POSTER: An Authorization Service for Collaborative Situation Awareness

Alexandru Ionut Egner, Duc Luu, Jerry den Hartog, Nicola Zannone
Eindhoven University of Technology
{a.i.egner, m.luu, j.d.hartog, n.zannone}@tue.nl

ABSTRACT

In international military coalitions, situation awareness is achieved by gathering critical intel from different authorities. Authorities want to retain control over their data, as they are sensitive by nature, and, thus, usually employ their own authorization solutions to regulate access to them. In this paper, we highlight that harmonizing authorization solutions at the coalition level raises many challenges. We demonstrate how we address authorization challenges in the context of a scenario defined by military experts using a prototype implementation of SAFAX, an XACML-based architectural framework tailored to the development of authorization services for distributed systems.

CCS Concepts

•**Security and privacy** → **Access control; Authorization;**

Keywords

Access control; XACML; Security-as-a-Service

1. INTRODUCTION

A Combined Joint Task Force (CJTF) is a complex military coalition created for a specific mission. It is typically composed of military branches (e.g., navy, air force) belonging to different authorities (e.g., several NATO members), which can cooperate with non-military bodies (e.g., civilian agencies, NGOs). With this type of highly collaborative missions, situation awareness is achieved from intel gathered from a large number of heterogeneous data sources (e.g., UAV, motion sensors) and analysis services (e.g., imagery analysis, HUMINT). Sharing information is thus essential for the success of a CJTF. However, due to the sensitive nature of data, each party within the coalition imposes stringent security constraints that govern sharing of information.

The operational model adopted by a CJTF enables each authority to retain control over their data. To this end, each authority employs its own authorization solution and defines its own policies to regulate who is allowed to access its resources. In particular, each authority can choose its own access control model, define policies according to its domain knowledge and best practices, augment policy evaluation with customized capabilities, etc. This, however, can have an impact on the exchange of information between parties within the coalition and can result in poor cooperation. Therefore, the success of a coalition requires harmonizing access rights across different authorities. In particular, the authorization systems employed by the authorities need to interoperate with each other and act like a global authorization system at the CJTF level.

Access control standards such as XACML [1] already provide a baseline for the development of authorization services suited for distributed systems. In particular, XACML has been proven to be suitable for the specification and enforcement of access control policies in open systems. One of XACML's strengths is that it supports augmenting policy evaluation through extensibility points such as User Defined Functions (UDF).

Existing implementations of XACML, however, are not flexible enough to face challenges characteristic to military coalitions. These implementations are usually monolithic and cannot be easily adapted to different deployment configurations, depending on the requirements of each authority. For instance, components for policy evaluation and management, which already exist as part of the access control systems of the authorities forming the coalition, cannot be reused for new missions. In addition, existing implementations do not support features fundamental to the military domain, such as delegation of authority [3, 7], concept alignment [8] and geolocation policies [2]. Although they can be extended to support additional capabilities, this is often done in an ad-hoc way. In particular, the logic of UDFs is typically implemented as part of the Policy Decision Point (PDP), which impacts both extensibility (PDP needs to be modified and redeployed) and separation of concerns (PDP should only be responsible for the evaluation of requests).

We have developed SAFAX [6], a novel XACML-based architectural framework. The driving idea underlying SAFAX is to provide authorization as a service. Each component of SAFAX is designed as a loosely-coupled service, thus allowing a variety of deployment configurations. In a major departure of existing XACML implementations, UDFs are realized as self-configuring clients that consume external services, making it possible to extend a PDP's capabilities without changing the PDP itself. In this paper we demonstrate the feasibility of creating secure collaborative situation awareness using SAFAX. To this end, we apply a prototype implementation of SAFAX to a scenario in the military domain within the context of the IN4STARS 2.0 project. It is worth noting that, although we demonstrate the framework in the context of a military scenario, it is domain independent and can be applied to different application domains (e.g., eHealth systems, cloud services).

CODASPY'16 March 09-11, 2016, New Orleans, LA, USA

ACM ISBN 978-1-4503-3935-3/16/03.

DOI: http://dx.doi.org/10.1145/2857705.2857740

2. AUTHORIZATION FRAMEWORK

SAFAX [6] is an XACML-based architectural framework tailored to the development of authorization services for open and distributed systems. An overview of the authorization framework is shown in Figure 1. The components forming the architecture are designed as loosely-coupled services and can be logically separated into three main blocks: *(i)* domain-specific components - Policy Enforcement Point (PEP), Context Handler (CH) and Policy Information Point (PIP), *(ii)* core components, which represent the baseline of the authorization service and *(iii)* external services, which can be used to evaluate custom constraints in the policies.

Domain-specific components depend on the application environment and are under the control of the coalition members. These components are responsible for handling the conversion between the attribute representation in the application environment and the attribute representation in the XACML format (CH), retrieving the information necessary for policy evaluation (PIP) and enforcing access decisions (PEP). The only implementation requirements are that they adhere to the XACML reference architecture [1] and that they are deployed as web services. External services provide additional capabilities for policy evaluation. For instance, they can be used to retrieve trust information from external sources (e.g., attributes certified by a trusted authority) or relocate the computation of complex functions relieving the burden on the PDP (e.g., geolocation function [2], concept alignment [8]).

The core components represent the baseline of the authorization service. Below, we briefly describe the core components of the authorization framework:

- **Policy Decision Point (PDP)** is responsible for evaluating access requests against policies and providing access decisions. SAFAX can support several PDP services for different versions of XACML or PDP implementations with additional functionalities (e.g., transparency [5], probabilistic decisions [4]). Each PDP service is identified by a unique URL (PDP-URL).
- **PDP Configuration (PDPC)** allows authorities to select the PDP service to be used for evaluation. Moreover, the PDPC allows the configuration of PDP services by setting a number of parameters including the root combining algorithm.
- **Router** is responsible for distributing access requests to the proper PDP service for evaluation based on the PDP-URL.
- **Policy Administration Point (PAP)** facilitates storage and management of access control policies, regardless of the location where resources are stored.
- **Service Repository** allows registered service to dynamically discover, bind and consume other services. Services can be components of the authorization framework (e.g., PDP, PIP) or external services registered as UDFs. In SAFAX, UDFs are decoupled from the PDP and implemented as external, pluggable services.

By designing SAFAX components as loosely-coupled services, the framework provides great flexibility in terms of deployment, configuration, and integration. In particular, the authorization service (i.e., core components) can be outside the control of an authority depending on the required deployment configuration. For instance, army branches such as infantry or armored brigades, can delegate part of the authorization decision making to a trusted authorization service (e.g., one provided by the army). This facilitates the integration of components that deploy different authorization solutions (e.g., based on different versions of XACML). Another strength of the framework comes from the implementation of the UDFs' logic as external services. This makes it possible to enrich the context of a request during evaluation and enable the computation of the complex functions, while keeping the system scalable.

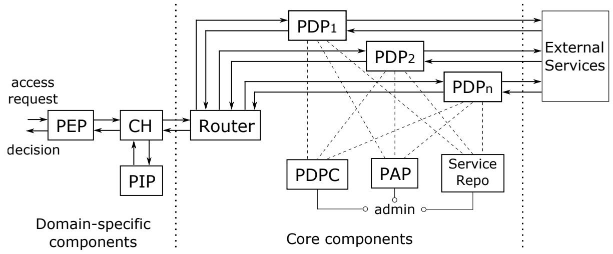


Figure 1: Overview of the Architectural Framework

3. DEMONSTRATION

The SAFAX framework has been realized as an authorization service[1], in which every component of the framework is implemented as a RESTful service. We demonstrate the feasibility of the authorization service in a collaborative situation awareness scenario elaborated by military experts in the context of the IN4STARS 2.0 project. In this case study, two instances of the service were deployed at authorities A_1 and A_2, under different deployment configurations. In both configurations, the core components of the authorization service were deployed within the boundaries of the authority. Each authority employs a credential-based trust management service based on GEM [7] to retrieve user credentials issued by the other authority. Moreover, A_2 deploys a geolocation service based on GeoXACML [2] for the evaluation of constraints on spatial information and a concept alignment service [8]; A_1 relies on these services for the evaluation of its policies.

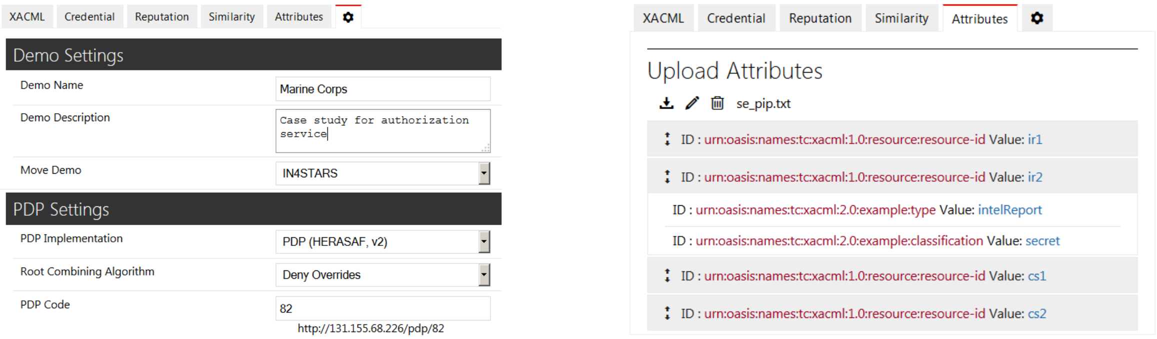
Both authorities define policies to regulate access to their resources and deploy them in their own PAPs. A_1 contributes to the coalition with intel reports. To protect these resources, A_1 defines a policy that grants access to reports providing intel about the mission's area only to *mission members* having *role* intel officer.

Authorities can customize the PDP to be used for the evaluation of their policies through the PDPC (Figure 2a). This makes it possible to reuse components in the context of other missions, which on the other hand would not be possible using existing monolithic implementations. In our scenario, A_1 chooses a PDP service for policies expressed in XACML v2. As confidentiality of data is considered to be of utmost importance, A_1 chooses deny-overrides as the root combining algorithm. The PDPC also assigns a unique identifier to the PDP (PDP Code), which is part of the PDP-URL (shown at the bottom of Figure 2a).

Suppose that A_1 receives a request from a mission member belonging to A_2 to access an intel report. The request, however, does not contain all the attributes needed for policy evaluation. One the one hand, the requester may not know which attributes are used in the evaluation. On the other hand, A_1 does not trust the value that requesters provide for certain attributes (e.g., resource *type* and *classification*). These attributes can be retrieved from the PIP deployed at A_1 (through the CH). Figure 2b shows the interface of the PIP service along with the attributes assigned to the resources shared by A_1. However, not all attribute values may be locally available for the evaluation of a request. For instance, the requester is within A_2's trust domain and A_1 has no knowledge of his *role*. Thus, the PDP retrieves the value of this attribute from A_2 through an external service (the credential-based trust management service) during policy evaluation.

A_1 consumes the concept alignment service at A_2 for harmonizing the terminology used within the coalition (e.g., the *role* ex-

[1] The authorization service is available at http://security1.win.tue.nl/safax/.



Demo Settings

Demo Name: Marine Corps

Demo Description: Case study for authorization service

Move Demo: IN4STARS

PDP Settings

PDP Implementation: PDP (HERASAF, v2)

Root Combining Algorithm: Deny Overrides

PDP Code: 82

http://131.155.68.226/pdp/82

Upload Attributes

se_pip.txt

ID : urn:oasis:names:tc:xacml:1.0:resource:resource-id Value: ir1

ID : urn:oasis:names:tc:xacml:1.0:resource:resource-id Value: ir2

ID : urn:oasis:names:tc:xacml:2.0:example:type Value: intelReport

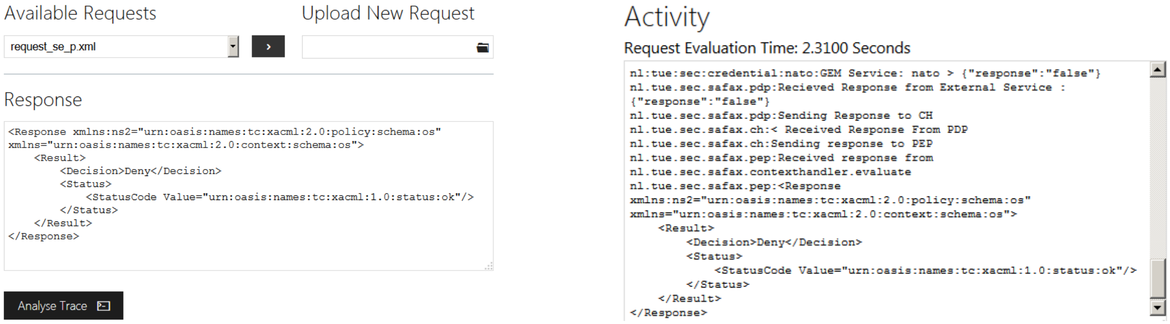
ID : urn:oasis:names:tc:xacml:2.0:example:classification Value: secret

ID : urn:oasis:names:tc:xacml:1.0:resource:resource-id Value: cs1

ID : urn:oasis:names:tc:xacml:1.0:resource:resource-id Value: cs2

(a) PDP Configuration (b) Policy Information Point

Figure 2: GUI: Management Configurations



request_se_p.xml

```
<Response xmlns:ns2="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os">
    <Result>
        <Decision>Deny</Decision>
        <Status>
            <StatusCode Value="urn:oasis:names:tc:xacml:1.0:status:ok"/>
        </Status>
    </Result>
</Response>
```

Request Evaluation Time: 2.3100 Seconds

```
nl:tue:sec:credential:nato:GEM Service: nato > {"response":"false"}
nl.tue.sec.safax.pdp:Recieved Response from External Service :
{"response":"false"}
nl.tue.sec.safax.pdp:Sending Response to CH
nl.tue.sec.safax.ch:< Received Response From PDP
nl.tue.sec.safax.ch:Sending response to PEP
nl.tue.sec.safax.pep:Received response from
nl.tue.sec.safax.contexthandler.evaluate
nl.tue.sec.safax.pep:<Response
xmlns:ns2="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os">
    <Result>
        <Decision>Deny</Decision>
        <Status>
            <StatusCode Value="urn:oasis:names:tc:xacml:1.0:status:ok"/>
        </Status>
    </Result>
</Response>
```

(a) Policy Evaluation (b) Evaluation Log

Figure 3: GUI: Evaluation

pected in the policy of A_1 is *intel officer*, while the same role provided with the request is defined as *intel analyst* by A_2). Moreover, A_1 consumes the geolocation service at A_2 to determine whether the requested intel report concerns the mission's area. The PDP's use of external services extends its capabilities in a flexible way, which is essential for enabling collaborations. This constitutes an advantage of SAFAX over other monolithic approaches, as it increases scalability by shifting part of the computational burden away from the PDP.

Upon receiving the request, the Router of the authorization service employed by A_1 forwards it to the PDP identified by the PDP Code contained in the PDP-URL. To manage multiple requests-responses, the Router augments each request sent to a PDP with a unique Transaction ID, internal to the SAFAX framework. In our example, the requester is not an intel officer. Thus, the request is denied as shown by the PDP response in Figure 3a. For debugging purposes SAFAX supports the visualization of the log messages generated during the evaluation of access requests. A snippet of the log can be found in Figure 3b.

4. CONCLUSION

In this paper we demonstrated the feasibility of secure collaborative situation awareness using an authorization service based on SAFAX. In particular, we have shown the deployment of the authorization service under two different configurations within the context of a military coalition. The demonstration highlights the importance of decoupling UDFs from the PDP and illustrates the benefits of relying on external services for attribute retrieval and relocation of computation for complex functions.

Acknowledgments This work has been funded by the EDA project IN4STARS2.0, ITEA2 projects FedSS and M2MGrid, EU FP7 project AU2EU and the Dutch program COMMIT under the THeCS project.

5. REFERENCES

[1] eXtensible Access Control Markup Language (XACML) Version 3.0. OASIS Standard, OASIS, 2013.

[2] OGC Geospatial eXensible Access Control Markup Language (GeoXACML) 3.0 Core. OGC Discussion Paper, Open Geospatial Consortium, 2013.

[3] XACML v3.0 Administration and Delegation Profile Version 1.0. OASIS Standard, OASIS, 2014.

[4] J. Crampton, C. Morisset, and N. Zannone. On missing attributes in access control: Non-deterministic and probabilistic attribute retrieval. In *Proc. of SACMAT*, pages 99–109. ACM, 2015.

[5] S. Damen, J. den Hartog, and N. Zannone. CollAC: Collaborative access control. In *Proc. of CTS*, pages 142–149. IEEE, 2014.

[6] S. P. Kaluvuri, A. I. Egner, J. den Hartog, and N. Zannone. SAFAX - An Extensible Authorization Service for Cloud Environments. *Frontiers in ICT*, 2(9), 2015.

[7] D. Trivellato, N. Zannone, and S. Etalle. GEM: A distributed goal evaluation algorithm for trust management. *TPLP*, 14(3):293–337, 2014.

[8] D. Trivellato, N. Zannone, M. Glaundrup, J. Skowronek, and S. Etalle. A semantic security framework for systems of systems. *Int. J. Cooperative Inf. Syst.*, 22(1), 2013.