Event Driven Adaptive Security in Internet of Things

Waqas Aman and Einar Snekkenes

Norwegian Information Security Laboratory (NISLab)
Gjøvik university College, Norway
Email: {waqas.aman, einar.snekkenes}@hig.no

***Abstract*—With Internet of Things (IoT), new and improved personal, commercial and social opportunities can be explored and availed. However, with this extended network, the corresponding threat landscape will become more complex and much harder to control as vulnerabilities inherited by individual *things* will be multiplied. Conventional security controls, such as firewalls, intrusion detection systems (IDS) etc., may show some level of resistance to this self-organizing network but, as standalone mechanisms, are not sufficient to analyze the threat in a particular context. They fail to provide the essential context of a threat and yields false positives-negatives which can trigger pointless re-configurations, service unavailability and end user discomfort. Such unwanted events can be very catastrophic, for instance, in an IoT enabled eHealth services. We need to have an autonomous adaptive risk management solution for IoT, which can analyze an adverse situation in a distinct context and manage the risk involved intelligently so that the end user, service and security preferences are well-preserved. This paper details an event driven adaptive security model for IoT to approach the objective specified and explicates how it can be utilized in an eHealth scenario to protect against a threat faced at runtime.**

***Keywords*—Adaptive Security; Internet of Things; Event Correlation; eHealth; Ontology.**

I. INTRODUCTION

According to an analysis conducted by the International Data Corporation (IDC), the IoT expected install base will consist of approximately 212 billion things among which 30.1 billion will be autonomous [1]. Indeed, IoT has the potential to create new huge opportunities for personal, business and social services. However, the research this far is still inconclusive on various topics, such as standardization, networking, QoS, etc., among which security and privacy are the most challenging [2].

Things carry inherited vulnerabilities and corresponding threats. Physical exposure, user lack of knowledge, unattended management, remote implementation, communicating wirelessly, low resources, etc., are the common weaknesses which are mostly exploited when devices at the edge of the network are attacked. Bringing them to the IoT will make the threat faced more complex and hard to control. Traditional controls, such as IDS, Antiviruses, etc., as standalone measures may provide protection to some level but are limited in providing a clear context of a situation. As a result, false positives and negatives are triggered and create service disruptions, unnecessary changes and sometimes panic [3]. For instance, an IDS trigger a critical alarm that someone is trying a port scan looking for an open File Transfer Protocol (FTP) port and suggest to close that immediately. This might take the administrator to a total panic situation, and he might close the port on the file server without the fact that it is adequately protected by a strong password. Thus, a simple lack of contextual information might yield to service disruption and panic.

An effective way to approach this problem will be to collect the appropriate network and system information (status or any changes), analyze them in a context and decide an action accordingly. This approach is called adaptive security or adaptive risk management. It is the process of understanding, analyzing and reacting to an adverse situation in a particular context [4] and can be seen in a number of proposals, such as, [5][6]. Common problems with these models are, either they focus on only one security service, such as authentication, or provide a generic architecture without detailing the methods used within each architectural component. Also, existing approaches are either focused on threat analysis or adaptation individually. We realize an absence of a model with specific methods to address and connect both analysis and adaptation as a holistic solution to the problem. Hence, we approach these issues as a set of two questions, i.e., *how to monitor and collect security changes in a real time and analyzed them in a specific context?* And, *how can the analyzed information be used to adapt security settings such that user and service preferences are preserved?*

In this paper, we address the first question by utilizing Open Source Security Information Management (OSSIM) [7], which provides a platform to filter and normalize primitive events collected from *things* in the monitored scope. Correlation directives are specified to model adverse situations in which security events are correlated and analyzed in a particular context. The adaptation question is addressed by utilizing a proposed Adaptation Ontology which leverages on the risk information from the event correlation and adapt security settings accordingly. Using the ontology an optimum mitigation action is selected from an action pool in a manner such that its utility, in terms of usability, QoS and security reliability, is maximum among the possible actions as per user requirements.

The main contribution of this paper is our autonomic security adaptation ontology. OSSIM does not provide such capability and relies on manual reconfigurations which may not address user and service requirements. Also, OSSIM is focused on the traditional computing environment including

 ISBN: 978-1-61208-353-7

servers, desktops and corresponding applications where event processing is relatively a common task. This paper extends event driven security to the IoT where environment becomes more complex due to things diversity and mobility for which traditional protocols and tools seem to be inefficient to approach event processing. Hence, the concept of the paper itself can be considered as contribution.

The rest of the paper is structured as follows: In Section II, work related to event monitoring, correlation and adaptation is presented. The proposed Event Driven Adaptive Security model is detailed in Section III. In Section IV, an eHealth case study will be presented to show how the model can be utilized to protect against a threat at runtime. Finally, the paper will be concluded in Section V along with an overview of our near future plan.

II. Related Work

The related work is categorized into three major areas of relevance, i.e., event monitoring, event correlation and security adaptation in order to get a clear understanding of the specific methods used.

A. Event Monitoring

The objective of monitoring is to collect primitive events from various sources in the environment, filter out the unwanted, categorize them into interested areas of investigation, such as authentication, routing, confidentiality, etc., and normalize them to a common language specification for further analysis. In most of the event driven architecture (EDA), this phase is considered to be a typical task yet, requires knowledge of the target system event specification.

1) Event Collection: The two common approaches are agent-based and agent-less collection. An agent is a small additional program that is installed on the monitored source in order to collect and send events or log files remotely [8]. Agents can be customized to accomplish more specific objectives. The agent-less approach does not require any additional component to be installed. Instead, it utilizes built-in protocols and services, such as System Log (Syslog), Windows Management Instrumentation (WMI), SNMP, etc., to store, access and communicate information at different levels of a monitored system in a standardized manner [9].

One has to address the attributes of flexibility, lightweight, platform in-dependency and management when either of these approaches is adopted. With agent-based, the first three properties can be somehow achieved using expert skills, open source tools and libraries; however, it will be quite a challenge to manage agents across a complex network. The management and control issues can be complex when it comes to a network like IoT. Agent-less approach faces the problem of detail customization thus lacks flexibility and might require additional tools for detail diagnosis [8].

Many commercial and open source event analysis tools, such as [7], use mixed strategies to overcome the flexibility and cross-platform issues. However, most of them use third party apps, for instance, [10][11], where updating and controlling is still a matter of discussion. In [8], the author presented an order-based approach which can provide all the mentioned properties by defining a monitoring scope and using system utilities. However, the method applies only to distributed computing environment where diagnosis utilities are supposed to be already in use. The approach apparently shows lacking when considered in the IoT environment where the monitored objects are more likely to be low-end and resource-less sensors.

2) Event Filtering : The objective of event filtering is to discard the redundant or unwanted events [12]. It defines the targeted event scope to be investigated. Filtering is normally achieved using regular expression where a pattern is matched against the collected events. Non matched events are dropped as redundant events. Two important issues that need to be addressed here are: what events are redundant and how to assure minimal information loss during the process? [13].

The authors in [14] explain that event redundancy scope can be defined using two approaches. Temporal filtration can be used to filter out events generated repeatedly over time with the same information. On the other hand, spatial filtration can provide a mechanism to remove similar event reported by a different system within a given time frame, t. They also propose casual filtration where events collected from different sources are removed based on the fact that they may have different syntax but conveys the same semantics.

Threshold values or time frames can be maintained in temporal or spatial filtration techniques to guarantee minimal information loss. Such flags and offsets will ensure that the information contained in the event will not change potentially and will also take into considerations, e.g., compression rates [13][15].

3) Event Classification: Event classification seems to be based on primitive knowledge about events. Every event generated and stored by a source has a unique set of attributes which can be used to classify an event, for instance, see event structures [16][17]. These attributes designate the event source/destination, timestamps, type, user IDs and the event severity level whose ranges changes as per the source event model and specification.

B. Event Correlation

Correlation is the heart of EDA. It aims to investigate a complex relationship among events and assist to provide enough contextual information to analyze errors, bugs and security threats . Broadly, correlation methods can be classified into two categories, Deterministic and Anomaly-based, either of which can observe events in spatial, temporal or both of the domains [18]. Both the approaches have their associated advantages and disadvantages. Thus, qualifying which of them is a better approach can be determined by evaluating them in a specific application domain [19].

1) Deterministic Approach: In deterministic approach, a predetermined knowledge is utilized to observe and evaluate a given situation. A knowledge base is maintained with application specific information, which is accessed whenever a

 ISBN: 978-1-61208-353-7

particular event pattern is matched. So as a fact, a more expert knowledge can analyze a given security threat, problem or situation more precisely. The knowledge itself and the control to it can be characterized in a number of ways as discussed underneath:

Rule-base Correlation: Rule-based event correlation or threat analysis is the most common way to implement deterministic approaches. Most IDS and security event monitoring tools, for instance [7][20][21], uses a rule based correlation to analyze a threat faced. The knowledge is represented in the form of a predefined rule set which dictates defined alarms and alerts when a specific condition during analysis is met.

State Machine Automata based Correlation: Finite State Machine (FSM) is used to study the behavior and state of underlying systems. In the context of event correlation, various defined states for a system behavior (normal and abnormal) are designed and stored as knowledge base as FSM tuples [22]. A runtime diagnosis engine observes user, application and device behavior and foresees the next system state. Alerts are generated as a flagged state is or about to be triggered. Some of the event correlation models proposed on FSM are [23][24].

The Codebook/Correlation Matrix Techniques: The codebook approach utilizes a symptom-problem relationship. Different suspected events (symptoms) are mapped to their associated abnormal behaviors (problems) and are stored as a knowledge base in a binary matrix, called correlation matrix or a codebook. Events generated are matched against this matrix to identify associated threats or problems. Event correlation models based on codebook techniques can be found in [18].

2) Anomaly-based Approach: Computing and networking environments are very dynamic and the attack vector changes frequently. Some events may not provide certain information and are thus subjected to probabilistic correlation and processing to resolve the uncertainty problem [19]. Unlike predetermined situations in deterministic methods, anomaly-based event correlation aims to identify anomalies without any prior knowledge and can be used to analyze unknown threats. However, they inherit the problems of generating false positive alarms.

Statistical Correlation: As mentioned earlier, events can be filtered, categorized and correlated in both time and space domains to extract rich contextual statistics. For instance, grouping the number of repeated login failure attempts events can provide credible statistics on whether the attempt is a legitimate or that somebody is trying to break-in using a guessing, dictionary or brute force method. High level events, such as alarm/alerts, generated by various security controls, such as IDS, can be used to perform statistical correlation. Statistical information can also be drawn from diverse events having similar attributes/parameters, such as event source, destination, timestamps, etc. Mostly used in anomaly based IDS, these attributes are used as random variables which are later utilized in statistical inferences [25][26].

Probabilistic Modeling: Bayesian networks tend to model relationship among interested random variables. Events can be mapped to random variables. Bayesian model can be illustrated as directed acyclic graphs where nodes represent events of interest and the connecting edges represent the relationships or inter-dependency between them. The probability of a node (situation or event) is inferred by utilizing conditional probability assigned to each node (event) in a given network (scenario) [27]. In most cases Bayesian modeling is coupled with other models techniques, such as Hidden Markov Model and Kalman filters, to investigate complex events in depth [18].

C. Security Adaptation

Assuming that during the analysis an adverse situation or a risk has been discovered, what choices do we have to adapt the security in accordance? How can we utilize the information or context of the analyzed risk to adapt our security? Following is a list of approaches that can be used to answer these questions.

1) Security Policies: Policies remained one of the earliest methods to dictate an action against a given situation. They are a set of rules specifying how a particular situation should be tackled. Edwards et al. in [28] pointed that security policies can be divided into three groups, fixed (e.g., kernel level implementation), customizable (e.g., firewall, router ACLs, etc.) and dynamic, based on the flexibility they offer. Dynamic policies can be detailed on individual user or service level thus providing more flexible adaptation. Some related work include [29][30].

Utility and Probabilistic Models: Utility expresses the measure of efficacy or profit of a choice for a given user or service. In event driven adaptive security, adaptive decisions can be expressed in utilities on the basis of user acceptance, accuracy, power usage, etc. for a given analyzed risk (event). For instance, Alia and Lacosta in [31] used various QoS and security properties corresponding to a required security service to manipulate the utility of an autonomic adaptive response using a non-probabilistic (utility) predictor function. Probabilistic models of utility, such as, [32][33], provides a fair understanding of how security and trust adaptation can be modeled with utilities.

Besides utility theory, probabilistic models such as Bayesian Networks have also been used in a variety of adaptive applications. Bayesian models can be used to select a suitable algorithm from available list [34]. They can also be advantageous in rules discovery [35] to resolve a conflict where an analyzed risk (high level event) two different rules under a given policy [36]. Game theoretic models have also been proposed where intrusion and defense are modeled as games to adapt and defend system security [37][38][39].

Ontologies: Ontologies are used to capture and structure the knowledge about entities, instances and their relationship within an organization. They can be used both for design and runtime purposes [40]. In [41], the author describes an ontology where the knowledge required for security adaptation such as risk, security services and metrics, etc., are related to be assessed at runtime. Denker et. al [42], the authors used security ontologies for annotating functional aspects of electronic resources. However, these ontologies did not discuss

 ISBN: 978-1-61208-353-7

how user requirements and preferences should be valued during the adaptation.

III. The Model

The model presented, Event Driven Adaptive Security (EDAS), addresses the notion of security adaptation in IoT as an EDA in feedback loop manner. We believe that the basic element of change available within the network is the event generated by various application and devices recorded into log files. They provide a primitive context about *who*, *when*, *where* and *what* of a change and contain vital information, such as timestamps, sources, destinations, user activity, severity levels, etc., necessary to reason about the risk situation associated with an event.

EDAS uses Open Source Security Information Management (OSSIM)[7] which provides a platform for writing scripts, called *plugins*, to filter and normalize primitive security events collected from the monitored sources. Correlation in OSSIM is supported with XML rules through which specific situations, in both temporal and spatial view, can be modeled to correlate and investigated events for potential security risks. The model utilizes a runtime adaptation ontology to adapt a best mitigation action from the available actions based on the stored user and service preferences and risk information produced by the correlation engine. A reference model is shown in Figure 1. It includes three major components Monitor, Analyzer and Adaptor. The input, method(s) utilized by individual component along with the details of the output they produced are explained below:

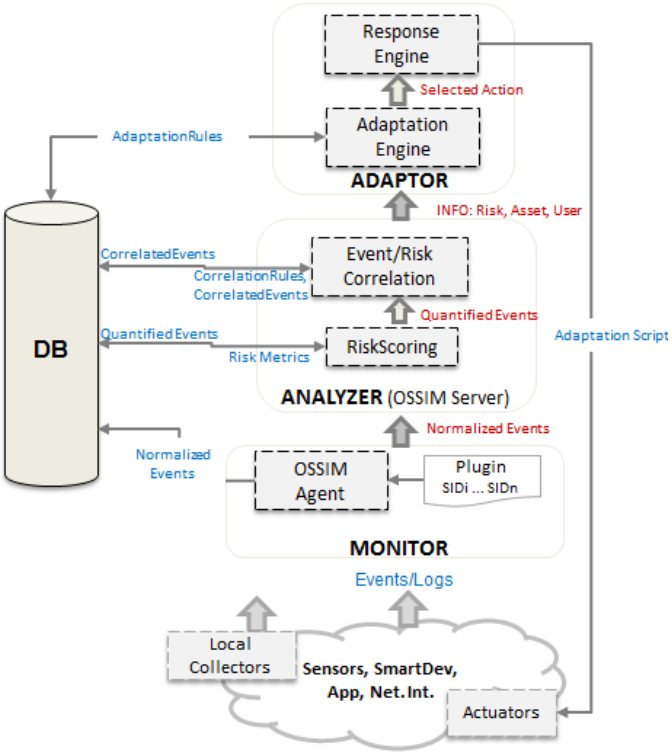

Fig. 1: Event Driven Adaptive Security-Reference Model

A. Monitor

The monitor, OSSIM Agent, collects various events (logs) from diverse things in the IoT, filters the unwanted events and normalizes them to a common language for correlation (analysis).

1) Event Collection: Events generated by monitored *things*, e.g., devices, applications, security tools, are collected remotely by the Monitor enabled with OSSIM Agent. Both, agent and agent-less, methods are used to collect methods. OSSIM uses a variety of methods for remote collection including Syslog and SNMP. These two protocols are only used when a device or application supports them otherwise; an agent is installed on the monitored object. OSSIM does recommend some agents, such as Snare [11] and OSSEC [10], which translate events onto the Syslog stream. However, these agents are not supported by devices at the edge of the network enabling IoT, for instance, smart devices and wireless or body sensors. Thus, we opt for an agent based on MQ Telemetry Transport (MQTT). MQTT is a lightweight M2M messaging transport protocol specifically designed for IoT with platform independence support [43]. The MQTT client hooks onto the event API of the device to collect security events generated and will transport them to the monitor component, the OSSIM Agent, where they are stored in a specific log file.

2) Event Filtration: Security events are extracted using a script, called *Plugin*, designed for individual event source. Writing the script requires some knowledge of the source and the events it is generating. Plugin, identified by a unique ID and other necessary parameters, is a configuration file that dictates from which queue events should be read and which of them needs to be filtered out. OSSIM utilizes a white-listing mechanism where only interested events are sent for further processing. A regular expression specifies these interested events. A match with the expressions is given a unique security ID (SID) which is further used in event correlation. An example plugin configuration is given in Figure 2 showing a specific SID corresponding to a login success event. A different SID can be defined for other events, for instance, a login failure event.

3) Event Normalization: Normalization is performed due to the fact that different *things* in the IoT will generate events in different formats. It is, therefore, necessary to transform them into a single common format for correlation and analysis. It is done during SIDs extraction and aims to extract vital attributes of an event transforming them into a common format for correlation. Attributes vary from event to event depending upon the primitive context they carry. In the above example, date and event source IP is normalized into a normalized common format and src_ip respectively.

B. Analyzer

1) Risk Scoring: Before the normalized events are correlated, they are assigned risk score. OSSIM uses three metrics used for the event (SID) risk quantification [44].

- Asset Value: Specifies the importance of event source or destination within the monitored scope. Ranges from 0-5.

```
[DEFAULT]
plugin_id=1008

[config]
type=detector
enable=yes
source=log
location=/var/log/mydevice.log

[my-device-login-success]
#Apr 2 12:45:12 192.168.5.18 my device:192.168.30.18
login success
event_type=event
regexp="(?P<date>\w{3}\s+\d{1,2}\s\d\d:\d\d:\d\d)\s+(?P
<sensor>\S+)\s+my\device\[\d{1,2}\]:+(?P<src>\d{1,3}\.\d
{1,3}\.\d{1,3}\.\d{1,3})\s+login\success"
date={normalize_date($date)}
sensor= $sensor
plugin_sid=1
src_ip={$src}
…
```

Fig. 2: Example Plugin

- Priority: Specifies the impact of the event. Ranges from 0-5.
- Reliability: Determines the probability or confidence of the fact the event will corresponds to a compromise. Thus, gives a weight to it false positivity. Reliability ranges from 0-10.

For each event, X, risk is quantified as:

$Risk(X) = (Priority * AssetValue * Reliability)/25$

The division of 25 is made to keep the risk values in the range of 0-10 which reflects the risk level of each event. These values are stored in the DB against each SID and are assigned as they arrive in the Risk Scoring engine. They can be changed as required manually. However, priority and reliability values can take different values automatically during event correlation as per the rules.

2) Event Correlation: The correlation engine investigates normalized events coming from the Monitor. It is done using correlation directives stored in XML. They are triggered when a specific SID is encountered, and thus a new event is generated with a new reliability value. The engine increases and decreases this value with respective to defined attributes within the directive rules. Hence, risk is dynamically assessed when SIDs are correlated over time. An SSH login failure example taken (simplified) from OSSIM wiki [45] is given in Figure 3.

```
<directive id="500000" name="SSH Brute Force Attack Against DST_IP" priority="4">
  <rule type="detector" name="SSH Authentication failure" reliability="0"
      occurrence="1" from="ANY" to="ANY" port_from="ANY" port_to="ANY"
      plugin_id="4003" plugin_sid="1,2,3,4,5,6,9,10,12,13,14,15,16,20">
      <rules>
         <rule type="detector" name="SSH Successful Authentication (After 1 failed)"
                        reliability="1" occurrence="1"
                        from="1:SRC_IP" to="1:DST_IP"
                        port_from="ANY" time_out="15" port_to="ANY"
                        plugin_id="4003" plugin_sid="7,8"/>
         <rule type="detector" name="SSH Authentication failure (10 times)"
                        reliability="2" occurrence="10" from="1:SRC_IP"
                        to="1:DST_IP"
                        port_from="ANY" time_out="40" port_to="ANY"
                        plugin_id="4003"
                        plugin_sid="1,2,3,4,5,6,9,10,12,13,14,15,16,20"
                        sticky="true"/>
      </rules>

   </rule>
</directive>
```

Fig. 3: Correlation Directive & Rules

It can be seen that rules can be defined up to n-levels of correlation depending upon the requirements. As the level is increased, more precise information is used, such as the time out, occurrence, source and destination, to validate the reliability and context of an event. In the mentioned example, reliability is increased which increases the risk level correspondingly. Similarly, using a rule, reliability during correlation can also be decreased if a login success event (SID) is encountered within the acceptable threshold range of the *occurrence* variable. Also, logical operators can be utilized when certain conditions are to be assured during the correlation.

Event correlation produces high level events which either goes for in-depth correlation or are flagged as alarms to be managed. Alarms are correlated events with risk level above risk acceptance threshold. Information carried by an alarm includes source and destination IDs, the user involved, risk level, threat details and the correlation directive responsible for generating it. This information is utilized during the adaptation process where the confronted risk is mitigated.

C. Adaptation

In order to utilize the available knowledge precisely and adapt security settings in an optimized manner, we propose an Adaptation Ontology. To be traversed at runtime, the ontology considers all the entities and their relationships necessary for optimal security adaptation. We will be utilizing this entire EDAS model in the IoT enabled eHealth scenario where a patient is remotely managed over the traditional internet or cellular network. To do so, we establish three different contexts in the proposed ontology as shown in Figure 4.

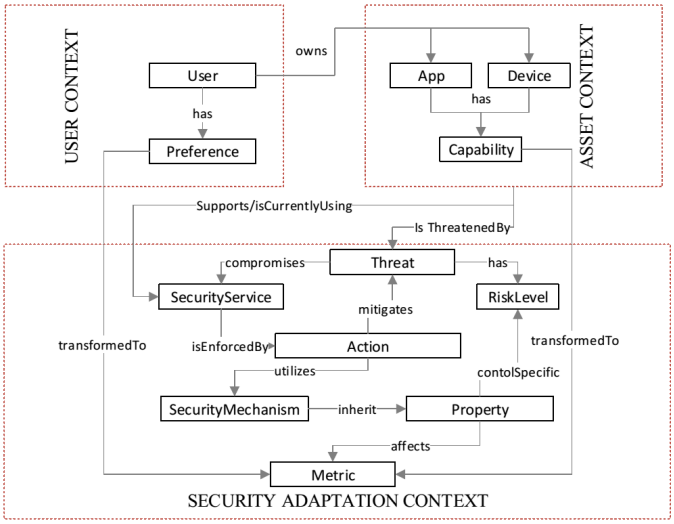

Fig. 4: Security Adaptation Ontology

- *User Context* corresponds to the patient and medical staff preferences which have to be considered before the adaptation
- Each user owns or utilizes a set of application, such as the eHealth app, Skype for patient-doctor communication,

etc. and devices, such as body sensors, smart device or desktop/Laptop, in the scope IoT-eHealth infrastructure. The corresponding information for instance, type, asset value, etc., along with their capabilities is contained in the *Asset Context*.

- The entities and associated settings required for optimized security adaption is grouped under the *Security Adaptation Context*.

An optimal mitigation action is selected from the actions pool following the procedure shown in Figure 5. The Response engine articulate a message based on the details of the action provided by the adaptation engine. Using MQTT transport, the message is sent to an actuator (MQTT Client) installed on the monitored *thing*. The actuator is hooked the specific component API, for instance a login API, and passes the message as variables to be reconfigured.

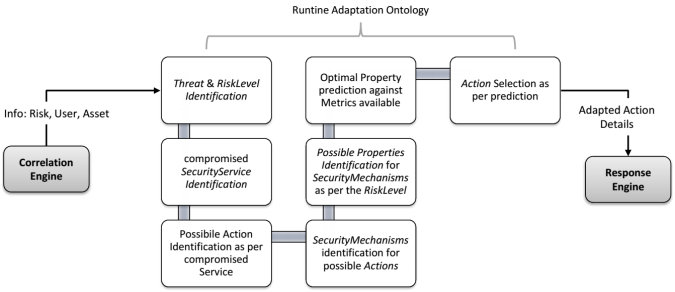

Fig. 5: Security Adaptation Process

A predictor function chooses the action with maximum utility. Subjective weights are assigned to affected metrics against each property, which correspond the overall utility of the property (to be used in the adapted action) for a specific user. Metrics reflect parameters, such as usability, reliability, service cost, etc., which can be negatively or positively influenced by a security property selection. For the time being, metrics are grouped into three categories, User, QoS and Security, to capture influences concerning user preferences, overall QoS and security reliability. However, we are still exploring metrics and measures, such as described in [46], to make our adaptation process more focused and convincing for user and service requirements besides dealing with security issues. A description of individual entities along with example instances is listed in Table I whereas, relations among them are detailed in Table II.

IV. eHEALTH CASE STUDY

IoT can substantially increase service quality and reduce cost, if enabled in the eHealth paradigm where patient vital signs are remotely diagnosed and managed via internet or cellular network. A number of projects, such as [47][48], aim to investigate different aspects of IoT-eHealth to make it more reliable and convenient. This section describes an IoT-eHealth home scenario in which a patient residing at home, Lynda, is equipped with various body sensors. Her vital signs are monitored through these sensors and are transmitted over a Wifi or cellular network to remote hospital site for further diagnosis. She frequently uses her smart phone, part of this infrastructure, installed with an eHealth app to keep track of health status as well as for billing payments besides personal use. We intend to explicate how our model fits into this scenario to defend against a security threat faced.

Home Scenario–Authentication: Lynda wants her credentials saved in the eHealth app to be protected. The app installed on her smart phone is protected with a password that is used to protect her credit card credentials, billing information and local Patient Health Information (PHI).

Adverse Situation: An insider having access to Lynda's smart phone with the intention of stealing her credit card information is trying to login into the eHealth app by guessing different passwords repeatedly.

Preferences: Lynda prefers medium level password instead of a complex one. She does not want her account to be locked out as she has to check her diabetes level frequently.

A generalized message sequence of the whole adaptation process as per the scenario is given in Figure 6. The defense against the situation is detailed as follow:

Model Go-Through – The Runtime Defense:

Event Collection & Monitoring: Smart phone login failure events will be collected by the MQTT client and will be sent to the Monitor. Plugin, e.g., `pluginID=20`, specified for the smart phone will read these events on the OSSIM Agent. The *login failure on eHealth App* SID, with `SID=3`, will extract and normalize the important attributes such as timestamps, user, source, and will add other attributes, such as the number of attempts made.

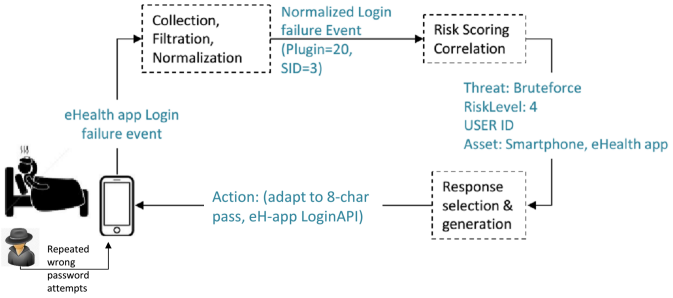

Fig. 6: Attack-Defense Case Study Message Diagram

Risk Quantification: Considering the risk acceptance level for repeated login failure is 4 let the smart phone be a critical asset, so `Asset Value=5`. To give space to for the accidental wrong attempts, let the `Reliability=0` for the first encounter and suppose the importance of the event is considerable so, `Priority=5`.

Event Correlation: The correlation directive shown in Figure 7 specifies 3 levels of correlation. The first wrong attempt is considered as normal so Reliability is not increased. For the next 5 wrong attempts, Reliability is increased to 2 and the engine waits for 10 seconds as a time out. Risk, as per the equation stated earlier, at this stage becomes 2. Similarly,

 ISBN: 978-1-61208-353-7

TABLE I. Ontology Entities

Context	Entity	Description	Example Instances
User	User	The registered user	Patient, Medical Staff, IT staff
	Preference	User preferences that affects or are affected by the adaptation decision	App/device usage knowledge, Current Health Status, Location, Environmental Context, etc.
Asset	App	Any soft components used in the IoT-eHealth infrastructure	eHealth app, communication software such as Skype, email, Security tools, etc.
	Device	Any hard components used to send receive and store User information	Body Sensors, Smart phones, Tablets, Laptops, Desktops
	Capability	The resources offered by individual **Asset**	Battery life time, CPU power, Memory, Supported Protocols etc.
Security Adaptation	SecurityService	The **security services** supported/Currently used by each **Asset**	e.g., Authentication, Encryption and Integrity modules
	RiskLevel	Event/Alarm Risk Level (analyzed by the event correlation/analysis engine) which threatens a **SecurityService** and **Asset**	Range(0-10)
	Threat	Threat information dictated by Correlation Directive	Brute Force, DoS, etc.
	Action	A list of adaptation actions (options) associated with a given **SecurityService**. Actions enforces a specific **SecurityService** in order to control a **Threat** faced	Changing Password, Locking a user for a specific time, changing encryption methods, Adapting a secure authentication protocol, etc.
	SecurityMechanism	Methods/algorithms associated with a given **Action** which are utilized in order to enforce a **SecurityService** challenged by a **Threat**	WEP, WP2, DES, AES, Captcha, SHA1, Disabling User Account etc.
	Property	Available attributes of a specific **SecurityMechanism** which can be adjusted for adaptation	AES (key length), Password (length, character type), captcha (image, audio), Account Locking time (seconds, minutes)
	Metric	Factors affecting security adaptation. Derived from user **Preferences**, device **capabilities** and the overall security against a given **Property** in terms of expected utilities.	Usability, PowerCost, ExecutionTime, ServiceLevelCost, Reliability, etc.

after 6 wrong repeated attempts Reliability is increased to 3 and so does the associated risk level. Finally, an alarm will be generated a risk of level 4 is raised after consecutive 20 attempts when Reliability is increased to 4. Risk is assessed dynamically and instances of the same events are correlated over a period of time as context becomes more evident.

```
<directive id="100" name="Password Brute Force against DST_IP" priority="5">
    <rule type="detector" name="eHealth APP Login failure" reliability="0"
       occurrence="1" from="ANY" to="ANY" port_from="ANY" port_to="ANY"
           plugin_id="20" sid="3">
           <rules>
                <rule type="detector" name="eHealth App Successful Login (After 1 failed)"
                   reliability="2" occurrence="1"
                   from="ANY" to="DST_IP"
                   port_from="ANY" time_out="10" port_to="ANY"
                   plugin_id="20" plugin_sid="3"/>
               <rule type="detector" name="eHealth App Login failure (6 times)"
                   reliability="3" occurrence="6" from="ANY"
                   to="DST_IP"
                   port_from="ANY" time_out="40" port_to="ANY"
                   plugin_id="20"
                   sid="3" />
              <rule type="detector" name="eHealth App Login failure (20 times)"
                   reliability="4" occurrence="20" from="ANY"
                   to="DST_IP"
                   port_from="ANY" time_out="60" port_to="ANY"
                   plugin_id="20"
                   sid="3" />
           </rules>

    </rule>
</directive>
```

Fig. 7: Correlation Directive & Rules for Repeated Login Failures

Security Adaptation: Proceeding logically with the procedure shown in Figure 5. An optimal mitigation action can be selected as:

- *Threat & Risk Level:* Password Brute Force
- *Compromised Security Service:* Authentication
- *Possible Actions:* Suppose, Password Change, Account Lockout & Enforcing Captcha
- *Security Mechanisms:* As per each action, Password Change (keyLength), Enforcing Captcha (Captcha), Account Lockout (Time Restriction)
- *Security Properties Metrics & Utilities:* As a hypothesis, consider Table III showing the affected metrics by individual properties with associated utilities (ranging from 1-10). The properties listed are considered to mitigate risk level 4 or above for password brute force attempts on the smart phone. Furthermore, it is assumed that the utilities are assigned as per service and user preferences.

TABLE III. Properties, Metrics & Utilities

Metric	PROPERTIES					
	KeyLength		Captcha		Time Restriction	
	8-char,	10-char.	Audio	Visual	15 min.	30 min.
Usability	8	5	6	7	6	3
QoS	8	7	5	5	6	6
Reliability	7	8	4	4	7	8
Total Utility	23	20	15	16	19	17

The predictor function will identify that the optimal action to circumvent this threat is to change the password on the smart phone eHealth app to an 8-characters. If it is already in use, it will go back and select the second best option. The selected action along with the user, concerned API and asset details will be given to the Response engine which will send a

 ISBN: 978-1-61208-353-7

TABLE II. Ontology Relations

Context	Relation	Classes Involved	Example Relations
User	has	User, Preference	Patient has a Preference of having easy to remember credentials Patient prefers service over security while being outside home Doctor prefers strict confidentiality while being outside hospital
User, Asset	owns	User, Asset	Patient owns a tablet to read his vital signs Patient owns (wears) ECG sensor Doctor owns a desktop machine to communicate with Patient over Skype
Asset	has	App, Device, Capability	Patient tablet has DualCore processor installed eHealth app installed on patient tablet has a medium level password ECG sensor does not support DES 128 bit algorithm Smart phone has 1 hour of talk time left
Asset, Security Adaptation	Supports, Currently Using	Asset, SecurityService	ECG Sensor supports/currentlyUsing Confidentiality, Authentication
	IsThreatenedBy	Asset, Threat	eHealth app is threatened by a password brute force attack In home Wifi network is threatened by DeAuth flooding
	compromises	Threat, SecurityService	Password Brute force compromises eHealth app Authentication WifiDeAuth flooding compromises network integrity
	has	Threat, RiskLevel	Password Brute force on eHealth app has a HIGH Risk Level
	isEnforcedBy	SecurityService, Action	eHealth App is authentication is enforced by a medium strength password Wifi Network authentication is enforced by WPA policy
	mitigates	Action, Threat	Changing user password mitigates a password brute force threat Restricting user login attempts to t-seconds mitigates a password brute force
	utilizes	Action, SecurityMechanism	A password change action utilizes the password length & complexity Restricting user login attempts utilizes the time limit Increase encryption level action utilizes AES
Security Adaptation	Inherit	SecurityMechanism, Property	Password length inherit the property of 6, 8 or 10 characters Password complexity inherit the property of character type
	controlSpecific	Property, RiskLevel	A password with 6 digit key length controls LOW level brute force attempts A password with 10 digit key length controls HIGH level brute force attempts
	affects	Property, Metric	10 character password affects (decreases) usability and (increase) security reliability 3G network affects (increases) Service Quality and (decreases) device battery
User, Security Adaptation **Asset, Security Adaptation**	transformedTo	Preference, Metric	User preferences are transformed to Usability User location is transformed to QoS, Security \& Privacy attributes
		Capability, Metric	Supported protocols (can be) transformed to QoS and Security metrices

message containing the instructions as appropriate variables to the MQTT client residing on the smart phone as an actuator. The actuator will identify the API mentioned and will pass the message variable. The API will implement the changes and will ask the user/adversary to enter a new 8-character password based on the older one.

V. Conclusion & Future Work

Existing detective and preventive controls as individual components seems to be inefficient in providing the required context to investigate security threats. We presented an event driven adaptive security model, EDAS, which leverages the capabilities of existing event models of diverse things in IoT and OSSIM correlation to adapt security settings by keeping the user and service utility at maximum. Primitive knowledge about security changes is collected and is analyzed in a definitive and established security context. The runtime adaptation ontology provides a structured knowledge of all the elements necessary to select appropriate mitigation action as user and service preferences. MQTT as a transport mechanism for the collection and actuation processes makes the model more extendable, platform independent and cost effective.

In the near future, we intend to develop a prototype for EDAS to test its processes as a real world IoT-eHealth artifact. Preliminary plans are to investigate the overall reliability, service response timings and building universal collectors and actuators for devices at the network edge, such as body sensors and personal smart devices. The prototype will be validated with confidentiality, availability, integrity and mobility scenarios as they are deemed to be the most critical aspects in remote patient management systems.

Acknowledgements

The work presented in this article is a part of the ASSET (Adaptive Security in Smart IoT in eHealth) project. ASSET (2012-2015) is sponsored by the Research Council of Norway under the grant agreement no: 213131/O70.

References

[1] "The internet of things is poised to change everything, says international data corporation," Press Release, October 2013, last access date: 31 May 2014. [Online]. Available: http://www.idc.com/getdoc.jsp?containerId=prUS24366813

[2] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, 2010.

[3] D. Shackleford, "Real-time adaptive security," SANS, Tech. Rep., December 2008, last Accessed on 4 April 2014. [Online]. Available: http://www.sans.org/reading_room/analysts_program/adaptiveSec_Dec08.pdf

[4] M. Nicolett and K. M. Kavanagh, "Magic quadrant for security information and event management," *Gartner RAS Core Reasearch Note (May 2009)*, 2011.

[5] RSA, "Rsa adaptive authentication. a comprehensive authentication and risk management platform," 2013, accessed on: 31 May 2014. [Online]. Available: http://www.emc.com/collateral/data-sheet/h11429-rsa-adaptive-authentication-ds.pdf

[6] H. Abie, "Adaptive security and trust management for autonomic message-oriented middleware," in *Mobile Adhoc and Sensor Systems, 2009. MASS'09. IEEE 6th International Conference on*. IEEE, 2009, pp. 810–817.

[7] "Ossim: the open source siem." last access date: 31 May 2014. [Online]. Available: http://www.alienvault.com/open-threat-exchange/projects

[8] L. Kufel, "Security event monitoring in a distributed systems environment," *IEEE Security Privacy*, vol. 11, no. 1, pp. 36–43, Jan. 2013.

[9] P. Bellavista, A. Corradi, and C. Stefanelli, "Java for on-line distributed monitoring of heterogeneous systems and services," *The Computer Journal*, vol. 45, pp. 595–607, 2002.

[10] "OSSEC: open source SECurity," last access date: 31 May 2014. [Online]. Available: http://www.ossec.net/

[11] "InterSect alliance - snare agents," last access date: 31 May 2014. [Online]. Available: http://www.intersectalliance.com/snareagents/index.html

[12] O. Etzion and P. Niblett, *Event Processing in Action*, 1st ed. Greenwich, CT, USA: Manning Publications Co., 2010.

[13] Z. Zheng, Z. Lan, B.-H. Park, and A. Geist, "System log pre-processing to improve failure prediction," in *IEEE/IFIP International Conference on Dependable Systems Networks, 2009. DSN '09*, Jun. 2009, pp. 572–577.

[14] A. Oliner and J. Stearley, "What supercomputers say: A study of five system logs," in *37th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, 2007. DSN '07*, Jun. 2007, pp. 575–584.

[15] M. F. Buckley and D. P. Siewiorek, "A comparative analysis of event tupling schemes," in *Fault Tolerant Computing, 1996., Proceedings of Annual Symposium on*. IEEE, 1996, pp. 294–303.

[16] "System logger: Syslog linux man page," last access date: 31 May 2014. [Online]. Available: http://linux.die.net/man/3/syslog

[17] "Event schema elements (windows)," last access date: 31 May 2014. [Online]. Available: http://msdn.microsoft.com/en-us/library/windows/desktop/aa384367(v=vs.85).aspx

[18] G. Jiang and G. Cybenko, "Temporal and spatial distributed event correlation for network security," in *American Control Conference, 2004. Proceedings of the 2004*, vol. 2, Jun. 2004, pp. 996–1001 vol.2.

[19] J. P. Martin-Flatin, G. Jakobson, and L. Lewis, "Event correlation in integrated management: Lessons learned and outlook," *Journal of Network and Systems Management*, vol. 15, no. 4, pp. 481–502, Dec. 2007.

[20] "Snort : Open source IDS/IPS," last access date: 31 May 2014. [Online]. Available: http://www.snort.org

[21] "The bro network security monitor," last access date: 31 May 2014. [Online]. Available: https://www.bro.org

[22] J. E. Hopcroft, R. Motwani, and J. D. Ullman, *Introduction to automata theory, languages, and computation*. Addison-Wesley, 2001.

[23] M. Sifalakis, M. Fry, and D. Hutchison, "Event detection and correlation for network environments," *IEEE Journal on Selected Areas in Communications*, vol. 28, no. 1, pp. 60–69, Jan. 2010.

[24] J. Tan, X. Pan, S. Kavulya, R. Gandhi, and P. Narasimhan, "SALSA: analyzing logs as StAte machines." *WASL*, vol. 8, pp. 6–6, 2008.

[25] P. Garca-Teodoro, J. Daz-Verdejo, G. Maci-Fernndez, and E. Vzquez, "Anomaly-based network intrusion detection: Techniques, systems and challenges," *Computers & Security*, vol. 28, no. 12, pp. 18–28, Feb. 2009.

[26] N. Ye, S. Member, S. M. Emran, Q. Chen, and S. Vilbert, "Multivariate statistical analysis of audit trails for host-based intrusion detection," *IEEE Transactions on Computers*, vol. 51, pp. 810–820, 2002.

[27] B. Gaudin, P. Nixon, K. Bines, F. Busacca, and N. Casey, "Model bootstrapping for auto-diagnosis of enterprise systems," in *International Conference on Computational Intelligence and Software Engineering, 2009. CiSE 2009*, Dec. 2009, pp. 1–4.

[28] W. K. Edwards, E. S. Poole, and J. Stoll, "Security automation considered harmful?" in *Proceedings of the 2007 Workshop on New Security Paradigms*, ser. NSPW '07. New York, NY, USA: ACM, 2008, pp. 33–42.

[29] D. Kulkarni and A. Tripathi, "Context-aware role-based access control in pervasive computing systems," in *Proceedings of the 13th ACM symposium on Access control models and technologies*. ACM, 2008, pp. 113–122.

[30] T. E. Maliki and J.-M. Seigneur, "A security adaptation reference monitor (SARM) for highly dynamic wireless environments," in *Proceedings of the 2010 Fourth International Conference on Emerging Security Information, Systems and Technologies*, ser. SECURWARE '10. Washington, DC, USA: IEEE Computer Society, 2010, pp. 63–68.

[31] M. Alia and M. Lacoste, "A QoS and security adaptation model for autonomic pervasive systems," in *Computer Software and Applications, 2008. COMPSAC '08. 32nd Annual IEEE International*, Jul. 2008, pp. 943–948.

[32] D. Quercia and S. Hailes, "MATE: mobility and adaptation with trust and expected-utility," *International Journal of Internet Technology and Secured Transactions*, vol. 1, no. 1, 2007.

[33] S.-W. Cheng, D. Garlan, and B. Schmerl, "Architecture-based self-adaptation in the presence of multiple objectives," in *Proceedings of the 2006 international workshop on Self-adaptation and self-managing systems*. ACM, 2006, pp. 2–8.

[34] H. Guo, "A bayesian approach for automatic algorithm selection," in *IJCAI 2003 Workshop on AI and Autonomic Computing, Mexico*. Citeseer, 2003, pp. 1–5.

[35] R. Sterritt, "Autonomic networks: engineering the self-healing property," *Engineering Applications of Artificial Intelligence*, vol. 17, no. 7, pp. 727–739, 2004.

[36] E. Lupu and M. Sloman, "Conflict analysis for management policies," in *Integrated Network Management V*. Springer, 1997, pp. 430–443.

[37] J. Stiborek, M. Grill, M. Rehak, K. Bartos, and J. Jusko, "Game theoretical adaptation model for intrusion detection system," in *Advances on Practical Applications of Agents and Multi-Agent Systems*. Springer, 2012, pp. 201–210.

[38] C. B. Simmons, S. G. Shiva, H. S. Bedi, and V. Shandilya, "ADAPT: a game inspired attack-defense and performance metric taxonomy," in *Security and Privacy Protection in Information Processing Systems*. Springer, 2013, pp. 344–365.

[39] W. Jiang, B.-x. Fang, H.-l. Zhang, Z.-h. Tian, and X.-f. Song, "Optimal network security strengthening using attack-defense game model," in *Information Technology: New Generations, 2009. ITNG'09. Sixth International Conference on*. IEEE, 2009, pp. 475–480.

[40] A. Evesti, E. Ovaska, and R. Savola, "From security modelling to run-time security monitoring," *Security in Model-Driven Architecture*, pp. 33–41, 2009.

[41] A. Evesti and E. Ovaska, "Ontology-based security adaptation at run-time," in *Self-Adaptive and Self-Organizing Systems (SASO), 2010 4th IEEE International Conference on*, Sept 2010, pp. 204–212.

[42] G. Denker, L. Kagal, and T. Finin, "Security in the semantic web using owl," *Information Security Technical Report*, vol. 10, no. 1, pp. 51–58, 2005.

[43] "Mq telemetry transport, mqtt," last access date: 31 May 2014. [Online]. Available: http://mqtt.org/

[44] "Ossim risk calculation," last access date: 31 May 2014. [Online]. Available: https://www.alienvault.com/wiki/doku.php?id=user_manual:dashboards:risk:risk_metrics#risk_calculation

[45] "Ossim-writing correlation directives," last access date: 31 May 2014. [Online]. Available: https://www.alienvault.com/wiki/doku.php?id=user_manual:intelligence:writing_correlation_directives

[46] R. M. Savola and H. Abie, "On-line and off-line security measurement framework for mobile ad hoc networks." *Journal of Networks*, vol. 4, no. 7, 2009.

[47] "Asset - adaptive security for smart internet of things in ehealth," last access date: 31 May 2014. [Online]. Available: http://asset.nr.no/asset/index.php/ASSET_-_Adaptive_Security_for_Smart_Internet_of_Things_in_eHealth

[48] "Butler ubiquitous, secure internet-of-things with location and context-awareness," last access date: 31 May 2014. [Online]. Available: http://www.iot-butler.eu/