Chapter 10
Automatic Security Analysis of SAML-Based Single Sign-On Protocols

Alessandro Armando
University of Genova, Italy & Fondazione Bruno Kessler, Italy

Roberto Carbone
Fondazione Bruno Kessler, Italy

Luca Compagna
SAP Research Sophia-Antipolis, France

Giancarlo Pellegrino
SAP Research Sophia-Antipolis, France & Eurécom, France

ABSTRACT

Single-Sign-On (SSO) protocols enable companies to establish a federated environment in which clients sign in the system once and yet are able to access to services offered by different companies. The OASIS Security Assertion Markup Language (SAML) 2.0 Web Browser SSO Profile is the emerging standard in this context. In previous work a severe security flaw in the SAML-based SSO for Google Apps was discovered. By leveraging this experience, this chapter will show that model checking techniques for security protocols can support the development and analysis of SSO solutions helping the designer not only to detect serious security flaws early in the development life-cycle but also to provide assurance on the security of the solutions identified.

INTRODUCTION

Single Sign-On (SSO) protocols are the cornerstones of Identity and Access Management systems as they enable companies to establish a federated environment in which users sign in once and yet are able to access to services offered by different organizations.

The *Security Assertion Markup Language* (SAML) 2.0 Web Browser SSO Profile (SAML SSO, for short) (OASIS, 2005a) is the emerging standard in this context: it defines an XML-based format for encoding security assertions as well as

DOI: 10.4018/978-1-61350-498-7.ch010

a number of protocols and bindings that prescribe how assertions should be exchanged in a wide variety of applications and/or deployment scenarios. This is done to the minimum extent necessary to guarantee the interoperability among different implementations. As a consequence, SAML SSO features many configuration options ranging from optional fields in messages, usage of SSL 3.0 or TLS 1.0 channels (SSL channels from here on) at the transport layer, application of encryption and/or digital signature on specific sensitive message elements which need to be instantiated according to the requirements posed by the application scenario and the available security mechanisms.

The security recommendations that are available throughout the bulky SAML specifications (OASIS, 2005a, 2005b) are useful in avoiding the most common security pitfalls but are of little help in ensuring their absence in specific instances of the protocol. Indeed the designer of a SAML-based SSO solution, while striving to meet the requirements posed by the application scenario, may overlook the security implications associated with the choice of some optional elements or may even decide to deviate from the SAML standard. Needless to say, this may have dramatic consequences on the security of the SSO solution.

The situation is exemplified by the SAML-based SSO for Google Apps. The protocol is inspired by the SAML standard but the version in operation until June 2008 deviated from it in a few, seemingly minor aspects (see Section «The SAML Web Browser SSO Profile» for the details). In May 2008 a severe security flaw was discovered and reported to Google and US-CERT (Armando, Carbone, Compagna, Cuéllar, & Tobarra, 2008, US-CERT, 2008). The vulnerability allowed a dishonest service provider to impersonate a user at another service provider. In reaction to the finding, Google immediately asked their customers to implement counter-measures to mitigate potential exploits and then to migrate to a new, patched solution of their SSO solution. Interestingly, the vulnerability was found by using a model checker for security protocols.

By leveraging this experience, this chapter will show that model checking techniques for security protocols can support the development and analysis of SSO solutions helping designers not only to detect serious security flaws early in the development life-cycle but also to provide assurance on the security of the solutions identified. By using SAML-based SSO protocols as running examples it will be illustrated how the design space can be iteratively explored with the aid of a model checker leading to the identification of vulnerabilities and hence to more secure solutions.

This work is part of a wider project aimed at the development of automated verification technologies for security protocols and, more generally, for security-sensitive, distributed applications (Armando et al., 2005, The AVANTSSAR Team).

Structure of the Chapter

In the next section the scene is set by briefly describing the state-of-the-art in model checking of security protocols. In Section «The SAML Web Browser SSO Profile» a brief introduction of the SAML SSO is given. In Section «Formal Modeling of the SAML Web Browser SSO Profile» the specification formalism is given in order to model security protocols, the properties of the transport protocols as well as those that the protocols are expected to meet. Section «Security Analysis of SAML-based SSO Protocols» presents the results of the analysis of SAML-based SSO protocols. In Section «Future Research Directions» the future research direction is discussed and Section «Conclusion» concludes the chapter with some final remarks.

BACKGROUND

SSO protocols belong to the wider family of security protocols, i.e. communication protocols