A Security Framework for Systems of Systems

Daniel Trivellato
Eindhoven University Of Technology
Email: d.trivellato@tue.nl

Nicola Zannone
Eindhoven University Of Technology
Email: n.zannone@tue.nl

Sandro Etalle
Eindhoven University Of Technology
University of Twente
Email: s.etalle@tue.nl

***Abstract*—Future IT systems will consist of a wide variety of dynamic, distributed coalitions of autonomous and heterogeneous systems that collaborate to achieve a common goal. While offering several advantages in terms of scalability and flexibility, this new paradigm has a strong impact on system interoperability and on the security requirements of collaborating parties. In this demo we present the prototype implementation of a security framework that addresses the security challenges of future IT systems.**

I. Introduction

Future IT systems will consist of dynamic coalitions of systems and services that collaborate to achieve a common goal. Examples of such coalitions, generally referred to as *systems of systems* (SoS), include Web Services, Mobile Ad-hoc Networks (MANETs), air traffic control systems, etc. Sharing sensitive information with other parties might be required for the success of a coalition; nevertheless, this information should be accessed exclusively by authorized parties, which may vary depending on the context (e.g., in emergency situations, or based on the location of the requester). Furthermore, when heterogeneous systems form dynamic coalitions that transgress the traditional boundaries between organizational and cultural units, parties will likely "speak" different languages and employ different organizational models.

Several security frameworks for SoS have been proposed. These frameworks can be divided into two categories: semantic frameworks [1], [2] and trust management (TM) frameworks [3], [4], [5]. Semantic frameworks rely on ontologies for the specification of access control policies and the definition of domain knowledge. This enables interoperability among parties at the cost of limiting the expressive power of the policy language, which does not allow the specification of several types of security constraints (e.g., separation of duty). On the other hand, TM frameworks rely on an attribute-based approach to access control where access decision are based on digital certificates, called credentials. TM frameworks employ expressive policy specification languages to ensure data confidentiality; however, they either require all parties in an SoS to use the same vocabulary [3], [6], or do not provide a mechanism to align different vocabularies [5].

In this demo we present the prototype implementation of the security framework for SoS that we are developing within the POSEIDON project (http://www.esi.nl/poseidon). The framework combines context-aware access control with TM and ontology-based services [7], [8] to guarantee confidentiality of information (both data and security policies), autonomy and interoperability among parties in an SoS. We show an application of the framework to a coast surveillance scenario, where parties need to exchange sensitive information to achieve situational awareness.

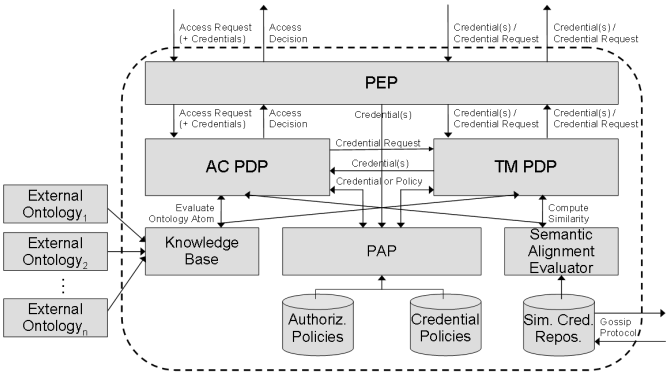


Fig. 1. Security Framework Architecture

II. Security Framework Architecture

This section presents the security framework that is employed by each party in the SoS to protect the local resources. An overview of the security framework's architecture is shown in Fig. 1; the dashed line separates the local components (i.e., the trusted environment of a party) from the external world.

The *policy enforcement point* (PEP) is the interface of a party with the external world, and has three main tasks: (1) intercepting incoming requests for local resources, (2) contacting the appropriate *policy decision point* (PDP) to evaluate those requests, and (3) enforcing the decision of the PDP. Two types of requests are allowed: *access requests* and *credential requests*, specified using XACML [9] and SAML [10] respectively. Access requests are processed by the *access control PDP* (AC PDP), while credential requests by the *trust management PDP* (TM PDP).

When it receives an access request, the AC PDP fetches the relevant authorization clauses through the *policy administration point* (PAP). If the clauses depend on some credentials, the AC PDP requests them to the TM PDP, which takes over the responsibility of retrieving them. Once all the necessary credentials have been collected, they are asserted together with the authorization clauses into the authorization engine to determine the access decision. Similarly to the AC PDP, upon receiving a request the TM PDP fetches the applicable credential clauses and the locally available credentials through the PAP. The policy evaluation algorithm within the TM PDP

defines the procedure to compute the answers to a credential request. In our framework we employ GEM [11], a policy evaluation algorithm that evaluates credential requests in a completely distributed way without disclosing the policies of parties, thereby preserving their confidentiality.

Both authorization and credential clauses are expressed in POLIPO [7], a logic-based policy language that relies on ontologies for enabling mutual understanding among parties. In particular, POLIPO uses ontologies in two ways: (a) to obtain domain and context information relevant for an access decision or credential release (e.g., the current location of the requester), by means of ontology atoms in the body of clauses; (b) to provide a semantics to the attributes certified by credentials, which enables the use of semantic alignment techniques to map attributes defined in different ontologies (i.e., to map an unknown attribute to a known one). Ontology atoms are resolved by requesting their evaluation to the *Knowledge Base* (KB) component, which consists of a set of ontologies defining the concepts and relationships employed in the party's policies and all the domain and context information. Attribute mapping requests are evaluated by the *Semantic Alignment Evaluator*, which implements the ontology alignment technique in [8].

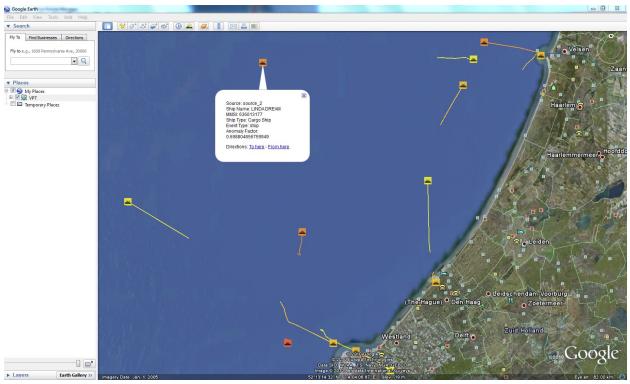
III. Prototype Implementation

We have deployed a prototype implementation of the security framework into an SoS in the Maritime Safety and Security (MSS) domain that has been developed within the POSEIDON project. The POSEIDON SoS consists of five types of systems: coastal AIS[1] receivers, sea-based AIS receivers, the Internet, a Maritime Security Center (MSC), and patrol vessels. The AIS receivers capture AIS messages broadcasted by the ships transiting in their coverage area and send those messages to the MSC for further processing. The MSC collects data from the various receivers, analyzes them (e.g., for detecting anomalous behavior of ships), and integrates them with further information from the Internet; the resulting information forms the KB of the MSC. The information in the KB is used by the operators of both the MSC and patrol vessels to analyze the maritime traffic.

In this demo we show an application of the security framework to a coast surveillance scenario, where the MSC and a patrol vessel of the coast guard collaborate to prevent illicit activities off the Dutch coast. Every request to access the MSC's KB, coming either from within the MSC or from the patrol vessel, passes through the MSC's security framework, which checks whether the requester possesses the required credentials (possibly initiating a credential discovery process), and filters the response based on the security policy of the MSC. In addition, access requests from operators of the patrol vessel go also through the security framework of the patrol vessel, which checks whether the request and the relative response are authorized by the local policy. Communication among parties is via HTTP. Accordingly, we developed the PEP of the security framework as a web proxy that intercepts all the HTTP requests and returns an HTTP response in the appropriate format; this allowed us to deploy the framework without modifying the rest of the POSEIDON SoS.

[1]The Automatic Identification System (AIS) is a short range coastal tracking system used for identifying and locating vessels.

(a) Data View for an MSC Operator



(b) Data View for a Patrol Vessel Operator

Fig. 2. Data Views Filtered by Security Policies

We use Google Earth as visualization software; the view is updated every 30 seconds to display the new data collected by the AIS receivers. Fig. 2(a) and 2(b) show the output of the visualization for an operator of the MSC and an operator of the patrol vessel respectively. In the visualization, icons represent the current position of ships, and the color of a ship's trajectory reflects the anomaly factor associated to that ship. In our scenario, MSC's operators (Fig. 2(a)) are authorized to see all the maritime traffic off the Dutch coast, while patrol vessel's operators (Fig. 2(b)) are allowed to see only ships with a high anomaly factor.

IV. Conclusion

We have presented a security framework that provides confidentiality of information, autonomy and interoperability of parties in dynamic coalitions of heterogeneous systems. The framework consists of a set of components implemented following the service-oriented paradigm. This facilitates the deployment of the framework into existing SoS, and allows for an easy integration of additional components to support the evaluation of policies and provide additional functionalities. Besides the MSS domain, we have also deployed the framework in an SoS in the employability domain [12].

ACKNOWLEDGMENT

This work has been carried out as part of the POSEIDON project under the responsibility of the Embedded Systems Institute (ESI). This project is partially supported by the Dutch Ministry of Economic Affairs under the BSIK03021 program.

REFERENCES

[1] L. Kagal, M. Paolucci, N. Srinivasan, G. Denker, T. Finin, and K. Sycara, "Authorization and Privacy for Semantic Web Services," *IEEE Intelligent Systems*, vol. 19, no. 4, pp. 50–56, 2004.

[2] A. Uszok, J. M. Bradshaw, M. Johnson, R. Jeffers, A. Tate, J. Dalton, and S. Aitken, "KAoS Policy Management for Semantic Web Services," *IEEE Intelligent Systems*, vol. 19, no. 4, pp. 32–41, 2004.

[3] N. Li, J. C. Mitchell, and W. H. Winsborough, "Design of a Role-Based Trust-Management Framework," in *Proc. of S&P'02*. IEEE Computer Society, 2002, pp. 114–130.

[4] W. Nejdl, D. Olmedilla, and M. Winslett, "PeerTrust: Automated Trust Negotiation for Peers on the Semantic Web," in *Proc. of SDM'04*, ser. LNCS 3178. Springer, 2004, pp. 118–132.

[5] A. J. Lee, M. Winslett, and K. J. Perano, "TrustBuilder2: A Reconfigurable Framework for Trust Negotiation," in *Proc. of IFIPTM'09*. Springer, 2009.

[6] M. Winslett, C. C. Zhang, and P. A. Bonatti, "PeerAccess: a logic for distributed authorization," in *Proc. of CCS'05*. ACM, 2005, pp. 168–179.

[7] D. Trivellato, F. Spiessens, N. Zannone, and S. Etalle, "POLIPO: Policies & OntoLogies for Interoperability, Portability, and autOnomy," in *Proc. of POLICY'09*. IEEE Computer Society, 2009.

[8] ——, "Reputation-Based Ontology Alignment for Autonomy and Interoperability in Distributed Access Control," in *Proc. of CSE '09*, vol. 3. IEEE, 2009, pp. 252–258.

[9] OASIS, "eXtensible Access Control Markup Language (XACML) Version 2.0," OASIS Standard, 2005, http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-core-spec-os.pdf.

[10] ——, "Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0," OASIS Standard, 2005, http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf.

[11] D. Trivellato, N. Zannone, and S. Etalle, "GEM: a Distributed Goal Evaluation Algorithm for Trust Management," Eindhoven University of Technology, Tech. Rep. CS 10-15, 2010.

[12] K. Böhm, S. Etalle, J. den Hartog, C. Hütter, S. Trabelsi, D. Trivellato, and N. Zannone, "Flexible Architecture for Privacy-Aware Trust Management," *JTAER*, vol. 5, no. 2, pp. 77–96, 2010.