Dynamic security management driven by situations: An Exploratory analysis of logs for the identification of security situations

Abdelmalek Benzekri, Romain Laborde, Arnaud Oglaza, Darine Rammal, François Barrère

Dynamic security management driven by situations: An exploratory analysis of logs for the identification of security situations

Abdelmalek Benzekri, Romain Laborde, Arnaud Oglaza, Darine Rammal, François Barrère
Institut de Recherche en Informatique de Toulouse
Université de Toulouse 3 Paul Sabatier
Toulouse, France
{surname.name}@irit.fr

Abstract— **Situation awareness consists of "the perception of the elements in the environment within a volume of time and space, the comprehension of their meaning, and the projection of their status in the near future". Being aware of the security situation is then mandatory to launch proper security reactions in response to cybersecurity attacks. Security Incident and Event Management solutions are deployed within Security Operation Centers. Some vendors propose machine learning based approaches to detect intrusions by analysing networks behaviours. But cyberattacks like Wannacry and NotPetya, which shut down hundreds of thousands of computers, demonstrated that networks monitoring and surveillance solutions remain insufficient. Detecting these complex attacks (a.k.a. Advanced Persistent Threats) requires security administrators to retain a large number of logs just in case problems are detected and involve the investigation of past security events. This approach generates massive data that have to be analysed at the right time in order to detect any accidental or caused incident. In the same time, security administrators are not yet seasoned to such a task and lack the desired skills in data science. As a consequence, a large amount of data is available and still remains unexplored which leaves number of indicators of compromise under the radar. Building on the concept of situation awareness, we developed a situation-driven framework, called dynSMAUG, for dynamic security management. This approach simplifies the security management of dynamic systems and allows the specification of security policies at a high-level of abstraction (close to security requirements). This invited paper aims at exposing real security situations elicitation, coming from networks security experts, and showing the results of exploratory analysis techniques using complex event processing techniques to identify and extract security situations from a large volume of logs. The results contributed to the extension of the dynSMAUG solution.**

Keywords— security situation; CEP; SIEM; SoC; IoC; APT;

I. INTRODUCTION

Security attacks that were launched by isolated hackers in search of glory have been replaced today by targeted attacks committed by well-organized criminal groups [1].

We moved from Script kiddies to Advanced Persistent Threats (APTs) for years now. If their commonalities are based on the exploitation of vulnerabilities present in the information system, their purpose as well as their means are very different. Script kiddies do not require any expertise and are undertaken by very little technical background, using "off-the-shelf" toolkits and/or mostly copy-and-paste scripts. They are based on zero-day or older vulnerabilities by taking advantage of systems and software lacks of updates. The motivation for the attackers was more the curiosity than the challenge to hack the system; the consequence, a web site defacement or an easy and fast servers compromise in order to scan other Internet servers or to launch spam campaigns. A main characteristic could be the fact that the attackers do not even try to hide their exploits and go unseen. These attacks remain gross and do not generally last in time. Once the goals reached, the attackers pass to the next targets.

APT are more complex. They are based on a sophisticated combination of commercial tools, malwares and rootkits, as well as back doors in order to escalade privileges on the client network and move to other servers. Such an implementation requires qualified technical skills and a far more advanced and professional organization. Based on vulnerabilities that exploits could take advantage of, as for Script kiddies, but also obfuscating the performed actions, and by covering up the traces thanks to anti-forensic techniques, the latter type of attacks has a more severe and destructive impact on the assets as well as on the brand image of the enterprise. The attacks being stealth are generally not detected and maintained for a while in the system. Often by the profit motivated, the pursued goal may be data theft, industrial espionage or sabotage resulting in the destruction of information.

Consequently, security has to evolve on an ongoing basis and the security management process should be flexible enough to quickly adapt the organization security to address these new threats. More precisely, security detection that consisted in manually analysing log files has given way to dedicated detection mechanisms. Security administrators had installed network and host based intrusion detection systems (IDS) that had the ability to perform real-time traffic analysis (e.g., SNORT, Surricata, Bro or OSSEC). However, new attacks being more sophisticated, these IDSs could not detect advanced attacks alone. Such complex attacks require correlating all the security events generated by each IDS in order to be detected. As a consequence, security information and event management systems (SIEM) were introduced on top of IDSs to aggregate/correlate security events and to propose dashboards. Dedicated SIEMS such as OSSIM, PRELUDE, QRadar, RSA or Splunk etc. have been deployed. However, with the growing number of security events to consider, the current strategy is to

978-1-7281-3949-4/19/$31.00 ©2019 IEEE

implement a SIEM by using big data solutions such as the ELK stack (ElasticSearch/ Logstash, Kibana). Indeed, this strategy fits with operational scalability issues by facilitating cloud deployments and by providing rich features for data analytics. We will discuss the needs for SIEMs in security operation centres in section II.

Signature-based detection allows to characterize and isolate many intrusions, but it leaves under the radars many of the advanced persistent threats inventoried so far such as APT32 or Wannacry… There are initiatives being conducted using behavioural analysis techniques to prevent from illegitimate traffics and persistent compromises of the IT services if not the whole information system. We will discuss them in Section III.

Even if an organization is following the best practices approaches, attackers can always exploit zero-day vulnerabilities, or make a social engineering attack to introduce themselves into the network of the organization. Then, in their next step, attackers attempt to persist in the targeted environment, and use different mechanisms to hide their activities and achieve their goals. Attackers use legitimate tools as well for achieving malicious goals, which is hard to detect using the actual available solutions. It is where Threat Intelligence collaborative solutions and platforms are supposed to help in getting valid indicators of compromise (IOCs); however, in the context of APTs, these indicators change quickly. In addition, inspecting the network traffic is useless, since it is encrypted. How to distinguish a successful password based authentication from a legitimate user initiating a remote session from an illegitimate resulting access, successful authentication but with a stolen password? That means how to detect that a user account is being hacked and how to react to such a situation. We will present such a scenario in section IV.

In Section V, we propose a situation-driven framework, called DynSMAUG, for dynamic security management, built on the concept of situation awareness. This approach simplifies the security management of dynamic systems and allows the specification of security policies at a high-level of abstraction (close to security requirements) as considered in section VI.

The effectiveness of the approach and the deployment of the defined protection measures are dealt with in section VI before concluding in section VII.

II. Security Information and Event Management

A complete protection against security incidents is almost impossible. On the contrary, incident detection may launch a fast response at least in order to contain the intrusions on the information system. The infrastructure devices generating large volumes of events, it is mandatory to process them automatically.

A SIEM is a tool dedicated to this task. It allows the monitoring of the information system events in real-time. The objective of SIEM systems is to monitor the whole IT infrastructure (network devices, PCs, data stores, application servers, smartphones etc.) in order to investigate and look for evidence when required [2]. A SIEM should provide the functionalities to:

- Aggregate security events from many sources. Standard formats for the exchange of security events have been proposed at the IETF level such as Intrusion Detection Message Exchange Format (IDMEF - RFC 4765) or Incident Object Description Exchange Format (IODEF - RFC 5070)
- Correlate a large mass of events to make real-time searches
- Safeguarding long-term security information especially for doing digital forensics
- Alert the network administrator automatically when a potential attack is detected
- Provide useful security information in dashboards to facilitate the automatically generated alerts display and assisting security administrators in manual/automated search when a potential attack has been detected

SIEMs are usually used for monitoring:

- authentication activities (abnormal authentication, non-working hours)
- shared accounts (session requests for the same user account)
- sessions activities
- connections details (closed ports, internal blocked)
- abnormal administrator's behaviour (on inactive admin accounts, unusual activities)
- information theft (data exfiltration, data leaks)
- vulnerabilities scans and correlation (suspicious events)
- statistical analysis (inbound/outband throughput, data per application)
- intrusion detection et infection (IDS data, IPS, antivirus, application anti-malware)
- system modifications (by the means of configuration change data)

To work efficiently, a SIEM needs to integrate different sources of logs and events (servers logs, routers logs, systems logs, applications/services logs etc.) by the means of so-called connectors. These connectors, in general, provide a standardised representation of the logged events and their recorded attributes (timestamps, addresses, etc.).

Rules detection describes the suspicious behaviour and the method of an attack against all monitored devices and increasingly end-systems or endpoints. The monitoring agents raise alerts based on IOCs – Indicators of Compromise. IOCs may be used to refer to specific artifacts left by an intrusion, or greater sets of information that allow for the detection of intrusions or other activities conducted by attackers.

The rules may consider attributes like Windows Security Logs events, a malicious IP address or domain name, URLs, a file hash, an upload exceeding a size limit etc. These rules are defined and set after a risk analysis to deploy the right security controls and measures.

SIEM analysts define also the investigation and forensic process. The aim of this paper is not to recall the processes of the incident & response platforms. But we think it is relevant to insist on the collaborative and shared initiatives allowing for an automated and better detection of malicious behaviors.

III. RELATED WORK

The GitHub web site APTnotes [3] intends to inventory various public documents, whitepapers and articles about APT campaigns since 2008 until 2017 with the purpose to provide the first threat intelligence exchange open platform publicly available. Most of the documents draw the attacks appearance and discuss their operational mode (strategy, intrusion, infection, propagation, mutation…), their lifecycle etc.

OpenIOC [4] is a threat information sharing standard that allows one to logically group forensic artifacts, and communicate this information in a machine readable format. The terms are sometimes used interchangeably, but an IOC is a logically grouped set of descriptive terms about a specific threat while OpenIOC is the language used to describe those specific sets (e.g. an incident response team would use the OpenIOC format to write multiple IOCs during the course of responding to an incident). SOC analysts and experts will have an XML-based schema framework allowing for advanced threat detection capability available.

The MITRE Att&ck project [5] consists in gathering the knowledge about adversary tactics and techniques as observed in the real-world. CybOX™ International [6], in scope and free for public use, is a standardized schema for the specification, capture, characterization, and communication of events or stateful properties that are observable in the operational domain. STIX™ [7] is a collaborative community-driven effort to define and develop a standardized language to represent structured cyber threat information. TAXII™ [8] defines a set of services and message exchanges that, when implemented, enable sharing of actionable cyber threat information across organization and product/service boundaries. CAPEC™ [9] for Common Attack Pattern Enumeration and Classification intends to create a community resource for identifying and understanding attacks. Understanding how the adversary operates is essential to effective cyber security. CAPEC™ helps by providing a comprehensive dictionary of known patterns of attacks employed by adversaries to exploit known weaknesses in cyber-enabled capabilities. It can be used by analysts, developers, testers, and educators to advance community understanding and enhance defenses.

Finally, the MISP open source threat intelligence platform and open standards for threat information sharing stands for Malware Information Sharing Platform. It is more and more deployed as the threat intelligence platform for sharing, storing and correlating Indicators of Compromise of targeted attacks, threat intelligence, financial fraud information, vulnerability information or even counter-terrorism information.

IV. USE CASE SCENARIO

Detecting a legitimate password-based authentication from an illegitimate one is challenging. This issue was raised by the security administrators from our University who are concerned by detecting hacked accounts.

Currently, there exist some markets on the dark web where anyone can easily buy credentials [10]. Universities are targeted by phishing attacks and some users' credential might be stolen and stored in such databases. Even if the security policy constrains the University staff to change the passwords regularly, it is possible that illegitimate users can successfully authenticate into the information system. Detecting these connections is complicated. A first idea could be blocking connections coming from specific countries. Indeed, some countries are known for being the origin of cyberattacks [11]. However, researchers are travelling in these countries for attending conferences which makes this approach impracticable.

As a consequence, the solution consists in analyzing the behavior of the accounts. When suspicious activities are detected, appropriate security measures are enforced (e.g., blocking the account). However, hackers do not launch immediately their attack right after buying the stolen account information. They first start by testing whether the account information is valid or not. Valid accounts will be used later in future attacks. Since there is no suspect activity on the accounts not already used for an attack, the task of the security administrators is much more complex. The current practice consists in, when a suspect activity is performed on an account, to retrieve all the accounts that were successfully authenticated by the same origin as the account with the suspect activity and enforce on these suspicious accounts the necessary security measures (e.g. change the password immediately).

V. EXPRESSING SITUATIONS USING COMPLEX EVENTS

We developed a situation-awareness approach [12, 13, 14] to deal with such complex security management issues. In this section, we specify this use case scenario as a security situation-based model. We also present how complex event processing techniques can compute security situations involving historical events to detect APTs.

A situation is a specific time frame during which the result of some computed relationships between parts of the collected security events is stable. A security situation focuses on specific entities of interest to protect, which we call *situation target*, and a particular *security concern*.

The situations related to the *same situation target* and the *same security concern* are in the same *situation class*. A situation class forms a weakly connected graph where vertices are the situations of the class and edges are context events impacting the stability of the situations. Situations of the same class are mutually exclusive, i.e., an entity can only be in a single situation of a given class at the same time. However, any entity can find itself in several situations in parallel provided all these situations belong to different classes.

A situation, being a particular time frame of interest, has a beginning, a life span and an end [15]. The beginning and the end of a situation are determined by combining multiple events coming from multiple sensors and occurring at different moments. Indeed, a situation involving multiple entities and multiple conditions, the beginning and the end of a situation cannot be simple events captured by a single sensor. In addition, events being instantaneous, combining multiple events requires complex temporal operators (event ordering, event existence/absence, time windows, etc.) to specify the beginning and the end of situations.

We follow the Complex Event Processing (CEP) approach for computing the beginning and the end of situations. CEP is "a defined set of tools and techniques for analysing and controlling

the complex series of interrelated events that drive modern distributed information systems" [16]. CEP solutions allow specifying complex events through complex event patterns that match incoming event notifications on the basis of their content as well as some ordering relationships on them. We choose Esper[1], the open source event processing implementation maintained by Espertech. Esper offers a stream-oriented language for specifying complex event patterns, called Event Processing Language (EPL), that is an extension of SQL for processing events (e.g., windows definition and interaction, timed-data arithmetic definition, etc.)

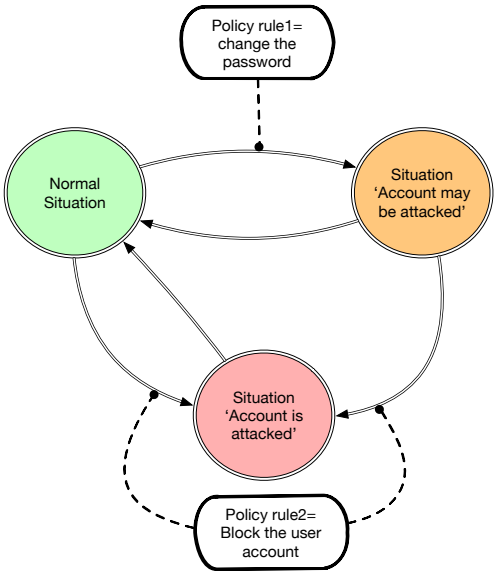


Figure 1. Situation class for account security level

```
new-situation-attacked-because-virus =
"insert into situation(accountName, hackerIP,
situationValue, state, reason)"+
"select accountName, hackerIP, 'account-attacked' as
situationValue, 'start' as state, 'detected by
antivirus' as reason from virus"

new-situation-may-be-attacked-because-dangerous-ip=
"insert into situation(accountName, hackerIP,
situationValue, state, reason)"+
"select cq.accountName as accountName, cq.ip_src as
hackerIP, 'account-maybe-attacked' as situationValue,
'start' as state, 'account contacted be hacker' as
reason from situation as cce left outer join"+
"sql:ESPER['select accountName, ip_src from LOGS
where ip_src = ${cce.hackerIP}'] as cq on cce.hackerIP
= cq.ip_src where cce.situationValue='account-
attacked' "
```

Figure 2. Sample of the situations specification in EPL

In our scenario, a user account can be in three different situations regarding the security issue (Figure 1):

- Situation 'normal' is the nominal behaviour where no abnormal activity or attack has been detected
- Situation 'account is attacked' occurs when the administrator has enough evidence that the account is being attacked and may be under control of a hacker
- Situation 'account may be attacked' happens when an abnormal or potentially dangerous activity related to this account has been detected

Specifying the situations using CEP consists in describing complex events by combining security events that constitute the edges of the situation class graph. For instance, the situation of a user account moves from situation 'normal' to situation 'account is attacked' when a strong security indicator is activated, an antivirus or anti-rootkit throwing security context events. This could be easily expressed in ESPER EPL by the statement `new-situation-attacked-because-virus` in Figure 2. This statement generates a new situation each time the CEP engine receives a `virus` event. Situations are stored as a table by the CEP engine. As a consequence, creating new situations consists in inserting new records in the situations table (`insert into situation(accountName, hackerIP, situationValue, state, reason)`). The transition between situation 'normal' and situation 'account may be attacked' is more complex. In our case, the security administrators have decided that a user account should be considered as potentially dangerous and requires specific security measures when a remote connection was successful in the past and the remote IP address is involved in a recent attack. This requirement is much more complex because it requires to combine recent and historical events. In our scenario, we consider that logs containing all the successful accesses are stored in an SQL database. The EPL rule `new-situation-may-be-attacked-because-dangerous-ip` defines the following statement. Each time a situation '`account-attacked`' occurs, the ESPER engine retrieves all the accounts successfully accessed by the same IP address (`cce.hackerIP = cq.ip_src`) from the LOGS table. The situation of all these accounts is changed to '`account-may-be-attacked`' by throwing a complex event for every account.

VI. Expressing Dynamic Security Measures Driven by Situations

In our approach, situations are specified outside the security policy and the security policy needs only to refer to them. Hence, security policies are defined in a generic way as: ***when*** *situation* ***and*** *some condition* ***then*** *authorization decision and/or obligation(s)*. This pattern allows the security administrator to specify both reactive and authorization rules:

- *reactive rules* : **when** situation **and** situation begins [and some condition] **then** obligation(s)
- *authorization rules* : **when** situation **and** some condition **then** authorization decision and/or obligation(s)

XACMLv3 [17] allows the security administrators to express such kind of policies. First, it follows the Attribute Based Access Control (ABAC) approach [18] where policies describe general access control requirements in terms of constraints on security attributes; attributes being any

[1] http://www.espertech.com/esper/

characteristics of entities. In addition, the XACMLv3 policy language includes obligations. Thus, XACMLv3 is not limited to PERMIT/DENY decisions only and can also describe complex decisions involving the modification of managed entities.

In our scenario, the security administrator should specify two reactive rules (Figure 1):

- Policy rule1 applies when the situation of an account changes to '`account-may-be-attacked`'. The security management system must react by launching the user password modification procedure. Following the situation driven policy pattern, this rule is specified as: **when** situation is 'account-may-be-attacked' **and** situation starts **then** obligation 'modify the password of the account'.
- Policy rule2 shall be enforced when the situation of an account becomes '`account-attacked`'. In this case, the security management system must block the account. Similarly as policy rule2, this rule is a reactive policy rule and can be specified as: **when** situation is 'account-attacked' **and** situation starts **then** obligation 'block the account'. Figure 3 shows the XACMLv3 specification of policy rule 2 ('`account-attacked`').

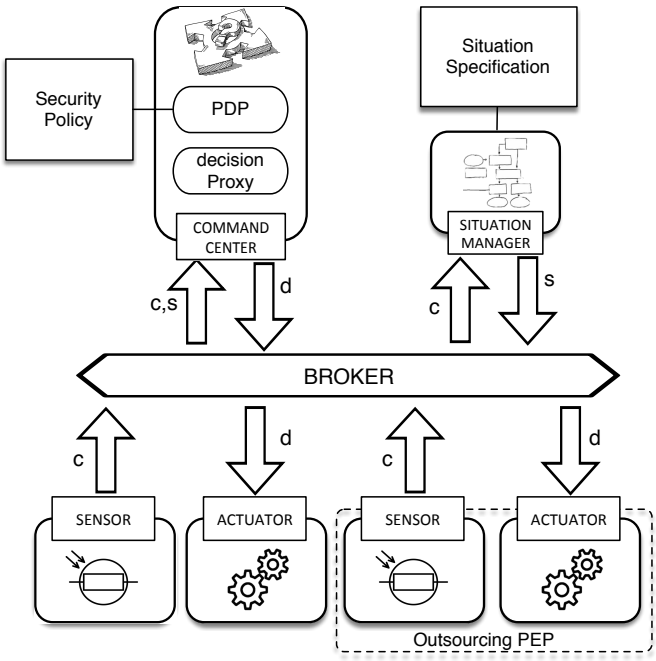
```
<xacml3:Rule Effect="Permit" RuleId="account_attacked">
    <xacml3:Description/>
    <xacml3:Target>
        <xacml3:AnyOf><xacml3:AllOf>
                <xacml3:Match
                    MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
                    <xacml3:AttributeValue
                        DataType="http://www.w3.org/2001/XMLSchema#string">
                            account_attacked
                        </xacml3:AttributeValue>
                    <xacml3:AttributeDesignator
                        AttributeId="urn:siera:situation_account:value"
                        Category="urn:siera:name:attribute-category:situation"
                        DataType="http://www.w3.org/2001/XMLSchema#string"
                        MustBePresent="true"/>
                </xacml3:Match>
                    <xacml3:Match MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
                    <xacml3:AttributeValueDataType="http://www.w3.org/2001/XMLSchema#string">
                            start
                        </xacml3:AttributeValue>
                        <xacml3:AttributeDesignator
                        AttributeId="urn:siera:situation_account:state"
                        Category="urn:siera:name:attribute-category:situation"
                        DataType="http://www.w3.org/2001/XMLSchema#string"
                        MustBePresent="true"/>
                </xacml3:Match>
        </xacml3:AllOf></xacml3:AnyOf>
    </xacml3:Target>
    <xacml3:ObligationExpressions>
        <xacml3:ObligationExpression
            FulfillOn="Permit"
            ObligationId="block_account">
            <xacml3:AttributeAssignmentExpression
                AttributeId="urn:siera:name:attribute:block_account:account-id"
                Category="urn:siera:name:attribute-category:attributes">
                <xacml3:AttributeDesignator
                    AttributeId="urn:siera:situation_account:account-name"
                    Category="urn:siera:name:attribute-category:situation"
                    DataType="http://www.w3.org/2001/XMLSchema#string"
                    MustBePresent="true"/>
            </xacml3:AttributeAssignmentExpression>
            <xacml3:AttributeAssignmentExpression
                AttributeId="urn:siera:name:attribute:block_account:reason"
                Category="urn:siera:name:attribute-category:attributes">
                <xacml3:AttributeDesignator
                    AttributeId="urn:siera:situation_account:situation-reason"
                    Category="urn:siera:name:attribute-category:situation"
                    DataType="http://www.w3.org/2001/XMLSchema#string"
                    MustBePresent="true"/>
            </xacml3:AttributeAssignmentExpression>
        </xacml3:ObligationExpression>
    </xacml3:ObligationExpressions>
</xacml3:Rule>
```

Figure 3. XACMLv3 policy rule '`account-attacked`'

VII. DEPLOYING DYNAMIC SECURITY MEASURES DRIVEN BY SITUATIONS

In [12], we developed dynSMAUG a situation-driven framework for dynamic security management. Its architecture aims at allowing the deployment of security policies including authorization and obligation policies. The deployment architecture consists in the following actors:

- The *broker* is the distribution middleware that transmits all the events between the actors following the publish-subscribe pattern. The broker divides events into three kinds of topics: the context events, the situation events and the decision events. It is also responsible for controlling the access to the dynSMAUG infrastructure (each dynSMAUG actor is authenticated by an X.509 certificate).
- The *sensors* produce context events (noted c in **Erreur ! Source du renvoi introuvable.**4). Context events are every instantaneous, detectable, and relevant security related information of the managed environment collected by sensors such as monitoring systems, intrusion detection systems, configuration management databases, etc. Each event is defined in XACMLv3 by a set of attributes of the form <identifier, type, value>. This solution has an advantage: it is possible to develop sensors in any programming language.
- The *situation manager* contains a CEP engine that calculates situations according to a situation specification. It consumes context events and produces situation events (noted s in **Erreur ! Source du renvoi introuvable.**4). Situation events have the same format as context events and are also carried in XACMLv3 format. Each time a new situation is calculated, the situation manager creates two situation events: the beginning of the new situation and the end of the last active situation.
- The *command centre* is the brain of our security management framework. It consumes both context and situation events and produces decision events (noted d in **Erreur ! Source du renvoi introuvable.**4). We specify security policies in XACMLv3. However, XACML PDPs only implement the outsourcing mode. Therefore, the command centre includes also a decision proxy for allowing the command centre to operate both authorization and obligation policies. Hence, the PDP acts in compliance with XACMLv3. When the decision proxy receives the decision from the PDP, it publishes the decision to the correct topic(s) based on attributes *recipientTarget* and *recipientType* to distribute it to the relevant actuators.
- Finally, the *actuators* only consume decision events. An actuator checks if it is the recipient of the decisions and enforces them if so. Like sensors, it is possible to develop actuators in any programming language.



c: context events / s: situation events / d: decision events

Figure 4. The dynSMAUG deployment architecture

In our scenario, the anti-virus or the anti-rootkits are sensors. When they detect a fraudulent activity, they send a context event on the broker. This event is received by the situation manager that executes the EPL statements described Figure 2. The situation manager generates an event indicating that the situation of the account is changing to 'account-attacked' on the broker (EPL statement `new-situation-attacked-because-virus`). In parallel, it retrieves the accounts that were accessed by the same IP address and generates an event informing that situation is now 'account-may-be-attacked' for each matching account (EPL statement `new-situation-may-be-attacked-because-dangerous-ip`).
When the command centre receives these events, it consults its policy and generates decision events to block the attacked account and modify the password of the suspicious accounts. Finally, the appropriate actuators enforce the decision on the managed system.

VIII. Conclusion

Detecting and thwarting complex cyber-attacks requires security administrators to retain a large number of logs in case problems are detected latter and involve the investigation of past security events. Indeed, benign activities might actually represent early stages of an attack. However, this approach generates massive data and appropriate techniques have to be employed to analyse it and react accordingly.

In this article, we explained how our situation-based security management approach can facilitate the detection of complex attacks and the enforcement of dynamic security measures accordingly. Complex event processing provides analytics tools to specify complex statements able to analyse current and historical events in a unified language. We built a security situation manager that can handle long range attacks. When it is coupled with a security management infrastructure, dynamic security measures can react automatically to multistage attacks. We applied this idea to detect and react to a legitimate password-based authentication from an illegitimate user use case.

The security decision-making process must consider that security sensors (logs, IDS, etc) provide intrinsically inaccurate, erroneous and ambiguous information. Indeed, bad or misinterpreted security information can lead to mistakenly believe the system is in a specific situation, which results in wrong decisions and irrelevant reactions at the end. Currently, SIEMs only consider basic information to calculate the reliability of security events. A more expressive representation is required for describing the complexity of the reliability of security event. We will propose a generic expression of security events reliability artifacts based on existing research on Quality of Context. Then, we will enhance the situation calculus to handle security events reliability and the decision making process to support different situation trust levels.

References

[1] Interpol. (2015). « Cybercrime » http://www.interpol.int/ Crime-areas/Cybercrime/ Cybercrime.
[2] Scarfone, K., and Mell, P. Intrusion Detection and Prevention Systems. In Handbook of Information and Communication Security, P. Stavroulakis and M. Stamp, Eds. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010, pp. 177–192.
[3] APTnotes, https://github.com/kbandla/APTnotes, last access September 2019
[4] OpenIOC, https://github.com/mandiant/OpenIOC_1.1, last access September 2019
[5] MITRE Att&ck, https://attack.mitre.org/, last access September 2019
[6] CybOX™ Cyber Observable eXpression, , https://cyboxproject.github.io/, last access September 2019
[7] OASIS STIX™ Structured Threat Information eXpression version 2.0, Part 1: STIX Core Concepts, OASIS standard, July 2017,
[8] OASIS TAXII™ Trusted Automated Exchange of Intelligence Information Version 2.0, Working Draft 02, OASIS, July 2017.
[9] Mitre CAPEC™ , Common Attack Pattern Enumeration and Classification, https://capec.mitre.org/, last access September 2019.
[10] Cal Jeffrey, "Credentials to an airport's security systems sold on the dark web for $10", techspot.com, 2018, Available: https://www.techspot.com/news/75462-credentials-airport-security-systems-sold-dark-web-10.html
[11] "Top 10 Countries Where Cyber Attacks Originate" 2013, available: https://www.govtech.com/security/204318661.html
[12] Laborde, R., Oglaza, A., Wazan, A. S., Barrère, F., & Benzekri, A. (2019). A situation-driven framework for dynamic security management. Annals of Telecommunications, 74(3-4), 185-196.
[13] Kabbani, B., Laborde, R., Barrere, F., & Benzekri, A. (2014, March). Specification and enforcement of dynamic authorization policies oriented by situations. In 2014 6th International Conference on New Technologies, Mobility and Security (NTMS) (pp. 1-6). IEEE.
[14] B. Kabbani, R. Laborde, F. Barrère, and A. Benzekri, "Managing Break-The-Glass using Situation-oriented authorizations," in 9ème Conférence sur la Sécurité des Architectures Réseaux et Systèmes d'Information-SAR-SSI 2014, 2014.
[15] A. Adi and O. Etzion, "Amit - the situation manager," The VLDB Journal—The International Journal on Very Large Data Bases, vol.

13, no. 2, pp. 177–203, 2004.
[16] D. Luckham, "The power of events: An introduction to complex event processing in distributed enterprise systems," in Workshop on Rules and Rule Markup Languages for the Semantic Web. Springer, 2008, p. 3.
[17] OASIS, "eXtensible Access Control Markup Language (XACML) Version 3.0," Tech. Rep., 2013. [Online]. Available: http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-en.pdf
[18] V. C. Hu, D. Ferraiolo, R. Kuhn, A. Schnitzer, K. Sandlin, R. Miller, and K. Scarfone, "Guide to Attribute Based Access COntrol (ABAC) Definition and Considerations," NIST, Tech. Rep. SP 800-162, 2016.