Implementing a Semantic Approach for Events Correlation in SIEM Systems

Tayeb Kenaza, Abdelkarim Machou, Abdelghani Dekkiche

Implementing a Semantic Approach for Events Correlation in SIEM Systems

Tayeb Kenaza, Abdelkarim Machou, and Abdelghani Dekkiche

Ecole militaire polytechnique, BP 17 BEB 16043, Alger, Algérie
ken.tayeb@gmail.com
https://orcid.org/0000-0002-4240-2978

Abstract. Efficient reasoning in intrusion detection need to manipulate different information provided by several analyzers in order to build a reliable overview of the underlying monitored system trough a central security information and event management system (SIEM). SIEM provides many functions to take benefit of collected data, such as Normalization, Aggregation, Alerting, Archiving, Forensic analysis, Dashboards, etc. The most relevant function is Correlation, when we can get a precise and quick picture about threats and attacks in real time. Since information provided by SIEM is in general structured and can be given in XML, we propose in this paper to use an ontological representation based on Description Logics (DLs) which is a powerful tool for knowledge representation and reasoning. Indeed, Ontology provides a comprehensive environment to represent any kind of information in intrusion detection. Moreover, basing on DLs and rules, Ontology is able to ensure a decidable reasoning. Basing on the proposed ontology, an alert correlation prototype is implemented and two attack scenarios are carried out to show the usefulness of the semantic approach.

Keywords: SEIM, Monitoring, Intrusion detection, Alert correlation, Description logic, Rules based reasoning, Ontology, OWL

1 Introduction

Information systems security requires the deployment of a rigorous security policy with several security mechanisms and tools. We generally start with prevention systems such as authentication where the goal is to prove the identity of users, access control where the goal is to define rights of users on data, and firewalls where the role is to control the access to the information system towards the outside world.

However, these mechanisms are not sufficient to fully protect systems against malicious attacks. Indeed, computer systems often exhibit vulnerabilities, which allow attackers to bypass preventive mechanisms. In addition, many security tools focus on the protection against external attacks, while attacks can be also internal. For example, client side attacks are a very common nowadays. Therefore, intrusion detection is necessary as a second layer of security after deploying

prevention systems. Unfortunately, Intrusion Detection is still imperfect for two reasons. First, intrusion detection systems (IDSs) generate a very large number of low-level alerts, where most of them are false positive; i.e, alerts generated in the absence of attacks. And second, IDSs suffer from false negative which is the absence of alerts in the presence of attacks.

In order to overcome these problems, a promising approach is the so-called cooperative intrusion detection [20, 4], which allows various intrusion detection tools to cooperate. In addition to IDS, other analyzers can be considered such as network and vulnerability scanners in order to correlate alerts by considering contextual information. This can be done by including for example topology and cartography. In fact, nowadays all security tools have to cooperate using a central security information and event management system (SIEM). A SIEM provide many functions to take benefit of the collected data, such as Normalization, Aggregation, Alerting, Archiving, Forensic analysis, Dashboards, etc. The most relevant function is Correlation, when we can get a precise and quick picture about the threats and attacks in real time. However, most of proprietary SIEM use its own data representation and its own correlation techniques which are not always favorable to share knowledge and to do custom reasoning.

In such situation, the use of common and extensible formalism to describe information in intrusion detection is a major concern. This information is generally structured and encoded in XML. For example, this is the case of alerts in IDMEF (for Intrusion Detection Message Exchange Format) and TAXII (Trusted Automated eXchange of Indicator Information) as well as the vulnerabilities in OVAL (Open Vulnerability and Assessment Language) and STIX (Structured Threat Information eXpression). However, information encoded here in XML is limited to a syntactic representation basing in different taxonomies. Consequently, in the absence of a semantic approach correlating this information is a fastidious task. Indeed, it is more interesting to move from taxonomies to ontology specification languages [12, 9], which are able to simultaneously serve as recognition, reporting and correlation languages.

Several existing knowledge representation models can be used in SIEM such as [8, 1, 2, 7]. In this paper, our contribution can be seen as an enhancement of existing representations by regrouping a large amount of information into a single ontology. This will offer a comprehensive and extensible knowledge representation which can be used in many event correlation systems.

On an other hand, given that tools used in SIEM are not totally reliable, usually conflicts appear between them [15, 19]. For example, one can easily see that IDSs are not fully reliable since they generate many false positives and false negatives. Therefore, it is very important to resolve these conflicts in order to exploit the cooperation. Hence, our second contribution is an ontological reasoning approach to correlate alerts in order to reduce the amount of alerts, especially false positives.

The rest of this paper is organized as follows. In Section 2 we briefly recall intrusion detection and some works of knowledge representation proposed in the context of intrusion detection, and then we present the proposed ontology.

Section 3 presents an architecture of an alert correlation system based on DLs reasoning. In section 4 experiments are conducted and results are discussed. In section 5, some related works are briefly discussed. Section 6 concludes this paper.

2 Related works

The automatic correlation of information from different security systems has been a vivid topic of research for over a decade [20, 4]. Numerous approaches have been developed for correlating alerts and other log entries to strength the power of intrusion detection systems. Here, we briefly discuss only related works regarding the use of ontology in computer security. Ontology can be used in many field in SIEM, such as to analyze user behavior and system activities, or to identify known attack patterns, or also to analysis abnormal behavior and activity of both systems and users. Notice that semantic approaches have many advantages over existing approaches, mainly two aspects: the formal and extensible knowledge representation capability and the decidable reasoning.

Using ontology in computer security is relatively new. The first research work was done by Jeffrey Undercoffer et al. [16]. They produced an ontology that specify a model of computer attack. Their ontology is based on attack strategies which is categorized according to targeted system components, tools of attacks, consequences of attacks, and location of attackers. They present their model as a target-centric ontology. Since the work of Jeffrey many other ontologies was proposed. In [17], Wang et al. propose an Ontology for Vulnerability Management (OVM) which contains several concepts about vulnerabilities, affected products, consequences and countermeasures, etc. Authors have used their own implementation of their ontology without referring to any languages. In [2], Azevedo et al. propose a domain-ontology with more generic and abstract concepts in the field of computer security, serving as the basis for the construction of other specific security-domain-ontologies called CoreSec. In [5], Jian-bo et al. provide an ontology-based attack model which is used to assess the information system security from attack angle. The proposed ontology consists of five dimensions, which include attack impact, attack vector, attack target, vulnerability and defense.

More recently, many semantic description methods for the security policy has been proposed. In [14], an ontology-based method is presented to solve the problem of the semantic description and verification of a security policy. Onto-ACM (ontology-based access control model), is a semantic analysis model proposed by Chang Choi et al. [3] to address the difference in the permitted access control between service providers and users. More over, in [18] ontologies are used to perform threat analysis and develop defensive strategies for mobile security. Autors has proposed on ontology-based approach that can identify an attack profile in accordance with structural signature of mobile viruses, and also overcome the uncertainty regarding the probability of an attack being successful, thanks to semantic reasoning.

3 Ontological based specification and reasoning for Alert Correlation

3.1 Knowledge Representation in Intrusion Detection

In front of an intrusion detection environment characterized by a very low detection rate, a high rate of false alerts, and a poor granularity of the information provided by alerts, a huge effort has been made by the intrusion detection community for the standardization of threats and attacks representation. The resulted data formalisms (e.g. IDMEF, TAXII, STIX, etc.) has provided a workspace for open communication between security tools and has been largely used in many alert correlation systems [4, 6].

Despite their different approaches, alert correlation systems have to share knowledge about attacks and the context in which they occur. However, many security tools do not care about how they represent their knowledge and how they use it. We think that having a coherent and formal model to represent knowledge is important for any correlation system. M2D2 is among the most important work in this area, it is a relational model that regroup essential information used in correlation, such as alerts, events, nodes, softwares, etc. In 2009, this model was revised by adding new concepts and by regrouping concepts into classes, this new model is called M4D4 [8]. In a recent work [10] proposed by Sadighian et al, authors have designed a set of comprehensive and extensible ontologies, and have implemented fusion and detection algorithms based on OWL-DL and SQWRL in order to allow reducing false positives.

3.2 The Proposed Ontology

Strassner defines the ontology as follows : "An ontology is a formal, explicit specification of a shared, machine-readable vocabulary and meanings, in the form of various entities and relationships between them, to describe knowledge about the contents of one or more related subject domains throughout the life cycle of its existence [13]. This meaning of ontology is used mostly in the context of knowledge sharing.

IDMEF and M4D4 are among the most important work in terms of knowledge representation in the domain of intrusion detection. However, IDMEF does not contain enough information because it describes just alerts, and M4D4 is proposed in the context of network intrusion detection including contextual information (cartography and topology) and the description of vulnerabilities.

In this section, we propose an ontological conceptualization that combines the representation of IDMEF, M4D4, TAXII and other information sources such as OVAL, STIX and NVD.Generally, we can divide knowledge in intrusion detection into 5 groups [8] : Analyzers, Events and alerts, Attacks and Vulnerabilities, Contextual information, and Users and Attackers. Figure 1 shows the main concepts and relations of the proposed ontology, baptized "ONTO-SIEM".

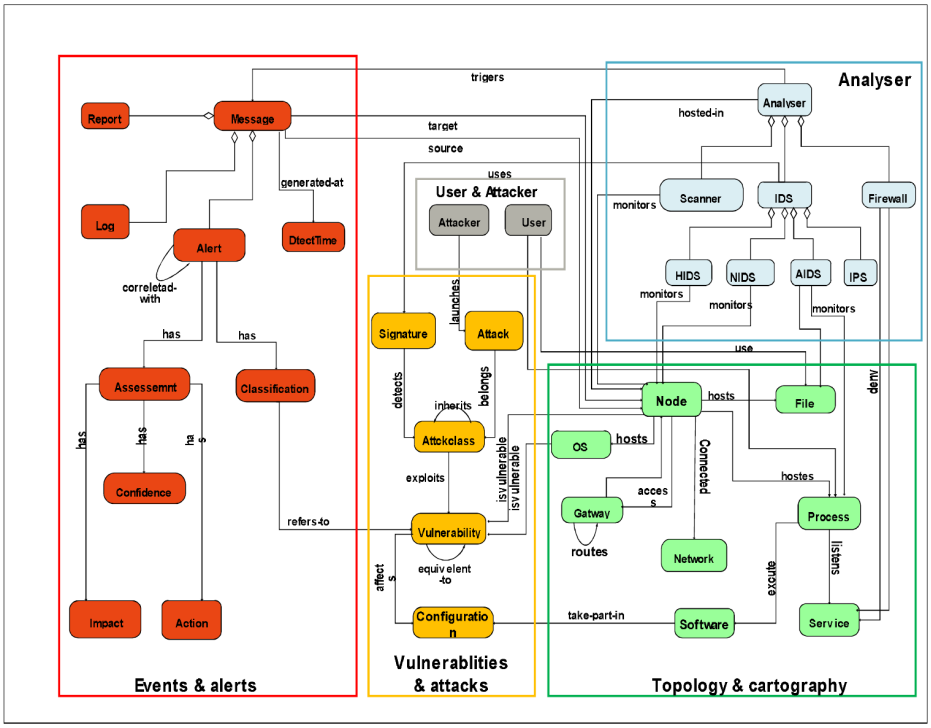

Fig. 1. Main concepts and relations of ONTO-SIEM

4 Ontology based Event Correlation System

The use ONTO-SIEM is very suitable for event correlation within a SIEM, when many tools have to cooperate and to exchange information. Indeed, we developed a prototype of alert correlation system to show the importance and usefulness of this ontology. The architecture of our system consists of two essential modules : the conversion module that puts reported alerts into the ontology, as well as contextual information (topology and cartography), and the correlation module that allows reasoning about the constructed ontology. Figure 2 summarizes the architecture of the correlation system.

In order to use an ontology within an application, it must be specified in a formal representation. Indeed, a variety of languages exists that are used to represent conceptual models, with varying expressiveness, ease of use and computational complexity. We used OWL, which is a recommendation of The World Wide Web Consortium (W3C), widely used in web semantic. OWL is based on Description Logics. Description Logics are known for their expressiveness and their clearly defined semantics that allow a decidable reasoning.

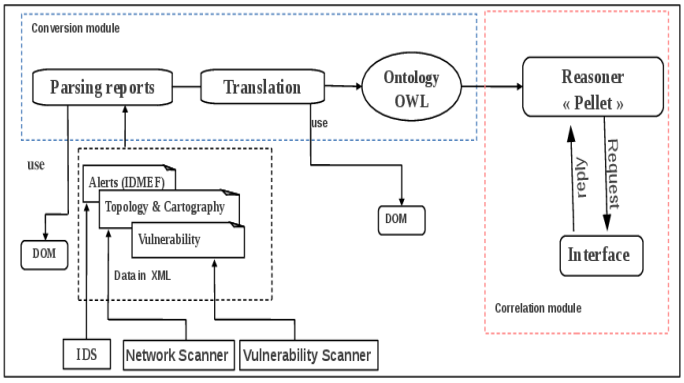

Fig. 2. ONTO-SIEM based alert correlation system architecture.

In this work, we build our ontology using the API Jena (http://jena.apache.org/), and the reasoning is provided by Pellet (http://clarkparsia.com/pellet/) which is a full OWL-DL reasoner.

4.1 Populating the Ontology

To populate our ontology we need to use several tools. Information about hosts and the network topology are given using Nmap (http://nmap.org/). This tool can provide many information such as running hosts and their operating systems, servers listening in these hosts with their corresponding version, and many further information. Information about the vulnerabilities of systems and applications are given using Nessus (http://www.tenable.com/products/nessus/). Information about attacks are given in real time by IDS/IPS, in our system we used Snort (http://www.snort.org/) with a set of VRT and community rules. Notice that it is also possible to insert directly information into ONTO-SIEM by security operators, namely add information about equipments, systems and applications.

4.2 Reasoning with the Ontology

Reasoning is important in ontology because it allows to ensure the quality of ontology. Indeed, through the use of a reasoner, it is possible to test whether concepts are non-contradictory, and also to derive implicit relations.

Filtering Events : In Table 1 and 2, we present some rules that can be used to filtering pertinent and not pertinent events. Most of theses rules are reused from the Pasagrada framework [10]. Notice that if an event is not classified as pertinent this does not means it is not pertinent. To decide so, the event must satisfy at least one rule from Table 2.

Table 1. Filtering Pertinent Events

Rule-1 (based analyser)	$Alert(?a) \wedge Node(?h) \wedge Analyser(?an) \wedge hasTarget(?a,?h) \wedge monitors(?an,?h) \wedge Trigers(?an,?a) \rightarrow sqwrl : select(?a)$
Rule-2 (based OS)	$Alert(?a) \wedge Node(?h) \wedge OS(?o) \wedge Vulnerability(?v) \wedge Classification(?cl) \wedge hasTarget(?a,?h) \wedge hasClass(?a,?cl) \wedge Hosts(?h,?o) \wedge isvulnerable(?o,?v) \wedge Refers-to(?cl,?v) \rightarrow sqwrl : select(?a)$
Rule-3 (based OS with vulnerability Equivalence)	$Alert(?a) \wedge Node(?h) \wedge OS(?o) \wedge Vulnerability(?v1) \wedge Vulnerability(?v2) \wedge Classification(?cl) \wedge hasTarget(?a,?h) \wedge hasClass(?a,?cl) \wedge Hosts(?h,?o) \wedge isvulnerable(?o,?v1) \wedge Refers-to(?cl,?v2) \wedge equivalent-to(?v1,v2?) \rightarrow sqwrl : select(?a)$
Rule-4 (based application)	$Alert(?a) \wedge Node(?h) \wedge Software(?sof) \wedge Vulnerability(?v) \wedge Classification(?cl) \wedge Process(?p) \wedge Configuration(?conf) \wedge hasTarget(?a,?h) \wedge hasClass(?a,?cl) \wedge Hosts(?h,?p) \wedge Execute(?p,?sof) \wedge Take-part-of(?sof,?conf) \wedge Affects(?v,?conf) \wedge Refers-to(?cl,?v) \rightarrow sqwrl : select(?a)$
Rule-5 (Application with vulnerability Equivalence)	$Alert(?a) \wedge Node(?h) \wedge Software(?sof) \wedge Vulnerability(?v1) \wedge Vulnerability(?v1) \wedge Classification(?cl) \wedge Process(?p) \wedge Configuration(?conf) \wedge hasTarget(?a,?h) \wedge hasClass(?a,?cl) \wedge Hosts(?h,?p) \wedge Execute(?p,?sof) \wedge Take-part-of(?sof,?conf) \wedge Affects(?v1,?conf) \wedge Refers-to(?cl,?v2) \wedge equivalent-to(?v1,v2?) \rightarrow sqwrl : select(?a)$

Rule 1 selects events generated by analyzers that can actually monitors the target. For example, an IDS can only detect events that occur in the network to which it is connected. This can be explicitly provided by the relation *monitors* or infered, for example for NIDS, as follows.

$$monitors \equiv hosted-in \wedge connected \wedge netNodes \tag{1}$$

Rules 2 and 4 select events based on the vulnerability of the OS and the Application, respectively. Some tools such as vulnerability scanners can confirm if an OS or an application is vulnerable or not to a given vulnerability. Obviously, this concern only known vulnerabilities, not zero-day vulnerabilities. Rules 4 and 6 are similar to rules 3 and 4, they just consider the equivalence between vulnerabilities reported by several organisms with different names. These tow rules deal with the case when different analyzers (IDS and scanners) refer to the same vulnerability with different names or references.

Table 2. Filtring Not Pertinent Events

Rule-6 (based analyzer)	$Alert(?a) \wedge Node(?h) \wedge Analyser(?an) \wedge hasTarget(?a,?h) \wedge Trigers(?an,?a) \wedge NotMonitors(?an,?h) \rightarrow sqwrl : select(?a)$
Rule-7 (based vulnerability)	$Alert(?a) \wedge Node(?h) \wedge Vulnerability(?v) \wedge Classification(?cl) \wedge hasClass(?a,?cl) \wedge hasTarget(?a,?h) \wedge IsNotVulnerable(?h,?v) \rightarrow sqwrl : select(?a)$

Rule 6 is the inverse of rule 1, it selects events reported by tools that does not actually monitor the target of the attack. Rules 7 selects events reported for target that is not actually vulnerable to the referred vulnerability. This concern both OS and Software vulnerabilities. The question now is haw to get such information, because traditionally scanners only report affected hosts not protected onces. For instance we admit that such information is explicitly given by the relation $IsNotVulnerable$.

Aggregating Events : Here we consider only pertinent event for which we try to group events together in order to generate meta-event. A meta-event represent a summarizing of a single malicious activity that causes multiple elementary events. We distinguish tow types, Host based meta-event and Network-based meta-event. In this latter we can distinguish three sub-classes [10], within certain time interval.

1. One-to-One (Rule-8). This can be an attack attempted by a single attacker against a single target, for example a SQL injection.
2. One-to-Many (Rule-9). This can be an attack attempted several time by a single attacker against many targets, for example a network or vulnerabilities scan.
3. May-to-One (Rule-10). This can be an attack attempted by several attackers against a single target, for example a DDoS.

Table 3. Network based Events Aggregating

Rule-8 (one to one)	$Message(?meg) \wedge Node(?h1) \wedge Node(?h2) \wedge hasSource(?meg, ?h1) \wedge hasTarget(?meg, ?h2) \wedge generated{-}at(?meg, ?t) \wedge biggerThan(?t, ?t1) \wedge lessThan(?t, ?t2) \rightarrow sqwrl : select(?meg)$
Rule-9 (many to one)	$Message(?meg) \wedge Node(?h1) \wedge Node(?h2) \wedge hasTarget(?meg, ?h1) \wedge generated - at(?meg, ?t) \wedge biggerThan(?t, ?t1) \wedge lessThan(?t, ?t2) \rightarrow sqwrl : select(?meg)^s qwrl : count(?meg)$
Rule-10 (one to many)	$Message(?meg) \wedge Node(?h1) \wedge Node(?h2) \wedge hasSource(?meg, ?h1) \wedge generated - at(?meg, ?t) \wedge biggerThan(?t, ?t1) \wedge lessThan(?t, ?t2) \rightarrow sqwrl : select(?meg)^s qwrl : count(?meg)$

For host-base meta-event, we consider several event' features to decide to group or not events. These features are Node (N), User (U), Process (P), Service (S), and File (F) [10]. Based on these features and in case of a complete availability of data, we can distinguish two main subclasses.

1. NUP (Rule-11), when many events have the same node, the same user and the same process.
2. NUF (Rule-12), when many events have the same node, the same user and the same file.

Table 4. Host based Events Aggregating

Rule-11 (Node-user-process)	$Message(?msg) \wedge Node(?h) \wedge User(?u) \wedge Process(?p) \wedge Analyser(?an) \wedge hasTarget(?msg, ?h) \wedge Launches(?u, ?p) \wedge Hosts(?u, ?p) \wedge Moniters(?an, ?p) \wedge Trigers(?an, ?msg) \wedge generated-at(?msg, ?t) \wedge biggerThan(?t, ?t1) \wedge lessThan(?t, ?t2) \rightarrow sqwrl : select(?msg) \wedge sqwrl : count(?msg)$
Rule-12 (Node-user-file)	$Message(?msg) \wedge Node(?h) \wedge User(?u) \wedge File(?f) \wedge Analyser(?an) \wedge hasTarget(?msg, ?h) \wedge Launches(?u, ?f) \wedge Hosts(?u, ?f) \wedge Moniters(?an, ?f) \wedge Trigers(?an, ?msg) \wedge generated - at(?msg, ?t) \wedge biggerThan(?t, ?t1) \wedge lessThan(?t, ?t2) \rightarrow sqwrl : select(?msg) \wedge sqwrl : count(?msg)$

5 Experimental results

ONTO-SIEM is implemented using Protégé which is a powerful editor supporting OWL-Dl, SWRL and other many reasoners such as HermiT, Pollet, etc. Protégé is powerful thanks to many plugins that can be add to it.

To evaluate the proposed rules we have used UNB-ISCX-2012 [11] which is an open Intrusion Detection Evaluation dataset. UNB-ISCX-2012 is an interesting benchmark because it provides a real labeled traffic which contains both attacks and normal activities. Moreover, this benchmark provides a complete capture of the traffic with a set of divers and multi-steps attack scenarios. Table 5 gives a summary about the benchmark and its attack scenarios.

Table 5. UNB-ISCX benchmark Description

Day	Scenario Description	Size (GB)
Friday	Normal activity	16.1
Saturday	Normal activity	4.22
Sunday	Infiltrating from the inside + normal activity	3,95
Monday	HTTP DDoS + normal activity	6,85
Tuesday	DDoS using an IRC Botnet	23,4
Wednesday	Normal activity	17.6
Thursday	Brute Force SSH + normal activity	12.3

We tested our approach using 2 scenarios, namely "Infiltrating from inside" and "HTTP DDoS". The first scenario consists to obtain access to a host inside the local network, and then the compromised host is used as a pivot to attack computers which are not accessible via the Internet. The second scenario consists of performing a stealthy, low bandwidth denial of service attack without the need to flood the network (for more details about the testbed architecture and the attack scenarios see [11]).

5.1 Populating ONTO-SIEM

The UNB-ISCX-2012 benchmark provides only the raw traffic collected during 7 days, and an xml file containing labeled attacks with their execution periods, but no thing about topology and the cartography of the testbed, as well as the Vulnerabilities. So, we did a had work to manually extract that information from the benchmark. For instance, in Table 6 we show some of used OSs and Softwares. Vulnerabilities are also manually insert into ONTO-SIEM.

Table 6. UNB-ISCX benchmark' OS and Softwares

OS	Softwares
4 Windows xp SP1, Windows xp SP2, Windows 7, Windows Server 2003, Ubuntu 10.04.	Acrobat Adobe Reader 8.1, Apache 2.2.9, Bind 9, Postfix, Dovecot, IIS 6, MSSQL Server, OpenSSH 5.3, Vsftp 2.2.2.

The raw traffic are analyzed using Prelude-Snort, and then reported alert are translated from xml to ONTO-SIEM. Prelude is an open SIEM which can easly be connected with many analyzers. Prelude' output are in xml, so it will be very simple to translate them to ONTO-SIEM.

5.2 Discussing results

Concerning the first scenario, snort has reported alerts which are translated into ONTO-SIEM using Prelude (https://www.prelude-siem.org/). Correlation process has given results shown in Table 7. The first filtering level has reduced the amount of alerts by 30% (3307 alerts are removed) which are alerts that refer to no existing hosts, while the second filtering level has reduced about 17% (1340 alerts are removed) of the amount of the remainder alerts. Therefore, after this preliminary filtering stage, more than 43% of alerts are reduced. Only 57% of the initial alerts will be concerned by further correlation processing using the ontology.

Table 7. Filtering alerts of scenario 1

Rules	Not pertinent alerts	
Rule 1	3307/11016	(30.02% of alerts are removed)
Rule 2 to 5	1340/7709	(17.40% of alerts are removed)

The same discussion is given for the second scenario. Table 8 shows correlation process results. The first filtering level has reduced the amount of alerts by 27.95% (1561 alerts are removed) which are alerts that refer to no existing hosts, while the second filtering level has reduced about 10% (410 alerts are removed) of the remainder alerts. Therefore, after this preliminary filtering stage, more

than 36% of alerts are reduced. Only 64% of the initial alerts will be concerned by further correlation processing using the ontology.

Table 8. Filtering alerts of scenario 2

Rules	Not pertinent alerts	
Rule 1	1561/5584	(27.95% of alerts are removed)
Rule 2 to 5	410/4023	(10.20% of alerts are removed)

6 CONCLUSION AND FUTURE WORK

We proposed in this paper a domain ontology for a cooperative intrusion detection based on several data sources such as IDMEF, TAXII, STIX, M4D4, OVAL, NVD, etc. This ontology is implemented with OWL which is recommended by W3C since 2004 for the representation of ontologies in the Web Semantic. OWL is based on Description Logics which are a decidable fragment of the first order logic and are well suitable to represent structured information.

We have illustrated the usefulness of this ontology through an application in the context of alert correlation. This application allows automatic translation of alerts generated by IDSs to OWL, as well as contextual information generated by network and vulnerability scanners. Furthermore, a set of rules proposed to be inferred over the constructed ontology, these rules aim mainly to remove not pertinent alerts. This is very important to reduce the amount of alerts by analyzing in priority pertinent alerts. Other actions can be performed in the perspective to complete this work. Indeed, the proposed ontology need to be completed by more concepts and relation to allow a more comprehensive correlation rules, and also by using other reasoning mechanisms provided by OWL-Dl such as the verification of consistency and the satisfiability of concepts.

References

1. F. Abdoli and M. Kahani. Ontology-based distributed intrusion detection system. In *14th International CSI Computer Conference.*, pages 65–70. IEEE, 2009.
2. R. Azevedo, E. Dantas, F. Freitas, C. Rodrigues, MJ Almeida, W. Veras, and R. Santosyi. An autonomic ontology-based multiagent system for intrusion detection in computing environments. *International Journal for Infonomics (IJI)*, 3:182–189, 2011.
3. Chang Choi, Junho Choi, and Pankoo Kim. Ontology-based access control model for security policy reasoning in cloud computing. *The Journal of Supercomputing*, 67(3):711–722, 2013.
4. H. T. Elshoush and I. Mohamed Osman. Alert correlation in collaborative intelligent intrusion detection systems a survey. *Applied Soft Computing*, 11:4349–4365, 2011.

5. Jian-bo Gao, Bao-wen Zhang, Xiao-hua Chen, and Zheng Luo. Ontology-based model of network and computer attacks for security assessment. *Journal of Shanghai Jiaotong University (Science)*, 18(5):554–562, 2013.
6. Seyed Ali Mirheidari, Sajjad Arshad, and Rasool Jalili. Alert correlation algorithms: A survey and taxonomy. In *Cyberspace Safety and Security*, volume 8300 of *Lecture Notes in Computer Science*, pages 183–197. 2013.
7. S. More, M. Matthews, A. Joshi, and T. Finin. A knowledge-based approach to intrusion detection modeling. In *2012 IEEE Symposium on Security and Privacy Workshops (SPW)*, pages 75–81. IEEE, 2012.
8. B. Morin, L. Mé, H. Debar, and M. Ducassé. A logic-based model to support alert correlation in intrusion detection. *Information Fusion*, 10(4):285–299, 2009.
9. B. Motik, P.F. Patel-Schneider, B. Parsia, C. Bock, A. Fokoue, P. Haase, R. Hoekstra, I. Horrocks, A. Ruttenberg, U. Sattler, et al. Owl 2 web ontology language: Structural specification and functional-style syntax. *W3C recommendation*, 27:17, 2009.
10. A. Sadighian, S. T. Zargar, J. M. Fernandez, and A. Lemay. Semantic-based context-aware alert fusion for distributed intrusion detection systems. In *2013 International Conference on Risks and Security of Internet and Systems (CRiSIS)*, pages 1–6, Oct 2013.
11. Ali Shiravi, Hadi Shiravi, Mahbod Tavallaee, and Ali A. Ghorbani. Toward developing a systematic approach to generate benchmark datasets for intrusion detection. *Computers and Security*, 31(3):357–374, 2012.
12. S. Staab and R. Studer. *Handbook on ontologies*. Springer, 2009.
13. J. Strassner. Knowledge engineering using ontologies. *Handbook of Network and System Administration*, 4, 2008.
14. Chenghua Tang, Lina Wang, Shensheng Tang, Baohua Qiang, and Jilong Tian. Semantic description and verification of security policy based on ontology. *Wuhan University Journal of Natural Sciences*, 19(5):385–392, 2014.
15. Elvis Tombini, Herve Debar, Ludovic Me, and Mireille Ducasse. A serial combination of anomaly and misuse idses applied to http traffic. In *20th Annual Computer Security Applications Conference*, pages 428–437. IEEE, 2004.
16. J. Undercoffer, J. Pinkston, A. Joshi, and T. Finin. A target-centric ontology for intrusion detection. In *18th International Joint Conference on Artificial Intelligence*, pages 9–15, 2004.
17. Ju An Wang and Minzhe Guo. Ovm: an ontology for vulnerability management. In *Proceedings of the 5th Annual Workshop on Cyber Security and Information Intelligence Research: Cyber Security and Information Intelligence Challenges and Strategies*, CSIIRW '09, pages 34:1–34:4, New York, NY, USA, 2009. ACM.
18. Ping Wang, Kuo-Ming Chao, Chi-Chun Lo, and Yu-Shih Wang. Using ontologies to perform threat analysis and develop defensive strategies for mobile security. *Information Technology and Management*, pages 1–25, 2015.
19. Safa Yahi, Salem Benferhat, and Tayeb Kenaza. Conflicts handling in cooperative intrusion detection: a description logic approach. In *22nd IEEE International Conference on Tools with Artificial Intelligence (ICTAI)*, volume 2, pages 360–362. IEEE, 2010.
20. C. Zhou, C. Leckie, and S. Karunasekera. A survey of coordinated attacks and collaborative intrusion detection. *Computers and Security*, 29:124–140, 2010.