New types of Alert Correlation for Security Information and Event Management Systems

Gustavo Gonzalez Granadillo, Mohammed El-Barbori, and Hervé Debar
Institut Mines Telecom, Telecom SudParis
CNRS UMR 5157 SAMOVAR, Evry, France
{first_name.last_name}@telecom-sudparis.eu

***Abstract*—Current Security Information and Event Management systems (SIEMs) constitute the central platform of modern security operations centers. They gather events from multiple sensors (intrusion detection systems, anti-virus, firewalls, etc.), correlate these events, and deliver synthetic views of the alerts for threat handling and security reporting. However, as the number of security incidents, and thus the diversity of alerts received by SIEMs increases, the need for appropriate treatment of these alerts has become essential. Alert correlation has been proposed in order to alleviate this problem. Current alert correlation techniques provide a better description of the detected incident and a concise view of the generated alerts, reducing their volume and thus their processing time. Although such techniques support administrators in processing a huge number of alerts, they remain limited, since these solutions do not provide information about the attacker's behavior and the defender's capability in reacting to detected attacks. In this paper, we propose two novel alert correlation approaches. The first is based on policy enforcement and defender capability models; and the second is based on information security indicators. We therefore enrich the current state of the art in alert correlation techniques with complementary approaches.**

I. Introduction

Security Information and Event Management systems (SIEMs) are tools that provide real time analysis of security events generated by network devices and applications. These systems acquire high volumes of alerts coming from multiple and heterogeneous sources and process them on the fly. Their deployment focuses, firstly, on writing ad-hoc collectors and translators to acquire information and normalize it, and secondly, on writing correlation rules to aggregate the information and reduce the amount of data. This operational focus leads SIEM implementers to prioritize syntax over semantics, and to use correlation languages poor in features. However, as the number of security incidents, and thus the diversity of alerts received by SIEMs increases, the need for appropriate treatment of these alerts has become essential.

Managing and analyzing such tremendous number of alerts is a challenging task for security administrators. Alert correlation has been designed as an approach to alleviate this problem. There is a variety of techniques used for alert correlation. Most of them fall into one of the following categories:

- **Similarity-based** aims at defining patterns to search for similarities among alerts, and thus, merging them with the closest similarity values among their attributes (e.g., IP source, IP destination, port source, port destination, timestamp) within a given period of time [1], [2].
- **Knowledge-based** is based on the definition of prerequisites and possible occurring results, as well as, the knowledge of the attack path. We distinguish between rule-based alert correlation, which explores the causality links between generated alerts; and scenario-based alert correlation, which requires a prior knowledge of the attacker's behavior to correlate the alerts [3], [4].
- **Statistical-based** stores causal relationships among different security incidents in order to analyze their occurred frequencies in a given period of time using statistical data [5].
- **Model-based** aims at supporting alert correlation and analyzing security alerts with a view on relevant contextual and topological information of the security incident. Such approach intends to represent the different relationships between system entities and components to facilitate the correlation process by a cooperative analysis of heterogeneous information [6], [7].

Current alert correlation techniques provide more comprehensive descriptions of the detected security incident and thus a more concise view of the generated alerts. As such, the amount of alerts gets reduced which automatically reduces the incident processing time. However, such techniques do not generally consider important information e.g., the knowledge about the attacker's behavior, the enforcement functionalities, and the defense perimeter of the protected network (Firewalls, Proxies, Intrusion Detection Systems, etc).

We propose, in this paper, new complementary approaches on alert correlation techniques focus on the following aspects: (i) Correlation of raw security events stored in the SIEM file management; (ii) Security alert correlation based on third-party information (e.g., performance, intrusion, vulnerability, ISS compliance, financial value, etc.) to generate global alerts; and (iii) Alert aggregation integrated to the correlation in order to facilitate the exploitation task.

For this, it is important to improve the knowledge about the attacker and to identify the policy enforcement mechanisms that are capable to process generated alerts. The main contributions of this paper are summarized as follows:

1) Defense-based alert correlation: this approach focuses on policy enforcement points and defender capability models;
2) Metric-based alert correlation: this approach considers standard information security indicators to integrate a set of default correlation rules (called metrics) into a SIEM, in order to qualify its deployment and correct deviations

978-1-5090-2914-3/16/$31.00 ©2016 IEEE

accordingly;

The rest of the paper is structured as follows: Section II introduces the state of the art in alert correlation. Section III presents our approach about new types of alert correlation techniques. Section IV presents a case study to illustrate the applicability of our approaches. Related work are presented in Section V. Finally, conclusions and perspectives for future work are presented in Section VI.

II. Event Classification Models

In order to study the possibility of integrating new correlation rules in a SIEM system, we must first analyze the different classification models of security events. This section introduces the main features of the following models: FIRST, CAPEC, and ETSI.

A. FIRST Model

FIRST stands for the Forum of Incident Response and Security Teams, an organization that manages security information incidents and provides incident prevention programs. Working groups, more limited in size exist in the forum to discuss and propose solutions to issues of general interest. Among them, we find the FIRST Common Vulnerability Scoring System (CVSS) Working Group, which is responsible for maintaining and updating the assessment of the criticality of the system's vulnerabilities. FIRST also offers a classification of security incidents which is used to establish communications between the Computer Emergency Response Teams (CERTs) and Computer Security Incident Response Teams (CSIRTs) worldwide, and to provide accurate reporting to management on a regular basis.

The classification takes into account the category (i.e., DoS, forensics, compromised information, compromised asset, unlawful activity, internal hacking, external hacking, malware, email, consulting, policy violations); criticality (i.e., highly critical, critical, non-critical); and sensitivity (i.e., extremely sensitive, sensitive, not sensitive) of each CSIRT event that will be provided when creating an event. More details are given in [8].

B. CAPEC Model

The Common Attack Pattern Enumeration and Classification (CAPEC) offers a publicly available catalog of attack patterns with a comprehensive schema and classification taxonomy created to support the needs of developers, testers, and educators to build secure software and to assist in enhancing security throughout the software development lifecycle. The catalog has been designed to help developers understand the point of view of an attacker exploiting vulnerabilities of software in order to implement appropriate defenses based on that knowledge.

Attack patterns are defined as a mechanism to describe the way a given attack is executed. They provide a description of the context where it is applicable and then, unlike typical patterns, they provide recommendations to mitigate the attack. In short, an attack pattern is a blueprint for an exploit [9]. CAPEC proposes that attack patterns should include the following information: (name, classification, prerequisites, description, related vulnerabilities or weaknesses, method of attack, motivation and consequences, attacker skill or knowledge required, resources required, solutions and mitigations, context description, references). Examples of attack patterns are: HTTP response splitting, SQL injection, Session fixation, Phishing. More details about the CAPEC schema is given in [10].

C. ETSI Model

The European Telecommunications of Standards Institute (ETSI) proposes a classification model of global security events and its associated taxonomy (based on existing results and expert knowledge), covering both incidents and security breaches [11]. The objective is to build a full taxonomy to describe IT security events through a set of attributes (different for incidents and vulnerabilities), and thus, to present a representation that leverages the current international best practices and enables diversified and complex uses.

The ETSI model is meant to support operational security staff in their effort to qualify and categorize the detected security events, and more generally to IT security managers in their needs to establish a common language. This representation provides precise information on the types of events that are within the competence of the law, either for the benefit or detriment of the organization, and thus to obtain an estimate of possible legal risks that this organization faces. The main security incident categories are the following: intrusions and external attacks (IEX), malfunctions (IMF), deviant internal behavior (IDB), behavioral vulnerabilities (VBH), software vulnerabilities (VSW), configuration vulnerabilities (VCF), general security vulnerabilities (technical - VTC, and organizational - VOR).

The model proposes a list of incident indicators (e.g., External malicious incidents, Accidental or unwitting incidents, Incidents with Availability consequences, Incidents with a specific impact, Incidents depending on the kind of vulnerabilities exploited or on their status, etc.). Each indicator has associated controls and a level of detection (from 1 very difficult to 3 relatively easy). More details about the model are found in [11].

III. Proposed Alert Correlation Approaches

Considering the previous event classification models, and the fact that the combination of different alert correlation methods improves alert correlation, as stated in [12], we propose complementary approaches that tackle two main drawbacks of alert correlation techniques. First, the lack of information about the attacker's behavior and the detected attack; and second, the role of alert correlation in policy enforcement points. The remaining of this section defines a policy enforcement point and details each of these approaches.

Definition 1 (Policy Enforcement Point)**.** *A Policy enforcement point (PEP) is a security entity that responds to access requests by a number of decisions and security policies. PEPs, once deployed and configured, are responsible for applying rules to specific network flows. As trespassing responses, decisions and measures are generally against changes that must be made to the configuration of these PEPs. If alerts are correlated*

based on rules configured in each PEP, treatment will be more effective [13].

A. Enforcement-based

This correlation technique aims at deriving correlation rules from the policy enforcement points (PEP) configuration. The technique is based on the information system's defense by focusing on two main aspects: (i) the ability to classify all possible countermeasures, and (ii) the ability to identify relevant Policy Enforcement Points that could block the attack.

PEPs work as gatekeepers or front doors to digital resources. When a user tries to access a given resource (e.g., file) on a computer network or server, the PEP will describe the user's attributes to the Policy Decision Point (PDP), request a security decision (e.g., allow, deny access to the resource), and enforce the access decision. Examples of PEPs are Web servers, portals, legacy applications, firewalls, LDAP Directories, SOAP Engines (e.g., Apache AXIS), and similar resources.

The PEP is capable to apply, on the triplet Subject (s), Action (a), Object (o), the enforcement decisions represented by $\{d_1, d_2, ..., d_k\}$, k being the total number of decisions that a PEP can apply, as shown in Equation 1.

$$PEP : S \times A \times O \rightarrow \{d_k\}\ k \in [1...n] \quad (1)$$

The enforcement-based approach involves the following steps:

1. Study the policy enforcement point: In this step we need to study and analyze the PEP characteristics, including the attributes and syntax used for all configurations. Let us consider, for instance, A firewall (e.g., IPtables), which is composed of three types of chains i.e., INPUT, FORWARD, OUTPUT, to which a given policy is affected (e.g., DROP, ACCEPT, REJECT). The firewall analyzes different parameters from the inspected packets (e.g., IP source, IP destination, Port source, Port destination, Protocol), each parameter is associated to a given attribute while creating the security rule (e.g., -s, -d, –sport, –dport, -p).

2. Transform configurations in countermeasure models: In this step we need to express the retrieved information of all PEPs in a common format. Since each PEP class could have its own set of attributes, we normalize them by using a standard representation. The Intrusion Detection Message Exchange Format (IDMEF) [14] defines data formats and exchange procedures for sharing information of interest to intrusion detection and response systems and to the management systems that may need to interact with them.

In order to find a transformation that allows expressing the PEP configuration in the IDMEF format, we defined a matching table that compares the attributes and selectors of each PEP class with the IDMEF format. Table I gives some examples of the corresponding values between IDMEF parameters and IPtables attributes.

Note that in the IDMEF standard, the source and target classes provide information about the event sources and targets

TABLE I. CORRESPONDING VALUES BETWEEN IDMEF PARAMETERS AND IPTABLES ATTRIBUTES

IDMEF Parameters	Type	IPtables Attribute	Type
Category	enum	ipv4-addr/ipv6-addr	-
Location	string	GeoIPdb	string
Name	string	-src/-dst	string
Address	string	-s/-d	string
netmask	string	-s-add/-d-add	integer
UserId name	string	-uid-owner	string
port	integer	–sport/–dport	integer
protocol	string	-p	string

that generated the alert. Selectors are defined as variables used to normalize the identification of a given PEP attribute. A selector S is a vector composed of five elements (i.e., ip_src, s_port, ip_dst, d_port, p). An example of the selector comparisons for a firewall class (e.g., IPtables) is presented in Table II.

TABLE II. SELECTOR COMPARISON IPTABLES AND IDMEF

Selector	IPtables Attribute	IDMEF Attribute
ip_src	-s	source(0).node.address(0).address
s_port	–sport	source(0).service.port
ip_dst	-d	target(0).node.address(0).address
d_port	–dport	target(0).service.port
p	-p	target(0).service.iana_protocol.name

3. Derive correlation rules from the countermeasure models: The derivation of correlation rules allows generating a test on alerts from a PEP selector. For this, we need to define a comparison process that returns a boolean value {0, 1}. Only if the process returns 1, we generate a correlated alert in the IDMEF format, indicating that the received alert will be treated by an identified PEP and if it is possible to integrate the countermeasure (blocking rule) in the given field (additional_data) of the IDMEF message.

As a result, it is possible to generate correlated alerts that indicate the policy enforcement point(s) that are susceptible of being modified as a reaction strategy to the detected security incident. We defined three functions to perform our approach: (i) comparison(); (ii) generate_correlation_alert(); and (iii) derive_correlation_alert(). The remaining of this section details each function.

1) Function 1: Comparison: The comparison() function takes alert A and selector S as input. This function tries to compare the destination IP, destination port, and the protocol between A and S. The function returns 1 if there is a match and 0 otherwise.

Function 1: comparison()

```
1: Input {alert: A, selector: S}
2: Output {boolean values 0 or 1}
3: result = 0
4: if((A.target(0).node.address(0).address == S.ip_dst)
     and (A. target(0).service.port == S.d_port)
```

```
     and (A. target(0).service.iana_protocol == S.p))
   then result = 1
5: return result
```

2) Function 2: Generate Correlation Alert: The generate_correlation_alert() function takes alert A as input, and generates part of the correlation alert (CA) in the IDMEF format based on alert attributes. It integrates the defense rule in the "additional_data" fields of the IDMEF format.

Function 2: generate_correlation_alert()

```
1: Input {alert: A}
2: Output {correlation alert: CA}
3: CA.source(0).node.address(0).address = A.source(0).
      node.address(0)
   CA.target(0).node.address(0).address = A.target(0).
      node.address(0)
   CA.target(0).service.port = A.target(0).service.
      port
   CA.target(0).service.iana_protocol = A.target(0).
      service.iana_protocol
   rule = "iptables -A INPUT -s A.source(0).node.
   address(0) -d A.target(0).node.address(0) --dport
   A.target(0).service.port -p A.target(0).service.
   iana_protocol -m state -- state NEW ESTABLISHED
   -j DROP"
   CA.additional_data(0).type = string
   CA.additional_data(0).meaning = defense rule
   CA.additional_data(0).data = rule
4: return CA
```

3) Function 3: Derive Correlation Alert: The derive_correlation_alert() function takes alert A and a list of PEPs as input. It calls the comparison function to check if A corresponds to selector S of the current PEP. If the returned value is 1, then the function generate_correlation_alert() is called to generate the first part of the correlation alert (CA). The second part consist of integrating the ID of the identified PEP in the additional_data field of the CA. This function returns an empty IDMEF field if there is no PEP able to deal with the alert.

Function 3: derive_correlation_alert()

```
1: Input {alert: A, list of PEP: PEPs}
2: Output {correlation alert: CA}
3: CA = NULL
4: for PEP in PEPs:
      for S in PEP:
        if (comparison (A, S) == 1) then:
          CA = generate_correlation_alert (A)
          CA.additional_data(1).type = string
          CA.additional_data(1).meaning = ID of PEP
          CA.additional_data(1).data = PEP.id
          return CA
5: return CA
```

B. Metric-based

This approach aims at deriving correlation rules from information security indicators, such as those developed and standardized by the European Telecommunications Standards Institute group (ETSI), making it possible to determine the effectiveness of a deployed SIEM, by ensuring its relevance on the current market and by measuring the improvement of its performance.

The innovation of this approach consists on integrating a set of default correlation rules (called metrics) into a SIEM to qualify its deployment using a representative reference of the current state of the art. These correlation rule metrics produce comparable indicators to the ETSI standard. SIEM operators analyze these indicators. They validate the proper operation of SIEM and interpret the evolution of indicators over time to learn how the performance of a SIEM evolves. A significant variation of these metrics may indicate an improvement in the security device, a change in the operation of sensors, or a change in the way the threat evolves.

Based on the security event classification model and taxonomy, we consider the indicators for which a sensor can produce alerts to a SIEM (e.g., Intrusions and external attacks, Deviant external behaviors, Behavioral vulnerabilities, General Security Vulnerabilities). We discard the indicators related to the presence of vulnerabilities in software and configurations, as well as, all types of accidental (breakdown or natural disaster) or unintentional (human error) security incidents.

The following example rules show the derivation of security indicators from easily available sources of information. We provide an example for a file illicitly transferred between the organization and the outside world. If all derived indicators remain null, it means that we do not detect anything. Other more elaborated formulas must then be implemented to show significant deviations from these indicators.

The rule below is written using the following formalism:

- && refers to a logical AND;
- || refers to a logical OR;
- $\rightarrow$ means consequence or deduction;
- variables begin with _ ;
- Arithmetic and parenthesis have the usual meanings.

File illicitly transferred

```
((a1.classification.impact = ILLICIT FILE TRANSFER ||
  a1.classification.text = untrusted file download ||
  a1.classification.text = illicit exchange files ||
  a1.classification.text = unencrypted file shared ||
  a1.classification.text = data leak prevention
 )||
 a1.classification.cve = _cve &&
 nvd1.cve = _cve &&
 a1.target.file.fileshare.permission = denied &&
 a1.target.file.fileshare = succeeded
)&&
a1.source.user.userid IN (internal network)&&
a1.target.user.userid NOT IN (internal network)&&
a1.source.node.address IN (internal network)&&
a1.target.node.address NOT IN (internal network)
-> metric.vbh_ftrc +=1
```

The previous example increases the metric counter in case of data leakage. The metric verifies if the impact classification corresponds to a file illicitly transferred, or if the classification text of the alert corresponds to unauthorized actions of file sharing. We verify if there is a CVE associated to the alert. In such a case, we create a variable named cve, which will be compared to the classification of the National Vulnerability Database (NVD). In addition, we verify the following conditions: (i) a file, whose sharing permission has been denied, is successfully shared, (ii) the source user ID belongs to the

internal network, (iii) the target user ID belongs to an external network, (iv) the source IP address belongs to the internal network, and (v) the target IP address belongs to an external network. As a result, we are able to calculate an indicator for the case of file illicitly transferred.

The metric based correlation technique is of great interest in the following scenarios:

1) A SIEM is deployed to supervise a given Information System. A monitoring strategy is applied over the IS. The indicators are initialized by the correlated metric and then compared to the standards. If this comparison reports an important inadequate functioning, the monitoring strategy is reviewed.
2) The Information System evolves, the entry points for attackers change too. The correlation rule metrics allow to visualize these changes through the defined indicators.
3) The monitoring strategy is updated accordingly, thus the correlation metric produces new indicators. Here, we can evaluate deviations that could indicate anomalies in the system. It is then possible to validate that each update of the monitoring strategy improves such indicators (by evaluating if the new reaction strategy makes decrease the counter of a given rule metric), and thus the supervision effectiveness of the security for the Information System.

IV. Use Case

This section presents an attack scenario against a user application, using a Cross-Site Request Forgery (CSRF)[1] attack to create a new administrator account in a Phplist platform. Our simulation platform is composed of four zones: (1) External network, which simulates malicious and legitimate traffic coming from the Internet; (2) Internal network, that regroups local deivces and simulate legitimate and malicious traffic coming from the local network; (3) DMZ, composed of vulnerable servers, IDS and SIEM components; and (4) a Security Operation Center, that integrates a SIEM (i.e., Prelude) and all its components (e.g., Manager, Correlator, LML, Database, Logger), as shown in Figure 1.

The SIEMs from the DMZ and the internal network areas analyze the log messages and send alerts to the SOC in case of an anomaly detection. We have set up a firewall configuration file (iptables) for each firewall (i.e., FW_1, FW_2). The final goal is to integrate a new firewalling rule in the IPtables configuration file of the identified firewall able to deal with the attack against the Phplist platform. This rule is obtained from the field "additional_data" of the correlated alert.

A. Enforcement-based Correlation

The following approach has been considered for the enforcement-based alert correlation:

1. Study the policy enforcement point: The platform zones are linked by a Policy enforcement Point (i.e., IPtables firewall). We can use a set of options to build the chains and to implement the filtering rules. examples of these options are as follows:

- $-s$, source of the packet;
- $-d$, destination of the packet;
- $--sport$, source port of the packet;
- $--dport$, destination port of the packet;
- $-p$, protocol of the rule, (e.g., icmp, tcp, udp, all);
- $-i$, incoming network interface;
- $-o$, outgoing network interface;
- $-state$, connection state (e.g., NEW, RELATED);
- $-j$, decision (e.g., ACCEPT, DROP)

2. Transform configurations in countermeasure models: Based on the countermeasure model, we present the configuration of each firewall as a set of selectors :

```
PEPs = [FW1, FW2]
     (ip_src, s_port, ip_dst, d_port, p)
FW1={(*,*,192.168.50.128/25,80,tcp);(S1)
     (*,*,192.168.50.128/25,80,udp);(S2)
     (*,*,192.168.50.200,5000,tcp); (S3)
     (*,*,192.168.50.0/25,*,icmp);  (S4)
     (*,*,192.168.50.0/25,*,ftp);   (S5)
     (*,*,192.168.50.129,8000,tcp); (S6)
    }

FW2={(192.168.50.0/25,*,192.168.10.100,80,tcp); (S1)
     (192.168.50.0/25,*,192.168.10.100,80,tcp); (S2)
     (192.168.128.0/24,*,192.168.10.100,80,tcp);(S3)
     (192.168.50.130,*,192.168.10.100,80,tcp);  (S4)
    }
```

In the configuration file of the firewall FW_1, selector S3 corresponds to IPtables rule that accepts the network traffic coming to machine 192.168.50.200 via the port 5000 and TCP protocol. Let us suppose that we receive an alert A with non-empty IDMEF fields that corresponds to the following values:

```
A.source(0).node.address(0).address = 172.16.128.254
A.target(0).node.address(0).address = 192.168.50.200
A.target(0).service.port = 5000
A.target(0).service.iana_protocol.name = tcp
```

A defense technique consists on blocking the IP source of the attacking machine in case of detecting the attack towards this machine and the field additional_data of the correlated alert will indicate two values: (i) the defense rule to integrate in the configuration file of the firewall that allows blocking the communication from the attacking machine (172.16.128.254) to the victim machine (192.168.50.200); and (ii) the ID of the firewall in which the new defense rule must be integrated.

3. Derive correlation rules from the countermeasure models: We have chosen the SIEM Prelude to show the applicability of our approach. In order to integrate the correlation defense in the SIEM Prelude, we have first integrated a new package called "firewall.py" in prelude-correlator. This package contains the previous functions and other methods that allow (i) recovering the set of selectors of each firewall, (ii) identifying the right firewall capable of processing the received alert, and (iii) proposing a new blocking rule to be integrated in the configuration file of the identified firewall.

In addition, we have created a new plugin (i.e., IptablesPlugin), which is responsible of receiving alerts and calling

[1]https://www.owasp.org/index.php/Cross-Site_Request_Forgery_(CSRF)

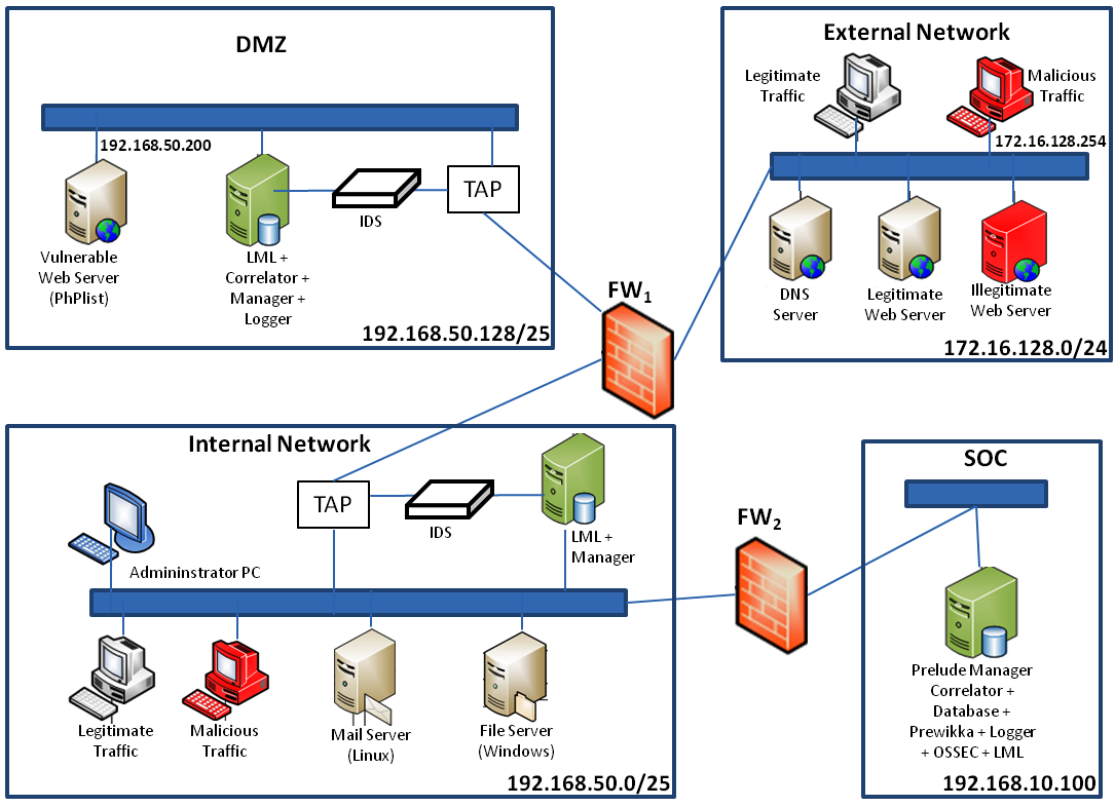

Fig. 1. Simulation Platform

the package "firewall.py". The plugin is configured so that it can be detected by prelude-correlator and consequently the generated correlation alert can be sent to prelude-manager.

An example of a derived correlation rule is given as follows:

```
If ((A.target(0).node.address(0).address==
     192.168.50.200) &&
   (A.target(0).service.port==5000) &&
   (A.target(0).service.iana_protocol.name==tcp))
-> CA.additional_data(0).type=string
   CA.additional_data(0).meaning=defense rule
   CA.additional_data(0).data='iptables -A INPUT
   -s 192.168.50.200 --dport 5000 -p tcp -m state
   --state NEW,ESTABLISHED -j DROP'
   CA.additional_data(1).type=string
   CA.additional_data(1).meaning=ID of PEP
   CA.additional_data(1).data=FW1
```

The previous firewall rule allows dropping all requests coming from the attacking machine (172.16.128.254) to the target machine (192.168.50.200) through destination port 5000. As a result, an error message appears, indicating an authentication issue, which makes it impossible to execute the CSRF attack in the Phplist application.

B. Metric-based Correlation

The analysis of the alert indicates a denial of service attack towards the machine with IP address 192.168.50.200. The corresponding metric is therefore updated with an increased value, as follows:

Denial of Service (DoS or DDoS)

```
(alert1.classification.impact = DOS ||
 alert1.classification.cve = _cve &&
 nvd1.cve = _cve &&(nvd1.cvss.score = A:P ||
 nvd1.cvss.score = A:C))&&
alert1.source.node.address NOT-IN
(internal network) -> metric.iex_dos +=1
```

In the previous indicator, we first check if the alert classification impact information is equal to DOS. If it is not the case, we verify if there is a CVE associated to the alert. In such a case, we create a variable named cve, which will be compared to the classification of the National Vulnerability Database (NVD). If they match, we check the availability impact of the incident. If this latter is either partial (P) or complete (C), then we verify that source IP address of the node does not belong to the internal network. If all previous criteria match, then we increase the metric counter. As a result, regardless of the configuration of the sensors, we will calculate the indicators, which will particularly characterize a denial of service.

For the previous scenario, the value of $metric.eix\_dos$ has increased by one. This value is then compared with the ETSI standard, which indicates a state of inadequate functioning by the system, therefore, the monitoring strategy must be reviewed. It is important to note that, as the monitoring strategy is updated, the correlation metric should produce new indicators. As a result, we can validate the effectiveness of the updated rule by analyzing the improvements of the indicators.

V. Related Work

Alert correlation is a process that integrates multiple components with the purpose of analyzing alerts in order to provide high-level insight view on the security state of the network surveillance [15]. Alert correlation aims at relating a set of alerts to build a more complete and bigger picture of the security incident, and therefore, it can be used to track the incident to its source.

A great variety of alert correlation techniques have been widely proposed to provide a higher level vision of the security incident. In [3], [4], for instance, authors propose alert correlation frameworks based on pre and post conditions of individual alerts. A main drawback of this technique is that new attacks are difficult to be matched with others since their prerequisites and consequences are not yet defined. Even for known attacks, it is not feasible to define all possible attributes in advance. In addition, such correlation approach requires extra knowledge about the defense capabilities deployed in the supervised system.

Other approaches, such as M2D2 [6], and M4D4 [7] represent the different relationships between system entities and components to facilitate the correlation process by a cooperative analysis of heterogeneous information. These correlation approaches, while useful in exploring additional information about the supervised network, present limited information about the attacker's behavior and the defender' capabilities that could be considered when correlating alerts.

In [16] authors propose a framework that reduces the number of processed alerts, discards irrelevant and false alerts, deals with unrelated alerts, and groups similar alerts. The intruders' intention is grouped into attack scenarios and thus used to detect future attacks. Such correlation approach, while useful to group alerts with high similarities within a given period of time, it is useless to evaluate the causality links between alerts, or to provide information about the origin of the incident.

VI. Conclusions

Alert correlation aims at relating a set of alerts to build a more complete and bigger picture of the security incident, and therefore, it can be used to track the incident to its source. Current approaches, while useful in providing a better understanding of the detected incident and a concise view of the generated alerts, they do not generally consider important information e.g., the knowledge about the attacker's behavior, the enforcement functionalities, and the defense perimeter of the protected network. Based on the aforementioned drawbacks, we propose new correlation techniques that can be used in complement with the current state of the art to enrich the alert information, in particular, about the attacker's behavior and the defender's capabilities.

The new alert correlation techniques have been integrated into a SIEM system (Prelude) and work as complementary approaches to traditional alert correlation. The first technique, Enforcement-based correlation, aims at classifying all possible countermeasures and their associated policy enforcement point that will implement the security rule as a defense mechanism. The second technique, Metric-based correlation, aims at deriving correlation rules from information security indicators, which allows for the analysis and evaluation of the SIEM effectiveness. As a result, it is possible to correlate alerts based on the policy enforcement point(s) responsible of implementing the security rule(s), which makes it possible to block attacks against web applications.

Acknowledgment

The research in this paper has received funding from the PIA Prelude NG project, with the support of the French National Agency for Information System Security and the French Ministry of Defense.

References

[1] S. A. Mirheidari, S. Arshad, and R. Jalili.: Alert Correlation Algorithms: A survey and taxonomy. In Cyberspace Safety and Security, (2013).

[2] Y. B. Mustapha, H. Debar, and G. Jacob.: Limitation of Honeypot/Honeynet Databases to Enhance Alert Correlation. In 6th International Conference on Mathematical Methods, Models and Architectures for Computer Network Security MMM-ACNS. Springer, (2012).

[3] F. Cuppens, F. Autrel, A. Miege, and S. Benferhat.: Correlation in an Intrusion Detection Process. In SEcurite des communications sur internet (SECI), pages 153-171, (2002).

[4] S. Cheung, U. Lindqvist, and M. Fong.: Modelling Multistep Cyber-Attacks for Scenario Recognition. In DARPA Information Survivability Conference and Exposition, pages 284-2921, (2003).

[5] J. Viinikka, H. Debar, L. Me, A. Lehikoinen, and M. Tarvainen.: Processing Intrusion Detection Alert Aggregates with Time Series Modelling. In Information Fusion, volume 10, pages 312-324, (2009).

[6] B. Morin, L. Me, H. Debar, and M. Ducasse.: M2D2: A Formal Data Model for IDS Alert Correlation. In 5th International Symposium on Recent Advances in Intrusion Detection (RAID), page 115-137, (2002).

[7] B. Morin, L. Me, H. Debar, and M. Ducasse.: M4D4: A Logical Framework to Support Alert Correlation in Intrusion Detection. In Information Fusion, page 285-299, (2009).

[8] D. Schieber and G. Reid.: CSIRT Case Classification (example for enterprise CSIRT). In Technical Report, FIRST, (2004).

[9] S. Barnum and A. Sethi.: Attack Patterns: Knowing your enemy in order to defeat them. In Technical Report. Black Hat DC, (2007).

[10] S. Barnum.: Common Attack Pattern Enumeration and Classification (CAPEC) Schema Description, Technical Report, Department of Homeland Security, (2008).

[11] ETSI.: Information security indicators (ISI event model) A Security Event Classification Model and Taxonomy. In Technical Report. European Telecommunications Standards Institute, (2013).

[12] F. J. Mora-Gimeno, F. Macia-Perez, I. Lorenzo-Fonseca, J. A. Gil-Martinez-Abarca, D. Marcos-Jorquera, and V. Gilart-Iglesias.: Security alert correlation using growing neural gas. In 4th International Conference on Computational Intelligence in Security for Information Systems, page 76-83. Springer-Verlag, (2011).

[13] Y. Ben Mustapha, H. Debar, G. Blanc,*Policy Enforcement Point Model*, 10th International Conference on Security and Privacy in Communication Networks, SecureComm, pp. 278–286, (2014)

[14] H. Debar, D. Curry, and B. Feinstein.: The intrusion detection message exchange format (IDMEF). RFC4765, Network Working Group, (2007).

[15] A. A. Ghorbani, W. Lu, and M. Tavallaee.: Network Intrusion Detection and Prevention: Concepts and Techniques. Springer, (2010).

[16] H. T. Elshoush and I. M. Osman.: Intrusion Alert Correlation Framework: An Innovative Approach. In IAENG Transactions on Engineering Technologies, pages 405-420, (2013).