspired by Autonomic Networking, as well as infrastructures for distributed network monitoring and security.



Simon Pietro Romano received the degree in Computer Engineering from the University of Napoli "Federico II", Italy, in 1998. He obtained a Ph.D. degree in Computer Networks in 2001. He is currently an Assistant Professor at the Computer Science Department of the University of Napoli. His research interests primarily fall in the field of networking, with special regard to QoS-enabled multimedia applications, network security and autonomic network management. He is currently involved in a number of research

projects, whose main objective is the design and implementation of effective solutions for the provisioning of services with quality assurance over Premium IP networks. Simon Pietro Romano is member of both the IEEE Computer Society and the ACM.