audit Web service (SAWS) as a general purpose audit tool.

As part of the EC TrustCoM integrated project, we have built a reputation management system capable of recording the reputations of users (for example, as performed by eBay). The next step is to link this to the PERMIS decision engine so that access control decisions can be based on the current reputation of a user (which is related to their trustworthiness). Currently users are either trusted or not to access a target resource, based on their X.509 ACs. Once reputations are included in the decision-making however, users' permissions may be removed if their reputation drops below a certain value. In addition, the TrustCoM project is defining standard protocols for credential validation (ie calls to getcreds) and the making of policy decisions (ie calls to decision). It is likely that WS-TRUST and XACML respectively will be used for these. PERMIS will be enhanced to support these protocols once they have been finalized by the consortium.

Link:
http://www.eu-trustcom.com

Please contact:
David W. Chadwick, University of Kent, UK
Tel: +44 1227 82 3221
E-mail: D.W.Chadwick@kent.ac.uk

Richard Sinnott, (for DyVOSE)
University of Kent, UK
E-mail: ros@dcs.gla.ac.uk,

Damian Mac Randal (for DyCom)
CCLRC, UK
E-mail: D.F.Mac.Randal@rl.ac.uk

Theo Dimitrakos (for TrustCoM), BT, UK
E-mail: theo.dimitrakos@bt.com

Interactive Access Control and Trust Negotiation for Autonomic Communication

by Hristo Koshutanski and Fabio Massacci

Recent advances in Internet technology and globalization of peer-to-peer communications offer organizations and individuals an open environment for rapid and dynamic resource integration. In such an environment, federations of heterogeneous systems are formed with no central authority and no unified security infrastructure. Considering this level of openness each server is responsible for the management and enforcement of its own security policies with a high degree of autonomy.

Dynamic coalitions and autonomic communication add new challenges: a truly autonomic network is born when nodes are no longer within the boundary of a single enterprise, which could deploy its policies on each and every node and guarantee interoperability. An autonomic network is characterized by the self-management and self-configuration of its constituent nodes. In an autonomic network, nodes are partners that offer services and lightly integrate their efforts into one (hopefully coherent) network.

Policy-based network access and management already requires a paradigm shift in the access control mechanism: from identity-based access control to trust management and negotiation, but even this is not enough for cross-organizational autonomic communication.

In an autonomic communication scenario, a client may have all the necessary credentials to access a service but may be unaware of this. Equally, it is unrealistic to assume that servers will publish their security policies on the Web so that clients can perform policy combinations and evaluations themselves. Rather, it should be possible for a server to ask a client, on the fly, for additional credentials: the client may then choose whether or not to disclose them. The server then re-evaluates the client's request taking the newly submitted credentials into consideration, and iterates the process until a final decision (of 'grant' or 'deny') is reached. We call this modality interactive access control.

While some of these challenges can be solved by using policy-based self-management of networks, this is not universally the case. Indeed, if we abstract away the details of the policy implementation, we can observe that the only reasoning service that is actually used by policy-based self-management approaches is deduction: given a policy and a set of additional facts, find out all consequences (actions or obligations) of the policy and the facts. We simply look at whether granting the request can be deduced from the policy and the current facts.

Autonomic communication needs another reasoning service: abduction. Loosely speaking, we could say that abduction is deduction in reverse: given a policy and a request to access a service, we want to know what credentials/events are missing that would grant access.

The contribution of the framework is in the way we bootstrap from the basic reasoning services of deduction, abduction and consistency checking, a comprehensive interactive access-control algorithm that computes on the fly the missing credentials needed for a client to get access. We extended the algorithm to cope with arbitrary access policies so that in cases of inconsistency it performs a recovery step and finds a set of excessing credentials banning the client to get a solution for the desired resource. Following this, a strengthened version of the algorithm is devised that is resistant to DoS attacks. We have modelled a fully fledged ac-

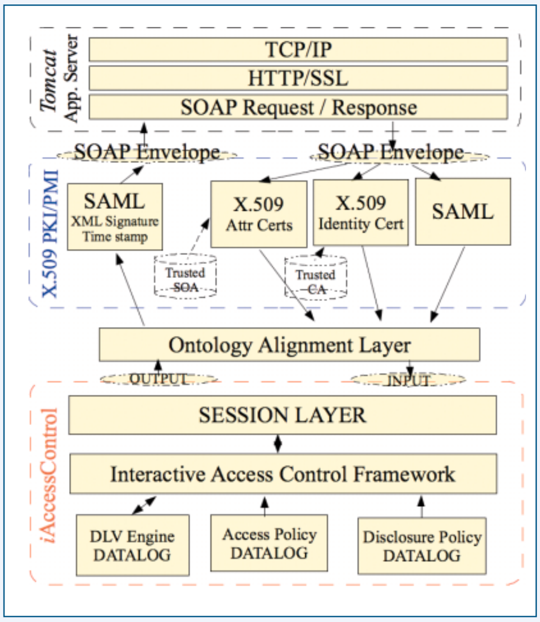

Figure 1: Interactive Access-Control Engine Architecture.

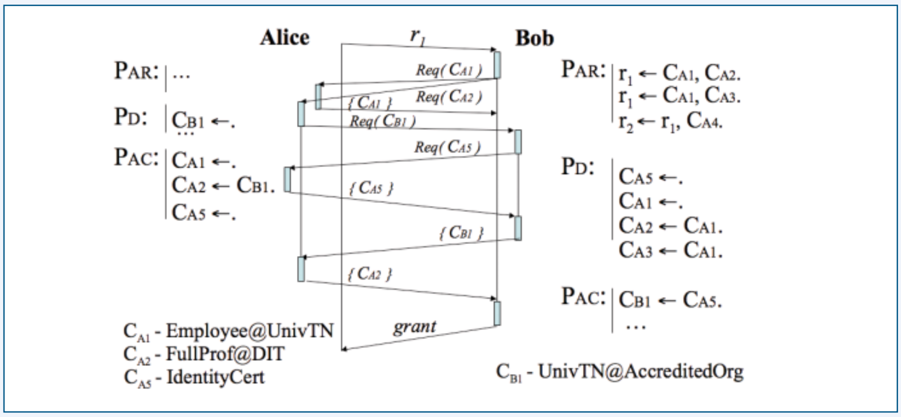

Figure 2: Example of Interoperability of the Negotiation Protocol.

P_{AR} denotes the policy for access to resources, P_{AC} denotes the policy for access to credentials and PD the policy for disclosure of (foreign) credentials. Credentials used in the policies are in the following notations: Alice's local credentials are marked with subscript 'A' and Bob's with 'B', respectively. Bob's access policy P_{AR} says that access to resource r_1 is granted if $\{C_{A1},C_{A2}\}$ or, alternatively, $\{C_{A1},C_{A3}\}$ are presented by Alice. Access to r_2 is granted if Alice satisfies the requirements for access to r_1 and presents C_{A4}.

Analogously, we read Bob's disclosure policy P_D as meaning that to disclose the need for credential C_{A2} there should exist already-disclosed credential C_{A1}, which by default is always disclosable. In contrast, the need for credential C_{A4} is never disclosed but is expected by Bob's access policy PAR when r_2 is requested.

The real interactions start when Alice requests r_1 from Bob. Then, suppose that the set $\{C_{A1},C_{A2}\}$ is minimal with respect to the other alternative $\{C_{A1},C_{A3}\}$, and say that C_{A2} contains a role lower in the hierarchy than the role in C_{A3}. Then, Bob replies with two counter requests to Alice. Alice, in her turn, runs the two requests in new threads and replies to Bob, according to her policy for access to sensitive credentials P_{AC}, with a counter-request for C_{B1} and the disclosure of C_{A2}.

The negotiation process continues, as shown in the figure, until Bob discloses all credentials requested by Alice and Alice, in her turn, discloses all credentials requested by Bob so that at the end the desired resource is granted.

cess-control framework and shown its correctness and completeness.

Based on the interactive access-control algorithm, we introduce a trust negotiation protocol that runs on both client- and server-sides. It automatically interoperates and negotiates missing credentials until either a final decision of 'grant' is taken and the negotiation is successfully completed, or one of the parties fails to negotiate the requirements and the service request is denied. Figure 2 shows a message-flow example of the negotiation protocol.

We have implemented the interactive access-control modality as a Web Service. For this purpose we used X.509 PKI/PMI and OASIS SAML standards as a unified way of conveying credentials and defining authorization statements respectively. We integrated the two standards with the W3C SOAP protocol so that our access control engine can be used and invoked in a platform-independent manner. We have done a Java wrapper for the DLV system that implements the interactive access-control algorithm. The DLV system is used as a core engine of the basic functionalities of deduction, abduction and consistency checking. The architecture of the access control engine is shown in Figure 1.

Future work will look at characterizing the complexity of the framework and extending it to cope with stateful systems and especially with the open issues of revocation of credentials.

Links:

X.509 PKI/PMI:
http://www.ietf.org/html.charters/pkix-charter.html

OASIS SAML:
http://www.oasis-open.org/committees/security

W3C SOAP: http://www.w3.org/TR/soap

DLV: http://www.dlvsystem.com

Please contact:

Hristo Koshutanski and Fabio Massacci,
University of Trento, Italy
E-mail: hristo@dit.unitn.it, massacci@dit.unitn.it