Toward an Efficient Ontology-Based Event Correlation in SIEM

Article *in* Procedia Computer Science · December 2016

DOI: 10.1016/j.procs.2016.04.109

2 authors:

Tayeb Kenaza
Ecole Militaire Polytechnique
23 PUBLICATIONS **45** CITATIONS
SEE PROFILE

Mahdi Aiash
Middlesex University, UK
62 PUBLICATIONS **338** CITATIONS
SEE PROFILE

Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 83 (2016) 139 – 146

Procedia Computer Science

The 7th International Conference on Ambient Systems, Networks and Technologies (ANT 2016)

Toward an Efficient Ontology-based Event Correlation in SIEM

Tayeb Kenaza[a], Mahdi Aiash[b]

[a]*Ecole militaire polytechnique, BP-17, Bordj El-Bahri, 16111, Alger, Algérie*
[b]*School of Science and Technologies, Middlsex Universiry, UK*

Abstract

Cooperative intrusion detection use several intrusion detection systems (IDS) and analyzers in order to build a reliable overview of the monitored system trough a central security information and event management system (SIEM). In such environment, the definition of a shared vocabulary describing the exchanged information between tools is prominent. Since these pieces of information are structured, we propose in this paper to use an ontological representation based on Description Logics (DLs) which is a powerful tool for knowledge representation. Moreover, DLs are able to ensure a decidable reasoning. An alert correlation prototype is presented using this ontology, and an illustrative attack scenario is carried out to show the usefulness of the proposed ontology.

Peer-review under responsibility of the Conference Program Chairs
Keywords: Intrusion detection, Alert correlation, Description logics, Ontology, OWL.

1. Introduction

Information systems security is a sensitive issue which requires the deployment of several security mechanisms and tools. We generally use prevention systems such as authentication, access control, firewalls, etc. 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, some of these systems focus on the protection against external attacks, while the majority of attacks are internal. Thus, the use of prevention systems only is not enough, hence a second layer of security is necessary, such as the intrusion detection. Unfortunately, IDSs are still imperfect for two reasons. First, they generate a very large number of low-level alerts, where most of them are false positive which is alerts generated in the absence of attacks. And second, they 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[21,4], which allows various intrusion detection tools to cooperate. The objective of such cooperation can be achieved

* Corresponding author. Tel.: +0-000-000-0000 ; fax: +0-000-000-0000.
E-mail address: ken.tayeb@gmail.com

Peer-review under responsibility of the Conference Program Chairs
doi:10.1016/j.procs.2016.04.109

through different detection approaches, such as misuse detection and anomaly detection which are complementary. One can also use several IDS based on the same approach, for example the misuse one, but with different rule bases. In addition to IDS, other analyzers should be considered in the cooperative intrusion detection such as network and vulnerability scanners in order to correlate alerts with contextual information, 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) .

In this case, the definition of a shared vocabulary to describe exchanged information is a major concern. In general this information is structured and is given in XML. For instance, this is the case of alerts in IDMEF (for Intrusion Detection Message Exchange Format)[1] and TAXII (Trusted Automated eXchange of Indicator Information)[2], as well as the vulnerabilities in OVAL (Open Vulnerability and Assessment Language)[3] and STIX (Structured Threat Information eXpression)[4]. However, information is generally based on different taxonomies, and given in XML which is limited to a syntactic representation. Given that XML representation is devoid of semantics, it is more beneficial to change from taxonomies to ontology specification languages [12,9], which are able to simultaneously serve as recognition, reporting and correlation languages.

Ontology specification languages such as OWL[5] and DAML+OIL[6] use a fragment of the first order logic, namely Description Logics (DLs for short). Indeed, DLs are convenient to represent structured information. They are decidable in the sense that reasoning can be achieved in a finite time. Also, a number of sophisticated DL-based reasoners have been developed such as Pellet [11] and FaCT ++ [16].

Based on several existing knowledge representation models used in SIEM such as works done in [8,1,2,7], our contribution in this paper is, on one hand, to enhance existing representations by regrouping a large amount of information into a domain ontology. This will offer a comprehensive and extensible knowledge representation which can be used in many event correlation systems.

On the other hand, given that tools used in SIEM are not totally reliable, usually conflicts appear between them [15,20]. For example, one can easily see that IDSs are not fully reliable since they generate many false positives and false negatives. Thus, 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 number of alerts, in particular false positives.

The rest of this paper is organized as follows. Section 2 presents a background on alert correlation. Section 3 briefly recall some works of knowledge representation used in intrusion detection and then presents the proposed ontology. Section 4 presents an architecture of an alert correlation system based on DLs reasoning with an illustrative experiment. In section 5 some related works are briefly discussed. Section 6 concludes this paper.

2. Background

The role of the intrusion detection is to monitor events that occur in computers or networks and to analyze them in order to discover signs of intrusions. These events are often defined as attempts to violate the security policy. Intrusions have several causes such as malware (e.g, Virus, Trojan, etc.), external attackers that access the information system via open networks such as Internet, unauthorized users that try to gain unauthorized privileges or users that abuse of their privileges [10].

Nowadays, IDSs play an important role in computer security. However, the large deployment of IDS in operational environments in the last two decades has showed their weaknesses. Their main problem lies mainly in the excess of reported alerts. The security operator is often quickly overwhelmed by the amount of alerts. Hence, he/she only examines alerts from time to time, which may cause missing of some critical attacks. In fact, the use of IDS become similar to the use of surveillance cameras which are viewed only when a problem occurs.

[1] http://www.ietf.org/rfc/rfc4765.txt
[2] https://taxii.mitre.org
[3] http://oval.mitre.org/
[4] https://stix.mitre.org
[5] http://www.w3.org/2004/OWL/
[6] http://www.daml.org/2001/03/daml+oil-index

Another weakness is the poor semantic of the reported alerts. The security operator generally cannot determine the severity of an alert without resorting to a manual analysis of the events that caused the alert. In addition, intrusions detection tools are faced to the problem of false negative, which is the absence of alerts in the presence of attacks.

To overcome these problems, it is essential to cooperate several security tools to build reliable capability of coordination and correlation. There are several key functions in the alert correlation process.

- Removing redundancy: one basic function is to determine whether two alerts have been generated according to the observation of the same event. Removing alert redundancy reduces the number of alerts to be processed.
- Aggregating alerts: some attacks cause more than one elementary event. Thus, the combination of elementary events reduces the flow of alerts.
- Merging alerts: after grouping alerts into clusters, an advanced function of correlation will be to produce global alerts summarizing the malicious activity reported by these groups of alerts.
- Recognizing attack scenarios (context correlation): this function is more advanced and requires more complex mechanisms to determine certain type of attacks which are carried out in several stages. Attacks are best understood as scenarios than individually.

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. The resulted data formalisms (e.g IDMEF, TAXII, STIX, etc.) has provided a space 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 contexts in which they occur. However, these approaches 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]. A part of our work in this paper can be seen as an extension of the M4D4.

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[7]. 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.

[7] `http://nvd.nist.gov`

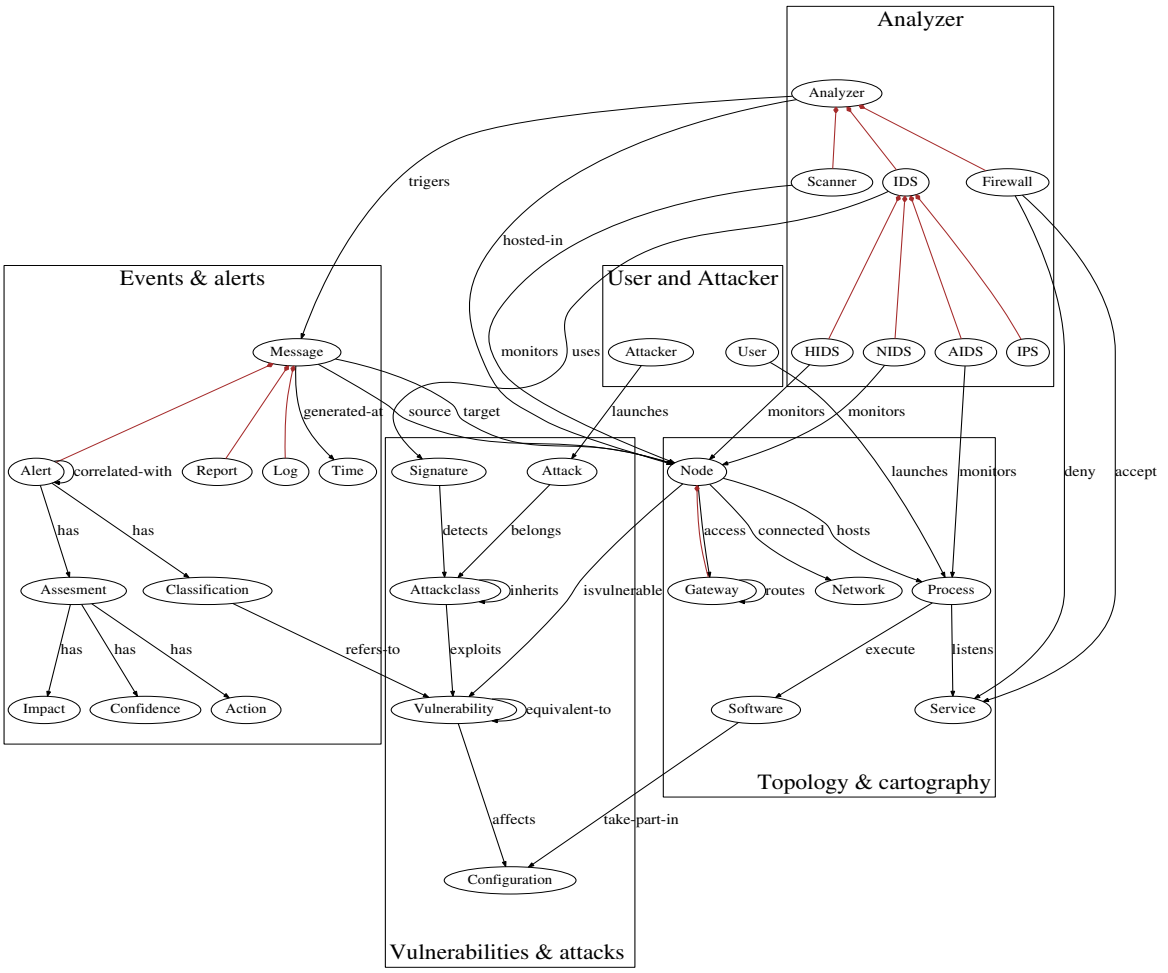

Fig. 1. A domain ontology for intrusion detection.

- Analyzers : this category contains information about several kind of security tools that can be used to protect an information system, such as IDS, Network Mapper, Vulnerability Scanner, Firewall, Integrity checker, Anti-malware, etc.
- Events and alerts : whatever the kind of the used security tools, they trigger a message when an event occur or at least a heartbeat message to report their activities. There are several kind of message such as alerts sent by detection tools (e.g. IDS), reports sent by scanner tools (e.g. vulnerabilities scanner), Logs sent by applications and devices (e.g. Firewall or Routers), etc. In general these messages are well structured and can be provided in XML.
- Attacks and Vulnerabilities : vulnerabilities refer to security flaws in softwares that can be used by an adversary to attack the information system. They can be also related to human errors and mistakes. In general, a vulnerability affects a product, have some consequences if it is successfully exploited, and some countermeasures may be applied to avoid it.
- Contextual information (topology and cartography) : this category involves information about the circumstance in which an attack is attempted, such as the configuration of the targeted host or product, the network topology, etc.
- Users and Attackers : this category concerns information about the users profiles and attacker intentions.

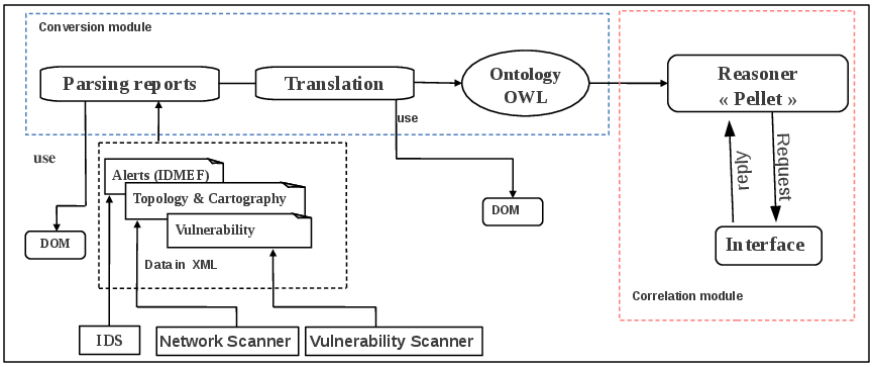

Fig. 2. Ontology based alert correlation architecture.

4. Ontology based event correlation

The use of the proposed ontology 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 event 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 some 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.

In this work, we build our ontology using the API Jena[8], and the reasoning is provided by Pellet[9] which is a full OWL-DL reasoner.

4.1. Populating our Ontology

To populate our ontology we need to use several tools. Information about hosts and network topology are given using Nmap[10]. This tool can provide many information such as the running hosts and their operating systems, the different softwares listening in these hosts with their corresponding version, and many further information. Information about the vulnerabilities of systems and softwares are given using Nessus[11]. Information about attacks are given in real time by IDS, in our system we used Snort[12] with a set of VRT and community rules. Note that it is also possible to insert directly information into the ontology by the security operators.

4.2. Reasoning with our 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 to derive implicit relations. For example, we defined a new concept *Plaussible_attack* as an alert that satisfies the following conditions: 1) the alert is generated by

[8] http://jena.apache.org/
[9] http://clarkparsia.com/pellet/
[10] http://nmap.org/
[11] http://www.tenable.com/products/nessus/
[12] http://www.snort.org/

an analyzer that actually monitors the target machine, and 2) the machine is actually vulnerable to the attack reported in the alert.

Formally, given an alert generated by an analyzer Z and reports a vulnerability V, a *Plaussible_attack* is defined as follows.

$$\begin{array}{l} Plaussible\_attack \subseteq \\ \hline \forall\ reported\text{-}by.\{Z\} \sqcap \forall\ has\text{-}classification.(\forall\ refers\text{-}to.\{V\}) \sqcap \forall\ has\text{-}target.(\forall\ isvulnerable.\{V\} \sqcap \\ \forall\ monitored\text{-}by.\{Z\}) \end{array} \quad (1)$$

Where,
monitored-by is the inverse of *monitors*, $monitored\text{-}by \equiv (monitors)^{-1}$.
reported-by is the inverse *trigers*, $reported\text{-}by \equiv (trigers)^{-1}$.
monitors is a relationship between *Node* and *Analyzer*, $monitors \equiv hosted\text{-}in \sqcap connected \sqcap netNodes$.
netNodes is the inverse of *connected*, $netNodes \equiv (connected)^{-1}$.

We also defined the concept $False\_alert$ which is the negation of the relationship *Plaussible_attack*. This concerns: 1) an alert generated by an analyzer that does not actually monitor the target of the attack, or 2) an alert where the target is not actually affected by the vulnerability reported in the alert. Formally, $False\_alert$ is defined as follows.

$$\begin{array}{l} False\_alert \subseteq \\ \hline (\forall\ has\text{-}classification.(\forall\ refers\text{-}to.\{V\}) \sqcap \forall\ has\text{-}target.(\forall\ isnot\text{-}vulnerable.\{V\})) \sqcup \\ (\forall\ reported\text{-}by.\{Z\} \sqcap \forall\ has\text{-}target.(\forall\ isnot\text{-}monitored\text{-}by.\{Z\})) \end{array} \quad (2)$$

These two new concepts will be used in the experiment of the next subsection. Note that many other inferred concepts can be proposed to improve our correlation system.

4.3. Illustrative experiment

This experiment consists on lunching a set of attacks against a linux vulnerable machine run on our simulation platform[13]. Here, we have used Metaspoitable[14] as a victim and the metasploit framework[15] as an attacker. Then, we process reported alerts during the experiment using our correlation system prototype. As shown in Figure 2, our architecture needs information from many analyzers, namely IDS, network scanner and vulnerability scanner. Snort has generated 13 alerts during the experiment. Reported alerts are given in Table 1.

We need also information about vulnerabilities, cartography and topology of network. To obtain this information, we have first scanned the target machine using Nmap, which has detected all services running in this node, namely ftp, ssh, telnet, smtp, domain, http, rpcbind, netbios-ssn, microsoft-ds, exec, login, shell, rmiregistry, ingreslock, nfs, ccproxy-ftp, mysql, postgresql, vnc, X11, and irc. Then, this victim machine was scanned by Nessus, which has reported 160 vulnerabilities. Note that we have tried all exploits in the Metasploit framework that targeted a linux machine, namely 44 exploits lunched against the services cited bellow.

Reports from IDSs and scanners are processed and translated into the ontology. Then, we launch the reasoner (Pellet) to infer the new concepts $False\_alert$ and *Plausible_attack*, and results are given in Table 2.

In this experiment, our system has correctly classified reported alerts as plausible attack or false alert, depending on: 1) if the victim machine is actually affected or not by the vulnerability referenced in the alert, or 2) if the victim machine is actually monitored by the IDS. Unfortunately, our system is not able to take a decision when information is missing. For example, the alert "NETBIOS SMB-DS IPC$ share access" do not refer to any vulnerability, so our system is not able to classify it. A solution to this problem would be to improve snort signatures by completing missed vulnerability references.

[13] for the lack of space, this platform is not presented in this paper.
[14] http://sourceforge.net/projects/metasploitable/. Metasploitable is an intentionally vulnerable Linux virtual machine. This VM can be used to conduct security training, test security tools, and practice common penetration testing techniques.
[15] http://www.metasploit.com/. Metasploit is an open source penetration test framework.

Table 1. Alerts detected by snort. The last column contains the decision of our correlation system.

Count	Alert message	CVE reference	Nessus scan	Correlation
1	(portscan) TCP Portscan	no ref	not nulnerable	Not classified
1	SNMP AgentX/tcp request	cve,2002-0012, cve,2002-0013	not nulnerable	False alert
3	COMMUNITY SIP TCP/IP message flooding directed to SIP proxy	no ref	not vulnerable	Not classified
1	SNMP request tcp	cve,2002-0012, cve,2002-0013	not vulnerable	False alert
1	WEB-PHP piranha passwd.php3 access	cve,2000-0322	not vulnerable	False alert
3	NETBIOS SMB-DS IPC$ share access	no ref	not vulnerable	Not classified
1	COMMUNITY WEB-CGI Twiki shell command execution	cve,2005-2877	vulnerable	Plausible Attack
1	SERVER-WEBAPP PHP-CGI remote file include attempt	cve,2012-1823	vulnerable	Plausible Attack
1	MALWARE-BACKDOOR UnrealIRCd backdoor command execution attempt	cve:2010-2075	vulnerable	Plausible Attack

Table 2. Alert correlation results.

False alert	Plausible attack	Not classified
3	3	7

5. RELATED WORKS

The automatic correlation of information from different security systems has been a vivid topic of research for over a decade[21,4]. 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.

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. Note 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.[17]. 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[18], 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[19] 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.

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 two new important concepts are inferred from the constructed ontology, the concept of *plausible_attack* and the concept of $false\_alert$. These two concepts are very important to reduce the amount of alerts by analyzing in priority *plausible_attack*, and by discarding $false\_alert$. 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. Moreover, we are now working to perform a more consistent experiment with more realistic and complex scenarios.

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. K. Scarfone and P. Mell. Guide to intrusion detection and prevention systems (idps). Technical report, National Institute of Standards and Technology, Special Publication 800-94, USA, 2007.
11. Evren Sirin, Bijan Parsia, Bernardo Cuenca Grau, Aditya Kalyanpur, and Yarden Katz. Pellet: A practical owl-dl reasoner. *Web Semantics: science, services and agents on the World Wide Web*, 5(2):51–53, 2007.
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. D. Tsarkov and I. Horrocks. FaCT++ description logic reasoner: System description. In *Proc. of the Int. Joint Conf. on Automated Reasoning (IJCAR 2006)*, volume 4130 of *Lecture Notes in Artificial Intelligence*, pages 292–297. Springer, 2006.
17. 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.
18. 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.
19. 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.
20. 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.
21. C. Zhou, C. Leckie, and S. Karunasekera. A survey of coordinated attacks and collaborative intrusion detection. *Computers and Security*, 29:124–140, 2010.