Automated reaction based on risk analysis and attackers skills

Wael Kanoun, Nora Cuppens-Bouhlahia, Frédéric Cuppens

Automated Reaction based on Risk Analysis and Attackers Skills in Intrusion Detection Systems

Wael Kanoun[1], Nora CUPPENS-BOULAHIA[1] and Frédéric CUPPENS[1]

TELECOM Bretagne, Cesson Sévigné 35576, France.
wael.kanoun, nora.cuppens, frederic.cuppens@telecom-bretagne.eu

Abstract. Current intrusion detection systems go beyond the detection of attacks and provide reaction mechanisms to cope with detected attacks or at least reduce their effect. Previous research works have proposed methods to automatically select possible countermeasures capable of ending the detected attack. But actually, countermeasures have side effects, can be as harmful as the detected attack and not adapted to the attacker's actions. In this paper, we propose to improve the reaction selection process by giving means to (i) quantify the effectiveness and select the countermeasure that has the minimum negative side effect on the information system by adopting a risk assessment and analysis approach, and (ii) assess the skill and knowledge level of the attacker from a denfensive point of view.
Keywords: Intrusion detection system, attack scenario, countermeasure, risk analysis, potentiality, impact, skill and knowledge.

1 Introduction

In intrusion detection approach, the primary objective is to detect and identify attacks, and then react to counter the detected attack to block it or to mitigate its impact on the Information System (IS). There are two different approaches for the reaction perspective: hot reaction [15] and policy based reaction [14, 11]. The first aims to launch a local action on the target machine to end a process, or on target network component to block a traffic, that are the cause of the launched alerts. The second acts on more general scope; it considers not only the threats reported in the alerts, but also constraints and objectives of the organization operating the IS and this by modifying the access control policy. Therefore a trade-off can be established between security objectives, operation objectives and constraints. Whatever the adopted approach, each countermeasure can have negative or positive side effects. The same countermeasure that was activated to end an attack can make the IS more vulnerable, expose it to other attacks, or even have an impact more disastrous than the attack itself. For example *Firewall reconfiguration* is effective against a DOS attack, but can be very harmful if valuable connections will be lost, therefore many questions emerge: Is it better to stand still? Or is the attack harmful enough to react? In this case, which countermeasure must be selected with minimum negative

side effects? To answer these questions, we adopt a Risk Assessment and Analysis approach. This approach is already used to analyze and evaluate the risks that threaten organization assets. Our approach is to use the same approach to evaluate the effectiveness of each countermeasure in real time and improve the automated reaction mechanism. The first step of a risk analysis method is to collect data that describes the system state in real-time. The second step is analyzing them and finding the potential threats and their severity. The final step is to study the countermeasure effectiveness to eliminate these threats or reduce their severity: The goal is not always to block the attack, but to minimize the risk incurred by target information system. Therefore a risk assessment method is used to evaluate and quantify the risk of an attack and its countermeasures. The method is useful to decide when it is suitable to react, and which countermeasure should be activated. Another important aspect is the Attackers Skills and Knowledge level *(SK_Level)*. Such data is useful for the automated reaction process. If a novice script kiddie attacker is trying to establish a remote session, a simple *TCP reset* will be enough to eliminate the detected attack. Otherwise, in the case of an experienced attacker, the *TCP reset* can be ineffective and a *firewall reconfiguration* may be needed. Therefore assessing the *SK_Level* make the reaction decision module more accurate and effective. We can assume that a Risk Assessment and Analysis approach combined with the assessment of the Attackers Skill and Knowledge level make the automated process of reaction and countermeasure selection more accurate, realistic, cost effective, and with minimum intervention of the human administrator. In section 2 our solution using risk analysis and skill and knowledge assessment approaches, and an implementaion is showed in section 3. In section 4 related works are presented. Finally section 5 concludes the paper.

2 Solution

To react against attacks, a fine and efficient diagnostic procedure to detect and identify the intrusions is needed. However, due to the limitation and unreliability of the intrusion detection probes like SNORT [8], only low-level events can be detected with high rates of false alarms. Therefore, to detect and recognize the current attack, an alerts correlation procedure is required for proper reaction. The correlation procedure recognizes relationships between alerts in order to associate these alerts into a more global intrusion scenario, and the intrusion objectives that violate the predefined organization security policies. There are many approaches that can be used for this purpose: implicit [19], explicit [16, 20] and semi-explicit [12, 21] correlations. The semi-explicit approach is based on the description of the elementary intrusions corresponding to the alerts. This approach then finds causal relationships between these elementary alerts and connects these elementary alerts when such a relationship exists. The correlation procedure then consists in building a scenario that corresponds to an attack graph of steps corresponding to the elementary intrusions.

Semi-Explicit Correlation Definition Two LAMBDA models A and B are anti-correlated if the post-condition of A matches the pre-condition of B. This semi-explicit [12, 21] approach is more generic and flexible because only the elementary steps are defined as entities and not the whole attacks scenario. The LAMBDA [13] language can be used to describe these elementary steps by defining their pre-conditions and post-conditions. Regarding reaction, it is also the most interesting because it provides a precise diagnosis of the ongoing intrusion scenario by construction the attack graph, predict the potential future steps and the intrusion objectives.

Using an approach similar to the one used to describe elementary intrusions, elementary countermeasures can be specified. In this case, anti-correlation [9] can be used to find the countermeasures capable of ending a detected scenario.

Anti-Correlation Definition Two LAMBDA models A and B are anti-correlated if the post-condition of A matches the pre-condition negation of B. The anti-correlation [9] approach is based upon finding the appropriate countermeasure that turn an elementary future step of an attack inexecutable due to preconditions value modifications. Therefore, using the anti-correlation approach, the administrator knows which countermeasures from a predefined library are capable of blocking the threat.

2.1 Risk Assessment Model

As explained in the previous section, the anti-correlation approach [9] can be used of generating a set of candidate countermeasre capable of ending the detected attack, but without assessing the impact of the detected attack nor these candidate countermeasure. Therefore, this reaction approach can be refined by combining it with the risk analysis model proposed in [17]. This model is used to evaluate the Total Gravity Risk of the IS once an attack is detected and after the simulated execution of the candidate countermeasure. Only the countermeasures that reduce the Total Gravity Risk are kept and a new set of Risk Efficient Countermeasures *(RISK_EFF_CM)* is instantiated. The Total Risk Gravity can be assessed after evaluating the Potentiality *(POT)* and the Impact *(Imp)* of the detected attacks. The structure of the model is described in Figure 1.

Potentiality *Pot* The major factor Potentiality Pot measures the probability of a given scenario to take place and achieve its objective with success. To evaluate Pot, we must first evaluate its minor factors: natural exposition $Expo$ and dissuasive measures $Diss$ and we have to take into account classification of the attack also. The minor factors can be evaluated after the appropriate audit clusters are calculated. These clusters are questions-tests that aim to evaluate the system state (active services, existent vulnerabilities, etc.). The value *zero* indicates that the studied scenario is impossible, and the value MAX_VALUE indicates that the occurrence and the successful execution of the scenario are inevitable.

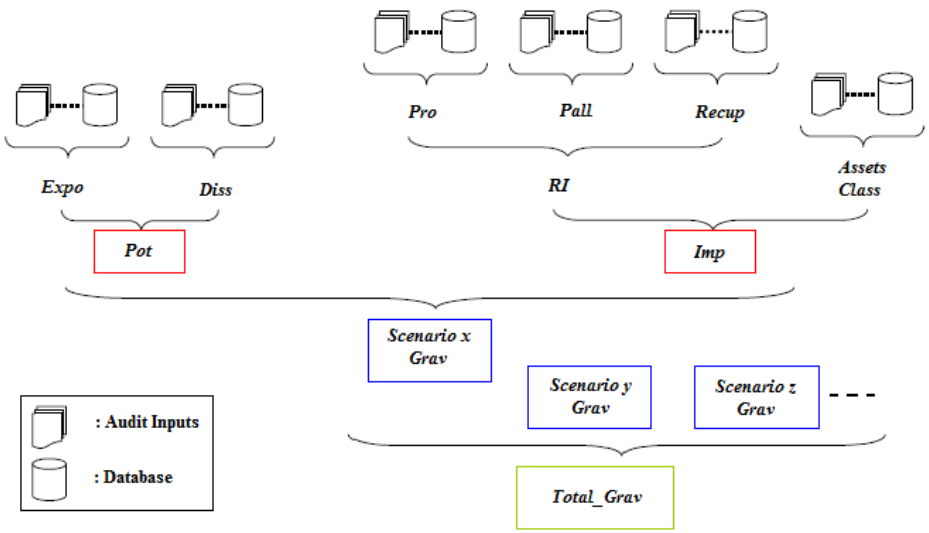

Fig. 1. Risk Assessment structure

Impact *Imp* The second major factor to evaluate Risk Gravity of an attack scenario is Impact *Imp*. $\overrightarrow{Imp}$ is defined as a vector with three cells that correspond to the three fundamental security principles: Availability *Avail*, Confidentiality *Conf* and Integrity *Integ*. Therefore, with each Intrusion Objective, a vector $\overrightarrow{Imp}$ is associated and should be evaluated. Actually, it is not possible to statically evaluate $\overrightarrow{Imp}$ of a scenario (or more precisely the $\overrightarrow{Imp}$ of the scenario's intrusion objective) directly because it depends on several dynamic elements. The impact depends on the importance of the target assets $\overrightarrow{Class}$, and the impact reduction measures level $\overrightarrow{IR}$ that are deployed on the system to reduce and limit the impact once the attack was successful.

Risk Gravity of an Attack Scenario or a Countermeasure *Grav* For each detected attack, the risk gravity must be evaluated to estimate the danger level of this attack. The risk is the combination of Potentiality and Impact using a predefined function f. An attack that occurs frequently with little impact may have the same risk level as another rare attack that have significant impact. If a scenario has *Pot* or *Imp* equal to *zero*, the scenario's gravity risk *Grav* will be null. To assess the risk Gravity of a candidate countermeasure CM_u, the same function f is used to assess the risk as shown in the following equation:

$$G_{CM_u} = f(Pot = MAX\_VALUE,\ Imp = CM_u.Impact) \tag{1}$$

The use of MAX_VALUE for the *Pot* parameter is justified by the fact that countermeasures, contrary to detected attack scenarii, do not have intrinsic potentiality. This can be explained by the fact that once a countermeasure is se-

lected, it must be activated successfully in the IS and therefore its impact must be considered with maximum potentiality. For the attack scenarii, each one has a proper potentiality that must be combined with the attack scenario impact to deduce the risk gravity.

Total Risk Gravity $Total\_Grav$ and $Total\_Grav'_u$ In most situations, the correlation and reaction modules do not deal with one specific scenario. Instead, these modules have to take into account many candidate and even simultaneous scenarios. Therefore, before estimating the total gravity of risk, we must evaluate the gravity of risk of each scenario separately. Then we define the Total gravity as an ordered vector containing the values of gravity risk of each candidate scenario. An order relation can be defined between the different instances of $\overrightarrow{Total\_Grav}$ using the lexicographic comparison. Therefore we are able to judge which graph has the highest risk gravity. We define also $\overrightarrow{Total\_Grav'_u}$ similarly to $\overrightarrow{Total\_Grav}$, where the difference is that $\overrightarrow{Total\_Grav'_u}$ is assessed with the new state of the information system after the simulated execution of the countermeasure CM_u.

Risk Efficient Countermeasures Set Once for each countermeasure u, G_{CM_u} and $\overrightarrow{Total\_Grav'_u}$ are evaluated, $\overrightarrow{Total\_Grav\_CM_u}$ can be evaluated :

$$\overrightarrow{Total\_Grav\_CM_u} = \overrightarrow{Total\_Grav'_u} \cup G_{CM_u} \tag{2}$$

Now, only the countermeasures from $Anticorrelated\_CM$ that decrease the total gravity risk are kept and a new set $Risk\_Eff\_CM$ is defined that contains only risk efficient countermeasures:

$$\forall CM_u \in Anticorrelated\_CM; \overrightarrow{Total\_Grav\_CM_u} \leq \overrightarrow{Total\_Grav} \tag{3}$$
$$\Rightarrow CM_u \in Risk\_Eff\_CM$$

2.2 Skill and Knowledge Assessment

To react properly against a detected attack scenario, the choice of countermeasure depends on the attacker's skill and knowledge level. To end an attack executed by a novice, a simple *close connection* can be effective, which it is not the case when facing an expert attacker where a *firewall configuration* is needed. Therefore, the assessment of the Attacker's Skill and Knowledge Level would be very useful to tune our reaction model. In fact, it would be useless to activate a complex countermeasure against a beginner; or to activate a simple countermeasure against an expert attacker who can easily bypass it. Another point is that the attacker can have internal knowledge of the information system. For instance, a remote attacker that have the proper credentials and is able to connect from the first attempt, or an attacker that is able to predict the tcp sequence of a connection that uses a complex algorithm (and not use the standard tcp sequence number incrementation algorithm), must be taken in consideration that they have internal knowledge and/or high level of expertise.

SK_Level Label To assess the Attacker's Skill and Knowledge, a defensive point of view must be adopted. The only information of the attacker's action accessible by the target information system is the generated alerts for each executed step in the attack scenario. Each step is described with a specific language like LAMBDA [13]. We propose to add a new label called *SK_Level* (Skill and Knowledge level). For the attack actions, this label indicates the minimum level of skill and knowledge required to execute this action-step successfully. For the countermeasure, it indicates the value of this level that the attacker can not bypass once it is activated. This new label can have the values shown in Table 1.

Table 1. SK_Level label values

SK_Level	**Skill**	**Internal Knowledge**
0	Low	No
1	Medium	No
2	Medium	Yes
3	High	No
4	High	Yes

Attacker's SK_Level and Skill and Knowledge Efficient Countermeasure Set We consider that an attacker capable of executing an attack step with high *SK_Level* is an expert attacker and thus, a more sophisticated countermeasure is required. The Attackers Skill and Knowledge Level *(SK_Level)* must be evaluated to refine the countermeasure selection. A first approach to assess the Attackers *SK_Level* is to retrieve the *SK_Level* maximum value among the successfully executed attack steps. The use of *Max* function could not be accurate enough and further advanced approaches must be explored. Once the Attackers *SK_Level* is assessed, only the countermeasures that have a *SK_Level* greater than the Attackers one will be kept. Hence, the selected countermeasure is adapted to the Attackers Level and a new set called Skill and Knowledge Efficient countermeasures *(SK_EFF_CM)* can be instantiated.

The correlation engine by determining the executed attack steps by an attacker, is capable of assessing the attacker's level of skill and knowledge *Attacker_SK_Level*. The first approach to evaluate *Attacker_SK_Level* can be done using the following equation:

$$Attacker\_SK\_Level = Max_i(SK\_Level_i) \tag{4}$$

$$where\ SK\_Level_i = Attack\ Step_i.SK\_Level\ and\ Step_i \in Executed\_Steps$$

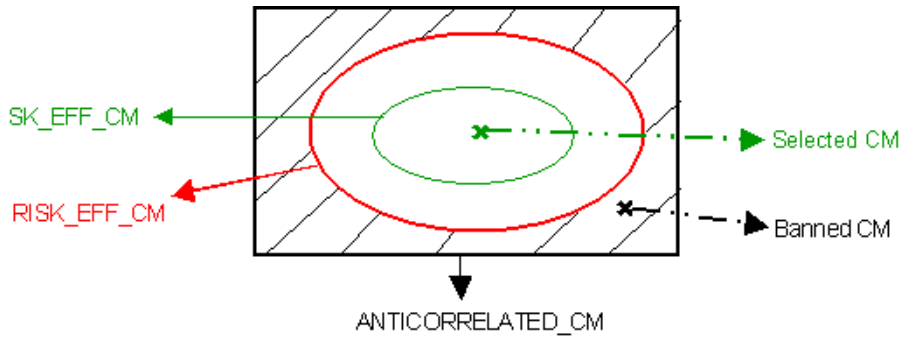

Fig. 2. *Risk_Eff_CM* and *SK_Eff_CM* sets

A possible method is to set honeypots [18, 23] and redirect the attacker to execute his/her attack steps on them. This will be used to collect the maximum number eof xecuted steps to assess accurately his/her *SK_Level*.

Once the *Attacker_SK_Level* had been assessed, only the countermeasures from *Risk_Eff_CM* such that the *SK_Level* is higher than *Attacker_SK_Level* are kept, and a new set called *SK_Eff_CM* (See Figure .2) can be defined and instantiated countermeasures:

$$\forall CM_u \in Risk\_Eff\_CM; Attacker\_SK\_Level \leq CM_u.SK\_Level \quad (5)$$

$$\Rightarrow CM_u \in SK\_Eff\_CM$$

2.3 Countermeasure Selection Procedure

Once the *Risk_Eff_CM* and *SK_Eff_CM* have been instantiated (See Figure 2), a clear automatic procedure can be applied to select the most appropriate countermeasure :

```
If SK_EFF_CM ≠ ∅
   Select(Min_SK_Level(CM_u))
If Risk_EFF_CM ≠ ∅
   Select(Min_Risk(CM_u))
Select(None)
```

3 Implementation of the Solution

CRIM (Correlation and Recognition of Malicious Intentions) [10] is a prototype that has been developed by Telecom Bretagne. It implements the fusion, semi explicit correlation and anti-correlation features using the LAMBDA language. It

collects the generated alerts and aggregates them. Then CRIM visualizes the detected attacks in real time, the future steps that can be executed by the attacker using the semi-explicit correlation principle, and the candidate countermeasure using the anti-correlation principle. As a proof of concept, a new module has been created to validate our proposal briefly described in the previous sections. As shown in 3, this module is used to assess the Risk Gravity of the detected attack scenarii and the candidate countermeasures, then it instantiates the *Risk_Eff_CM* countermeasure set. A first version of this module has been developped, but no public version is yet released. Another module takes in charge of assessing the Attackers *SK_Level* and compare it to the *SK_Level* of the countermeasures that belong to the *Risk_Eff_CM*, then it instantiates the *SK_Eff_CM* set. Once the two sets are instantiated, the selection procedure can be applied to activate the most appropriate and effective countermeasure. Works has begun to develop a first version of the module.

The Figure 3 shows the output of the CRIM prototype corresponding to the detection of the Mitnick attack.The mitnick attack aims to gain remote illegal shell by causing a DoS to a legal machine and then "stealing" its pre-established tcp connection with the target machine. The attack graph generated by CRIM using the LAMBDA language [13] is composed of four elementary steps . We suppose that the attacker was capable to execute successfully the first three steps. Therefore one final step remains before the attacker achieves his or her intrusion objective *Illegal Remote Shell* on a critical machine. Dark green circles represent elementary steps of the intrusion scenario, light yellow losanges correspond to candidate reaction and boxes are intrusion objectives.

The attacker has executed successfully the first three steps, and thus only one step remains. Therefore the potentiality and therefore the *Total_Risk_Grav* have high values (*=4*). In other hand, the Attacker's *SK_Level = Max(1,3,2) = 3*.

There two candidate countermeasures are capable of reducing the total gravity risk from 4 to 1 (See *Total_Risk_CMx in Figure 3*, therefore the two countermeasures are in the *Risk_Eff_CM*. The Attacker's skill and knowledge level is three and that indicates the fact he or she is not a novice. Thererfore a *block connection* that has a *SK_Level = 2* could be not efficient and a *firewall reconfiguration* that has a *SK_Level = 3* is needed. Hence only the second countermeasure is in *SK_Eff_CM* and recommended to be launched.

4 Related Works

Intrusion detection systems with reaction capabilities like SNORT [8] already exist. SNORT offers reflex reaction when a given attack is detected like blocking packets, sending visible warning and logging. No advanced reasoning on the reaction consequence is conducted, and side effects could appear with devastating consequences. Many industrial solutions exist like IBM Internet Security [5] and Cisco Secure IDS [2]. These solutions are efficient in intrusion prevention and offer protection against well known attacks with the corresponding impact using

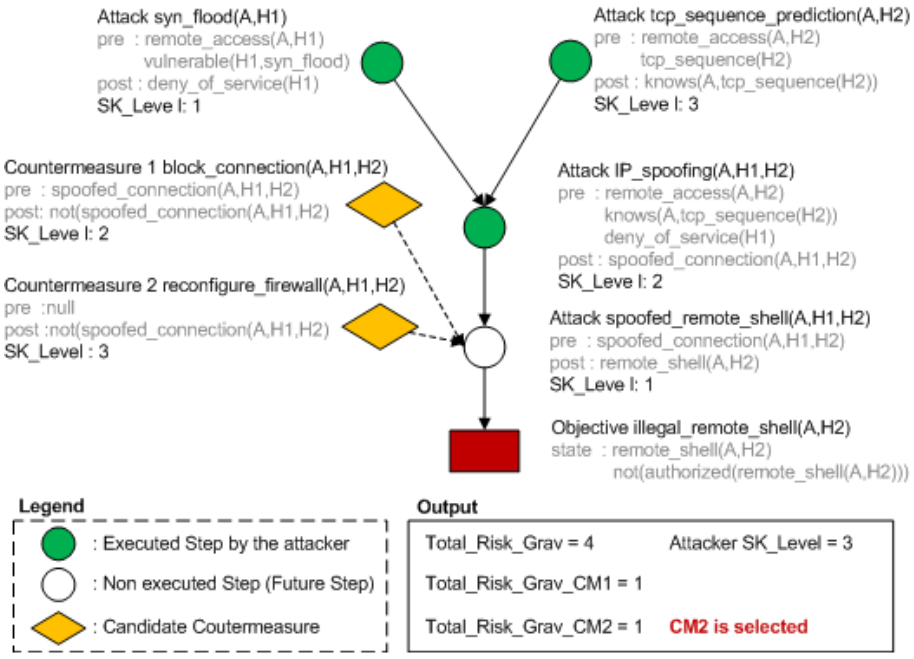

Fig. 3. CRIM output snapshot

database like CVSS [6]. The main drawback is that there is no following up and no monitoring of the detected attacks to assess their risks and impacts in real time once they bypassed the security and prevention measures, dependably on the target organizations and assets. Another limitation is that these solutions handle vulnerabilities exploit without considering the complete scenarii. There are several Risk Assessment methods like EBIOS [7, 1], MARION [3], MEHARI [4], etc. These methods are used to manage system assets and evaluate the risk that threatens these assets; they are unfortunately abstract, informal and incompatible with intrusion detection and computer systems: Many elements and parameters are related to physical and nature disasters (fire, earthquake, failure, etc.). There are also elements that need redefinition to be compatible with the intrusion detection systems like potentiality and impact of a threat. As suggested in [22], the risk exposure can be evaluated in terms of business perspective by using financial metrics. Another problem is that these methods can not be used in real-time. Our goal is to evaluate the system and the available countermeasure actions in real-time to help the administrator to chose the best action, or to make the reaction process automatic with minimum human intervention.

5 Conclusion and Future works

A first version of the Risk Analysis module has been implemented, and current works is being conducted to develop the Skill and Knowledge Assessment module. Series of tests will be conducted to evaluate the effectiveness and the performance of these added modules.
First, the tests will handle classical attacks (e.g. Mitnick Attack, Password Guessing, etc.). In the future, attack and countermeasure will be modeled with LAMBDA models, and attacks simulations will be conducted for VoIP services: CRIM will control and supervise the VoIP services status, detect and recognize attacks in real time, then conduct a Risk analysis and Skill and Knowledge assessment to propose to the administrator, or to the automated reaction module, the most effective countermeasure. Intrusion detection systems aim to detect attacks, however such detecion is not quite useful without reaction. Against given attacks, there could be many possible countermeasres. Our approach will help administrators taking their decisions and selecting the proper countermeasure(s) by assessing the impact of the detected attacks and the candidate countermeasre. We are even conducting further researches and tests to turn the reaction selection and activation process fully automated. As we know, no similar approaches exists in the litterature.

References

1. Certification of ebios method with iso 27001 : `www.cases.public.lu/publications/recherche/these_jph/NMA-JPH_MISC27.pdf`.
2. Cisco secure systems official website: `www.cisco.com/en/US/products/sw/secursw/ps2113/index.html`.
3. Description of MARION method published by clusif : `http://i-a.ch/docs/CLUSIF_Marion.pdf`.
4. Description of MEHARI method published by clusif : `www.clusif.asso.fr/fr/production/ouvrages/type.asp?id=METHODES`.
5. IBM internet security systems official website : `www.iss.net`.
6. National vulnerabilities database official website : `http://nvd.nist.gov/cvss.cfm`.
7. Official document describing EBIOS published by sécréteriat general de la defense nationale rpublique francaise : `www.ssi.gouv.fr/fr/confiance/documents/methodes/ebiosv2-memento-2004-02-04.pdf`.
8. Snort official website :`www.snort.org`.
9. *Anti-correlation as a criterion to select appropriate counter-measures in an intrusion detection framework*, volume 61, chapter Annals of Telecommunications. January 2006.
10. *CRIM : un module de corrélation d'alertes et de réaction aux attaques*, volume 61, chapter Annals of Telecommunications. September-October 2006.
11. N. Cuppend-Boulahia and F. Cuppens. Specifying intrusion detection and reaction policies: An application of deontic logic. In *Ninth Workshop on Deontic Logic in Computer Science (DEON'08)*, Luxembourg, July 2008.

12. F. Cuppens, F. Autrel, and A. Miege et S. Benferhat. Recognizing malicious intention in an intrusion detection process. In *Second International Conference on Hybrid Intelligent Systems*, Santiago, Chili, December 2002.
13. F. Cuppens and R. Ortalo. Lambda: A language to model a database for detection of attacks. In *Third International Workshop on Recent Advances in Intrusion Detection (RAID'2000)*, Toulouse, France, 2000.
14. H. Debar, Y. Thomas, F. Cuppens, and N. Cuppens-Boulahia. Enabling automated threat response through the use of a dynamic security policy. *Journal in Computer Virology (JCV)*, 3(3), August 2007.
15. S. Gombault F. Cuppens and T. Sans. Selecting appropriate counter-measures in an intrusion detection framework. In *Computer Security Foundation Workshop*.
16. M. Huang. A large-scale distributed intrusion detection framework based on attack strategy analysis. Louvain-La-Neuve, Belgium, 1998.
17. W. Kanoun, N. Cuppens-Boulahia, and F. Cuppens. Advanced reaction using risk assessment in intrusion detection systems. In Springer, editor, *Second International Workshop on Critical Information Infrastructures Security (CRITIS07)*, Malaga, Spain, 2007.
18. S. Krasser, J. Grizzard, and H. Owen. The use of honeynets to increase computer network security and user awareness. School of Electrical and Computer Engineering.
19. R. Lippmann. Using key string and neural networks to reduce false alarms and detect new attacks with sniffer-based intrusion detection systems. In *Second International Workshop on the Recent Advances in Intrusion Detection (RAID'99)*, October 1999.
20. B. Morin and H. Debar. Correlation of intrusion symptoms: an application of chronicles. In *Proceedings of the Sixth International Symposium on the Recent Advances in Intrusion Detection (RAID'02)*, Pittsburg, USA, September 2003.
21. Peng Ning, Yun Cui, and Douglas S. Reeves. Constructing attack scenarios through correlation of intrusion alerts. In *ACM Conference on Computer and Communications Security*, 2002.
22. J.-P. Sauvé, R. A. Santos, R. R. Almeida, and J. A. B. Moura. On the risk exposure and priority determination of changes in it service management. *Lecture Notes in Computer Science*, pages 147–158, September 2007.
23. C. Yin, M. Li, J. Ma, and J Sun. Honeypot and scan detection in intrusion detection system. School of Electronic InformationEngineering.