An Ontology-Based Model for SIEM Environments

Gustavo Gonzalez Granadillo, Yosra Ben Mustapha, Nabil Hachem, and Herve Debar

Telecom Sudparis, SAMOVAR UMR 5157
9 rue Charles Fourier, 91011 EVRY, France
{G.Granadillo,Y.Mustapha,N.Hachem,H.Debar}@it-sudparis.eu

Abstract. The management of security events, from the analysis of attacks and risk to the selection of appropriate countermeasures, has become a major concern for security analysts and IT administrators. Furthermore, network and system devices are designed to be heterogeneous, with different characteristics and functionalities that increase the difficulty of these tasks. This paper introduces an ontology-driven approach to address the aforementioned problems. The proposed model takes into account the two main aspects of this field, the information that is manipulated by SIEM environments and the operations that are applied to this information, in order to reach the desired goals. We present a case study on Botnets to illustrate the utilization of our model.

Keywords: SIEM, Ontology, Data Model.

1 Introduction

A Security Information and Event Management (SIEM) system[1] is an integrated, information security oriented platform offering the following services:

- Log management (log collection, storage, organization and retrieval)
- IT regulatory compliance (audit, validation or violation identification)
- Event correlation (normalization, fusion, verification, analysis)
- Active response (decision analysis, counter-measure response, prioritization)
- Endpoint security (monitoring, updating, configuration)

SIEM platforms need to acquire high volumes of information from heterogeneous sources and manipulate them on the fly. SIEM deployments thus focus on writing ad-hoc collectors and translators to acquire information and normalize it, and on writing correlation rules to manipulate the normalized information. This operational focus leads SIEM implementers to operate on information syntax rather than semantic, and to use feature-poor operations (counts, and sequences) in their correlation languages.

This paper addresses the previous issues by proposing a system solution for modeling SIEM-related data structures and operations to ensure interoperability. It abstracts the most important concepts from pre-existing formats (i.e.,

H. Jahankhani et al. (Eds.): ICGS3/e-Democracy 2011, LNICST 99, pp. 148–155, 2012.

IDMEF[RFC4765], IODEF[RFC5070], Syslog[RFC5424], M4D4[2], ...), to focus on information *semantics* and define a clean formal model for reasoning and modeling purposes. This preserves the capability of interacting with these formats while relaxing operational limitations that would artificially alter the data model. We thus specify an abstract model and will implement translations (on an as needed basis, using for example XSLT transformations) to push the concepts developed to operational environments at a later stage.

The proposed solution provides a general framework that formally models SIEM information and operations. It is developed using Protégé-OWL 4.1 beta platform, which incorporates the advantages and characteristics of the OWL2 language [3] into a flexible architecture. The remainder of the paper is structured as follows: section 2 briefly describes ontologies and the relevant state of the art; section 3 describes our Security Information Model; section 4 presents a use case of the proposed model; finally, some conclusions and future work are given in section 5.

2 Ontology

An Ontology can be seen as a mechanism to define the knowledge associated to a specific subject in a way that it can be interpreted by machines and shared by scientists and researchers. Some authors [4], [5] agree on the fact that Ontologies allow the reuse of knowledge from a specific domain, separate domain knowledge from operational knowledge and make inferences on the data. Furthermore, they can be used to find the right data at the right time while dealing with several information models; it is possible to use them in real environments (a use case is provided in section 4); and they enable the extensibility of attack and signature languages to be used among heterogeneous systems.

2.1 Elements

Ontologies are generally composed of instances, classes and properties. Instances are representations of individuals or objects in the domain of interest (e.g. Botnet, Network, Probe, etc). Classes generally group two or more instances according to common characteristics (e.g. the class Attack has Botnet, Phishing and Trojan horse as instances). Properties are used to describe features and/or attributes to link instances (e.g. directed_to, results_in, etc). They are shared by classes in order to give to the inheriting class (subclass) a more restrictive definition than the one provided by the ascendant class (superclass).

2.2 Related Work

Ontologies have been used in many disciplines such as: artificial intelligence, semantic web, systems engineering, biomedical informatics, information architecture, network security, and many others. Lopez et al. in [4], propose, for instance, the use of Ontologies to share alerts among SIM systems. As a result

the Ontology enables a mapping from the format message to the corresponding ontology instances, making possible the translation of the information contained on each message into an instance of alert.

Similarly, Cuppens et al. in [6], propose an Ontology framework to react to network attacks by using format languages such as IDMEF and OrBAC. As a consequence, it is possible to know which context rules are to be active as a reaction for a given attack. Other authors [7], introduce the use of Ontologies to detect and counteract to computer attacks, which enables the system to send suitable alarms so that appropriate reactions are implemented. Furthermore, Razzaq et al.[8] present an Ontology solution against web application attacks. As a result, violation of rules is efficiently undertaken, information retrieval is well performed and malicious inputs are correlated intelligently.

It is important to mention that some concepts and relationships from the aforementioned ontologies have been considered in the development of our information model. However, we designed a novel ontology-basis model that introduces more elaborated relations among concepts and that can be implemented as an abstraction of several format languages.

3 Proposed Security Information Model

Our solution provides an ontological model composed of two concepts: the *Information Class*, which models all the necessary information regarding the system and network configuration, as well as the security logs and events; and the *Operation Class*, which models treatments that are needed to propose the security policies and countermeasures. The distinction between these two classes makes the SIEM efficient and more consistent.

The Operation and Information classes are primarily related by an *Analysis* property that consists of requesting convenient data to ensure its functionalities. The instantiated individual within the Operation class will eventually update or modify the available information throughout the *Decision* property.

3.1 Information Class Model

In our Ontology, the information class is composed of two subclasses: the *configuration class*, which includes all the system and network information (such as protocols, machines, services, users and operating system); and the *security class*, which includes all the information regarding events, vulnerabilities, signatures, impact and security policies (figure 1).

The *event class* models abnormal or suspicious situations that need to be analysed by a security device since it can be considered as an attack or as an unusual activity performed by an authorised user. Its definition and classification is essential for the appropriate performance of the SIEM model, which eliminates ambiguities and misinterpretation of the processed data.

The *impact class* denotes the cost that a response action may have over a detected intrusion. It is modeled as TimeImpact and MonetaryImpact. The *policy class* defines all the information related to the security polices used to mitigate

the effects of an intrusion. It includes the *expectation class*, which conveys to the impact report the actions that the sender may request, such as: doing nothing; contacting source, target or sender; blocking port, network or host; etc.

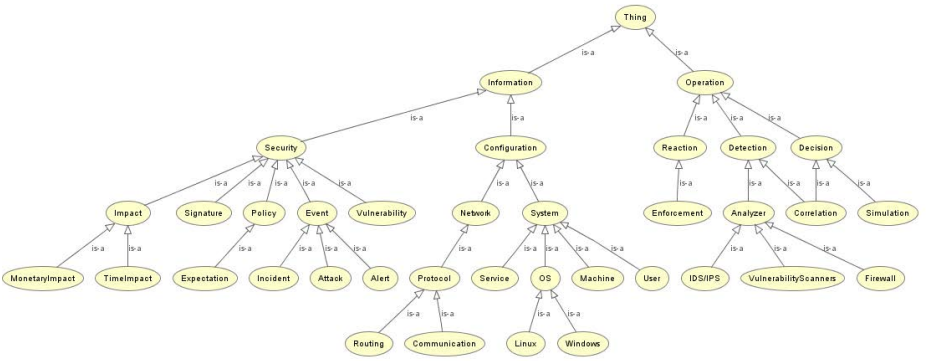

Fig. 1. Information Class Model

The *configuration class* has two subclasses: the network and system classes, which define a topology and a cartography model respectively, as proposed in [2]. The *network class* mainly focuses on the protocols (e.g., routing information protocols), necessary for the correlation process to calculate the path followed by a packet in the monitored domain. The *system class* describes a system involved in an event and it is categorized according to the role it played in the incident through the category attribute.

Some instances of our proposed Ontology may have multiple inheritances, meaning that one instance can inherit properties from more than one class. As a matter of fact, system or network instances (a server configuration, a network protocol, etc), which inherit all the properties from the configuration class, can also inherit properties from the vulnerability class, which is defined in the security information concept.

3.2 Operation Class Model

SIEM operations are generally composed of several activities and processes such as: detection, decision, and reaction (see figure 1).

Detection: This operation includes the subclass Analyzers, (i.e., IDS, vulnerability scanners, firewall, sensors, and other security devices), which generate a huge amount of heterogeneous events. Processing these events directly would be too complex and inefficient. Thus, an alert correlation engine, which essentially provides a higher level view of the occurring intrusion, will be important to explore.

Decision: This class contains two subclasses: *correlation* and *simulation*, whose primary goal consists on providing system support for a global view of network attacks by analyzing raw alerts in a simulated environment. As a result, the countermeasure impact is evaluated in terms of money, user/root access, damage, attack efficiency, etc., and an *impact report* is generated. This report can be a support for security administrators to evaluate and guarantee the respectfulness of security policies and to design or implement the counter-measures to react against defined attacks.

Reaction: This operation consists of an enforcement point that transforms on the first phase the selected countermeasures by the correlator and the evaluator from a high to a low specification level. In the second phase, this process enforces the configurations and rules on the different system and network components. These countermeasures include not only access control policies (i.e. blocking a user), but also management actions (i.e. taking down server, changing sensor configurations).

Table 1 summarizes the inputs, outputs, and goals of the different modules.

Table 1. Operation functions

Operations	**Input**	**Output**	**Goals**
Detection	Events related information	Events from heterogeneous sources	Efficient attack detection
Decision	Correlated data and countermeasures	Countermeasure costs and impact report	Optimal countermeasure decisions
Reaction	Countermeasures and information	Configurations and rules	Rules and configurations enforcement based on countermeasure decisions

4 Case Study

The first part of this section provides a general model for SIEM operations, while in the second section we explain the procedure to detect Botnet attacks and the actions to be taken as counter-measures.

4.1 General Model of SIEM Operations

We can model the SIEM utilization by using the following expressions, where all the elements that take place in the process are expressed between curly brackets ({}); the arrow (→) represents the definition of the global process; the addition process, expressed by the plus (+) symbol, refers to the activities that need to be performed in order to combine the different set of elements; and the equal symbol (=) denotes the resulting product of the process.

$$\mathbf{A} \rightarrow \{\mathbf{Config}\} + \{\mathbf{Information}\} = \{\mathbf{Events_{(1..n)}}\} \quad (1)$$

An analyzer (A) take a combination of a set of predefined system configuration parameters plus the information received, which generates, as a result, one or many events for a particular incident.

$$\mathbf{R/SIEM} \rightarrow \{\mathbf{Config_{(t)}}\} + \{\mathbf{Events}, \mathbf{Policies}\} = \{\mathbf{Config_{(t+1)}}\} \tag{2}$$

Similarly, a Reaction process (R) in a SIEM environment can be modeled as the addition of a system configuration at time t and the detection of an event or policy, which in turn produces a new configuration at time t+1.

$$\mathbf{C/SIEM} \rightarrow \{\mathbf{Alerts_{(1..n)}}\} + \{\mathbf{Config_{(t)}}\} = \{\mathbf{Policy_{(1..n)}}\} \tag{3}$$

The Correlation Process (C) in a SIEM environment takes into account one or more alerts along with the configuration information of the system at time t. As a result, one or many security policies can be generated.

The following equation can be proposed as a model to evaluate the impact of an attack:

$$\mathbf{Min_i} \rightarrow \{\mathbf{Config_{(i,usage)}}\} = \{\mathbf{Cost}\} \tag{4}$$

By dynamically evaluating attack impacts, the same attack is considered to have different impacts (i), depending on the pre-established system configuration and usage. All the resulting impacts are then quantified, meaning that their corresponding cost is calculated, the minimum (Min) of which is then proposed as the best security policy.

4.2 Botnet Use Case

Botnet attack life cycle generally consists of three main phases: Spreading/ Infection, Control, Usage [9].

Infection: During the Spreading/Infection phase, analyzers including HIDS (ex: antivirus, vulnerabilities scanners) and NIDS (ex Snort) generate events when detecting or suspecting any malicious actions. this is modeled in the following equation, where the system and network configuration are instances of the configuration class; and signature, vulnerabilities and rule sets (polices) are instances of the security class.

$$\begin{aligned} \mathbf{A_{Antivirus}} &\rightarrow \{\mathbf{SystemConfig}\} + \{\mathbf{Signature}\} = \{\mathbf{Events}\} \\ \mathbf{A_{V.Scanner}} &\rightarrow \{\mathbf{SystemConfig}\} + \{\mathbf{Vulnerabilities}\} = \{\mathbf{Events}\} \\ \mathbf{A_{Snort}} &\rightarrow \{\mathbf{NetworkConfig} + \mathbf{Traffic}\} + \{\mathbf{Rulesets}\} = \{\mathbf{Events}\} \end{aligned} \tag{5}$$

Control: Through the control phase, masters might use different ways to control their distributed bots. Analyzers on the network and system level might generate different events that might be useful for the detection of a Botnet attack and its corresponding C&C covert channel. These analyzers include Filters, Routers, Firewalls, Port scanners and others.

$$\mathbf{A_{portscanner}} \rightarrow \{\mathbf{SystemConfig}, \mathbf{Connections}\} + \{\mathbf{Portsets}\} = \{\mathbf{Events}\} \quad (6)$$
$$\mathbf{A_{Netfilters}} \rightarrow \quad \{\mathbf{NetworkConfig}, \mathbf{Traffic}\} + \{\mathbf{Rulesets}\} = \{\mathbf{Events}\}$$

Usage: The same case in the usage phase; Bots perform different services which can be categorized in DDoS attacks, Spamming and Spreading malwares, espionage and hosting malicious activities. Each category can be identified by different Analyzers.

$$\mathbf{A_{spamfilters}} \rightarrow \{\mathbf{NetworkConfig}, \mathbf{Traffic}\} + \{\mathbf{Rulesets}\} = \{\mathbf{Events}\} \quad (7)$$
$$\mathbf{A_{DDoS}} \rightarrow \{\mathbf{NetworkConfig}, \mathbf{Traffic}\} + \{\mathbf{Rulesets}\} = \{\mathbf{Events}\}$$

These analyzers might be able to take actions by themselves as for the firewall and the spam filtering situation.

$$\mathbf{R/SIEM_{firewall}} \rightarrow \{\mathbf{Config_{(t)}}\}_{\mathbf{IPopened}} + \{\mathbf{Alert}\} = \{\mathbf{Config_{(t+1)}}\}_{\mathbf{blockIP}} \quad (8)$$

Detection-Decision: These systems taken separately, cannot detect attacks or incidents in which networks and systems are involved, nor distinguish false positives and negatives. In order to detect this false rates and attacks, a correlation phase is needed to filter and aggregate the events (alerts, logs...) received from different analyzers. These events are then associated in the alert fusion process during the correlation. This phase will send countermeasures to the reactor or suggest different countermeasures (e.g. participation in taking down C&C server, blocking traffic, blocking port ...) to the simulation phase depending on the countermeasure's clarity and attack's severity.

$$\mathbf{C/SIEM} \rightarrow \{\mathbf{Events}\}_{\mathbf{Antiviruses,Snort,DDoS,...}} + \{\mathbf{Network}, \mathbf{SystemConfig}\} \quad (9)$$
$$= \{\mathbf{Policies}\}_{\mathbf{TakingdownC\&Cserver,Blockingtraffic,Port,Website...}}$$

The Simulation phase studies the impact of these countermeasures (policies). By collaboration with the correlation process, they can decide which actions to be taken and forward the best security policy(s) to the reactor (ex: Blocking access to a specific website, cleaning hosts).

$$\mathbf{Min} \rightarrow \{\mathbf{Config_{Takedownserver,Blockwebsite...}usage}\} = \{\mathbf{Cost}\} \quad (10)$$

Reaction: When the reaction process receives these countermeasures, it applies them on the system and network using different configurations and rules depending on the predefined configurations (Environment) and the corresponding data models. In this case, the reaction can be: blocking a website on the DNS server, forwarding the info to other network domains (in cooperative detection scenario) and cleaning the infected hosts.

$$\mathbf{R/SIEM} \rightarrow \{\mathbf{Config_{(t)}}\}_{\mathbf{WebsiteOpened}} + \{\mathbf{Policies}\}_{\mathbf{BlockWebsite}} \quad (11)$$
$$= \{\mathbf{Config_{(t+1)}}\}_{\mathbf{BlockingWebsite,BlockingDNSServer...}}$$

5 Conclusions and Future Works

In this document we have proposed a first attempt to define a global unified security information model to share information from heterogeneous sources in a SIEM infrastructure. Our proposal uses Ontologies as a shared vocabulary between elements and classes, which can ensure interoperability among the system components (i.e. services, machines, and users) and constant processes.

We defined two main classes: the information and operation classes, as well as the derived subclasses and the relationships among them. The information class describes the configuration and security information related to the different system and network devices; and the Operation class describes three main processes: Detection, Decision and Reaction, necessary to enable the detection of upcoming security threats and trigger appropriate mitigation actions.

An example on the applicability of the proposed model over a Botnet attack is provided at the end of the document, showing the functionality of the main operations that integrate the defined security information model. Future work will concentrate on the implementation of this architecture and the analysis of the obtained results.

Acknowledgements. The work in this paper has been sponsored by the EC Framework Programme as part of the ICT MASSIF project (grant no. 257644).

References

1. Miller, D., Harris, S., Harper, A., Van Dyke, S., Blask, C.: Security Information and Event Management (SIEM) Implementation. Mc Graw Hill (2010)
2. Morin, B., Me, L., Debar, H., Ducasse, M.: M4D4: A Logical Framework to Support Alert Correlation in Intrusion Detection. Information Fusion Internationale (2008)
3. Web Ontology Working Group: M4D4: OWL 2 Web Ontology Language, `http://www.w3.org/TR/owl2-overview/`
4. Lopez, J., Villagra, V., Holdago, P., De Frutos, E., Sanz, I.: A semantic web approach to share alerts among. Security Information Management Systems (2010)
5. Undercoffer, J., Joshi, A., Pinkston, J.: M4D4: Modeling Computer Attacks: An Ontology for Intrusion Detection. In: 6th International Symposium on Recent Advances in Intrusion Detection, pp. 113–135. Springer (2003)
6. Cuppens-Boulahia, N., Cuppens, F., Lopez, J., Vasquez, E., Guerra, J., Debar, H.: An ontology-based approach to react to network attacks. International Journal of Information and Computer Security 3, 280–305 (2009)
7. Abdoli, F., Kahani, M.: Ontology-based Distributed Intrusion Detection System. In: Proceedings of the 14th International CSI Computer Conference, pp. 65–70. IEEE (2009)
8. Razzaq, A., Hur, A., Ahmed, H., Haider, N.: Ontology based Application Level Intrusion Detection System by using Bayesian Filter (2009)
9. Hachem, N., Ben Mustapha, Y., Gonzalez Granadillo, G., Debar, H.: Botnets: lifecycle and taxonomy. In: 6th Conference on Network Architecture and Information Systems Security, IEEE (2011)