WebSand: Server-Driven Outbound Web-Application Sandboxing*

Martin Johns[1] and Joachim Posegga[2]

[1] SAP Research, Germany
martin.johns@sap.com
[2] ISL, University of Passau, Germany
jp@sec.uni-passau.de

1 Motivation

The Web started in 1990 as a simple, stateless delivery mechanism for static hypertext documents; it evolved over time into a fully-fledged run-time environment for distributed, multi-party applications. Even today, new features and capabilities are added continuously, which drives the Web's chaotic evolution.

Security became increasingly important, in particular with the commercialization of the Web around 2000; but even today security is typically only an afterthought in the Web's evolutionary process. Today's primarily server-centric solutions provide a rich and stateful client-centric paradigm, but barely any manageable security: Data and services from multiple heterogeneous domains, aggregated both on the server-side and on an end-user's clients, demand a novel, comprehensive security solution, addressing fundamental security requirements. This is what the Websand project is about.

2 Project Objective

WebSand tackles security beyond dealing with low-level vulnerabilities at a higher level of abstraction: The technical strategy is to deal with security in a server-driven fashion. Clearly, security preferences and requirements from end-users at the client-side need to be taken into account, but primarily service developers at the server-side have the required expertise and context information to define adequate policies to be enforced. Moreover, server-driven security can be deployed relatively easily, since the need for updating the client-side platform is minimized.

Since WebSand strands for "Server-driven Outbound *Web*-application *Sand*-boxing", the project's overall goal is –along with this strategy– to empower web application developers, service providers, and users in designing, implementing, and running secure applications: Developers and service providers can develop

* WebSand is funded by the EU under FP7-256964. The WebSand consortium consists of SAP AG, Katholieke Universiteit Leuven, Chalmers Tekniska Hoegskola AB, University of Passau, Siemens AG. http://websand.eu

S. Fischer-Hübner, S. Katsikas, G. Quirchmayr (Eds.): TrustBus 2012, LNCS 7449, pp. 217–218, 2012.

and deploy secure web applications on their application servers; users will benefit from the project's results by transparently receiving a suitable security platform for their applications. WebSand aims to deliver this non-disruptively, i.e. by building upon existing web application technologies wherever possible to allow a seamless, immediate adoption of results in existing and future web applications.

3 Approach, Organisation, and Challenges

Websand identified three main focal areas for it's technical objectives:

Secure Web Interaction: The public interface of a Web application consists of the set of incoming HTTP request that it handles. Consequently, security properties, such as authentication and authorization, are directly linked to properties of incoming HTTP traffic. However, the original one-to-one browser/server relationship of early Web applications has been replaced recently with application scenarios spanning multiple clients and severs, interacting within one application context. The established concepts for authentication, cross-domain interaction, and control-flow integrity must hence be revisited and adapted to meet the security challenges of the evolving Web application interaction.

Secure Composition: Web 2.0 applications –unlike any other application model– frequently mix data and executable code from different service providers. Web browsers were initially not designed to cater for such scenarios, and application developers frequently encounter situations where the current trust model of the Web browser's same-origin Policy in insufficient: it only allows either full or no trust at all between components. WebSand's secure composition policies are much more expressive, allowing to specify privileges of each component, including behavioral capabilities and interaction constraints. This enables least-privilege composition and the enforcement of secure multi-origin policies.

Secure Information Flow Control: If application components from different sources are executed in a shared context, as in multi-party, mash-up driven Web applications, unintended and potentially insecure flow of sensitive information can occur. Information flow control governs sensitive and public data, possibly originating from multiple content providers in multiple trust domains; such data can be used in data aggregations or client-side and server-side processing as typically seen in mashups. Particular challenges for this task arise from Web browser's flexible nature and JavaScript's dynamic characteristics.

4 Summary

WebSand aims at developing a foundation for developers to build multi-party Web applications with robust security guarantees in non-trivial settings. The project defines fine-grained security policies and applies novel sandboxing techniques to the application, to enable a client-side enforcement of the given security policies. Whenever applicable, WebSand will build upon emerging Web standards; for its novel contributions, WebSand targets compatibility to such standards. This should enable the use of WebSand techniques together with these standards and support future inclusion of WebSand's contributions.