Making the Internet of Things More Reliable Thanks to Dynamic Access Control

Anne GALLON[a], Erkuden RIOS[b], Eider ITURBE[b], Hui SONG[c], Nicolas FERRY[c]
[a] *EVIDIAN, Les Clayes-sous-Bois, France*
[b] *FUNDACIÓN TECNALIA RESEARCH & INNOVATION, Derio, Spain*
[c] *SINTEF Digital, Oslo, Norway*

Abstract. While the Internet-of-Things (IoT) infrastructure is rapidly growing, the performance and correctness of such systems becomes more and more critical. Together with flexibility and interoperability, trustworthiness related aspects, including security, privacy, resilience and robustness, are challenging goals faced by the next generation of IoT systems. In this chapter, we propose approaches for IoT tailored access control mechanisms that ensure data and services protection against unauthorized use, with the aim of improving IoT system trustworthiness and lowering the risks of massive-scale IoT-driven cyber-attacks or incidents.

Keywords. Internet-of-Things, Trustworthiness, Access Control, Context, Dynamicity, Security, Privacy.

1. Introduction

By 2021, Gartner envisions that 25 billion Internet-of-Things (IoT) endpoints will be in use[1], representing great business opportunities. However, complex challenges remain to be solved to efficiently exploit the full potential of the rapidly evolving IoT technologies. The performance and correctness of such systems will be critical, ranging from business critical to safety critical. Thus, aspects related to trustworthiness such as security, privacy, resilience and robustness, are still unsolved challenges of paramount importance for next generation of IoT systems [1].

Access control and identity governance mechanisms are cornerstones of security and privacy, which is today focused on addressing people accessing IT applications. In the context of IoT, access control needs to be extended to address not only people accessing the Internet of Things, but also to manage the relationships between connected things. This requires designing and building new access control mechanisms for authorizing access to and from connected things, with ad hoc protocols while still being able to address traditional access to IT applications.

The key challenge for access control in IoT is dynamicity. IoT systems are changing all the time: Devices keep entering and exiting the system; The same devices may be used in different context; New connections emerge among the devices; etc. For such

[1] https://www.gartner.com/en/newsroom/press-releases/2018-11-07-gartner-identifies-top-10-strategic-iot-technologies-and-trends

highly dynamic IoT systems, access rights from people to devices, and from devices to devices, are not immutable. On the one hand, the access right may vary according to the context change. Take an eHealth scenario as an example, where senior adults use IoT devices to monitor their physiological data such blood pressure. In the normal, day-to-day context, only the user himself should have the access right to the data, due to the privacy concern. However, in a special context, such as under emergency rescue, medical staff should be granted with the access right the journal with historical physiological data. Therefore, the decision of access right in IoT systems must be made with awareness of the context. On the other hand, the distribution of IoT systems and the instability of connections between devices require that those decisions must be made in a distributed way. To answer this challenge, the objective we have set ourselves is to develop an authorization server with dynamicity that is tailored to context and architecture changes of IoT systems. Today, no protocol can deliver dynamic authorization based on context for both IT and OT (operational technologies) domains.

Our work described in this chapter proposes to deal with these considerations, by providing dynamic access control mechanisms for IoT systems based on context awareness and risk identification, in order to ensure data protection by controlling access to (personal) data and resources which are usually distributed in the IoT environment. Our solution controls the access of all the actors (end-users, services, devices, administrators) to the data managed by smart IoT systems (SIS) which contributes to system trust by ensuring integrity and confidentiality of the operated data and resources, and by providing data security and privacy to Operation phase of IoT system engineering process.

We present two complementary dynamic access control mechanisms tailored to IoT which can be adopted individually or combined depending on the needs of the IoT system resources access policies: i) a Context-aware Access Control to be used in cases when the security policy requires reasoning over changing context conditions that impact permissions to access resources and ii) a distributed Access Control mechanism based on distributed agents able to evaluate independent access policies close to the target resources. Both approaches are based on industrial standards, i.e., XACML [13] and OAuth 2.0 [16].

We implement the approaches into proof-of-concept tools, as part of the ENACT toolset. ENACT [18] is a research project with the overall goal to enable DevOps in the realm of trustworthy smart IoT systems. ENACT will provide an integrated DevOps Framework composed of a set of loosely coupled enablers that can be easily integrated with existing IoT platforms via plug-in mechanism. As shown in Figure 1, the ENACT enablers are categorized into three groups as follows: (i) the toolkit for the continuous delivery of smart IoT systems, (ii) the toolkit for the agile operation of smart IoT systems, and (iii) the ENACT facilities for trustworthiness. The dynamic access control tools are part of the third group, and provide the reusable facilities for trustworthiness solutions, which can be integrated into IoT applications under development. The dynamicity achieved by the access control facilities also provide the other tools with the capability of continuously evolving the security and privacy policies of the IoT applications.

These advanced and IoT tailored context-aware access control and authorization mechanisms and tools for trustworthy smart IoT systems will advance state-of-the-art techniques for managing the accesses of both devices and users. The success of this work will lead to bring the potential to accelerate the adoption of the IoT, by improving smart IoT system trustworthiness and lowering the risks of massive-scale IoT-driven cyber-

attacks infringes. This will enable the full potential of IoT systems in the future digitalized society.

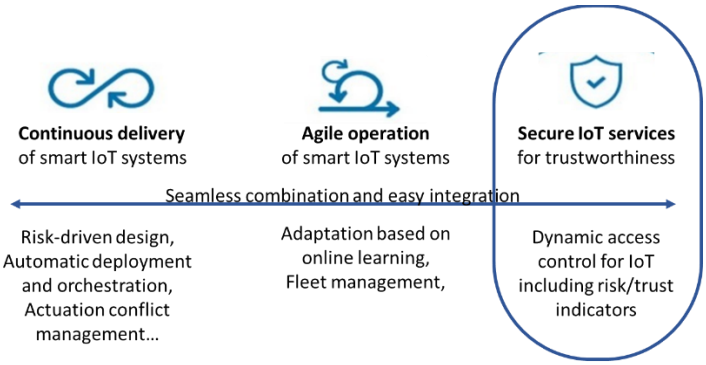

Figure 1. The dynamic access control tool in the ENACT framework

The rest of the chapter is organized as follows. Section 2 introduces the industry standards of access control protocols which we use as the basis of our approach. Section 3 and Section 4 presents our approaches of context-aware and distributed access control tailored for IoT, and the proof-of-concept implementations. Section 5 compare the approach with related work. Section 6 concludes the chapter with our future plans.

2. Background: Industry standards of Access Control protocols

2.1. The traditional dynamic access control chain based on the XACML model

Although in 2013, a Forrester analyst wrote a blog [2] proclaiming that XACML (eXtensible Access Control Markup Language) was dead, in fact some years later this is not so obvious, and therefore a first approach is to study how the traditional dynamic access control chain based on the XACML model could help to answer the challenge of securing the Internet of Things.

XACML is a policy-based management system that defines a declarative access control policy language implemented in XML and a processing model describing how to evaluate authorization requests according to the rules defined in policies. As a published standard specification, one of the goals of XACML is to promote common terminology and interoperability between authorization implementations by multiple vendors.

XACML is primarily an Attribute-Based Access Control (ABAC) system, where attributes associated with an Entity are inputs into the decision of whether a given Entity may access a given resource and perform a specific action.

The XACML model supports and encourages the separation of the authorization decision from the point of use. When authorization decisions are baked into client applications, it is very difficult to update the decision criteria when the governing policy changes. When the client is decoupled from the authorization decision, authorization policies can be updated on the fly and affect all clients immediately.

[2] https://go.forrester.com/blogs/13-05-07-xacml_is_dead/

The access control chain based on the XACML model is depicted in Figure 2.

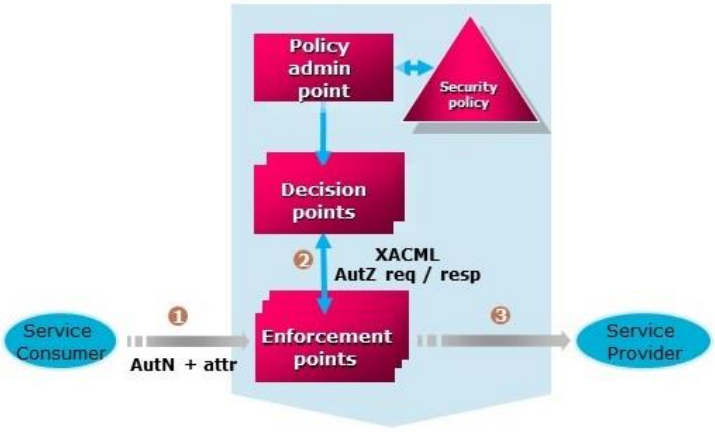

Figure 2. The dynamic access control chain based on the XACML model

In this chain:

- The Policy Decision Point (PDP) evaluates access requests against authorization policies before issuing access decisions.
- The Policy Enforcement Point (PEP) intercepts the user's access request to a resource, makes a decision request to the Policy Decision Point to obtain the access decision (i.e. access to the resource is approved or rejected), and acts on the received decision.

In fact, this approach is dynamic by essence, since the access control decisions are made based on attributes associated with relevant entities. In addition, it offers a powerful access control language with which to express a wide range of access control policies.

But the following points make this approach prohibitive:

- An approach based on rules is difficult to administer. Defining policies is effort consuming. You need to invest in the identification of the attributes that are relevant to make authorization decisions and mint policies from them. In addition, the ABAC system introduces issues, most notably the 'attribute explosion' issue and, maybe more importantly, the lack of audibility.
- Although Service-Oriented Architecture and Web Services offer advanced flexibility and operability capabilities, they are quite heavy infrastructures that imply significant performance overheads.
- Since XACML has been designed to meet the authorization needs of the monolithic enterprise where all users are managed centrally, this central access control chain is not suitable for Cloud computing and distributed system deployment, and it doesn't scale the Internet.

2.2. The new approach based on OAuth 2.0

Another approach has been studied, based on the OAuth 2.0 industry-standard protocol for authorization. This new approach is depicted in Figure 3.

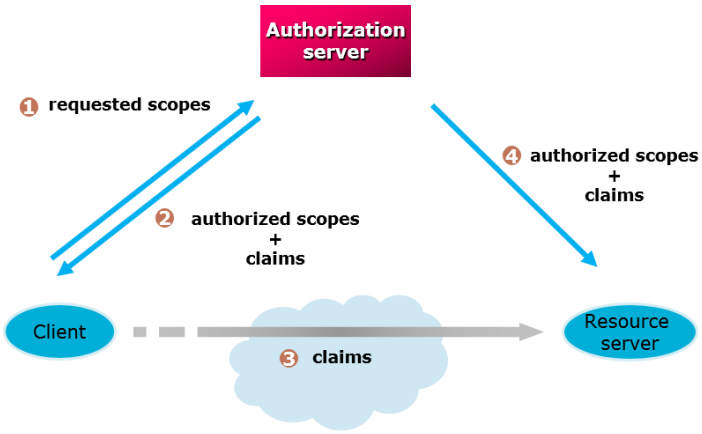

Figure 3. A new approach based on OAuth 2.0

In this approach, a client can access a resource on behalf of a user through an authorization delegation mechanism. This assumes that the user has given his consent for the requested scopes.

As a major advantage, this protocol can be implemented in a light way, by leveraging HTTP and REST-based APIs. In fact, OAuth 2.0 supports the mobile device application endpoint in a lightweight manner. Its simplicity makes it the de-facto choice for mobile and also non-mobile applications. Due to the growing importance of Cloud technologies and APIs, the REST architecture is now heavily favored.

In addition, OAuth 2.0 allows a fluid integration with role management: OAuth 2.0 scopes can be used to provide role-based authorization.

But this protocol does not have the granularity of XACML in terms of rules. And another point is still an obstacle to meet the need of an IoT Context-aware Access Control: the dynamicity, for situation awareness, is not delivered by design.

3. A Context-aware access control approach for IoT

3.1. Objectives

Even if identification and authentication of both users and nodes (things, edge, etc.) are fundamental for trustworthy IoT systems, the focus of this work will be on authorization. Thus, the objective is to offer an innovative access control mechanism to meet the security needs of the Internet of Things.

Our aim is to provide mechanisms for controlling the security, privacy and trustworthiness behavior of smart IoT systems. This includes reaction models and mechanisms that address the adaptation and recovery of the IoT application operation on

the basis of the application context, in order to deliver dynamic authorization based on context for both IT and OT (operational technologies) domains.

The Internet of Things links many devices such as sensors, cameras or smartphones to the Internet. These devices have the capacity to act as sensors or actuators in their environment, while the context can continuously change and evolve. The environmental data are considered as dynamic and give crucial information about a context (state of devices, user's behavior and location, etc.). The traditional mechanisms of access control do not use these contextual data while doing authorization decisions.

The context-aware access control mechanisms described in this chapter will provide context-aware risk & trust-based dynamic authorization mechanisms, through an IAM (Identity and Access Management) gateway for IoT that includes next-generation authorization mechanisms. The aim is to ensure (i) that an authenticated IoT node accesses only what it is authorized to and (ii) that an IoT node can only be accessed by authorized software components.

Access authorizations will be adapted according to contextual information. *Context* may be for instance the date and time an access authorization is requested, or the geolocation of this request; it may be also composed of a set of information about the status of the underlying infrastructure, the physical system status, SIEM alerts, for example to make certain information more widely available in the case when an alarm has been triggered.

3.2. The solution

Due to the disadvantages observed on the traditional dynamic access control chain based on the XACML model, we turned to a solution based on OAuth 2.0.

But to achieve the goal we set ourselves, which is to provide an IoT context-aware access control mechanism, we must fill the gap to deliver dynamic authorizations based on context by using the OAuth 2.0 protocol.

Starting from security features for identity management and access control based on the protocols OAuth 2.0 and OpenID Connect (OIDC), our approach is to develop an evolution of these authentication and authorization mechanisms intended for the Internet of Things. Due to the dynamicity of the data concerning the environment of a person, this contextual information must be used to manage and adjust the security mechanisms, i.e. consider contextual information in the identification of the entity requesting access and in the evaluation of the conditions to grant access.

By assessing the applicability of OAuth 2.0, our IoT context-aware access control will leverage it as a key protocol for interoperability. Our work will address problems of adding dynamicity to the authorization decisions produced by OAuth 2.0 even if it is not meant for that. This dynamic capability will be in charge of evaluating contextual information and insert it in authorization decisions.

3.3. Proof of concept: the ENACT Context-aware Access Control tool

The context-aware access control tool provides an authorization mechanism that issues access tokens to the connected objects after successfully authenticating their owner and obtaining authorization. This authorization mechanism uses the OAuth 2.0 protocol, which provides authorization delegation mechanism. Following this protocol, an object can access a backend API by using an access token containing the list of claims (i.e. user's attributes, e.g., user name, email address, etc.) and scopes (read-only, read/write)

that an authenticated user has consented for this object to access. This mechanism may be coupled with contextual information to adapt the access authorizations according to them (for example to make certain information more widely available in some urgent case).

The context-aware access control tool provides access tokens that allow a reverse proxy working as an API Gateway to control the access to applications and APIs. The scopes and claims contained in the access tokens are used to restrict accesses to the backend server APIs to a consented set of resources.

The authorization mechanism can be coupled to a multi-level, multi-factor Authentication Server that provides strong authentications mechanisms to the users. This mechanism mitigates the level of authentication required depending on the user's environment context and an external context. The risk is a value computed either statically, depending on a defined configuration, or dynamically by using a REST API to dialog with an external decision engine. The input used to compute the risk is the user's session context, which contains the browser DNA (i.e. the user's browser fingerprinting which consists in uniquely identifying a web browser through its configuration: time zone, screen resolution, character font, user-agent), the service the user wants to access and the configured trust level of this service, the access time and the trust planning associated to the service, and the IP address and its geolocation. Depending on the evaluated risk of the user's session, the level of the required authentication will be leveled up, or, if the risk is too high, the connection will be refused.

Figure 4 shows a use case example of the context-aware access control tool integrated in Evidian Web Access Manager (WAM[5]). The numbers associated to arrows in this schema are mentioned in parentheses in the next paragraph.

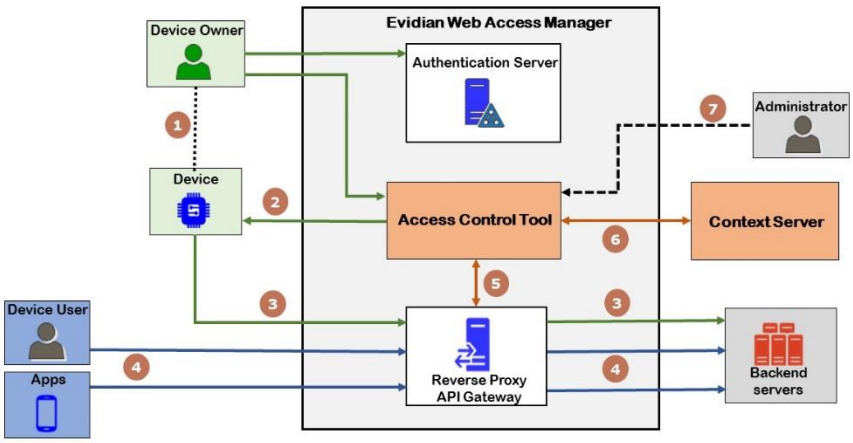

Figure 4. Use case example of the Context-aware Access Control

During the enrolment phase, a connected object is associated with a user (1); then the connected object can push data to a backend server in a controlled way (3) by using

[5] https://www.evidian.com/products/web-sso/

the access token it received (2) from the access control tool. The backend server stores this data and can display it within an application. The authorized users and applications can retrieve the data from the backend server (4). WAM plays a pivotal role between all these exchanges by making authorization decisions depending on the context (5).

The context-aware access control tool includes a scoping system. The scopes are used by an application to authorize access to a user information. In the OAuth 2.0 protocol, the scopes system only returns a set of static user attributes. The context-aware access control tool extends the protocol to also consider dynamic attributes which provide contextual data on the user and his devices. That way, the access control can be adapted depending on the context which can be continuously evolving, in order to make the access rights more secure and efficient in function of the current environment.

The access control tool directly communicates with a Context Server (6) to make dynamic access controls based on the context information during the authorization phase. For example, it can reject the authorization if the access token is valid while other context information does not respect the authorization policy.

The authorization policy is a set of rules that define whether a user or device must be permitted or denied accessing to a resource. An administrator can control this adjustment and create special authorization rules based on risk levels computed from the context data provided (7).

In this architecture, two components are providing the Context-aware Access Control mechanisms:

- **The context server.** The Context Server exposes a REST API that provides contextual data on the user and his devices. These data are dynamic attributes and come from other external sources (sensors, other applications, etc.).
- **The access control Tool.** The access control tool is composed of an authorization server associated to a post authorization plugin, to add more controls during the authorization phase. Its purpose is to check if the request is authenticated and is authorized to access the backend server.

Indeed, each time a device sends a request to a backend server, the access control tool can check the dynamic scopes about the user associated to the device that performs the request and realize special actions according to this information like blocking the request or limiting the accessible scopes.

The Post authorization plugin extends the basic authorization phase and is entirely customizable. Any operation can be executed during the authorization phase, including calling external programs, and in particular the context server. From the provided contextual information, the idea is to compute a risk value that will be used to apply context-aware dynamic scopes. The post authorization plugin can create injection variables that can be reused and injected in the initial request sent to the backend server.

The Administrator can configure the access control tool to adjust the authorization security rules according to the dynamic attributes.

4. A distributed access control approach for IoT

Considering the large number and great diversity of interconnected resources in IoT systems, it would likely be the case that one access control solution does not fit all the

possible scenarios or needs. Therefore, for some IoT systems it would be necessary to adopt complementary mechanisms that together ensure secure access to resources.

In this chapter we describe a distributed access control solution conceived as a mechanism that could be easily integrated in the IoT system components and activated at runtime when needed. These control mechanism responds to the need of securing the access to resources and services in the distributed IoT system components, while having the enforcement of security policies continuously controlled.

The solution is based on enforcement agents developed in ENACT as preventive security mechanisms or controls that are managed by the ENACT Framework. These agents are an IoT-tailored evolution of MUSA Enforcement agents [11] which in turn were built on top of existing open source solutions. The major innovation resides in having a tool in the ENACT Framework as the single point of management for orchestrating multiple agents and mechanisms that address diverse security properties on the IoT system. In this chapter we only describe those enforcement agents related to Access Control (AC).

The AC agents developed rely on XACML policy specification standard by OASIS explained in previous section. The AC agents check whether the policy rules evaluate to true or false and the enforcement of the access (grant or deny the access) will be done by an external entity (e.g. the IoT platform) according to the result. Note that the power of the access control performed depends on the granularity of attributes taken into account in the XACML rules. The finer the granularity the richer the possibilities.

To this end, an actionable AC agent-based distributed enforcement has been developed so as the AC agents can be deployed by the GeneSIS Orchestration tool (within the ENACT Framework) together with the components of the IoT system. These AC control agents will be external to the SIS components and managed externally by the operators. The agents are able to send events at the IoT application level that serve a double purpose: i) allow the continuous control of the good performance of the agent and ii) detect security anomaly by processing and correlating agent events with other data from system and network layers. In the following subsections we elaborate architectural details of the solution.

4.1. Proof of concept: the ENACT Agent-based Access Control

4.1.1. Architecture

We prototyped the solution as a distributed Access Control tool part of the ENACT framework for the DevOps of smart IoT systems. The overall architecture of the tool is depicted in Figure 5. The AC agents are deployed to work together with the IoT application system components, so they can enforce the authorisation policies to the resources in these components. Every AC agent is controlled through control actions (e.g. access policy updates) that can be sent either by the *Control Manager* or by the SIS operator through the GUI displayed in the *Dashboard*. The *Control Manager* can be set with predefined rules based on learned patterns of cyber threats. In addition, the distributed AC agents send events to the *Streaming bus*, which will be later stored and displayed in the *Dashboard*. In the rest of this section, we elaborate the main components of this architecture.

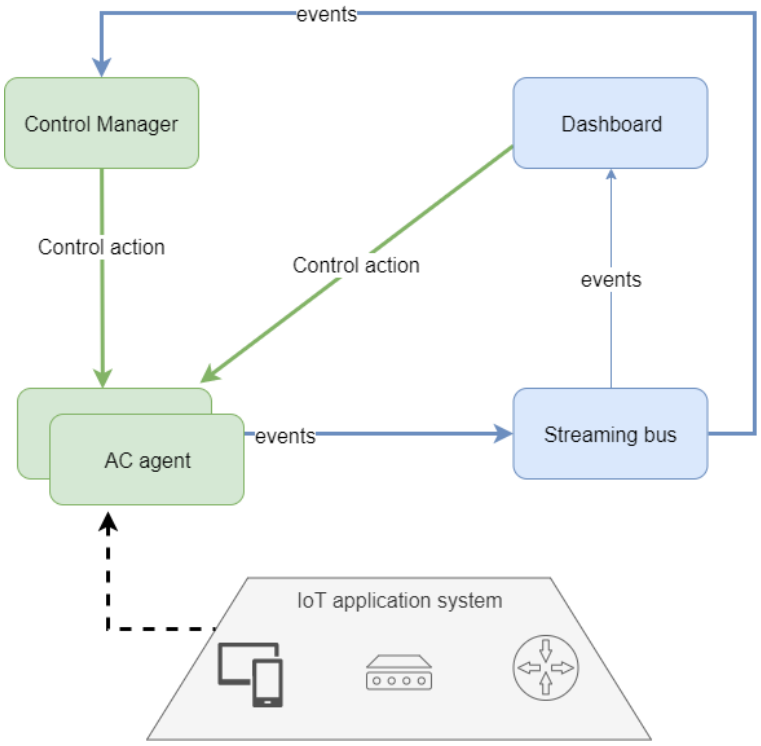

Figure 5. The agent-based distributed Access Control tool in the ENACT framework

4.1.2. The Access Control agents

The AC mechanisms are deployed as agents distributed in the IoT system that provide authorisation services to IoT system resources. They can be considered as security controls that can be activated, deactivated or configured when needed at runtime. These agents are deployed to work together with IoT application components, while they can be installed on the same infrastructure (smart device, IoT platform, etc.) where the application component is deployed or in another, the closer to the resource, the faster the access.

AC agents of two types are designed: a) agents that can be used when HTTP communications are used between the access requester and the resource (service or data), and b) agents that are tailored to non-HTTP type of communications.

It is interesting to note that both types of agents rely on the XACML technology, though the non-HTTP AC agent can be easily adapted to interpret policies in other languages. More technical details on both classes of AC agents follow:

a) AC agent for HTTP communications:

The AC agent developed in Node.js [12] is intended to be integrated into the application as a reverse proxy that intercepts all requests to a target and ensures that the requester has the rights to reach/consume the target. The agent includes a rule engine based on XACML policies [13]. The agents use the JSON Web Token open standard RFC 7519 [14] as self-contained security resource for securely transmitting information between communication parties in the form of a JSON object. The mechanism includes

the creation of access tokens that are used to assert a number of claims in the service requests.

When a user needs to consume a service protected by the AC agent, she needs to include the token as HTTP Header on the service request. The request will be intercepted by the AC agent, and according to the user attributes in the token, the agent will evaluate the XACML policy and grant or reject the access to the service.

The XACML model supports and encourages the separation of the access decision, the point of use, and the management of the policies. In this implementation the access decision (XACML PDP) and the point of use (XACML Policy Enforcement Point, PEP) resides on the same instance to improve the performance.

The AC agent includes the following features:

- A **XACML PEP (Policy Enforcement Point)** that intercepts access request and ensures XACML PDP decision.
- A **XACML PDP (Policy Decision Point)** that evaluates XACML policies.
- A **XACML PIP (Policy Information Point)** that provides external information to the PDP, such as LDAP attributes. It is implemented as data retrievers that get information from the request (origin, date, IP address, user email, user name...).
- A **JWT client** to verify JWT tokens.

The AC agent does not offer an external REST API, but it works as a reverse proxy that intercepts all the requests to the protected service, i.e. offers an access control mechanism to REST services based on XACML rules.

In order the agent can apply the XACML rules, each request must include a JWT token with the requester information.

To protect an IoT application component service, for each request sent by clients to consume this service, the AC agent will execute the following steps:

1. Intercept any request to the Backend service.
2. Extract user information from JWT Token located in request header.
3. Get context data from request.
4. Make access control decision based on policies, user information and context data.
5. If decision is *permit* it will redirect the request to Backend service.
6. If decision is *not permit* the request will be rejected.

The AC agent can evaluate all and only the attributes included in the JWT. Therefore, the power of the Attribute Based Access Control performed by the AC agent depends on the granularity of attributes taken into account in the XACML rules.

Note that the management of the policies (XACML Policy Administration Point, PAP) is not supported inside the agent. The XACML access policies would be defined in the Policy Administration Point (PAP). As explained below, the Dashboard within the ENACT Framework provides a PAP functionality to view and edit the XACML rules (in the form of JSON files) as well as automatically communicate them to the AC agent.

b) AC agent for non-HTTP communications:

When the communications are not HTTP it is not possible to adopt the reverse HTTP proxy paradigm and therefore, the AC agent includes only the part of the PDP which evaluates pre-defined security policy rules over access requests where the attributes of the request context (e.g. requesting IP, user role, etc.) are no longer taken from the HTTP protocol but rather they need to be sent to the agent together with the rule evaluation order. This is exactly the case of the IoT systems of the ENACT project use cases. Therefore, the PDP agent does not perform the interception of the access request but requires an explicit order to evaluate the access policy.

The interface of the PDP Agent being developed in ENACT will offer at least the following services:

- updatePolicy(policyID): updates the indicated XACML policy stored in the agent.
- evaluatePolicy(policyID, attributes): boolean evaluates the rules in the XACML policy for a specific set of attributes sent as parameters.
- start: starts the agent running.
- stop: stops the agent.

4.1.3. The Control Manager

The Control Manager acts as a centralized management hub for the distributed enforcement agents. This component is in charge of analyzing the events sent by the distributed agents and controlling whether the policies are being correctly applied.

When an enforcement agent is launched, the first thing that it does is to inform the Control Manager that a new agent is alive, and the Control Manager registers the ID of the new agent which will be used for identifying the agents in the Dashboard.

4.1.4. The streaming bus

This component is the typical message broker or bus where agents subscribe and unsubscribe to topics, so they can communicate their events to the Control Manager within the ENACT Framework. The event bus is implemented in Apache Kafka [15] .

The events emitted by the internal services of the agents and exchanged through this bus are:

- Agent events: On every transition, a public event is emitted by the agent to notify on internal state transitions.
- Proxy events: events fired by the internal proxy service of the agent, e.g. when a request to a resource is proxied, i.e. these are actually access control events.
- Log events: events fired by internal logger service of the agent, e.g. log a value of a token.

4.1.5. The Dashboard

This module refers to the main Enforcement GUI that allows the management of different preventive enforcement agents to be deployed for the IoT system in order to work with the enforcement agents and apply at runtime diverse security mechanisms such as Identity Management and Access Control in the resource usage.

The Dashboard allows for registering and configuring the agents (endpoints to protect, etc.) as well as setting up the XACML policies that each agent will enforce. The policies

need to be in JSON file format and the Dashboard includes a JSON Editor that gives support to the editing of the file. The manual edition by the operators is optional and automatic updates of the JSON files is also supported.

5. Related work

The academic and industrial state-of-the-art solutions for IoT Access Control lacks dynamic and adaptive capabilities, through IoT system context-awareness, as well as with risk and trust as potential sources of context information.

Dynamic access control. Mahmud et al. [2] have stated that several IoT-centric security issues might be unnoticed or poorly addressed by the security researchers, as this paradigm is not full-fledged yet. A key requirement they identified is access control: the act of ensuring that an authenticated IoT node accesses only what it is authorized to. Cvitic et al. [3] analyzed the security aspects for each layer of the IoT architecture: the biggest security risk is at the perception layer of the IoT architecture due to the specific limitations of devices and the transmission technology used at this layer, followed by the middleware layer based on cloud computing and inherited vulnerabilities of that concept. Fall et al. [4] have learned that cloud computing infrastructures do not use dynamic access control, but static traditional mechanisms, despite the highly dynamic nature of cloud computing capabilities. Farooq et al. [5] confirmed that, in the future, more security techniques (such as risk assessment) must be explored in each architectural layer. More approaches can be found in a survey by Ravidas, et al. [20]

Context awareness. Ramos et al. [21] summarized different types of context for IoT devices, and their impact of security and privacy. Jagadamba et al. [6] studied adaptive security schemes based on context. Context-awareness enhances the effectiveness of the mechanisms by incorporating contextual data into a decision-making process. This capability of taking grey decisions instead of black-or-white is particularly key in environments where perimeter security is not enough anymore, especially for cloud and IoT infrastructures. Habib et al. [7] have identified 3 types of context (physical, computing, user-related), with 4 approaches (category, context-awareness, context learning, context modelling). Interestingly they identified active or passive context awareness (contextual changes are automatically discovered or statically presented), as well as sensed (taken from the processes' environment) and derived (computed on the go). Our approach follows the same direction and provide a solution of context awareness for access control in IoT.

Risk-based access control. Dankar et al. [8] learned that different risk classes are identified ahead of time and each class is matched with a protection level. In their solution, an access request to a resource undergoes automated risk assessment and it is accordingly classified into one of the predefined classes. The appropriate protection level is then applied to the requested data. While analyzing competing smart home frameworks, Fernandes et al. [9] refined this by considering that device operations are inherently asymmetric risk-wise, and a capability model needs to split such operations into equivalence classes. An on/off operation pair for a light bulb is less risky than the same operation pair for an alarm. They proposed splitting/grouping objects' capabilities based on risk, hence with the possibility to select the granularity. From the range of granularities observed, none was risk-based. Atlam et al. [] use user context, resource sensitivity and risk history to analyse the security risk for each access request, and adjust access control accordingly. Fall et al. [4] learned that many researchers define a risk

formula for a given user or object, but on an insufficient set of parameters (e.g., focusing on requestor but not on the resource accessed). They learned also that the main issue with risk-aware access control is the cost of computation. The benefit is that risk is evaluated for each access request, but this is costly in terms of computation. Their proposition does not solve the issue. In our approach, we introduce risk analysis into Context-aware Access Control as a post authorization step to adjust access control scope.

Privacy concerns. Privacy is an essential part of planning for cyber secure systems, and is getting more and more important for IoT systems as they are moving closer to personal users. Authors in [17] exemplify hands on the "most severe, yet easy to abuse" IoT threats, namely: leakage of the personally identifiable information (PII), leakage of sensitive user information and unauthorised execution of functions. Hiller et al. [10] put a focus on involving privacy in risk management, while analysing the NIST Privacy Framework, and confirmed that adaptive capability is a cornerstone for the resilience of privacy. Our approaches focus on the first safeguard towards IoT privacy protection, by preventing unauthorized and risky data access, and achieve the adaptive capability of access control according to the context and system changes.

Contribution of trust. Jagadamba et al. [6] learned that conceptually *trust* is a parameter used to exchange information regarding the entity's actions through belief and faith. Positive behaviors increase the trust while negative behaviors decrease the trust upon the entity. Trust is classified into *proofs* (certified information –such as identity, property and authorization- issued by a certification authority or from other central controlled systems) and *indicators* (possible factors collected from various sources). Dynamic access control is an attempt to combine proofs and indicators, considering not only the identify of data requestor, but also the context and system status during runtime.

6. Conclusion

We presented the IoT Access control mechanisms, including both Context-aware Access control and distributed access control agents, developed as part of the ENACT H2020 project DevOps Framework which offers novel solutions to address challenges related to the development, operation, and quality assurance of trustworthy smart IoT systems. The presented Access control mechanisms are under development and the Context-aware Access control solution will be integrated in the Evidian standard offer as "an IAM gateway for IoT", while access control agents are part of Tecnalia open source solution portfolio.

The lesson learned so far is that we are committed on a powerful and promising approach that offers authorisation dynamicity to different types of scenarios. We are comforted in the idea that the OAuth 2.0 framework remains the foundational industry approach for providing authentication and authorization to REST-based APIs, but it is a mistake to think of OAuth 2.0 as a simple protocol by underestimate its complexity.

Furthermore, building a generic solution for multiple situations requires a sophisticated design, compared to deploying OAuth 2.0 for one use case only. The management of scopes and claims is key. Too many scopes make administration difficult, whereas too few scopes degrade security with over-entitlements.

At this stage, the key topics of further research in context-aware Access Control are as follows:

- A robust and scalable approach for scope management must be defined, by injecting scopes in OAuth 2.0 "Device flow", applicable to IoT use cases.

- Contextual information inside scopes must be exploited, to be able to apply dynamic scopes via notions of risk/trust, and so enable situation-aware dynamic access control behaviors.

The ENACT use cases (eHealth, Smart Buildings and Intelligent Transportation Systems) will give us the means to validate these concepts.

References

[1] IEC: IoT 2020: Smart and secure IoT platform. IEC white paper (2016).
[2] H. Mahmud, F. Maziar. and H. Ragib (2015), "Towards an Analysis of Security Issues, Challenges, and Open Problems in the Internet of Things".
[3] I. Cvitic, and Vujic, M. and Husnjak, S. (2015), "Classification of Security Risks in the IoT Environment", 26th DAAAM International Symposium on Intelligent Manufacturing and Automation, p731-740.
[4] D. Fall, T. Okuda, Y. Kadobayashi, and S. Yamaguchi (2016), "Risk Adaptive Authorization Mechanism (RAdAM) for Cloud Computing, Journal of Information Processing, Vol 24 No.2, p371-380.
[5] M. U. Farooq, M. Waseem, A. Khairi and S. Mazhar (2015), "A critical Analysis on the Security Concerns of Internet of Things (IoT)", International Journal of Computer Applications, Volume 111 No.7.
[6] G. Jagadamba, and B. Sathish Babu (2016), "Adaptive Security Schemes based on Context and Trust for Ubiquitous Computing Environment: A Comprehensive Survey", Indian Journal of Science & Technology, Vol 9 (48).
[7] K. Habib and W. Leister (2015), "Context-Aware Authentication for the Internet of Things", ICAS 2015: The Eleventh International Conference on Autonomic and Autonomous Systems, p134-139.
[8] F. Dankar, and R. Badji (2017), "A risk-based framework for biomedical data sharing", Journal of Biomedical Informatics, Vol 66, p231-240.
[9] E. Fernandes, J. Jung and A. Prakash (2016), "Security Analysis of Emerging Smart Home Applications".
[10] J. Hiller and R. Russel (2017),"Privacy in Crises: The NIST Privacy Framework", Journal of Contingencies and Crisis Management, Volume 25 Number 1, p31-38.
[11] E. Rios, E. Iturbe, W. Mallouli and M. Rak (2017, October). Dynamic security assurance in multi-cloud DevOps. In 2017 IEEE Conference on Communications and Network Security (CNS) (pp. 467-475). IEEE.
[12] The Node.js project. Available at: https://nodejs.org/en (Retrieved July 2019)
[13] eXtensible Access Control Markup Language (XACML) Version 3.0. Available at: http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-os-en.html (Retrieved July 2019)
[14] JSON Web Token (JWT) standard by Internet Engineering Task Force (IETF). Available at: https://tools.ietf.org/html/rfc7519 (Retrieved July 2019)
[15] Apache Kafka® distributed streaming platform by Apache Software Foundation. Available at: https://kafka.apache.org/ (Retrieved July 2019)
[16] Hardt, Dick. "The OAuth 2.0 authorization framework." (2012).
[17] C. Kolias, A. Stavrou, J. M. Voas, I. V. Bojanova and D. R. Kuhn. "Learning Internet of Things Security "Hands-on"", 2016.
[18] ENACT: Development, Operation, and Quality Assurance of Trustworthy Smart IoT Systems. https://www.enact-project.eu/
[19] H.F. Atlam, A. Alenezi, R.J. Walters, G.B. Wills and J. Daniel (2017). Developing an adaptive Risk-based access control model for the Internet of Things. In 2017 IEEE international conference on internet of things (iThings) and IEEE green computing and communications (GreenCom) and IEEE cyber, physical and social computing (CPSCom) and ieee smart data (SmartData) (pp. 655-661). IEEE.
[20] S. Ravidas, A. Lekidis, F. Paci, and N. Zannone (2019). Access control in Internet-of-Things: A survey. Journal of Network and Computer Applications, 144, 79-101.
[21] J. L. H. Ramos, J. B. Bernabe, and A. F. Skarmeta (2015). Managing context information for adaptive security in iot environments. In 2015 IEEE 29th International Conference on Advanced Information Networking and Applications Workshops (pp. 676-681). IEEE.