Automated VAlidatioN of Trust and Security of Service-oriented ARchitectures

FP7-ICT-2007-1, Project No. 216471

www.avantssar.eu

Deliverable D6.3
Migration to standardisation bodies

Abstract

The migration to standardisation bodies provides a means to discuss the results of AVANTSSAR with standardisation organizations. This includes the result of AVANTSSAR problem cases, and in particular of SAML SSO. The migration to standardisation bodies requires a certain level of maturity of the AVANTSSAR results and this is why this sub-workpackage has been carried out only for the last year of the project.

Deliverable details

Deliverable version: *v1.0*
Classification: *public*
Date of delivery: *03.01.2011*
Due on: *31.12.2010*
Editors: *SIEMENS, UNIVR, UGDIST, SAP*
Total pages: *11*

Project details

Start date: *January 01, 2008*
Duration: *36 months*
Project Coordinator: *Luca Viganò*
Partners: UNIVR, ETH Zurich, INRIA, UPS-IRIT, UGDIST, IBM, OpenTrust, IEAT, SAP, SIEMENS

Contents

1 Introduction

Quite often, insecurities arise with complexity: interactions of sub-protocols and functionalities, adjustment of policies, low-level decisions, etc. Open standards define protocols or services that are interoperable by design. A particular instance may be composed using modules created by different vendors and operated by different organizations or users. In order for the modules to be able to work together and to provide multi-lateral security to the different stakeholders, it is necessary that different software versions co-exist, that each of them implements different security options for different security policies, and that they support the negotiation of those options, policies, versions, etc.

This usually is bad news for privacy and security. Even using a carefully defined service, the presence of many options and functionalities, many configuration possibilities, many implementations, and/or many low-level decisions is likely to give rise to vulnerabilities due to difficulties in composing and configuring an instance, and understanding the consequences of single decisions in a complex environment.

Moreover, comprehensive security analyses of protocols or services are seldom available: the security aspects are usually presented in a constraint-based manner and only an attack-by-attack description of countermeasures is provided. This can lead developers to overlook security-critical aspects of the services or to fail to understand its impact on the service security.

The main contribution of AVANTSSAR regarding standards is that it helps, with a comparatively small effort, to find errors in standard specifications or in hand-crafted models of particular implementation choices. The AVANTSSAR project has been contributing to standardisation through its partners in several organisations. Our results regarding the SAML standards have been discussed directly with main members of

- the Organization for the Advancement of Structured Information Standards (OASIS),
- the Internet Engineering Task Force (IETF),
- the World Wide Web Consortium (W3C),
- the Internet Architecture Board (IAB), and,
- the ETSI Interest Group on Identity and Access Management for Networks and Services (ETSI ISG INS), via Nokia Siemens Networks (NSN).

Our work has already influenced and secured some SAML SSO de-facto industry standards, like the Google SAML SSO and the simpleSAMLphp, which have changed their specifications based on our input (see [15, 14] and [8]). Similarly, and based on the discussions we have conducted and the feedback we have received, we trust that our results will soon be integrated in the OASIS SAML standards.

Besides our results regarding SAML, we have also discussed our project methods and tools with the mentioned standardisation organizations. As shown, for instance, in [13], our tools can be used to find vulnerabilities in many different standard specifications (in this case, PKCS#11).

In the following sections, we give more details on the standardisation topics, as well as on related activities and contributions.

2 SAML-SSO

2.1 The importance of SAML

SAML lies in the core of Web Services Security. It is arguably the most important single standard in the landscape of security services and protocols for Web services. SAML 2.0 is the de-facto standard for SSO for Web services [11] and Google offers a SAML-based SSO service for its Google Apps. An Identity Provider (IdP) authenticates and issues authentication assertions for the clients of the SSO service. These assertions are consumed by Service Providers (SPs). By means of the SSO service, the client can authenticate once with the IdP and get transparent access to all the services offered by the available SPs.

Most of the main security standard protocols, such as WS-Security or XACML, depend on SAML for gaining access to the user authentication and credentials, and rely on the information contained in the SAML assertions. WS-Security is used to provide integrity and confidentiality of web-service messages, whereas XACML is used to express and convey authorization-related information.

But as soon as an attacker is able to steal the SAML assertions of a trusted (and trustworthy) user, say *Alice*, the attacker is able to masquerade as *Alice* within all different contexts and purposes: WS-Security allows the attacker to counterfeit "secure" messages and make them pass as original messages of *Alice*; XACML allows the attacker to pass the authentication credentials of *Alice* to any service and gain access in her name.

SAML is used everywhere, in many specifications of varying degrees of maturity, and is maintained and supported by different standardisation bod-

ies and entities. The standards based on SAML complement, overlap, and compete with each other; they include Microsoft's federated identity platform, .NET, Active Directory, WS-Federation, WS-Privacy, WS-Policy, WS-Trust, WS-SecureConversation, Shibboleth, and many others. An attack to SAML renders WS-Security, XACML and all other protocols insecure.

Hence, if SAML fails, most of the security of web services breaks down. The alternatives to SAML — Kerberos and X.509 — are not easy to embed in web services since they are not XML-based standards and do not provide single-sign for web services as seamlessly as SAML. Also for this reason, contributing to the security of SAML is securing the whole edifice of web services standards and specifications.

2.2 SAML-based SSO for Google Apps

A highlight of the effectiveness of the AVANTSSAR methods and tools is the detection of a serious flaw in the SAML-based SSO solution for Google Apps [3].

SAML is indeed well specified and thoroughly documented, but the standard is complex. It encompasses several specifications in interconnected documents, including many different options for assertions, protocols, bindings, profiles, metadata, authentication context, and conformance requirements. The options and choices are interrelated and interdependent. Since the specifications are written in natural language, they are often subject to interpretation. It may be difficult to establish when some message fields are mandatory in a given profile and context and which ones not, or to understand what are the consequences of such choices. On top of this, producers of SAML-based solutions have their own internal requirements that may result in small deviations from the standard.

For instance, internal requirements (or DoS considerations) may induce the Service Provider to avoid checking that the ID field in the AuthResp coincides with that included in the previously sent AuthReq.

The technical overview document provided by OASIS SAML as an addendum non-official document increases the clarity in this respect. Still, things went wrong when Google designed and developed its SAML-based SSO solution for Google Apps. The flaw allowed a dishonest service provider to impersonate the victim user on Google Apps thereby granting unauthorized access to private data and services (email, docs, etc.). The vulnerability was discovered with SATMC (see [4, 1]), a fully automatic security protocol analyzer that is part of our AVANTSSAR Validation Platform and was reproduced in an actual deployment of SAML-based SSO for Google Apps. Google and the US Computer Emergency Readiness Team (US-CERT) were

informed and the vulnerability was kept confidential until Google developed a new version of the authentication service and Google's customers updated their applications accordingly. The severity of the vulnerability has been rated High in a note issued by the National Institute of Standard and Technology (NIST).

2.3 An authentication flaw in the SAML standard

In [2], we show that the prototypical SAML SSO use case (as described in the SAML Technical Overview) suffers from an authentication flaw that, under some conditions, allows a malicious service provider to hijack a client authentication attempt and force the latter to access a resource without its consent or intention. It also allows an attacker to launch Cross-Site Scripting (XSS) and Cross-Site Request Forgery Attacks (XSRF). This last type of attack is even more pernicious than classic XSRF, because XSRF require the client to have an active session with the service provider, whereas in this case, the session is created automatically hijacking the client's authentication attempt. This may have serious consequences, as witnessed by the new XSS attack that we identified in the SAML-based SSO for Google Apps and that could have allowed a malicious web server to impersonate a user on any Google application. In our paper, we describe solutions that can be used to mitigate and even solve the problem. These possible solutions are being discussed with OASIS.

3 The use of formal methods in standardisation

Designers and developers, while striving to meet the requirements posed by their own application scenarios, have hard time to assess the security and privacy impact of a selected option, a seemingly minor deviation, a specific combination of functionalities, etc. This often results in the release of flawed products to end-users.

This issue can be significantly mitigated by empowering designers and developers with custom verification tools, i.e., tools that offer usable graphical interfaces and user-friendly notations, and employ, behind the scenes, established verification techniques (e.g., model checking) to efficiently tackle specific industrial relevant problems (e.g., SAML SSO).

One example is the enriched Alice-and-Bob-style specification language developed by IBM for security protocols [6, 9, 10].

As a second example, SAP Research has devised the tool *SAML Modelling Environment* (SAML ME) [5] as a Java application that analyzes the different choices in SAML Federated environments. It combines a simple graphical user interface, a rule engine that checks if the chosen configuration options are coherent with the OASIS SAML specification, and a security validation process that formally analyzes the configured federated environment via the AVANTSSAR Validation Platform. Our prototype is able to establish whether a decision can be source of security flaws in the overall federated environment.

As a last example, M. Bortolozzo, M. Centenaro, R. Focardi and G. Steel have used the tool Tookan, see [7, 13], based on our tool SATMC [4, 1], to automatically find vulnerabilities in PKCS#11-based products by Aladdin, Bull, Gemalto, RSA, and Siemens among others. PKCS#11 [12] specifies an API for performing cryptographic operations such as encryption and signature using cryptographic tokens (e.g., USB tokens or smart cards). Sensitive cryptographic keys, stored inside the token, should not be revealed to the outside world and it should be impossible for an attacker to change those secret, unextractable keys. The attacks found show that in many implementations this is not the case: the compromise of a key allows an attacker to clone the token and, more generally, to perform the same security-critical operations as the legitimate token user.

4 Conclusion

AVANTSSAR, through the Industry Migration workpackage (WP 6), has taken current industrial best practice languages and models into account. Tools and languages have been carefully defined and provided to designers to assist them in extending their models with the augmentations required for validation. This same path is open and available to the community in general, and to the standardisation organizations in particular, in order to be able to exploit the AVANTSSAR results in more real-world industrial settings. The AVANTSSAR approach does not require disruptive changes, but its Industry Migration allows for a smooth integration in existing environments.

AVANTSSAR tools are starting to make a difference for de-facto industry standards and open technical standards. Some key messages to the standardisation bodies are:

- Tools can help to assess the impact of a decision, configuration, etc.: they cope with the complexity (e.g., combinatorial issue) where humans can not.

- They can be used during standardisation. They are easy-to-use, so that it is not necessary to have an expert to use the tools. They are efficient so that designers will receive answers in due time, and they can be customized for special needs.
- For instance, the tool created for SAML SSO analyzes the possible consequences of different choices in a SAML service and mitigates, by means of automated formal analysis, the risk of deploying flawed versions of the service.

References

[1] A. Armando, R. Carbone, and L. Compagna. LTL Model Checking for Security Protocols. In *Journal of Applied Non-Classical Logics, special issue on Logic and Information Security*, pages 403–429. Hermes Lavoisier, 2009.

[2] A. Armando, R. Carbone, L. Compagna, J. Cuéllar, G. Pellegrino, and A. Sorniotti. From multiple credentials to web-browser single sign-on: Are we more secure? submitted for publication, 2011.

[3] A. Armando, R. Carbone, L. Compagna, J. Cuéllar, and L. Tobarra Abad. Formal Analysis of SAML 2.0 Web Browser Single Sign-On: Breaking the SAML-based Single Sign-On for Google Apps. In *Proceedings of 6th FMSE*. ACM Press, 2008.

[4] A. Armando and L. Compagna. SATMC: a SAT-based model checker for security protocols. In *Proceedings of the 9th European Conference on Logics in Artificial Intelligence (JELIA'04)*, volume 3229 of *LNAI*, pages 730–733, Lisbon, Portugal, 2004. Springer-Verlag.

[5] AVANTSSAR. Deliverable 1.4: Progress/Assessment Report for Year 2 (Period P2: 01.01.09 – 31.12.09). Available at http://www.avantssar.eu, 2010.

[6] AVANTSSAR. Deliverable 6.2.3: Migration to industrial development environments: lessons learned and best practices. Available at http://www.avantssar.eu, 2010.

[7] M. Bortolozzo, M. Centenaro, R. Focardi, and G. Steel. Attacking and Fixing PKCS#11 Security Tokens. In *Proceedings of the 17th ACM Conference on Computer and Communications Security (ACM CCS 2010)*, 2010.

[8] Feide RnD. A security fix for simplesamlphp-1.6.3, 2012. https://rnd.feide.no/2010/12/20/simplesamlphp-1-6-3-is-available-with-a-security-fix/.

[9] S. Mödersheim. Algebraic Properties in Alice and Bob Notation. In *Proceedings of Ares 2009*, pages 433–440. IEEE Computer Society Press, 2009. DOI:http://doi.ieeecomputersociety.org/10.1109/ARES.2009.95. An extended version is available as Technical Report no. RZ3709, IBM Zurich Research Lab, 2008, domino.research.ibm.com/library/cyberdig.nsf.

[10] S. Mödersheim. Algebraic Properties in Alice and Bob Notation. In *Proceedings of ARES 2009*, 2009. Extended version available as IBM Research Report RZ3709 at domino.research.ibm.com/library/cyberdig.nsf.

[11] OASIS. Assertions and protocols for the OASIS security assertion markup language (SAML) v2.0, 2005.

[12] RSA Laboratories. PKCS#11: Cryptographic Token Interface Standard. http://www.rsa.com/rsalabs/node.asp?id=2133.

[13] Tookan: TOOl for cryptoKi ANalysis. http://secgroup.ext.dsi.unive.it/projects/security-apis/pkcs11-security/tookan/.

[14] US-CERT. Google information for vu#612636, 2008. http://www.kb.cert.org/vuls/id/MIMG-7FQGWU.

[15] US-CERT. Vulnerability note vu#612636: Google saml single sign on vulnerability, 2008. http://www.kb.cert.org/vuls/id/612636.