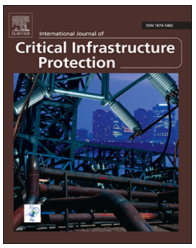
INTERNATIONAL JOURNAL OF CRITICAL INFRASTRUCTURE PROTECTION ▮ (▮▮▮▮) ▮▮▮–▮▮▮

Available online at www.sciencedirect.com

ScienceDirect

www.elsevier.com/locate/ijcip



A novel security information and event management system for enhancing cyber security in a hydroelectric dam

Cesario Di Sarno[a,*], *Alessia Garofalo*[b], *Ilaria Matteucci*[c], *Marco Vallini*[d]

[a]*Department of Engineering, University of Naples "Parthenope," Centro Direzionale Isola C4, 80143 Naples, Italy*
[b]*Computer Science Research Group (COSIRE), 81031 Aversa, Italy*
[c]*Institute for Informatics and Telematics, National Research Council (CNR), via Giuseppe Moruzzi 1, 56124 Pisa, Italy*
[d]*Department of Control and Computer Engineering, Polytechnic University of Turin, Corso Duca degli Abruzzi 24, 10129 Turin, Italy*

ARTICLE INFO

Article history:
Received 15 April 2015
Received in revised form
11 February 2016
Accepted 29 February 2016

Keywords:
Security information and event management (SIEM) Systems
Decision support systems
Resilient event storage
Hydroelectric dam

ABSTRACT

Security information and event management (SIEM) systems are increasingly used to cope with the security challenges involved in critical infrastructure protection. However, these systems have several limitations. This paper describes an enhanced security information and event management system that (i) resolves conflicts between security policies; (ii) discovers unauthorized network data paths and appropriately reconfigures network devices; and (iii) provides an intrusion- and fault-tolerant storage system that ensures the integrity and non-forgeability of stored events. The performance of the enhanced system is demonstrated using a case study involving a hydroelectric dam. The case study considers an attack model that affects portions of the information technology infrastructure of the hydroelectric dam and demonstrates that the security information and event management system is successfully able to detect and respond to attacks.

1. Introduction

The U.S. Department of Homeland Security [20] defines the critical infrastructure as "assets, systems and networks, whether physical or virtual, so vital to the United States that their incapacitation or destruction would have a debilitating effect on security, national economic security, national public health or safety, or any combination thereof." The protection of critical infrastructures is a priority to avoid disasters that could affect government, industry and society. President Obama's Presidential Policy Directive – Critical Infrastructure Security and Resilience (PPD-21) of 2013 [15] identifies 16 critical infrastructures that must be monitored and protected. The U.S. Industrial Control Systems Cyber Emergency Response Team (ICS-CERT) has noted that the energy sector, which includes hydroelectric dams, is one of the most attractive targets for cyber attacks. In 2013, the media reported that U.S. intelligence agencies traced a compromise of the U.S. Army Corps of Engineers National Inventory of Dams (NID) to Chinese government or military entities [8]. The compromised database stored vulnerabilities of major dams that could be exploited in future cyber attacks against the U.S. electric power grid.

*Corresponding author.
E-mail address: cesario.disarno@uniparthenope.it (C. Di Sarno).

http://dx.doi.org/10.1016/j.ijcip.2016.03.002
1874-5482/

Security information and event management (SIEM) systems are an emerging technology that can significantly enhance critical infrastructure protection. These systems are designed to analyze security information from the monitored infrastructures to discover security breaches. Existing SIEM systems lack several important features such as the abilities to detect and resolve conflicts between security policies, to identify and control network data paths existing in the monitored infrastructures and to securely store data while ensuring its integrity and non-forgeability.

This paper describes an enhanced SIEM system that overcomes these limitations by integrating a decision support system and a resilient event storage system. The enhanced system is customized for a specific critical infrastructure, namely a hydroelectric dam. An attack model that affects various portions of the information technology infrastructure of the hydroelectric dam is employed to demonstrate that the SIEM system can significantly enhance the cyber security of the monitored dam infrastructure.

2. SIEM systems

SIEM systems are widely used to perform real-time monitoring and control of critical infrastructure assets. A SIEM system integrates two formerly heterogeneous systems – a security information management (SIM) system and a security event management (SEM) system [3]. A security information management system focuses on the analysis of historical data to improve the long-term effectiveness and efficiency of cyber security mechanisms [21]. A security event management system, on the other hand, aggregates data into a manageable amount of information to enable the rapid handling of security incidents [21].

SIEM technology aggregates event data produced by security devices, network infrastructures and information technology systems and applications. The data fed to a SIEM system comprise log entries generated by devices and components installed within the monitored infrastructure (e.g., routers, servers and applications). Several protocols (e.g., Syslog, SNMP and OPSEC) are available for transferring log entries from data sources to a SIEM system. If a device or component does not support such a protocol, then an "agent" is required to translate (or normalize) the log data to a format that is recognized by a SIEM system. Also, an agent may provide filtering functionality to prevent irrelevant data from being sent to a SIEM system, helping reduce network bandwidth, storage space and SIEM processing resources. The task of distinguishing useful data from irrelevant data in a SIEM application is an important, albeit challenging, task.

Each agent outputs events that contain relevant data. The events are sent to a correlator that performs complex security analysis using attack signatures. If an attack is detected, the correlator generates an alarm containing information about the security breach. The events and alarms are saved in a storage system. A Gartner report [14] provides an overview of SIEM technologies; two of the most widely used SIEM systems are OSSIM and Prelude.

SIEM systems have three principal weaknesses when used in critical infrastructure protection applications:

- Critical infrastructure protection invariably involves the implementation of multiple – and conflicting – security policies. However, while SIEM systems permit the definition of security policies, they often do not provide mechanisms for resolving policy conflicts.
  A search of the literature reveals that several researchers have proposed conflict resolution strategies and mechanisms. Matteucci et al. [12] have developed a conflict resolution strategy based on the prioritization of the most specific privacy policies customized for the e-health domain. Cuppens et al. [5] employ an OrBAC methodology to manage conflicts involving permissions and prohibitions. Lupu and Sloman [10] define and review policy conflicts, discuss precedence relationships that enable inconsistent policies to coexist and present a conflict analysis tool that is part of a role-based management framework. Syukur et al. [19] have investigated policy conflict resolution in pervasive environments using standard strategies such as role hierarchy overrides and obligation precedence. Masoumzadeh et al. [11] consider attributes related to subjects, objects and environments, grouping them under a unique context; a conflict resolution strategy is then used to prioritize authorization rules according to the specificity of the context as a whole. Dunlop et al. [7] present four strategies for solving conflicts based on the evaluation of the role of the requester. Unfortunately, while all these conflict resolution approaches show promise, none of them has been integrated in a SIEM architecture.
- Critical infrastructure monitoring is performed by deploying communication networks that enable the exchange of information between the monitored facilities and the control system. In order to control connections between external networks and internal networks, security policies that place strong limitations on data flows are established. For example, sensor firmware updates can only be performed by specific hosts located in an authorized local-area network that has privileged accounts and limits access to trusted employees. Current SIEM systems are unable to identify and control all possible data paths existing in a monitored infrastructure. The OSSIM SIEM system, for example, allows certain actions for controlling a monitored scenario, such as sending an email containing an alarm to the system administrator or executing a specific command.
  Network reachability analysis is required to identifying allowed and disallowed traffic between network entities. Over the years, several dynamic approaches (e.g., using network tools such as `ping`) and static approaches (e.g., using router and firewall configurations) have been proposed. Some approaches rely on graph-based representations to model the routing and filtering features of computer networks. Xie et al. [22] have proposed a unified model for analyzing static reachability based on two views: (i) a graph that describes the physical network topology, where the nodes are routers and the edges are network links and (ii) a graph that models the routing process, where the nodes are routing processes and the edges are adjacencies that implement a routing policy. The composition of these views makes it possible to evaluate reachability by combining routing policies that govern the

Please cite this article as: C. Di Sarno, et al., A novel security information and event management system for enhancing cyber security in a hydroelectric dam, International Journal of Critical Infrastructure Protection (2016), http://dx.doi.org/10.1016/j.ijcip.2016.03.002

distribution of routers with packet filtering policies that mandate which packets can traverse network links.

Al-Shaer et al. [2] have developed a model that represents a network using a state machine. A transition between states depends on the packet header, packet location (i.e., device) and policy semantics. The semantics of access control policies are represented using binary decision diagrams (BDDs). Computation tree logic (CTL) and symbolic model checking are then adopted to identify the past and future states of packets. This approach makes it possible to represent network configurations (based on access control policies) that help identify security violations (e.g., backdoors and broken IPsec tunnels). As in the case of the conflict resolution strategies discussed above, these approaches show promise, but they have not as yet been integrated in SIEM systems.

- A SIEM system generates alarms when attack signatures are detected. Alarms are stored along with related events in a database. Alarm information can be used for forensic purposes to obtain details of attack execution and impact as well as to identify the attackers. Thus, it is vital to ensure the integrity and non-forgeability of alarms. At this time, very few commercial SIEM systems ensure these requirements, typically by incorporating a module that signs the alarms using a cryptographic algorithm. However, the signing module is not designed to be resilient to attacks; thus, alarm information is subject to tampering as well as deletion. The SIEM system described in this paper engages the approach proposed by Afzaal et al. [1], which uses threshold cryptography to construct a storage system that is resilient to faults and intrusions.

3. Enhanced SIEM architecture

The proposed SIEM architecture is shown in Fig. 1. The source block represents the infrastructure to be monitored.

Probes are software components that are designed to: (i) collect information generated by hardware and software components in the monitored infrastructure; (ii) generate events that are useful for monitoring purposes (e.g., temperature and pressure measurements); (iii) perform a preliminary security analysis based on the available information; (iv) generate alerts when anomalies are detected based on the incoming information (e.g., the measured temperature exceeds a threshold); and convert the collected information to a common format to enable it to be processed by a correlator and a decision support system (DSS).

The correlator analyzes the events and alerts provided by the probes against known attack signatures. The correlator can perform fairly sophisticated security analysis because it considers the events and alerts produced by all the probes. The attack signatures, which are encoded as schematic rules, are stored in the rule database of the SIEM system. If a security breach pattern is found, then the correlator generates an alarm that is passed to the decision support system and resilient event storage.

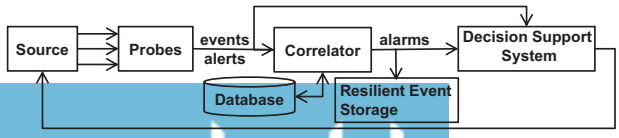


Fig. 1 – Enhanced SIEM architecture.

The decision support system (based on an XACML engine) is designed to: (i) ensure that the established security policies are not violated; (ii) implement a resolution strategy when policies are in conflict; and (iii) perform reachability analysis. Reachability analysis monitors all the network data paths in the infrastructure and verifies their compliance with the established security policies. Each data path is created by configuring network components (hardware and/or software). If the decision support system discovers a misconfiguration, then it performs a control action on the monitored source (e.g., an unauthorized data path is closed by modifying firewall rules). The DSS-solver is a decision support system component that implements policy conflict detection and resolution while the DSS-analyzer is a decision support system component that performs reachability analysis.

The alarms generated and stored by the correlator can be used as evidence of malicious activity; for this reason, it is important to guarantee their integrity and non-forgeability. The resilient event storage is an intrusion- and fault-tolerant system that is designed to satisfy these two requirements even if some components are compromised by an attack.

3.1. *Probes and correlator*

The probe modules are software components that process incoming data and perform security analyses. Each probe is installed at a specific location of the monitored infrastructure and operates in one of two ways: (i) passive, if the host/device is designed to generate and send log entries, then the probe processes and analyzes incoming logs and (ii) active, if the host/device does not generate log entries, then the probe actively gathers information, for example, by sniffing and analyzing network traffic at the monitored host/device.

As mentioned above, a probe is designed to generate events and alerts for monitoring purposes. Events are messages that contain information about the measurements of key indicators; the key indicators, which are parameters defined by infrastructure experts, facilitate infrastructure monitoring. An alert is a message that contains information about a detected anomaly (e.g., a threshold defined for a key indicator is exceeded). Events and alerts generated by the probes must be normalized (i.e., converted to a common message format). This allows the correlator to process logs generated by heterogeneous hardware and software components.

The correlator in Fig. 1 is a software component that analyzes the events and alerts sent by the probes to detect known attack signatures. The attack signatures are defined by correlation rules. Each correlation rule describes a specific pattern that identifies an anomaly or attack. A pattern is defined in terms of the attribute values of the analyzed information. When a correlation rule is matched, an anomaly or attack is detected and an alarm is raised according to the correlation rule. Each alarm contains the textual description of the detected anomaly or attack and the information that matched the correlation rule pattern. Thus, an alarm is

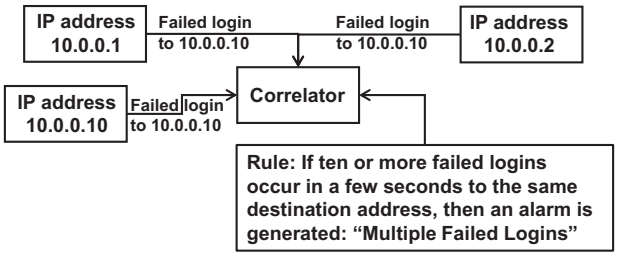


Fig. 2 – Correlation rule for detecting a malicious user who makes 10 or more login attempts to discover an administrator password.

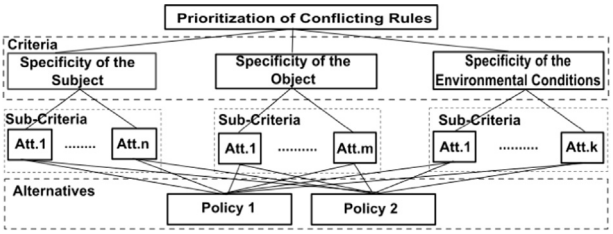


Fig. 3 – Hierarchy for policy conflict resolution.

semantically richer than a single event – it also contains the history of the events that matched the correlation rule.

Fig. 2 presents a correlation rule that detects when a malicious user performs 10 or more login attempts in a short period time to obtain an administrator password and gain access to an important server. The following correlation rule incorporated in the SIEM system detects this malicious activity:

```
if ((IPDestination==10.0.0.10) AND (numberOfFailedLoginOccurences
>=10) AND (data=.failed login.) AND (deltaTime<50))
then create new alarm .MultipleFailedLogin.
```

A MultipleFailedLogin alarm is generated when the analyzed information matches the correlation rule.

3.2. *Decision support system*

The decision support system incorporates two main components: (i) DSS-solver, which implements a customized policy conflict resolution strategy that addresses conflicts between XACML-based policies that are applied contemporaneously, but allow conflicting actions and (ii) DSS-analyzer, which discovers unauthorized network access and allows the redefinition of network configurations. In addition, the decision support system incorporates other components such as a repository that contains high-level policies and configuration policies used for security analysis.

3.2.1. *DSS-solver*

After the correlator detects a possible attack and raises an alarm, certain decisions have to be made to preserve the security and functionality of the system. The decisions depend on the policies specified for each component of the monitored system.

Each policy, which is specified using XACML, is expressed in terms of (i) subject; (ii) object (or resource); (iii) action; and (iv) environment (element of the policy) that are specified through their attributes. Policies are divided into two main classes based on their effects: (i) authorization, which express the actions that a subject is allowed to perform on an object in an environment and (ii) prohibition, which expresses the actions that a subject is not allowed to perform on an object in an environment. Note that the above assumptions are not necessarily restrictive; as a matter of fact, XACML relies on similar assumptions.

Thus, a security policy is a set of rules that are evaluated for each access request. The purpose is to decide whether or not a subject is allowed to perform an action on a resource in an environment. Policy rules include conditions on element attribute values to determine the rules that are applied to each access request. Security policies expressed in this manner may conflict with each another. Two policies are potentially in conflict when they are contemporaneously applied to allow or disallow access to some resources.

XACML introduces combining algorithms to solve conflicts that could arise among rules in a policy and among policies in a policy set. Standard combining algorithms are: (i) deny-overrides; (ii) permit-overrides; (iii) first-applicable; and (iv) only-one-applicable. XACML also permits the specification of customized combining algorithms.

This paper employs the AHPPolicyCombiningAlgorithm, which is based on the analytic hierarchy process (AHP) developed by Saaty [16]. The algorithm engages a multi-criteria decision making technique that has been used in several fields of study.

Given a decision problem, a hierarchy is constructed as shown in Fig. 3. Various alternatives at the bottom of the figure can be chosen to reach the goal at the top of the hierarchy. The analytic hierarchy process returns the most relevant alternative with respect to a set of criteria located at the second level of the hierarchy. This approach requires the division of a complex problem into a set of sub-problems equal in number to the chosen criteria. The solution is then obtained by merging the local solutions of all the sub-problems.

In the hierarchy shown in Fig. 3, the goal is to rank the conflicting policies that correspond to the alternatives in the analytic hierarchy. Note that, in the figure, a conflict is assumed to exist between two policies (policies 1 and 2), but the alternatives may be more than two. The criteria, which represent the second group of boxes from the top of the hierarchy in Fig. 3, express the specificities of elements of a policy. Since the elements are defined in terms of their attributes, the specificity of each element is expressed using the attributes that comprise its definition. In general, attribute a_1 of element e is more specific than attribute a_2 of the same element e if a condition on attribute a_1 is likely to identify a more homogeneous and/or smaller set of entities within e.

The analytic hierarchy process allows further refinement of each criterion in the sub-criteria by considering the attributes that identify each element. Examples of subject attributes are identification number (ID) and organization of the subject. Note that the set of considered attributes

Table 1 – Fundamental scale for the analytic hierarchy process.

Intensity	Definition	Explanation
1	Equal	Two elements are equally relevant
3	Moderate	One element is slightly more relevant than another
5	Strong	One element is strongly more relevant than another
7	Very strong	One element is very strongly more relevant than another
9	Extreme	One element is extremely more relevant than another

depends on the chosen scenario. In Section 4, this approach is customized to a hydroelectric dam case study and the customized hierarchy is used to solve a specific conflict raised in the case of the hydroelectric dam.

After the hierarchy is constructed, the next step is to compute the local priorities of: (i) each alternative with respect to each sub-criterion; (ii) each sub-criterion with respect to the relative criterion; and (iii) each criterion with respect to the goal. The computations involve pairwise comparisons from the bottom to the top. Each pairwise comparison is expressed as a pairwise comparison matrix, which has positive entries that are reciprocal values (i.e., $a_{ij} = \frac{1}{a_{ji}}$). The value of each a_{ij} is chosen according to a scale typical to the analytic hierarchy process (Table 1), which indicates how much an alternative is more relevant than another.

A pairwise comparison matrix is defined as being consistent if $a_{i,j} \cdot a_{j,k} = a_{i,k}$ for all i, j, k. The satisfaction of this property implies that if x is more relevant than y and y is more relevant than z, then z cannot be more relevant than x. In practice, creating a perfectly consistent matrix may not be possible because the judgments are left to humans. According to Saaty [17], the inconsistency of an $m \times m$ reciprocal matrix can be expressed using a consistency index CI given by:

$$CI = \frac{\lambda_{max} - m}{m - 1} \tag{1}$$

where λ_{max} is the maximum eigenvalue of the reciprocal matrix.

In the case of a consistent matrix, $CI = 0$. A matrix is considered to semi-consistent if $CI < 0.1$. If this condition does not hold, then the comparison values must be reevaluated.

Consider the hierarchy shown in Fig. 3 and assume that only two policies are in conflict. First, the local priorities of each alternative must be computed (two in this case) with respect to each sub-criterion by computing k 2×2 pairwise comparison matrices, where k is the number of sub-criteria ($k = 9$ in this case). The matrices are constructed according to the attributes present in the policies. Let a_{ij} be a generic element of one of the matrices, then the following properties hold:

- If Policy 1 and Policy 2 contain (or do not contain) attribute A, then $a_{12} = a_{21} = 1$.
- If only Policy 1 contains attribute A, then $a_{12} = 9$ and $a_{21} = \frac{1}{9}$.
- If only Policy 2 contains attribute A, then $a_{12} = \frac{1}{9}$ and $a_{21} = 9$.

After a comparison matrix has been defined, the local priority corresponds to the normalized eigenvector associated with the largest eigenvalue of the matrix [16]. Next, moving up the hierarchy, the relevance of each sub-criterion is quantified with respect to the corresponding criterion. This evaluates the relevance of the attributes to identifying the subject, object and environment (e.g., in the case of a subject, its ID is more relevant than its role and organization, and its role and organization have the same relevance).

Finally, the relevance of the three criteria to achieving the goal of solving conflicts is quantified. The global priority is calculated as a weighted summation of the local priorities. Specifically, the local priorities are calculated as pairwise comparisons between entities at one level of the hierarchy with respect to entities at the upper level: (i) comparisons of the alternatives with respect to sub-criteria; (ii) comparisons of the sub-criteria with respect to the criteria; and (iii) comparisons of the criteria with respect to the goal.

The following equation is used to calculate the global priority of the alternatives with respect to the goal:

$$P_g^{a_i} = \sum_{j=1}^{n1} \sum_{k=1}^{q(w)} p_g^{c_w} \cdot p_{c_w}^{sc_k^w} \cdot p_{sc_k^w}^{a_i} + \sum_{j=1}^{n2} p_g^{c_j} \cdot p_{c_j}^{a_i} \tag{2}$$

where the leftmost $n1$ criteria in the hierarchy tree each have a set of sub-criteria while the rightmost $n2$ criteria have no sub-criteria below them; $n1 + n2 = n$ is the number of criteria; $q(w)$ is the number of sub-criteria for criterion c_w; $p_g^{c_w}$ is the local priority of criterion c_w with respect to the goal g; $p_{c_w}^{sc_k^w}$ is the local priority of sub-criterion k with respect to criterion c_w; and $p_{sc_k^w}^{a_i}$ is the local priority of alternative a_i with respect to sub-criterion k of criterion c_w. Note that $p_{c_w}^{sc_k^w}$ and $p_{sc_k^w}^{a_i}$ are also computed using the pairwise comparison matrices as discussed above.

3.2.2. *DSS-analyzer*

The DSS-analyzer invokes the reachability analysis process (RAP) whose objective is to discover unauthorized network access. An unauthorized network access occurs when firewall rules are modified by an unauthorized individual (e.g., attacker) or by an authorized individual (e.g., configuration error). Reachability analysis is used to detect unauthorized traffic. Specifically, the filtering rules derived from the policies are compared with the firewall configuration rules. However, as mentioned above, the policies are XACML-based and are defined using a few elements (e.g., subject, action [options], object) and are topology independent (i.e., network topology is not considered during policy creation).

Three types of actions are considered: (i) reach, which specifies the authorized network interactions between a subject and object; (ii) log, which specifies when the interactions (i.e., a subject contacts an object using a particular protocol and port) are logged either locally or remotely; and (iii) mirror, which specifies when the traffic of an interaction is duplicated and forwarded to another host (useful for traffic

Please cite this article as: C. Di Sarno, et al., A novel security information and event management system for enhancing cyber security in a hydroelectric dam, International Journal of Critical Infrastructure Protection (2016), http://dx.doi.org/10.1016/j.ijcip.2016.03.002

analysis). Both log and mirror support an option to specify when the traffic is logged/captured locally or forwarded to a remote host (in which case, the option requires the IP address of the remote host). This approach simplifies policy management. For example, undefined interactions are prohibited by default and policy conflicts are avoided; however, some anomalies must be addressed (e.g., equivalent rules).

In contrast, firewall rules are represented using a common format: (source IP address, source port, destination IP address, destination port, protocol, action, [options]). These rules depend on the network topology (e.g., where a firewall is positioned in the network and which hosts are protected by the firewall). Therefore, policies must be transformed into a concrete format and this operation must be executed for every filtering device in the network before reachability analysis can be performed.

The reachability analysis process uses a set of rules and an inference engine to detect unauthorized network access. In the beginning or when policies are modified, the reachability analysis process starts the refinement to generate the sets of rules for filtering devices. This process is organized in terms of a set of policy refinement tasks in which the policies and system description are analyzed and a graph-based network topology representation is generated. During the policy analysis, anomalies (e.g., redundancy) are detected and addressed.

The system is described using XML. The description comprises hosts and their related information (e.g., IP addresses), capabilities (e.g., packet filtering), services and the network topology.

The network analysis task identifies the set of firewalls that enforce each policy. Specifically, it analyzes the graph-based representation to discover the network paths that have at least one firewall between the subject and object of a policy. Since the default action of a firewall is to deny all traffic, each firewall contained in a path must be configured to permit the policy traffic. Hence, for each firewall, a set of filtering rules is generated to enforce the policies.

When at least one firewall does not exist to implement a policy, the policy is not enforceable. This typically occurs when the subject and object belong to the same subnet and their traffic does not traverse a firewall. In such a situation, any type of traffic between the subject and object is permitted, potentially creating a security breach. This situation is managed by the module that logs the security issue and saves it to the internal models repository.

After the refinement process is completed, the reachability analysis process evaluates the filtering rules. The rules include those generated by the previous process (i.e., generated rules) and those deployed by firewalls (i.e., deployed rules).

The overall process is organized into four phases:

- *Translation:* For each deployed rule, the fields structured as firewall-specific statements are translated to the common format: (srcIP, srcPort, dstIP, dstPort, action, [options]), where srcIP and dstIP are single IP addresses or address ranges, srcPort and dstPort are single ports or ranges of ports, action is either accept (used when the destination is the current device), accept/forward (used when the destination is not the current device and traffic must be forwarded), log or mirror, option (available for log and mirror actions) is of the form: local, local interface, remote IP (where local interface identifies the destination interface for logged or mirrored traffic). Since firewalls have different features and language-specific statements, a set of adapters are required to translate the rule set for a particular device to the common format. The adapter-based approach makes the module simple, flexible and extensible.
- *Expansion:* For each generated rule and deployed rule (i.e., rules obtained after translation), the fields that contain ranges (IP addresses, ports, etc.) are expanded by considering the network description (i.e., hosts and services in the system description) and creating new rules. Suppose rule r_1 is given by: (*srcIP*:192.168.0.1, *srcPort*:∗, *dstIP*:192.168.10.10, *dstPort*:80,443, *protcl:TCP*, *action:accept*), where the destination IP address refers to a host that offers a web service on ports 80 and 443. The expansion phase transforms r_1 into rules $r_{1,1}$ and $r_{1,2}$; the first rule matches port 80 and the second rule matches port 443. The same approach is followed for the IP address ranges (i.e., subnets). Note that, before the expansion operation is applied, the deployed rules are analyzed to detect and address anomalies.
- *Composition:* This phase creates the reachability matrices. Each firewall i has two rule sets, one for the generated rules ($R_{g,i}$) and the other for the deployed rules ($R_{d,i}$). An equivalent rule set for firewall i ($R_{e,i}$) is introduced that contains the generated and deployed rules, i.e., $R_{e,i} = R_{g,i} \cup R_{d,i}$. Two partitions are created for $R_{e,i}$: the first partition contains the source IP address and port fields ($S_{IP,port}$) and the second contains the destination IP address, port, protocol, action and option ($D_{IP,port,protcl,action,option}$), where option is applicable only to the log and mirror actions; otherwise, it is null. A two-dimensional reachability matrix for firewall i (M_i) has $S_{IP,port}$ elements in its rows and $D_{IP,port,protcl,action,option}$ elements in its columns. Therefore, the equivalent rule set ($R_{e,i}$) has rules with four types of actions: accept, accept/forward, log and mirror. To simplify the analysis (e.g., comparisons of rules with the same actions), the rule set is organized into four matrices, one for each action type.

The composition phase performs the following tasks for each firewall i:

1. Create four matrices for the generated rules: $M_{g,a,i}$ (contains rules with accept actions), $M_{g,af,i}$ (contains rules with accept/forward actions), $M_{g,l,i}$ (contains rules with log actions) and $M_{g,m,i}$ (contains rules with mirror actions). Each matrix contains $S_{IP,port}$ entries in its rows and $D_{IP,port,protcl,action,option}$ entries in its columns. For the accept and accept/forward actions, the option is set to null.
2. Create four matrices for the deployed rules. As in the previous step, this generates $M_{d,a,i}$, $M_{d,af,i}$, $M_{d,l,i}$ and $M_{d,m,i}$.
3. Compute $M_{g,a,i}$, $M_{g,af,i}$, $M_{g,l,i}$, $M_{g,m,i}$. For each rule r in $R_{e,i}$, if r is a part of the $R_{g,i}$ rules (i.e., $r \in R_{g,i}$), set the corresponding row and column to one; otherwise, set them to zero.
4. Compute $M_{d,a,i}$, $M_{d,af,i}$, $M_{d,l,i}$, $M_{d,m,i}$. For each rule r in $R_{e,i}$, if r is a part of the $R_{d,i}$ rules (i.e., $r \in R_{d,i}$), set the corresponding row and column to one; otherwise, set them to zero.

- *Analysis:* This phase compares the reachability properties of the generated rules and deployed rules. For each firewall i, the following computations are performed:
  (a) $M_{\rho,a,i} = M_{g,a,i} - M_{d,a,i}$ for accept rules.
  (b) $M_{\rho,af,i} = M_{g,af,i} - M_{d,af,i}$ for accept/forward rules.
  (c) $M_{\rho,l,i} = M_{g,l,i} - M_{d,l,i}$ for log rules.
  (d) $M_{\rho,m,i} = M_{g,m,i} - M_{d,m,i}$ for mirror rules.

  If $M_{\rho,x,i} = 0$, where x is a placeholder for a, af, l or m (i.e., when all the elements are equal to 0), the reachability for the generated and deployed rules are the same for a particular action. When $M_{\rho,a,i}, M_{\rho,af,i}, M_{\rho,l,i}, M_{\rho,m,i}$ are equal to 0, no security issue is identified. Otherwise (i.e., $M_{\rho,x,i} \neq 0$), at least one element is equal to 1 or -1. In the first case (equal to 1), the corresponding rule is not deployed in the firewall. Therefore, the firewall configuration drops a packet that must be permitted by the policy. This situation is reported as an anomaly. Otherwise (equal to -1), the corresponding rule is enforced by the firewall configuration, but is prohibited by the policy. In this situation, the firewall contains a misconfiguration and the reachability analysis process logs it as security issue.

When an element (i.e., rule) of $M_{\rho,a,i}$ or $M_{\rho,af,i}$ is equal to 1, a misconfiguration or an attacker block certain traffic (by removing the related firewall rule) authorized by the policy, but dropped by the firewall. If the element is equal to -1, then a rule is added or modified (when no rule is added or removed, at least one field is modified) to permit certain traffic. The values of $M_{\rho,l,i}$ are useful to detect when logging rules are removed, modified or added to track certain traffic. The deletion of logging rules is a typical approach employed in masquerade attacks where evidence is removed to hinder forensic analysis. Similarly, the modification of a logging rule (e.g., changing the remote IP address of the logging server) could redirect log traffic (e.g., send data to a different server that discards traffic instead of performing analysis, or to a malicious endpoint to track network traffic). The insertion of a new logging rule can be considered to be a misconfiguration or an attack that seeks to gain information about network communications. Traffic mirroring (whose rules are represented by matrix $M_{\rho,m,i}$) is typically used to perform analysis on network content (e.g., by an intrusion detection system). Similarly, for logging, network content could be redirected to a malicious endpoint (e.g., to capture sensitive data) or the content could be suppressed by an attacker to hinder analysis (e.g., detection of traffic anomalies). Finally, the reachability analysis process reports any detected anomalies or security issues and proposes remediation approaches. These may include suggestions for modifying firewall rules or positioning a filtering device (e.g., personal firewall) to enforce the policy.

3.3. Resilient event storage

The resilient event storage (RES) is an infrastructure designed to: (i) tolerate faults and intrusions; (ii) generate signed records containing events and alarms related to security breaches; and (iii) ensure the integrity and non-forgeability of stored events and alarms.

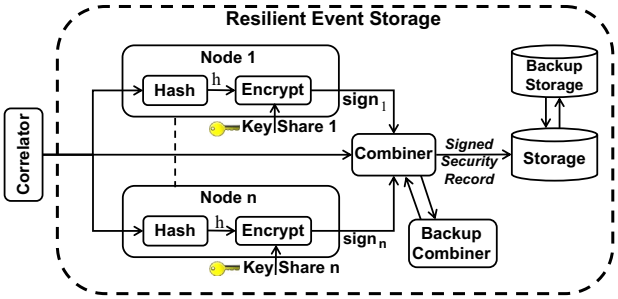


Fig. 4 – Resilient event storage.

Fig. 4 presents the conceptual architecture of the resilient event storage. Its fault- and intrusion-tolerant functionality enables it to create securely signed records even when some components of the system are compromised. The basic principle is to use more than one secret key. In fact, only one secret key is used, but it is divided into n parts (shares) with each share is stored at a different node. This approach can be realized by the Shoup threshold cryptography algorithm [18]. The most important characteristic of the algorithm is that the attacker has zero knowledge about the secret key if less than $k-1$ secret key shares are compromised ($k \leq n$). The algorithm is characterized by two parameters: (i) n, the number of nodes and (ii) k, the security threshold. The output of a cryptography algorithm and its threshold version are equivalent.

The resilient event storage has a component named dealer that generates n secret key shares, n verification key shares and one verification key from a main secret key. This component is not shown in Fig. 4 because it is only used during the initialization phase. After the dealer has generated the keys, it sends a secret key share to each node; the n verification key shares and the verification key are sent to the combiner. Each verification key share is used to check the correctness of the signature share generated by each node with its own secret key share. After the combiner puts together the signature shares provided by nodes, the verification key is used to check the correctness of the entire signature. Input data to the resilient event storage is provided by the correlator (Fig. 1) because the alarms, which contain information about security breaches, must be stored in a secure manner. Incoming alarms are sent to all the nodes and to the combiner. Each node computes a hash digest of the received alarm; the digest is denoted by h in Fig. 4. Finally, each node encrypts the digest h with the secret key share and generates a signature share that is sent to the combiner.

After the combiner receives at least k signature shares (from the nodes) for the same alarm, it can assemble the partial signatures to obtain the complete signature. Then, the combiner verifies the complete signature using the verification key. If the verification process fails, then the combiner verifies the correctness of each signature share using the corresponding verification key share. When a node is identified as having sent the wrong signature share, it is flagged as

Please cite this article as: C. Di Sarno, et al., A novel security information and event management system for enhancing cyber security in a hydroelectric dam, International Journal of Critical Infrastructure Protection (2016), http://dx.doi.org/10.1016/j.ijcip.2016.03.002

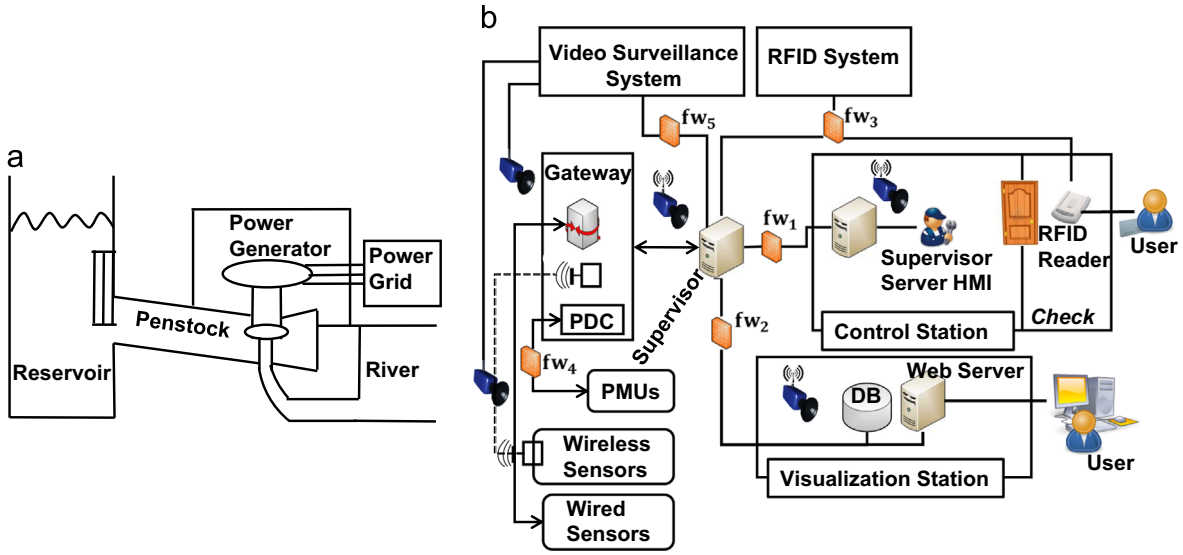


Fig. 5 – (a) Hydroelectric dam and (b) simplified IT infrastructure for dam monitoring and control.

being corrupted. The next time, if new signature shares are available for the same alarm, the combiner uses the already-validated signature shares and the new signature shares to create a new set of k signature shares. Then, the combiner generates a new complete signature and repeats the verification process. If the verification process is successful, then the complete signature, the original alarm and the identifiers of the corrupted nodes are stored in the resilient event storage. To improve the fault- and intrusion-tolerance of the resilient event storage, replication and diversity are employed in the media storage and combiner.

4. Hydroelectric dam case study

The SIEM system described in the previous sections was used to monitor and control a hydroelectric dam. Fig. 5(a) shows a schematic diagram of the hydroelectric dam, which feeds power to the grid as in the case of any common generator. A hydroelectric power generation dam requires a reservoir that is fed by a river. A certain amount of water from the reservoir flows through penstocks in the dam. The water flow moves turbines, which are connected to alternators that convert mechanical energy into electrical energy. The generated electricity is injected via transmission lines to the power grid.

The power generated by hydroelectric dam primarily depends on the water flow rate Q (m^3/s) provided to a turbine via a penstock and the difference Δh (m) between the water level in the reservoir and water level in the turbine. The power P generated by the turbine rotation is given by:

$$P = \rho * \eta * g * \Delta h * Q \tag{3}$$

where ρ is the density of water (1000 kg/m^3), η is the turbine efficiency and g is the gravitational constant (9.81 m/s^2). If Δh is assumed to be constant in Eq. (3), then the generated power is only a function of the water flow rate Q. Under this assumption, it is possible to increase the generated power by increasing the flow rate Q. The upper bound on the water flow rate that feeds the turbine is related to the penstock geometry or/and turbine features. The dependence between water flow rate and the generic penstock geometry is described by the Hazen–Williams equation:

$$Q^{1.852} = \frac{\Delta h * c^{1.852} * d^{4.8704}}{l * 10.675} \tag{4}$$

where l is the length of penstock (m), c is the penstock roughness coefficient and d is the inside penstock diameter (m).

A hydroelectric plant turbine has three design parameters as described in the following equation:

$$n_c = n * \left(\frac{\sqrt[2]{P}}{\Delta h^{1.25}} \right) \tag{5}$$

The three parameters: (i) Δh, the difference in water levels (m); (ii) n, the turbine rotation speed (rpm); and (iii) P, the output power to be provided (kW). Note that the term n_c is a dimensionless number called the specific speed.

The value of n depends on the turbine and alternator features. If Δh and the alternator features in Eq. (5) are fixed, then n_c depends only on the output power P. The values computed for n_c is used to choose the turbine to be deployed.

In the scenario under consideration, an emergency state alarm is raised when a violation occurs for any critical key indicator established by expert operators of the monitored infrastructure. For example, a critical key indicator could be the water flow rate. In fact, in Eq. (3), an increase in the water flow rate implies an increase in the generated power. In Eq. (3), more power implies higher turbine rotation. If the number of rotations per minute exceeds a fixed threshold, then the turbine could be damaged and/or the electric power generated could exceed the security threshold.

Fig. 5 (b) shows a simplified view of the information technology (IT) systems used to monitor and control the dam (including the hydroelectric power generation station). The information generated by the devices/systems shown in Fig. 5(b) feed the enhanced SIEM system. In particular, the visualization station (VS) allows the monitoring of the infrastructure and the viewing of statistics related to the collected data.

Table 2 – Testbed configuration for the enhanced SIEM system.

Type	Description/configuration	Data rate
Sensor	Water flow rate measurement	1 sample/min
Sensor	Penstock gate opening measurement	Event-based
Phasor measurement unit	Generated power measurement	60 samples/s
Probes	Gather data from visualization station control station, sensors and phasor measurement units Each probe generates events (in IDMEF [6] format)	Event-based
Probe security thresholds	Expected output power P_e = 550 kW Expected water flow rate Q=0.6 m^3/s Expected max power P_{max}=650 kW Maximum safe water flow rate Q=0.7 m^3/s Default gate opening=80%	None
Correlator	Prelude-OSS SIEM correlator Correlation rules written in Python	Event-based
Resilient event storage	Configured with n=5 and k=3	Event-based
Decision support system (conflict resolution)	Analytic hierarchy configured with attributes of each policy element A1: if a user has no administrator role and is not in the control station, then the user cannot reprogram the sensors A2: if a user is at the visualization station and an emergency state is raised, then the user is allowed to reprogram the sensors A3: if a user is at the visualization station and an alarm is raised, then the user is not allowed to reprogram the sensors	Event-based
Decision support system (reachability analysis)	Reachability analysis rules (see Table 3)	Event-based

When an emergency state alarm is raised (i.e., a key indicator exceeds a security threshold), some control actions can be initiated from the visualization station (e.g., remote sensor reprogramming). However a software component (not shown in Fig. 5(b)) monitors each request from the visualization station to the gateway and accepts or denies the requests according to the established high-level policies.

The control station (CS) enables the monitoring and control of the dam infrastructure (i.e., it receives data from sensors and sends commands to actuators). Wireless/wired sensors and phasor measurement units (PMUs) measure the key indicators and send the measurements to the gateways. A special gateway is the phasor data concentrator (PDC), which collects measurements generated by the phasor measurement units to evaluate the power grid status. Video surveillance and radio frequency identification (RFID) systems help maintain the physical security of the infrastructure.

5. Attack model

The attack model assumes that the attacker seeks to access sensitive data such as environmental measurements related to the hydroelectric dam. It is assumed that the attacker has a device within the wireless sensor network (WSN) range (i.e., WSN gateway). The attacker cannot penetrate the network directly because it is properly secured. However, the attacker has physical access to firewall fw_4 shown in Fig. 5(b) and can modify its firewall rules.

The attack involves the following steps.

1. The attacker configures a false phasor data concentrator and modifies a firewall fw_4 rule to mirror data generated by a phasor measurement unit under attack to the false phasor data concentrator. The traffic to and from phasor measurement units is not encrypted [13], so the attacker is able to read unencrypted data from the false phasor data concentrator.
2. From the false phasor data concentrator, the attacker gathers data and information about the attacked phasor measurement unit (e.g., ID and IP address). Next, the attacker sends a command to the attacked phasor measurement unit asking it to stop sending measurements. The attacker then uses a software phasor measurement unit that is configured to emulate the attacked phasor measurement unit; this false phasor measurement unit injects forged data corresponding to a power overload on the transmission line to the authentic phasor data concentrator. This action creates a false emergency state, in which power generation by hydroelectric dam exceeds the security threshold. As described in the previous section, the emergency state allows additional actions from the visualization station (e.g., sensor reprogramming).
3. The attacker is not allowed to access the control station, so the attacker reprograms a sensor in the wireless sensor network from the visualization station (the attacker is allowed to do this because of the false emergency state triggered in Step 2). The new malicious program performs a sinkhole attack that redirects wireless sensor network traffic to the compromised node to access and compromise legitimate packets [4].

Please cite this article as: C. Di Sarno, et al., A novel security information and event management system for enhancing cyber security in a hydroelectric dam, International Journal of Critical Infrastructure Protection (2016), http://dx.doi.org/10.1016/j.ijcip.2016.03.002

4. The attacker successfully compromises the wireless sensor network. In fact, the malicious node can send all the data to the malicious gateway, which has visibility of the wireless network. At this point, the attacker does not need to use any device in the monitored infrastructure to analyze wireless sensor network traffic. Also, any corrective actions issued from the visualization station (e.g., disabling traffic from/to the visualization station) are ineffective.

6. System setup and validation

This section describes the setup of the enhanced SIEM system and its validation using the attack model described in the previous section.

6.1. System setup

Table 2 shows the system configuration. A wireless network was employed for the sensors. This is because deploying wired sensors on a dam is a very difficult task – many kilometers of Ethernet cables would have to be laid out, much of the cabling in a hostile environment.

The water flow sensor data rate was chosen to trade-off the monitoring requirements versus the energy requirements of battery-supplied wireless sensor nodes. The data rate used for phasor measurement unit monitoring was chosen based on the IEEE C37.118-2011 standard. The default value chosen for the gate opening ensures that the expected water flow rate value is not exceeded. This water flow rate also ensures that the power generated by the hydroelectric dam does not exceed the security threshold. The correlator chosen was a standard component embedded in the Prelude-OSS SIEM system, one of the most popular SIEM systems.

The conflict resolution strategy and the policies themselves are useful for dealing with the attack scenario. The approach of Lunardelli et al. [9], which uses an open source implementation of the XACML engine from Sun Microsystems, was employed. The abstract class PolicyCombiningAlgorithm in the XACML library was extended with a new class named AHPPolicyCombiningAlgorithm. This new class overrides the original combining algorithm with a new method that implements the analytic hierarchy process described in Section 3.2. It receives as input the evaluation context (which includes the access request), parameter list (empty in the prototype) and policy set and outputs the result of the evaluation.

The first step in the analytic hierarchy process is the instantiation of the hierarchy (Fig. 3) with the attributes of each element of each policy. In the hydroelectric dam scenario, the subject attributes are the identification number (ID), role and location within the dam; the object attributes are ID, category of data and producer of the data (i.e., sensor); and the environment attributes are alarm (indicates that a communications problem has occurred), alert (refers to an emergency state, i.e., physical problem) and time. To simplify the presentation, the policies in Table 3 are expressed in natural language although they are specified in XACML in the DSS-solver.

Firewalls fw_1, fw_2 and fw_4 contain the rules listed in Table 3. In particular, r_1 accepts and forwards TCP traffic from the control station to the gateway (e.g., to manage sensors); r_2 accepts and forwards TCP traffic from the visualization station to the gateway (e.g., to manage sensors during the emergency state) and r_3 duplicates and forwards (i.e., mirrors) all traffic to firewall fw_4, which performs network content analysis. Considering these rules at setup, the matrices $M_{g,af,fw1}$, $M_{d,af,fw1}$, $M_{g,af,fw2}$, $M_{d,af,fw2}$, $M_{g,m,fw4}$ and $M_{d,m,fw4}$ are given by:

$$M_{g,af,fw1} = M_{d,af,fw1} = r_{1,1}^{r_{1,2}}(1) \tag{6}$$

$$M_{g,af,fw2} = M_{d,af,fw2} = r_{2,1}^{r_{2,2}}(1) \tag{7}$$

$$M_{g,m,fw4} = M_{d,m,fw4} = r_{3,1}^{r_{3,2}}(1) \tag{8}$$

where $r_{x,y}$ is partition y of rule x in Table 3. For example, $r_{1,1}$ is: (IP_{CS}, any). On the other hand, $r_{1,2}$ is: ($IP_{Gateway}$, any, TCP, accept/forward, null). At setup, $M_{\rho,af,fw1} = M_{\rho,af,fw2} = M_{\rho,m,fw4} = 0$.

6.2. Validation

Based on the attack steps described in Section 5, the enhanced SIEM system detects the attack as follows:

- In Step 1, the attacker updates the firewall fw_4 rules and no event is generated by any probe. The attacker adds a mirror rule (r_4) to capture and forward traffic coming from the compromised phasor measurement unit (and directed to the legitimate phasor data concentrator) to the malicious phasor data concentrator (attacker device). The rule r_4 is defined by the following tuple: (Src IP: IP_{PMU}, Src Port: any, Dst IP: IP_{PDC}, Dst Port: any, Protcl: any, Action: mirror, Option: remote IP_{AD}). During this step, neither the correlator nor the DSS-solver are able to detect the attack because no alert is raised by any SIEM system component.
- In Step 2, the probe monitoring the attacked phasor data concentrator is misled because it captures the forged phasor measurement unit data sent by the attacker. The probe merely compares the incoming data against the thresholds, so it is unable to detect the forged data and it generates a new alert indicating an emergency state:-

```
idProbe=probe_12 timestamp=2014-08-10 T 10:44:30 UTC
data=OverloadedTransmissionLine location=GatewayStation
SensorIp=192.168.10.5 role=# state=emergency type=alert
```

  However, neither the correlator rules nor the DSS-solver policies trigger alarms or reactions. As shown in Table 2, this information does not match any anomalous conditions with respect to the implemented cyber security rules and policies.
- In Step 3, the attacker attempts to reprogram a wireless sensor node from the visualization station because reprogramming is allowed in the emergency state. The probe connected to the visualization station used by the attacker

detects the reprogramming command and generates the event:-

```
idProbe=probe_20 timestamp=2014-08-10 T 10:44:50 UTC
data=sensorReprogramming location=VS
SensorIp=10.0.0.1 role=employee state=# type=event
```

The event is sent to the correlator and decision support system. The event and the alert are translated to the IDMEF format before they are sent from the probe to the correlator and decision support system.

The DSS-solver is already aware of the emergency state and it also receives the sensor reprogramming event from the probe. Thus, policies A1 and A2 in Table 2 are true. However, the two policies are in conflict with each other because A1 denies the reprogramming of the sensor under the current conditions while A2 permits sensor reprogramming.

The correlator also receives the reprogramming event from the probe and evaluates the following correlation rule:

```
if idmef.Get("state")=='emergency'
   rawEvent=idmef.Get("rawEvent")
   ctx=Context(("Emergency-Raised-Probes", rawEvent),
        {"expire": 180} update=True)
   numberOfEmergencies += 1
if idmef.Get("data")=='SensorReprogrammingEvent' and
   idmef.Get("location")=='VS' and numberOfEmergencies >= 1
   rawEvent=idmef.Get("rawEvent")
   updateContext(ctx,"Emergency-Raised-Probes", rawEvent)
if ctx.expire==true
   boolean found = search(ctx,"SensorReprogrammingEvent")
   if found==true
     [emergencyProbesIPList] = GetProbes('emergency', ctx)
     if emergencyProbesIPList.size==1
       ctx.Set("correlation.name", "Unauthorized-Sensor-
        Reprogramming-Attempt")
       ctx.alarm()
   ctx.destroy()
```

The first IF-statement checks if the incoming alerts are related to the emergency state. If an emergency state is detected, then the raised alert is saved in a context. The Prelude-correlator uses the context data type to group events and alerts that have common features. Each context identified by a label is created the first time that an event or alert is added to it and it is updated when the next event or alert are processed. A context is destroyed when the time threshold expires (180 s in the case study).

The second IF-statement checks if the event is related to a sensor reprogramming action performed from the visualization station. When this occurs and an emergency state was previously raised, then the context is updated with the sensor reprogramming event.

The third IF-statement is true when the time threshold has expired (180 s). When this occurs, the correlation rule checks if a sensor reprogramming event was received within the last 180 s. The goal of the correlator is to establish if this action is related to an actual emergency or to an abuse of the emergency state (cyber attack). Thus, the correlator counts the number of probes that have raised the emergency state. This is because the effects of an emergency state are detected by different probes within the same time window as a real emergency. If only one probe raises an emergency state – as in this scenario – a cyber attack is indicated. In this case, an alarm is generated and sent to the DSS-solver and resilient event storage. The alarm stored in the resilient event storage contains the traceback of the previous events and alerts.

The DSS-solver receives the alarm from the correlator; this information activates policy A3 referring to the alarm. All the

Table 3 – Testbed filtering rules.

Rule	Src IP	Src Pt	Dst IP	Dst Pt	Protcl	Action	Options	Firewall
r_1	IP_{CS}	Any	$IP_{Gateway}$	Any	TCP	acc/for	Null	fw_1
r_2	IP_{VS}	Any	$IP_{Gateway}$	Any	TCP	acc/for	Null	fw_2
r_3	Any	Any	Any	Any	Any	Mirror	local fw_4	fw_4

Table 4 – Comparison matrices and local priorities for the sub-criteria w.r.t. the criteria.

SUBJ	ID	role	locn.	$\overline{P}_{Subj}$
ID	1	9	9	0.8181
role	$\frac{1}{9}$	1	1	0.0909
locn.	$\frac{1}{9}$	1	1	0.0909
OBJ	**ID**	**prod.**	**cat.**	$\overline{P}_{Obj}$
ID	1	5	7	0.7454
prod.	$\frac{1}{5}$	1	43	0.1454
cat.	$\frac{1}{7}$	34	1	0.1091
ENV	**alarm**	**alert**	**time**	$\overline{P}_{Env}$
alarm	1	3	7	0.61963
alert	$\frac{1}{3}$	1	1	0.32390
time	$\frac{1}{7}$	1	1	0.05644

conflicting policies are given as inputs to the conflict solver along with additional information needed to evaluate the attributes. Then, pairwise comparisons from the bottom to the top are performed to compute the local priorities. Thus, each criterion is statically evaluated with respect to the goal.

The local priorities of the uppermost two levels of the analytic hierarchy in Fig. 3 are defined when the policies are created for a specific scenario. The local priorities for the uppermost level (i.e., criteria with respect to the goal) are all assumed to be equal to 0.33. The local priorities for the middle level (i.e., sub-criteria with respect to criteria) are specified in Table 4.

The comparison matrices are created according to the following relevance conditions:

- *Subject attributes:* ID is more relevant than role and location. Role and location have the same relevance.
- *Object attributes:* ID is more relevant than producer. Producer is more relevant than category.
- *Environment attributes:* Alarm is slightly more relevant than alert. Alert is more relevant than time.

The local priorities at the lowest level of the analytic hierarchy are evaluated at runtime (e.g., when data access is attempted). The evaluation is simply based on the presence or absence of an attribute in the conflicting rules. For example, policy A1 identifies a subject using role and location while policies A2 and A3 only identify a subject using location. Furthermore, policy A1 does not have constraints related to an environment while policy A2 identifies an environment using an alert and policy A3 identifies an environment using an alarm.

The global priorities $P_g^{A_n}$ are calculated according to Eq. (2). The final results are:

$$P_g^{A1} = 0.27, \quad P_g^{A2} = 0.33, \quad P_g^{A3} = 0.4 \tag{9}$$

Hence, in the case of an alarm, the user cannot reprogram the sensor using the visualization station. It is worth noting that, in the case of a real emergency and without any raised alarm, the conflict resolver returns policy A2 as the policy to be applied. Indeed, in this case only policies A1 and A2 are compared and the global priorities are:

$$P_g^{A1} = 0.46, \quad P_g^{A2} = 0.54 \tag{10}$$

Next, the DSS-solver activates the DSS-analyzer and indicates the IP address of the visualization station node that requested the sensor reprogramming action. The DSS-solver looks for mismatches among the policies chosen at the design stage and the policies currently available for the network devices. In practice, the reachability analysis process computes the matrices for every firewall to detect security issues and anomalies. In the case of firewall fw_4, the matrices are:

$$M_{g,m,fw4} = \begin{matrix} \\ r_{3,1} \\ r_{4,1} \end{matrix} \begin{matrix} \begin{matrix} r_{3,2} & r_{4,2} \end{matrix} \\ \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \end{matrix} \quad M_{d,m,fw4} = \begin{matrix} \\ r^{3,1} \\ r^{4,1} \end{matrix} \begin{matrix} \begin{matrix} r_{3,2} & r_{4,2} \end{matrix} \\ \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \end{matrix} \tag{11}$$

$$M_{\rho,m,fw4} = \begin{matrix} \\ r^{3,1} \\ r^{4,1} \end{matrix} \begin{matrix} \begin{matrix} r_{3,2} & r_{4,2} \end{matrix} \\ \begin{pmatrix} 0 & 0 \\ 0 & -1 \end{pmatrix} \end{matrix} \tag{12}$$

where partition $r^{4,1}$ represents (IP_{PMU}, any) and $r^{4,2}$ represents (IP_{PDC}, any, any, mirror, remote IP_{AD}). The reachability analysis detects a mismatch related to Step 1 of the attack: traffic from the phasor measurement unit is allowed to be read by an unauthorized device (identified by value -1 in $M_{\rho,m,fw4}$). The DSS-analyzer then removes rule r_4 that allowed this traffic and thus recovers the original configuration. Also, the DSS-analyzer knows the IP address of the visualization station node (IP_{VS}) that attempted the unauthorized reprogramming action (since the IP address was sent from the DSS-solver). Thus, the DSS-analyzer can notify the administrator of the violation (via SMS) and proceed to close the data path from the given IP address to the wireless sensor network.

- The reprogramming action is not authorized by the DSS-solver, which prevents the malicious code from being injected into the wireless sensor network by the attacker. Thus, the attack is not successful and Step 4 of the attack is not reached. The administrator is also notified about the unauthorized action.

7. Conclusions

The enhanced SIEM system described in this paper is designed to handle cyber security problems encountered in critical infrastructure assets. The SIEM system addresses several limitations of existing systems by leveraging a DSS-solver that resolves security policy conflicts, a DSS-analyzer that discovers unauthorized network data paths and reconfigures network devices when misconfigurations and anomalies occur, and resilient event storage that ensures the integrity and non-forgeability of stored data. The performance and utility of the enhanced SIEM system are demonstrated using a hydroelectric dam. The case study considers an attack model that affects various portions of the information technology infrastructure of the hydroelectric dam and demonstrates that the SIEM system is able to detect and respond to the attacks.

Future research will conduct extensive experimental investigations to analyze the effectiveness of the SIEM

Please cite this article as: C. Di Sarno, et al., A novel security information and event management system for enhancing cyber security in a hydroelectric dam, International Journal of Critical Infrastructure Protection (2016), http://dx.doi.org/10.1016/j.ijcip.2016.03.002

system in critical infrastructure applications. Another research task is to analyze and manage false-positive events in order to enhance the accuracy of attack detection and response.

Acknowledgments

This research was partially supported by the TENACE PRIN Project (No. 20103P34XC) funded by the Italian Ministry of Education, Universities and Research. Alessia Garofalo conducted her research while she was with the Department of Engineering, University of Naples "Parthenope," Naples, Italy.

REFERENCES

[1] M. Afzaal, C. Di Sarno, L. Coppolino, S. D'Antonio, L. Romano, A resilient architecture for forensic storage of events in critical infrastructures, *in: Proceedings of the Fourteenth IEEE International Symposium on High-Assurance Systems Engineering*, 2012, pp. 48–55.

[2] E. Al-Shaer, W. Marrero, A. El-Atawy, K. Elbadawi, Network configuration in a box: towards end-to-end verification of network reachability and security, *in: Proceedings of the Seventeenth IEEE International Conference on Network Protocols*, 2009, pp. 123–132.

[3] D. Carr, Primer: security information and event management, *Baseline*, no. 47, 2005, p. 83.

[4] L. Coppolino, S. D'Antonio, A. Garofalo, L. Romano, Applying data mining techniques to intrusion detection in wireless sensor networks, *in: Proceedings of the Eighth International Conference on P2P, Parallel, Grid, Cloud and Internet Computing*, 2013, pp. 247–254.

[5] F. Cuppens, N. Cuppens-Boulahia, Meriam Ben Ghorbel, High level conflict management strategies in advanced access control models, *Electronic Notes in Theoretical Computer Science*, vol. 186, 2007, pp. 3–26.

[6] H. Debar, D. Curry, B. Feinstein, The Intrusion Detection Message Exchange Format (IDMEF), RFC 4765, 2007.

[7] N. Dunlop, J. Indulska, K. Raymond, Methods for conflict resolution in policy-based management systems, *in: Proceedings of the Seventh IEEE International Conference on Enterprise Distributed Object Computing*, 2003, pp. 98–109.

[8] B. Gertz, The cyber-dam breaks, *The Washington Free Beacon*, May 1, 2013.

[9] A. Lunardelli, I. Matteucci, P. Mori, M. Petrocchi, A prototype for solving conflicts in XACML-based e-health policies, *in: Proceedings of the Twenty-Sixth International Symposium on Computer-Based Medical Systems*, 2013, pp. 449–452.

[10] E. Lupu, M. Sloman, Conflicts in policy-based distributed systems management, *IEEE Transactions on Software Engineering*, vol. 25 (6), 1999, pp. 852–869.

[11] A. Masoumzadeh, M. Amini, R. Jalili, Conflict detection and resolution in context-aware authorization, *in: Proceedings of the Twenty-First Conference on Advanced Information Networking and Applications*, vol. 1, 2007, pp. 505–511.

[12] I. Matteucci, P. Mori, M. Petrocchi, Prioritized execution of privacy policies, in: *Data Privacy Management and Autonomous Spontaneous Security*, R. Di Pietro, J. Herranz, E. Damiani and R. State (Eds.), Springer, Berlin Heidelberg, Germany, 2012, pp. 133–145.

[13] T. Morris, S. Pan, U. Adhikari, N. Younan, R. King, V. Madani, Phasor measurement unit and phasor data concentrator cyber security, in: *Optimization and Security Challenges in Smart Power Grids*, V. Pappu, M. Carvalho and P. Pardalos (Eds.), Springer-Verlag, Berlin Heidelberg, Germany, 2013, pp. 141–159.

[14] M. Nicolett, M. Kavanagh, Magic Quadrant for Security Information and Event Management, Gartner Technical Report, Gartner, Stamford, Connecticut, 2011.

[15] B. Obama, Presidential Policy Directive – Critical Infrastructure Security and Resilience, PPD-21, The White House, Washington, DC, 2013.

[16] T. Saaty, A scaling method for priorities in hierarchical structures, *Journal of Mathematical Psychology*, vol. 15(3), 1977, pp. 234–281.

[17] T. Saaty, How to make a decision: The analytic hierarchy process, *European Journal of Operational Research*, vol. 48(1), 1990, pp. 9–26.

[18] V. Shoup, Practical threshold signatures, in: *Proceedings of the Nineteenth International Conference on the Theory and Application of Cryptographic Techniques*, 2000, pp. 207–220.

[19] E. Syukur, S. Loke, P. Stanski, Methods for policy conflict detection and resolution in pervasive computing environments, in: *Proceedings of the Workshop on Policy Management for the Web*, 2005, pp. 10–14.

[20] U.S. Department of Homeland Security, What is critical infrastructure? Washington, DC ⟨http://www.dhs.gov/what-critical-infrastructure⟩, 2016.

[21] A. Williams, Security information, *SiliconIndia*, vol. 10(1), 2006, pp. 34–35.

[22] G. Xie, J. Zhan, D. Maltz, H. Zhang, A. Greenberg, G. Hjalmtysson and J. Rexford, On static reachability analysis of IP networks, in: *Proceedings of the Twenty-Fourth Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 3, 2005, pp. 2170–2183.