Achieving Data Privacy and Security Using Web Services

Alfred C. Weaver
University of Virginia
Department of Computer Science
Charlottesville, VA 22904 U.S.A.
weaver@cs.virginia.edu

Data Privacy and Security. A recurring concern with intelligent, distributed systems is the privacy and security of process data and software exchanges. As factory monitoring and control migrate from proprietary, closed systems to open, intranet- and internet-based systems, the risk of information leakage, malicious invasion by hackers, and damage due to software viruses increases to unacceptable levels unless system designers are proactive with regard to security. We propose a security architecture based upon web services that provides information security for factory data exchange by both human users and software processes.

The key steps are authentication, authorization, and federation, and are applicable to both human users and software that request access to any protected resource. Authentication results in a security token that conveys both the identity of the requester and the trust level of the identification technology. Authorization determines what objects are accessible by a user given his identity token, request, role, context, and privileges. Federation, using both direct and indirect trust, addresses the problem of how identity, once legitimately established in one trust domain, can be reliably exported to another cooperating trust domain. In this talk we describe the design, implementation, and lessons learned from using this security architecture for data protection.

Web Services. Our core idea is to enforce data privacy and security by means of web services, which is consistent with today's trend toward service-oriented architectures. Access to protected resources such as factory data occurs through the factory data portal, which allows the system to authenticate the user and then authorize the user's access based upon the requester's identity, role, and context. Access and authorization policies are devised by administrators, converted by the system into web service policy documents, stored in XML format, and then retrieved and interpreted on demand as data requests are received.

Authentication. Humans are authenticated by conventional methods such as PINs and passwords; by biometric identification techniques such as fingerprints, iris scans, signature recognition, and voice recognition; and by digital techniques such as e-tokens, RFID, and two-factor approaches (e.g., a combination of a password and a random number that changes once per minute). In all cases the result of a successful identification is the generation of an authentication token issued by a Secure Token Service (STS). The authentication token is stored on the user's access device after generation, and is presented with each subsequent web portal access and data request. All tokens carry expiration times and must be periodically renewed. Further, each token carries a numerical representation of its trust level—a quantification of how much the system trusts the authentication technology.

Authorization. Each data request arriving from the factory web portal is accompanied by its authentication token. The authorization web service consults its authorization policies (stored as WS-Policy documents in XML format) to determine, based upon the details of the request and the requester's identity, role, and context, whether or not the request should be granted. Conventional Role-Based Access Control (RBAC) models are not sufficiently powerful for modern applications, and thus have been extended by adding context-based access control. This allows the authorization policy to interrogate the local circumstances of the request (e.g., time of day, IP address of the access point, whether the request originated from a wireless device) so that the authorization rules can be applied dynamically. The Policy Decision Point is empowered to gather all data needed to make an informed decision, which is then passed to the Policy Enforcement Point for implementation.

Federation. Data requests that originate within the local trust domain can be handled locally by the STS and the authorization web service. However, requests originating in one domain (e.g., company X) can be legitimately intended for another domain (e.g., a partner company Y). In this case, company X has no control over the security architecture or policies implemented by company Y, so requests emanating from X need special processing. The goal is that identity, once legitimately established in one trust domain, can be reliably exported to another one, consistent with the web service policies of both. The authentication token from domain X must now be authenticated by domain Y. One approach is to implement direct trust, in which a verified token from one domain is exchanged for a valid token in the other domain. This is practical when the number of interacting domains is modest and trust can be established and maintained through human administrators. For larger-scale interactions, interactions across domains must be administered automatically. This requires indirect trust whereby domains X and Y must locate another domain Z that they both trust; security tokens are then verified and exchanged through domain Z.

Research Challenges. While the security architecture outlined above is consistent with the emerging standards for web services, there are nevertheless significant implementation challenges, including:

(1) The authentication trust level is a simplistic quantification of trust. Within a single authentication technology, say biometrics, differing trust levels can be assigned and justified based upon evidence such as the number of degrees of freedom in the underlying technology (this is what makes retina scans more reliable than iris scans, which in turn are more reliable than fingerprints). But this concept is hard to extend outside a technology type, say when trying to compare the trust levels of fingerprints and RFID-equipped ID badges.

(2) Authorization rules can be implemented by the authorization web services if they can be reduced to Boolean equations. This is easier to assert than to achieve in practice. Human organizations are complex, and thus tend to generate complex rules. A prime example of complexity is role delegation wherein the privileges of one person are temporarily granted to another, but under controlled circumstances. Complexity must be reduced and contradictions eliminated for reliable construction and implementation of machine-readable policies.

(3) Federation is in its infancy, and the standards bodies have yet to agree on techniques and formats. Yet it is essential that legitimate requests from one domain be recognized, verified, and accepted by another; without this capability, effective cooperation across institutions is impossible.

(4) The tools and techniques for implementing all the above are still experimental. The human/computer interface is especially important. For example, humans can articulate access policies, but it is unrealistic to expect them to input them as Boolean equations. The research community has much work to do to make these ideas accessible and effective.

(5) Systems of the complexity described can not be simply implemented and then trusted. Instead, we need formalisms to structure our thoughts, processes, and implementations. Formal proofs of correctness will be required to justify trust and cooperation across domains.