RECLAMO: Virtual and Collaborative Honeynets based on Trust Management and Autonomous Systems applied to Intrusion Management

Manuel Gil Pérez*, Verónica Mateos Lanchas†, David Fernández Cambronero†,
Gregorio Martínez Pérez*, and Víctor A. Villagrá†

* *Departamento de Ingeniería de la Información y las Comunicaciones,*
University of Murcia, 30071 Murcia, Spain
Email: mgilperez@um.es, gregorio@um.es

† *Departamento de Ingeniería Telemática,*
Universidad Politécnica de Madrid, 28040 Madrid, Spain
Email: vmateos@dit.upm.es, david@dit.upm.es, villagra@dit.upm.es

***Abstract*—Security intrusions in large systems is a problem due to its lack of scalability with the current IDS-based approaches. This paper describes the RECLAMO project, where an architecture for an *Automated Intrusion Response System* (AIRS) is being proposed. This system will infer the most appropriate response for a given attack, taking into account the attack type, context information, and the trust and reputation of the reporting IDSs. RECLAMO is proposing a novel approach: diverting the attack to a specific honeynet that has been dynamically built based on the attack information. Among all components forming the RECLAMO's architecture, this paper is mainly focused on defining a trust and reputation management model, essential to recognize if IDSs are exposing an honest behavior in order to accept their alerts as true. Experimental results confirm that our model helps to encourage or discourage the launch of the automatic reaction process.**

***Keywords*-autonomous systems; reaction networks; trust and reputation; collaborative systems; virtualization**

I. Introduction

The Internet plays nowadays an important role in our daily life, but its increasing usage has also arisen many potential risks. To protect the availability, authenticity, confidentiality, integrity, and reliability of the network components, many researchers are working in the field of network security.

In this area, access control includes a variety of security technologies, from authentication and identity management technologies for controlling who can access the provided services to firewall technologies for filtering traffic from/to the organization network. In addition to firewalls, there are a number of components that help in detecting and mitigating remote attacks, like *Intrusion Detection Systems* (IDS) that are capable of monitoring security parameters in order to detect malicious or unexpected behaviors.

IDS technologies have rapidly evolved in recent years and now there exist very mature tools with a high extent of reliability in the intrusion detection area. But IDSs are mainly passive components and their effectiveness is not enough for complex attacks; the common response of IDSs is passive, such as the notification to other components.

As the number of security incidents increases, becoming more sophisticated and widespread [1], there is a strong need of automating the detection and reaction processes to face them. *Automated Intrusion Response Systems* (AIRS) provide the best possible defense, as well as shortening or eliminating the delay before administrators come into play.

AIRSs are security technologies whose main goal is to choose and trigger automated responses against intrusions detected by IDSs, in order to mitigate them or reduce their impact. Unfortunately, the state of the art in AIRSs is not as mature as with IDSs. Reactions against intrusions are not optimal, and IDSs have difficulty detecting intrusions in real time and triggering automated responses.

In this context, we describe in this paper the approach taken by the RECLAMO project (*Virtual and Collaborative Honeynets based on Trust Management and Autonomous Systems applied to Intrusion Management*), an ongoing R&D project funded by the Spanish Ministry of Science and Innovation. This project proposes an autonomous response system able to infer the most appropriate response for a given intrusion, taking into account not just the intrusion, but also other parameters related to it such as the context or the trust and reputation of the network source.

Figure 1 depicts the main functional blocks of the RECLAMO proposal. Some of them are briefly addressed throughout the paper, although others, like the trust and reputation management system, are thoroughly presented as one of the first results in the ongoing RECLAMO project.

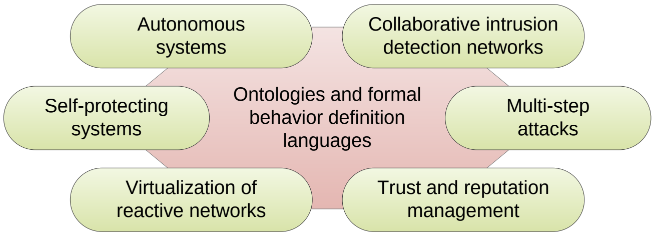
Autonomous systems
Collaborative intrusion detection networks
Self-protecting systems
Ontologies and formal behavior definition languages
Multi-step attacks
Virtualization of reactive networks
Trust and reputation management

Figure 1. Main functional blocks proposed in the RECLAMO project

One of the most promising approaches in the project proposal is based on the inclusion of the so-called *deception responses*; as a response to some type of intrusion, the attack might be diverted to a specific honeynet in order to confine and analyze it. RECLAMO addresses how to dynamically define, generate, and deploy a honeynet built ad-hoc for the attacker preferences. These honeynets will be implemented on top of a network virtualization platform.

Nonetheless, the decision of building a specific honeynet in response to a security threat is subjected to the information gathered from the monitoring system. The generation of bogus attack information, usually coming from malicious IDSs with a dishonest or misbehaving attitude, can lead the AIRS to react mistakenly. It can create and deploy an unnecessary honeynet, misusing system resources, as a consequence of a failed detection because of making use of untruthful information. Hence, bogus alerts and attacks must be discarded to avoid errors in detection processes.

As a promising solution, this paper presents the design of a trust and reputation model, as well as giving a global vision of the proposed autonomous response system, that assesses the behavior of IDSs by taking all security information (alerts and attacks) generated by them in the past.

The remainder of this paper is organized as follows. Section II outlines the current state of the art in AIRSs and their key features. The collaborative model designed for intrusion management in multi-domain environments is defined in Section III, while Section IV presents the designed trust and reputation model. Section V tackles the diverse responses that the system may trigger, with a special emphasis on the deception responses based on the dynamic honeynets generated on virtualized platforms. Section VI presents the RECLAMO project where all the above concepts are applied to achieve a novel automated response system to attacks. Section VII presents some experimental results to demonstrate the benefits of using the designed trust and reputation model. Section VIII discusses the main related work and finally, Section IX summarizes our contributions.

II. AUTOMATED INTRUSION RESPONSE SYSTEMS

Dr. Natalia Stakhanova et al. presented in [2] a taxonomy of autonomous intrusion response systems, together with a review of current trends in intrusion response research.

According to such a paper, AIRSs can be classified in different ways according to various characteristics:

- By ability to adjust: *Static* and *adaptive*. Adaptability is a powerful feature that can automatically modify the chosen response according to other external factors, like the previous response effectiveness.
- By response selection mechanism: *Static mapping*, *dynamic mapping*, and *cost-sensitive mapping*. There is an increasing interest in developing cost-sensitive models for response selection recently. The primary goal of such models is to ensure an adequate response without sacrificing the normal system functionality.
- By time of response: *Proactive* and *delayed*. Proactivity is the ability of the AIRS to react against an intrusion or attack before it takes place.
- By cooperation capabilities: *Autonomous* and *cooperative*. Network-based IRSs are often built in a cooperative fashion, because they provide more effective responses than autonomous systems.

To achieve an optimal response in the shortest time, it is required that AIRSs are adaptive, cost-sensitive mapping, proactive, and cooperative. But there is another feature (*semantic coherence*) that is not present in this taxonomy and is crucial in heterogeneous detection networks.

Semantic coherence is the system ability to understand the syntax and semantic of the intrusion report, with independence of the intrusion source. The intrusion response system would understand intrusion alerts or events with diverse syntaxes from different IDSs, and it would be able to determine whether two alerts refer to the same or different intrusions. This would improve system performance and efficiency, because the system only performs an action to respond to different alerts concerning the same intrusion.

In addition to the taxonomy presented in [2], several AIRSs have been proposed in recent years such as

- AAIRS (*Adaptive Agent-based Intrusion Response System*), a methodology for adaptive and automated intrusion response using software agents [3];
- ADEPTS (*Adaptive Intrusion Tolerant Systems*) that models intrusions by using attack graphs to identify possible attack targets, and provides methods to automatically trigger a suitable response [4];
- EMERALD (*Event Monitoring Enabling Responses to Anomalous Live Disturbances*) [5] and CSM (*Cooperating Security Managers*) [6], host-based and distributed intrusion detection and response systems;
- SARA (*Survivable Autonomic Response Architecture*), a system developed to provide an effective defense against fast and distributed information attacks by using coordinated autonomic responses [7];
- and MAIRF (*Mobile Agents-based Intrusion Response Frame*), a system based on mobile agents that is focused on the source of attackers [8].

The functionalities of these systems, according to the previously mentioned features, are mapped in Table I.

None of these AIRSs offers ways to achieve *semantic coherence* among diverse signs of the same incident. For example, ADEPTS relies on specific formats and syntaxes of intrusion notifications. Instead, the autonomous system proposed by RECLAMO sorts out semantic coherence by using ontologies, formal behavior specification languages, and reasoning mechanisms as a working technology, as well as fulfilling the rest of the requirements.

	AAIRS	ADEPTS	EMERALD	CSM	SARA	MAIRF
Adaptive	√	√				√
Cost-sensitive	√	√	√			
Proactive		√		√		√
Cooperative			√		√	
Semantic coherence						

Table I
FUNCTIONALITIES OF THE AIRSS

One of the main advantages in using ontologies is the formalization of the information semantics. This is important when dealing with heterogeneous information sources that can represent the same resource with different formats and syntaxes. Within the scope of this work, using ontologies helps to support inclusion of different and heterogeneous IDSs, with different intrusion formats and syntaxes, and different sources of network and system context.

There are several ontology languages such as KIF (*Knowledge Interchange Format*), SHOE (*Simple HTML Ontology Extensions*), RDF (*Resource Description Framework*), OIL (*Ontology Inference Layer*), OWL (*Web Ontology Language*), and OWL2. The latter is the main ontology language used nowadays in Semantic Web in order to formally describe information definitions [9]. OWL2 is a knowledge definition language that structures the information into classes and properties –nominal or relation among objects–, with hierarchies, and range and domain restrictions.

OWL2 introduces some improvements in the knowledge representation aspect, but its competence to define behavior related to the defined information is limited. So it is necessary to use additional rule languages; for example, KAoS (*Knowledge-able Agent-oriented System*), Rei, SWRL (*Semantic Web Rule Language*) or OWL-Services. Among them, SWRL is the most widely used language of rules in Semantic Web, which includes a type of axiom, called Horn clause logic, of the form *if... then...*, to specify the behavior of the system [10].

III. COLLABORATIVE INTRUSION DETECTION NETWORKS

Attackers have changed the execution of their malicious practices towards a new mode of operation more global and distributed. Their main goal is that their attacks go unnoticed by exploiting current drawbacks inherent to existing IDSs.

Because of their widespread nature in execution, current information and communication systems are being attacked, and thence compromised, by more sophisticated threats. This drawback is innate to existing IDSs, where the alerts generated by them are viewed as isolated incidents with no relevance when they are analyzed individually [11].

As a solution, alerts generated by IDSs have to be treated as a whole from a more global point of view in order to know what is happening in the entire network. A much more global perception about the system will provide a better opportunity of detecting complex threats, such as distributed intrusions or attacks. Hence, the deployment of multiple IDS instances among all security domains can achieve a better detection coverage and accuracy in multi-domain environments.

The security information detected by IDSs –alerts and attacks– is gathered from all the security domains belonging to the AIRS that is being monitored. In addition, further information from other administrative domains can also be incorporated in the detection processes in order to reach a better knowledge base in detecting distributed attacks.

The union of all IDSs entails to have a close collaboration among them when they are placed in more than one security or administrate domain, this being a critical factor in the success of detecting distributed attacks [12]. Thus, relationships among all IDSs form an overlay layer for exchanging security information among peers; mainly, alerts, incidents, and attacks detected by each IDS in an individual fashion. This cooperative system is named *Collaborative Intrusion Detection Network* (CIDN) that allows building a collective knowledge base of isolated alerts [13].

Placement of IDSs is a key issue for the proper exchange of information among them. An in-depth survey about how to distribute IDSs to obtain the best results in performance (centralized, hierarchical, and fully distributed architectures) is provided in [11]. Instead, we propose in this paper to follow a partially-decentralized approach so as to tackle the drawbacks implicit in each [14]. Partially-decentralized schemes address the problems of having a single point of failure and the lack of scalability, derived from centralized approaches, and the overhead and management difficulty, derived from hierarchical and decentralized schemes [15].

Figure 2 illustrates an example of the system architecture that we propose in this paper, which is based on a partially-decentralized scheme.

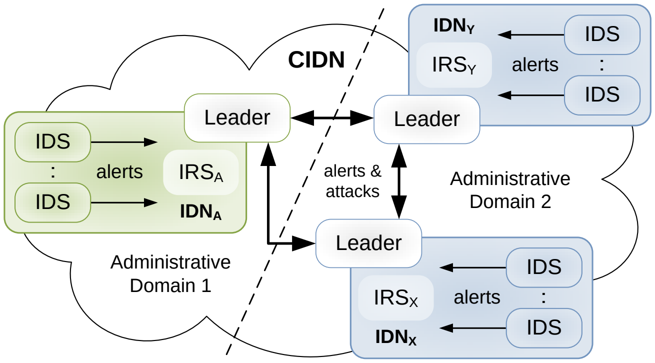

Figure 2. Partially-decentralized scheme of a CIDN

The partially-decentralized scheme proposed in this work is built by using a *supernode* (or *superpeer*) that acts as the head or leader of its security domain [16]. This is in charge of sharing the collective (*intra-domain*) knowledge base built in its domain with other domains, through their leaders, thereby constructing the global (*inter-domain*) knowledge base at CIDN level.

As seen in Figure 2, the CIDN is composed of three security domains, or *Intrusion Detection Networks* (IDN), belonging to two administrative domains. The intra-domain knowledge base is built internally in each IDN through the alerts detected by its IDSs, while the inter-domain knowledge base is achieved by sharing the alerts and attacks detected in each IDN separately. In this sense, it provides a common and holistic view for intrusion management among IDSs, working either in the same or in different administrative domains.

The exchange of security information among IDSs of the same IDN, and also among IDNs working collaboratively to detect distributed attacks, will be made only with those that are considered trustworthy enough according to a trust a reputation model, as the one presented below.

IV. Trust and reputation management system

As previously mentioned, IDSs deployed in the same domain, or IDN, exchange security information among them in order to share a collaborative (intra-domain) knowledge base of alerts. Furthermore, the leaders of each IDN also exchange security information with each other in order to share a global (inter-domain) knowledge base about alerts and attacks [17]. In both cases, the accuracy of such a security information has to be assessed to confirm whether IDSs and/or IDNs have exhibited an honest or malicious behavior in their detection capabilities.

This proof of confidence checks if the security information corresponds to an actual incident occurred in the CIDN. Depending on this assessment, the alert or attack will be shared with the rest of the parties, or directly discarded as the reporting IDSs or IDNs are behaving maliciously.

An IDS, as any other kind of software, might be compromised by an attacker or to have an anomalous behavior due to malfunctioning. For example, if a bogus alert, generated from scratch by a compromised IDS, conveys real data about a legitimate user as the event source, the IDN's IRS might react by blocking any activity from/to such a user or even in the context of this work, by creating an unnecessary honeynet. Thus, it is required to measure the *goodness* of IDSs and IDNs before sharing their alerts and attacks.

As a solution, trust and reputation management systems can be used to measure the confidence that an IDS or IDN can deposit on others to accept their alerts and attacks as benevolent, and thence valid, either IDSs or IDNs running in the same or different administrative domains [18]. In the latter case, the use of ontologies and a common language and format in order to exchange security information, as introduced in Section II, is a key requirement to make the proposed trust and reputation mechanism a reality.

In [13], a trust-based framework is presented to avoid sharing bogus alerts within a CIDN. Moreover, an admission control algorithm is proposed for the Host-based IDSs (HIDSs) to select their collaborators. However, this proposal is restricted to the management of HIDSs, leaving Network-based IDSs (NIDSs) out of scope. Instead, a complete trust and reputation system for HIDSs and NIDSs, including their management in intra- and inter-domain environments, is proposed in [19]. Nevertheless, the reputation model designed for inter-domain scenarios is quite generic, and no experimental results are provided.

Bearing in mind both works, we propose in this paper that the trust of a generic domain or IDN Ω on a new alert or attack a, denoted as $T_\Omega(a)$, and generated by one or more IDSs regardless of their placement, is computed by using (1). Note that $T_\Omega(a)$ is quantified in $[-1, 1]$.

$$T_\Omega(a) = \alpha \cdot \mu_\Omega(a) + \beta \cdot \left(\bigoplus_{i=1, i\neq\Omega}^{n} \mu_i(a) \times T_{\Omega,i} \right) \qquad (1)$$

where $\mu(a) \in [-1, 1]$ is the confidence that a domain has on its IDSs with the proper configuration to detect the alert or attack a; $\oplus$ is an aggregation operation such as a mean or minimum function; n is the total number of domains with which Ω maintains a trust relationship; $T_{\Omega,i} \in [0, 1]$ represents the trustworthiness that Ω has on the i-th domain; and $\alpha, \beta \in [0, 1]$, with $\alpha+\beta = 1$, are two weights to balance the importance between local and external evidences.

The right-hand part of (1) can be seen as a recommendation about the own experiences of the Ω's reliable domains in detecting the same alert or attack a. Note that they will only share by those, through their leaders, that detected a.

A neutral trust on a, i.e., $T_\Omega(a) = 0$, reflects an absolute lack of confidence in detecting such an alert or attack. This uncertainty indicates that the IDSs that generated a have the same belief in this detection that those that did not.

On the other hand, $\mu(a)$ can be computed as described below, according to the number and reputation –denoted as $Rep(ids) \in [0, 1]$– of the IDSs that generated a and those that did not when they were properly configured for it. The former IDSs are part of the set G –generation–, while the latter are part of the set R –rejection–, with $A = G \cup R$.

$$\mu(a) = \frac{max\{\varphi_G\} \cdot \overline{\varphi_G} \cdot |\varphi_G| - max\{\varphi_R\} \cdot \overline{\varphi_R} \cdot |\varphi_R|}{\overline{\varphi_A} \cdot |\varphi_A|}$$

where φ represents the list of reputation values of all the IDSs involved in detecting a, i.e., $\varphi = \{Rep(ids)\}$, $\forall ids$, belonging to the set G, R, or A; $max\{\varphi\}$ is the highest reputation among all the IDSs belonging to a given set; $\overline{\varphi}$ is the average reputation of the IDSs with respect to a set; and $|\varphi|$ is the total number of IDSs in G, R, or A.

The maximum reputation of a given set, $max\{\varphi\}$, has been included in this equation to limit the confidence on a to the IDS with the highest reputation value; $\mu(a)$ cannot exceed the reputation of the most trustworthy IDS.

On the other hand, the weights in computing $T_\Omega(a)$, both α and β, can be set by the Ω's administrator as predefined and fixed values (e.g., by setting $\alpha = 0.75$ and $\beta = 0.25$). Even though using fixed weights can be considered as valid, we propose in this paper to follow a dynamic approach. Both weights can be computed according to the number of local instances of the alert or attack, generated by the IDSs deployed in the generic domain Ω, and those instances generated in other reliable domains for Ω as recommendation sources. Hence, α and β can then be computed as

$$\alpha = \frac{\Phi_\Omega(ids)}{\Phi_\Omega(ids) + \sum\limits_{i=1}^{n} \Phi_i(ids) \times T_{\Omega,i}}, \ \beta = 1 - \alpha \quad (2)$$

where $\Phi(ids)$ represents the percentage of instances generated by the IDSs in a given domain, i.e., $|ids| \in G$ divided by $|ids| \in A$, $\forall ids$; n is the total number of domains with which Ω has a trust relationship; and, as in (1), $T_{\Omega,i}$ is the trustworthiness that Ω has on the i-th security domain.

In this sense, as the number of alert or attack instances is higher in remote domains, the greater the importance their recommendations is in contrast to the local opinion, always depending on the trustworthiness that Ω has on them.

Once computing the trust on the new alert or attack a, by using (1), the domain has to decide if it is reliable or not. For example, a simple and suitable solution is to check if $T_\Omega(a) > T_{\Omega,\sigma}$, where $T_{\Omega,\sigma}$ is a predefined threshold value set by the Ω's administrator.

On the other hand, and provided that a is considered as reliable, each domain should update its trust on

1# the rest of the domains involved in detecting the alert or attack a, by computing the new $T_{\Omega,i}$ from the perspective of Ω; and

2# the new reputation of the IDSs depending on their behavior exhibited in sharing the alert or attack, by computing the new $Rep(ids)$ for each IDS involved in the detection of the alert or attack a.

Both calculations for this update (options 1# and 2#) can be made at time t, from the perspective of Ω, as

$$T^{(t)}_{\Omega,i} = T^{(t-1)}_{\Omega,i} + Sat_{\Omega,i}(a) \quad (3)$$

$$Rep^{(t)}_\Omega(ids) = Rep^{(t-1)}_\Omega(ids) \pm |(T_\Omega(a) - T_{\Omega,\sigma}) \cdot \mu_\Omega(a)| \quad (4)$$

where $(t-1)$ is the previous time when both values were computed; $Sat_{\Omega,i}(a) \in [-1, 1]$ is the satisfaction of Ω with the i-th reliable domain in detecting the same alert or attack; $|(T_\Omega(a) - T_{\Omega,\sigma}) \cdot \mu_\Omega(a)|$ indicates the absolute success of the IDS in detecting a; and ± is the reward or punish on the basis of the behavior exhibited by the IDS in its detection.

The satisfaction between two domains, from the perspective of the domain Ω, relies on the assessment made by the i-th domain in comparison with the local opinion of Ω, but also taking into account external recommendations about the same alert or attack. This calculation is made through

$$Sat_{\Omega,i}(a) = T_\Omega(a) - \left(\gamma \cdot \mu_\Omega(a) + \delta \cdot \bigoplus_{j=1, j\neq i}^{n} \mu_j(a) \times T^{(t-1)}_{\Omega,j} \right)$$

where $T_\Omega(a)$ has been computed in (1); $\mu(a)$ represents the confidence of a domain in its IDSs to detect a; $\oplus$ is the same aggregation operation used in (1); n is the total number of domains with which Ω maintains a trust relationship, except the i-th domain that is being assessed; $T^{(t-1)}_{\Omega,j}$ is the previous trustworthiness of Ω on the j-th domain; and $\gamma, \delta \in [0, 1]$ have the same meanings that the weights α and β, but extracting the i-th reliable domain in (2).

Finally, it is worth mentioning that those domains that have not informed about a, from the perspective of Ω, should be excluded from the satisfaction computation among domains. That is, $T^{(t)}_{\Omega,i}$ will not be updated if the domain i has not shared the alert or attack a.

V. Intrusion responses

After detecting a new threat by the IDSs deployed in a distributed fashion, as shown in Section III, the autonomous intrusion response system has to deploy the most appropriate response to mitigate and/or learn about the threat, provided that it is true according to the trust and reputation model of Section IV. This system must be able to rapidly react against intrusions and attacks in an autonomous and optimal way, without the administrator intervention. The reaction includes to infer the optimal response (*diagnosis phase*) and the deployment of such a response (*reaction phase*).

As mentioned in Section II, several AIRS solutions have been proposed in recent years to improve the limited response of current IDSs. They enrich the security information obtained from IDSs with other information sources, like the correlation of alerts or information about systems. The optimum response will be inferred to deploy and create the specifications needed to carry it out. To do so, the use of ontologies and formal policy languages is essential to define the behavior of the autonomous system and provide it with the ability of the self-protection required.

Classical reactions typically consist on deploying new firewall rules, whereas AIRS opens new possibilities in reaction; for example, by creating honeynets specifically adapted to the attack being detected. In this context, RECLAMO proposes a reaction based on the configuration and automatic deployment of honeynets optimized and adapted to each attack, according to the specifications provided by the autonomous system. A honeynet is basically a set of honeypot systems –servers, clients, and routers– as similar as possible to the production ones, but specifically prepared to be attacked and equipped with silent tools to allow monitoring attacks that go unnoticed to attackers.

All systems in a honeynet are typically interconnected by following a network topology similar to the production one that is being protected or studied. Given the complexity in launching a honeynet, virtualization is an essential technique that greatly facilitates its deployment and management. Its ability to build and deploy multiple logical systems over a single physical box drastically reduces the number of physical systems needed to create a honeynet, also allowing increasing the complexity of the honeynets created.

There are many projects and initiatives that have used virtualization as a basic tool to dynamically create honeynets. One of the most advanced ones is Collapsar that combines a powerful distributed traffic capture system with a server farm [20], where the interesting traffic is redirected to dynamically create honeynets that process it.

The use of complex honeynets results in the need for specialized tools to manage them. They should facilitate their definition –topology, addresses, types of systems, among others–, their deployment, and their monitoring, hiding all the complexity of the underlying virtualization platforms. There are several tools available such as VNX (*Virtual Networks over linuX*), Netkit, MLN (*Manage Large Networks*), and vBET (*VM-Based Emulation Testbed*). Among them, VNX emerges as the most powerful solution due to i) its ability to deploy virtual network scenarios over a cluster of servers; ii) its large scalability in creating high interaction honeynets; and iii) the creation of very complex honeynets, even over distributed cluster infrastructures [21].

VI. The RECLAMO project

RECLAMO is a research project aimed at designing and creating an advanced framework for enhancing existing intrusion detection and reaction proposals. To reach this ambitious objective, this project deals with the different key technologies mentioned before and combine them in a single solution to provide an automated response system to attacks.

The concept of self-protection, as one of the four features of any autonomous system, is the key concept driving the main component of the RECLAMO architecture. This provides the ability to infer the most appropriate response for a given intrusion, taking into account not just the intrusion, but also other parameters related to it, such as the context or the trust and reputation of the network source.

The proposed autonomous system will use information models, formally defined with ontologies, to combine intrusion data, self-evaluation parameters, trust and reputation of the different involved elements –IDSs and IDNs–, and security information coming from collaborative IDS-based systems in the same or different administrative domains. This security information will be assessed with a set of security metrics, represented with formally-defined behavior specification languages, in order to reason and infer the most appropriate response, taking into account all the inputs and other criteria specified in the security metrics.

The most promising approach in RECLAMO is the one based on advanced reaction techniques, with a special focus on the so-called *deception responses*. They are based on the dynamic generation and deployment of honeynets, where the attacks will be diverted, that will be created ad-hoc for each attack. The most optimal reaction will be optimized to gather as much information as possible from each attack.

The diversion of the attack to a dynamically ad-hoc generated honeynet is to be adequately confined in order to mitigate it and learn from it. The dynamic honeynet generation will be done by using advanced virtualization techniques able to generate large-scale heterogeneous honeynets.

An envisaged architecture of the system, including all the components and solutions presented in previous sections, is schematically shown in Figure 3.

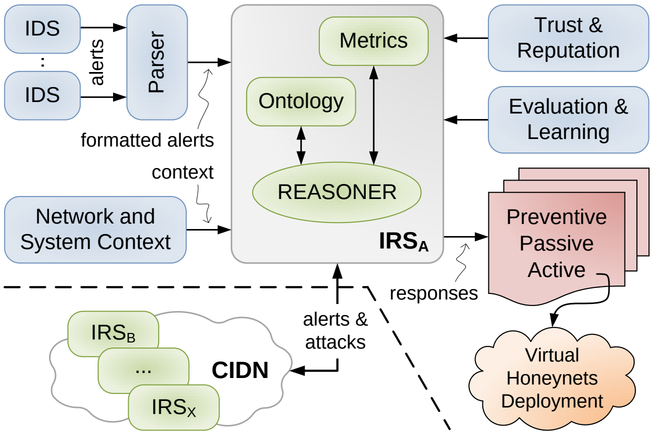

Figure 3. System architecture of RECLAMO

Intrusions or attacks detected from the alerts individually generated by each IDS are analyzed in real-time by using a model of intrusions, responses, and security metrics that allow triggering an inference process from the detected intrusion. Concepts like autonomous system, ontologies, trust and reputation management, collaborative intrusion detection and prevention networks, self-protection, and virtualized honeynets are clearly identified in Figure 3. All of these concepts are considered as a key part of the novel automated response system to attacks proposed in RECLAMO.

VII. Experimental results

We discuss in this section some experimental results obtained with a prototype implementation. The fundamental aim is to assess how the designed trust and reputation model, presented in Section IV, can strengthen the RECLAMO's automated intrusion response system in order to avoid unnecessary responses against bogus or fictitious threats. For example, this will avert the creation of an unnecessary honeynet in the context of this work.

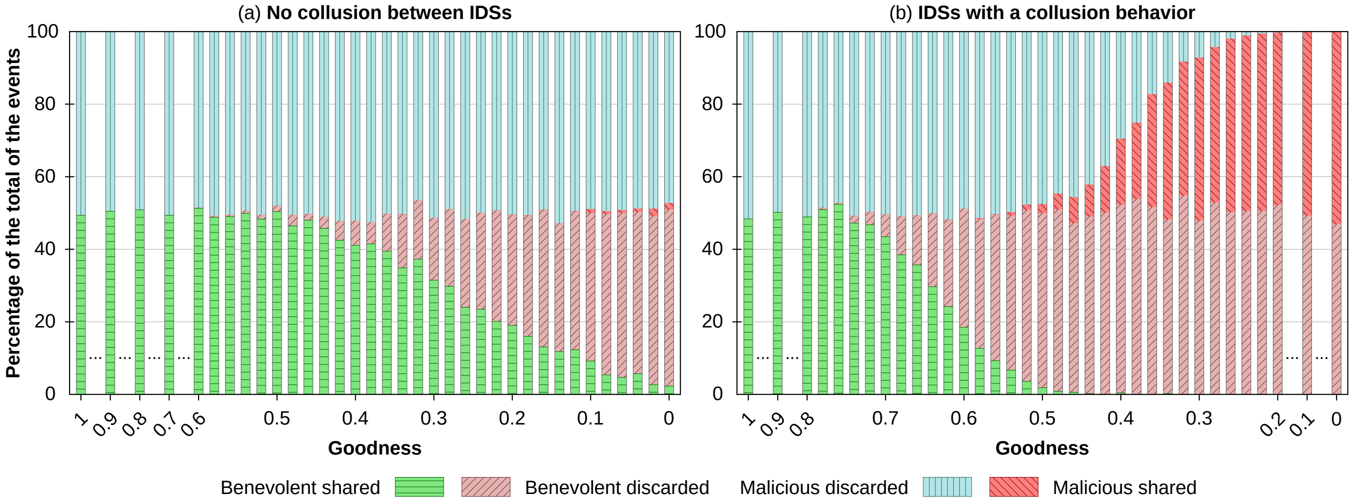

Figure 4. Events shared or discarded by IDSs with a benevolent or malicious behavior while varying their goodness over time

A. Experiment 1: Assessing Resilience against Malicious or Misbehaving IDSs

In this first experiment, we have simulated a single IDN to evaluate the events –alerts and attacks– shared by its IDSs as their behaviors or goodness vary throughout time. This IDN is composed of 50 IDSs with an initial reputation of 1. In order to decide if events are reliable, we have set the IDN's threshold to 0.25 (i.e., $T_{IDN,\sigma} = 0.25$). In this experimental setting, we have injected 1000 random events for each of the goodness values exhibited by the IDSs (around 50 % of reliable events and the rest of bogus).

It is worth emphasizing that no external evidences, or recommendations, are taking into account in computing the trust on the diverse events, as a single IDN has been simulated. That is, just the left-hand part of (1) is considered. Moreover, the update process of the trustworthiness values among IDNs –see (3) in Section IV– is not carried out for the same reason as before; just the update of the IDSs reputation is performed –see (4) in Section IV.

Figure 4 depicts two outcomes of this first experiment. In both charts, the y-axis represents the percentage of the events shared and discarded by IDSs with a benevolent or malicious attitude, while the x-axis represents the goodness when they generate events. In this latter case, a goodness of 1 indicates that IDSs properly classify the events as reliable or bogus; that is, they expose a complete benevolent behavior. In the opposite case, a goodness of 0 indicates that IDSs exhibit a complete malicious behavior.

In this experiment, malicious IDSs have been simulated from two different perspectives. First, Figure 4a shows the outcomes when IDSs exhibit a usual random behavior in their detection tasks (they sometimes share the events, while other times do not report anything). Secondly, Figure 4b shows the outcomes when IDSs present a collusive behavior, thereby forming a malicious collective, always taking the same decision about sharing the events or not. In both cases, malicious IDSs share the events if they are bogus, or not when they are actually reliable.

As seen in Figure 4a, the proposed trust and reputation system is able to detect up to a 50 % of IDSs with a malicious attitude without any collusion among them; there is only a negligible error in discarding a 1.6 % of reliable or benevolent events (98.4 % of success). From that 50 %, more and more reliable events are discarded as the IDSs behavior is getting worse. Nevertheless, events generated by malicious IDSs are not shared although their behaviors decay in time; such IDSs lost their reputations previously and only a 1.8 % of bogus events are finally shared.

Results are worse when all the IDSs form a malicious collective for deceiving the IDN. These outcomes are shown in Figure 4b. In this case, reliable events begin to be discarded from a 20 % of malicious IDSs, although up to a 30 % of these IDSs indicate good results by only discarding a 6.3 % of benevolent events.

However, as opposed to the previous case, bogus events are increasingly shared as the goodness value of the IDSs worsens. This fact is due to malicious IDSs begin to behave in the same way, i.e., there is no diversity in their attitudes, leading the IDN to believe that its IDSs are detecting the events in a proper fashion. At the end of this second test, all IDSs have the highest reputation so that the IDN cannot discriminate them as malicious.

As a main conclusion of this first experiment, we can affirm that the proposed trust and reputation system is capable of, at least, supporting a 30 % of malicious IDSs ($goodness = 0.7$) without the RECLAMO architecture being compromised. Our trust and reputation system gets even better results when there is no collusion among such IDSs, achieving a very promising performance with up to a 50 % of malicious IDSs.

B. Experiment 2: Detecting Malicious IDNs

We have extended the previous experiment towards a multi-domain environment, at CIDN level, to assess how the behavior of the IDNs varies as their goodness worsens in time. In this case, our intention is to extract the number of necessary events until deciding if one or more IDNs are behaving in a malicious fashion.

In this experiment, we have followed the same setting as before, but simulating 25 IDNs with 50 IDSs in each and injecting 100 sequential events in the CIDN for each goodness value. Figure 5 depicts the outcomes, where each line represents the percentage of malicious IDNs, while the goodness indicates the behavior of their IDSs. Note that we assume that an IDN is considered as malicious when its trustworthiness $T_{IDN,\sigma}$ is below 0.25.

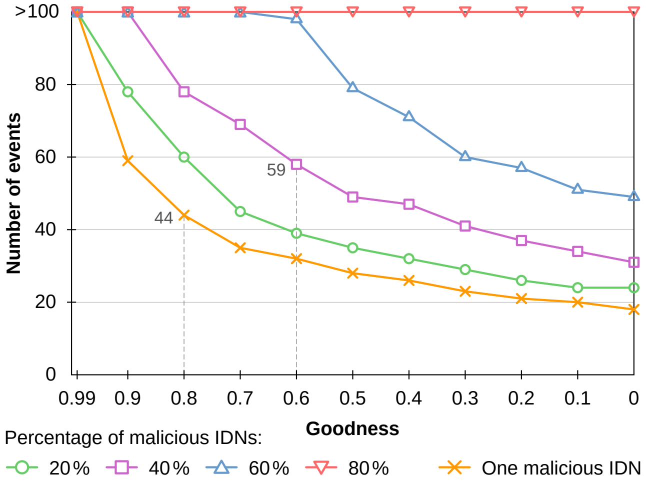

Figure 5. Number of events needed to detect malicious IDNs

As can be observed in Figure 5, only 44 events are required to classify an IDN as unreliable, when it is the only malicious one in the CIDN and there is a 20 % of IDSs in it ($goodness = 0.8$) with a malicious behavior. As their goodness increases, it is easier and faster to identify such an IDN as malicious. Nevertheless, this identification is "more difficult" as the percentage of malicious IDNs increases.

On the other hand, less than 60 events are required to classify a 40 % of the IDNs as unreliable, when they have more than 40 % of malicious IDSs ($goodness = 0.6$). Nevertheless, the number of events begins to be quite high when there are more than a 40 % of unreliable IDNs, with up to a 50 % of malicious IDSs, being this number above 100 events when there are an 80 % of unreliable IDNs regardless of the amount of malicious IDSs in them.

As a main conclusion of this second experiment, we have demonstrated that our trust and reputation system needs a relatively small number of events in order to identify IDNs with a malicious behavior, which can be isolated later for not accepting their alerts as reliable.

Finally, and after analyzing the results of both experiments, we can assert that the trust and reputation management system proposed in Section IV is tough enough to avoid launching unnecessary responses, i.e., the creation of a honeynet, despite the presence of malicious IDSs or IDNs.

VIII. Related work

Along this paper, we have analyzed some related works in the context of intrusion response systems with special automated features in reaction. However, none of them offers support for *semantic coherence*, a key feature when operability is a crucial requirement in multi-domain environments. In addition, George Kurtz is currently leading an expert team to implement CrowdStrike [22]; an industrial solution for building IDS-based honeynets to learn and react against Advanced Persistent Threats. This is a still rather immature solution whose its main weakness, in comparison with our proposal, is that CrowdStrike does not rely on a trust and reputation management system in order to isolate the malicious components of its architecture.

On the other hand, we have also analyzed and discussed a few works with another key requirement in the success of an AIRS. This is the assessment of the information sources, when IDSs share alerts and attacks stating that they have detected an actual (distributed) threat. Without this evaluation, IDSs with malicious attitudes can lead the AIRS to react mistakenly against a bogus threat. Both [13] and [19] are the first initiatives in this context, although none of them presents an entire model for the trust and reputation management in multi-domain environments.

To the best of our knowledge, RECLAMO is the first proposal where a complete AIRS, with automated reaction capabilities, is presented to build IDS-based honeynets by considering the assessment outcomes from a multi-domain trust and reputation model. This avoids improper responses when a (distributed) threat has not actually happened.

IX. Conclusion

In this paper, we have briefly reviewed the main objectives and relevant topics behind the RECLAMO project. This is a research project aimed at designing and creating an advanced framework for enhancing existing intrusion detection and reaction proposals. To reach this ambitious objective, RECLAMO deals with different key technologies, analyzed in this paper, and combine them in a single solution to provide an automated response system to attacks.

As one of the first results in the ongoing RECLAMO project, we have presented a complete trust and reputation model for multi-domain environments. This model allows assessing the behavior of reporting IDSs in order to accept or reject their alerts and attacks as reliable security information. The experimental results have confirmed the importance of a trust and reputation model before an AIRS infers and deploys the most appropriate response.

Acknowledgment

This work has been partially funded with support from the Spanish MICINN (project RECLAMO, *Virtual and Collaborative Honeynets based on Trust Management and Autonomous Systems applied to Intrusion Management*, with codes TIN2011-28287-C02-01 and TIN2011-28287-C02-02) and the European Commission (FEDER/ERDF). Thanks also to the Funding Program for Research Groups of Excellence granted by the Séneca Foundation with code 04552/GERM/06.

References

[1] Symantec Corp., "Internet security threat report, vol. 17," April 2012.

[2] N. Stakhanova, S. Basu, and J. Wong, "A taxonomy of intrusion response systems," *International Journal of Information and Computer Security*, vol. 1, no. 1/2, pp. 169–184, January 2007.

[3] C. A. Carver, "Adaptive agent-based intrusion response," Ph.D. dissertation, Texas A&M University, May 2001.

[4] B. Foo, Y.-S. Wu, Y.-C. Mao, S. Bagchi, and E. Spafford, "ADEPTS: Adaptive intrusion response using attack graphs in an e-commerce environment," in *Proceedings of the 2005 International Conference on Dependable Systems and Networks*, pp. 508–517, June 2005.

[5] P. A. Porras and P. G. Neumann, "EMERALD: Event monitoring enabling responses to anomalous live disturbances," in *Proceedings of the 20th National Information Systems Security Conference*, October 1997.

[6] G. B. White, E. A. Fisch, and U. W. Pooch, "Cooperating security managers: A peer-based intrusion detection system," *IEEE Network*, vol. 10, no. 1, pp. 20–23, January 1996.

[7] S. M. Lewandowski, D. J. Van Hook, G. C. O'Leary, J. W. Haines, and L. M. Rossey, "SARA: Survivable autonomic response architecture," in *Proceedings of the DARPA Information Survivability Conference & Exposition II*, vol. 1, pp. 77–88, June 2001.

[8] Z. Q. Wang, Q. Zhao, H. Q. Wang, and L. J. Yu, "MAIRF: An approach mobile agents based intrusion response system," in *Proceedings of the 1st IEEE Conference on Industrial Electronics and Applications*, pp. 1–4, May 2006.

[9] W3C Recommendation, "OWL Web ontology language - Guide," February 2004.

[10] W3C Member Submission, "SWRL: A semantic Web rule language combining OWL and RuleML," May 2004.

[11] C. V. Zhou, C. Leckie, and S. Karunasekera, "A survey of coordinated attacks and collaborative intrusion detection," *Computers & Security*, vol. 29, no. 1, pp. 124–140, February 2010.

[12] T. Gamer, "Collaborative anomaly-based detection of large-scale internet attacks," *Computer Networks*, vol. 56, no. 1, pp. 169–185, January 2012.

[13] C. Fung, J. Zhang, I. Aib, and R. Boutaba, "Trust management and admission control for host-based collaborative intrusion detection," *Journal of Network and Systems Management*, vol. 19, no. 2, pp. 257–277, June 2011.

[14] L. Mekouar, Y. Iraqi, and R. Boutaba, "Reputation-based trust management in peer-to-peer systems: Taxonomy and anatomy," in *Handbook of Peer-to-Peer Networking*, pp. 689–732, March 2010.

[15] T. Dohi and T. Uemura, "An adaptive mode control algorithm of a scalable intrusion tolerant architecture," *Journal of Computer and System Sciences*, vol. 78, no. 6, pp. 1751–1774, November 2012.

[16] T. Dimitriou, G. Karame, and I. Christou, "SuperTrust - A secure and efficient framework for handling trust in super peer networks," in *Distributed Computing and Networking*, pp. 350–362, January 2008.

[17] A. Durresi, M. Durresi, and L. Barolli, "Network trust management in emergency situations," *Journal of Computer and System Sciences*, vol. 77, no. 4, pp. 677–686, July 2011.

[18] M. Gil Pérez, F. Gómez Mármol, G. Martínez Pérez, and A. F. Gómez Skarmeta, "Mobility in collaborative alert systems: Building trust through reputation," in *NETWORKING 2011 Workshops, Workshop on Wireless Cooperative Network Security*, ser. Lecture Notes in Computer Science, vol. 6827, pp. 251–262, May 2011.

[19] M. Gil Pérez, F. Gómez Mármol, G. Martínez Pérez, and A. F. Skarmeta Gómez, "RepCIDN: A reputation-based collaborative intrusion detection network to lessen the impact of malicious alarms," *Journal of Network and Systems Management*, vol. 21, no. 1, pp. 128–167, March 2013.

[20] X. Jiang and D. Xu, "Collapsar: A VM-based architecture for network attack detention center," in *Proceedings of the 13th Conference on USENIX Security Symposium*, vol. 13, August 2004.

[21] F. Galán, D. Fernández, J. E. López de Vergara, and R. Casellas, "Using a model-driven architecture for technology-independent scenario configuration in networking testbeds," *IEEE Communications Magazine*, vol. 48, no. 12, pp. 132–141, December 2010.

[22] G. Kurtz, "Crowdstrike," http://www.crowdstrike.com.