

A Collection of Real World (JavaScript) Security Problems

Examples from 2½ Applications Areas of JavaScript

Achim D. Brucker
achim.brucker@sap.com

SAP AG, Central Code Analysis Team

July 2, 2014

SAP

Agenda

- 1 Motivation and Basics
- 2 SAP UI5: Client-side JavaScript
- 3 Apache Cordova: JavaScript on Mobile
- 4 HANA XS Engine: Server-side JavaScript

A Collection of Real World (JavaScript) Security Problems

Abstract

JavaScript is gaining more and more popularity as an implementation language for various applications types such as Web applications (client-side), mobile applications, or server-side applications. We outline a few security challenges that need to be prevented in such applications and, thus, for which there is a demand for analysis methods that help to detect them during during development.

Agenda

- 1 Motivation and Basics
- 2 SAP UI5: Client-side JavaScript
- 3 Apache Cordova: JavaScript on Mobile
- 4 HANA XS Engine: Server-side JavaScript

What We Want to Find

Programming Patterns That May Cause Security Vulnerabilities

Mainly two patterns

Local issues (no data-flow dependency), e. g.,

- Insecure functions

```
1 var x = Math.random();
```

- Secrets stored in the source code

```
1 var password = 'secret';
```

Data-flow related issues, e. g.,

- Cross-site Scripting (XSS)

```
1 var docref = document.location.href;
2 var input = docref.substring(
3     docref.indexOf("default=")+8);
4 var fake = function (x) {return x;}
5 var cleanse = function (x) {
6     return 'hello_world';}
7 document.write(fake(input));
8 document.write(cleanse(uinput));
```

- Secrets stored in the source code

```
1 var foo = 'secret';
2 var x = decrypt(foo,data);
```

Functions as First-Class Objects

```
1 var href = document.location.href;
2 var unsafeInput = href.substring(href.indexOf("default=")+8) // unsafe input
3 var safeInput = "1+2"; // safe input
4
5 // aliasing eval
6 var exec = eval;
7 var doit = exec;
8
9 var func_eval1 = function (x) {eval(x)};
10 var func_eval2 = function (x,y) {eval(y)};
11
12 var func_eval_eval = function (x) {func_eval1(x)};
13 var func_doit = function (x) {doit(x)};
14 var func_exec = function (x) {exec(y)};
15 var run = func_eval1;
16 var inject_code = func_exec;
17
18 doit(safeInput); // secure
19 doit(unsafeInput); // code injection
```

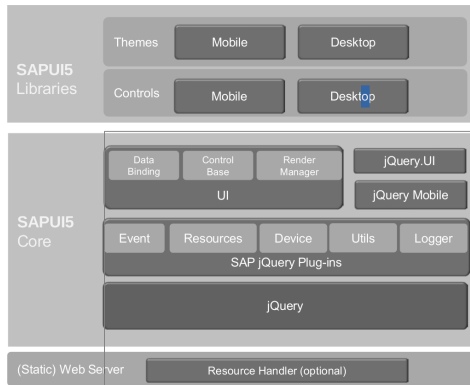
Where is The Code of my Application?

```
1 var input = document.location.href.substring(document.location..indexOf("default=")+8);
2 var fake = function (x) {return x;}
3 var cleanse = function (x) {return 'hello_world'};
4
5 var uinput = unknown(input); // unknown is nowhere defined
6 document.write(uinput); // secure!?
7
8 var finput = fake(input);
9 document.write(finput); // not secure
10
11 var cinput = cleanse(input);
12 document.write(cinput); // secure
13
14 var extfinput = extfake(input); // defined externally (part of scan)
15 document.write(extfinput); // not secure
16
17 var extcinput = extcleanse(input); defined externally (part of scan)
18 document.write(extcinput); // secure
19
20 var nobodyKnows = toCleanOrNotToCleanse(input); multiply defined (underspecified)
21 document.write(nobodyKnows); // not secure!?
```

Agenda

- 1 Motivation and Basics
- 2 SAP UI5: Client-side JavaScript
- 3 Apache Cordova: JavaScript on Mobile
- 4 HANA XS Engine: Server-side JavaScript

The SAP UI5 Architecture



Prototype-based Inheritance

```
1 var vl = new sap.ui.commons.layout.VerticalLayout();
2 sap.ui.core.Control.extend("foo.Label", {
3   metadata : {
4     properties : {
5       "text" : "string"
6     }
7   },
8   renderer : function(oRm, oControl) {
9     oRm.write("<span>XSSLabel:_"");
10    oRm.write(oControl.getText());
11    oRm.write("</span>");
12  }
13 });
14 var p = jQuery.sap.getUriParameters().get("xss");
15 vl.addContent(new foo.Label({text:p}));
16 return vl;
```

CSRF Prevention

You need to know your frameworks

```
1 var request = {
2   headers : {
3     "X-Requested-With" : "XMLHttpRequest",
4     "Content-Type" : "application/atom+xml",
5     "X-CSRF-Token" : "Fetch"
6   },
7 };
8 if (Appcc.CSRFToken)
9   var request = {
10    headers : {
11      "X-Requested-With" : "XMLHttpRequest",
12      "Content-Type" : "application/atom+xml",
13      "X-CSRF-Token" : Appcc.CSRFToken
14    },
15  };
16 else var request = {
17   headers : {
18     "X-Requested-With" : "XMLHttpRequest",
19     "Content-Type" : "application/atom+xml",
20     "X-CSRF-Token" : "etch" // secure?
21   },
22 };
```

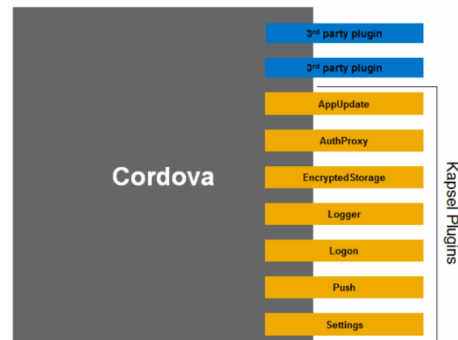
Agenda

- 1 Motivation and Basics
- 2 SAP UI5: Client-side JavaScript
- 3 Apache Cordova: JavaScript on Mobile
- 4 HANA XS Engine: Server-side JavaScript

Apache Cordova (SAP Kapsel): Overall Idea

An integrated platform for developing hybrid mobile apps

- Apache Cordova plus
 - App management
 - Encrypted Storage
 - Authentication
 - Logging
 - ...
- Application management (SMP)
- Can be used with device management solutions



Agenda

- 1 Motivation and Basics
- 2 SAP UI5: Client-side JavaScript
- 3 Apache Cordova: JavaScript on Mobile
- 4 HANA XS Engine: Server-side JavaScript

Exploiting the JavaScript to Java Bridge

- We can expose Java methods in JavaScript

```
foo.addJavaScriptInterface(new FileUtils(), "FUtil");
```

- And use them in JavaScript easily

```
1 <script type="text/javascript"> // 
2 filename = '/data/data/com.livingsocial.www/' + id + '_cache.txt';
3 FUtil.write(filename, data, false);
4 // ]]&gt;&lt;/script&gt;</pre></div><div data-bbox="525 361 760 379" data-label="List-Group"><ul><li>• Which might expose much more than expected</li></ul></div><div data-bbox="531 386 853 442" data-label="Text"><pre>1 function execute(cmd){
2     return
3     window._cordovaNative.getClass().forName('java.lang.Runtime').
4     getMethod('getRuntime', null).invoke(null, null).exec(cmd);
5 }</pre></div><div data-bbox="516 470 619 482" data-label="Page-Footer"><p>© 2014 SAP AG. All Rights Reserved.</p></div><div data-bbox="925 470 973 482" data-label="Page-Footer"><p>Page 14 of 18</p></div><div data-bbox="513 525 775 547" data-label="Section-Header"><h2>The HANA XS Engine Architecture</h2></div><div data-bbox="513 548 560 562" data-label="Text"><p>Overview</p></div><div data-bbox="575 581 894 813" data-label="Diagram"><img alt="Diagram of the HANA XS Engine Architecture. It shows three layers: Client: Browser or Mobile (Presentation logic), SAP HANA (Control flow logic), and Data (Calculation logic). Front-end Technologies (http/s, HTML5 / SAPUI5, Client-side JavaScript) are associated with the Client layer. Control Flow Technologies (OData, Server-Side JavaScript, XMLA) are associated with the SAP HANA layer. Data Processing Technologies (SQL / SqlScript, Calculation Engine Functions, Application Function Library (AFL)) are associated with the Data layer."/></div><div data-bbox="516 831 619 843" data-label="Page-Footer"><p>© 2014 SAP AG. All Rights Reserved.</p></div><div data-bbox="925 831 973 843" data-label="Page-Footer"><p>Page 16 of 18</p></div>
```

History Repeats: SQL Injection

```
1 $.response.contentType = "text/html";
2 var userInput = $.request.parameters.get('userStuff');
3
4 // We assume
5 // - $.db.getConnection().prepareStatement(x0, ..., xn) is secure iff x0 is *not*
6 //   influenced by user input
7 // - sqlSanitize() safeguards us against SQL injections.
8 // - any other preparedStatement call is evil regardless if it is influenced by
9 //   user input or not
10
11 if (userInput) {
12
13     var sql = "select_*_from_SFLIGHT.SNVOICE_where_CustomID_=" +
14             + userInput + "'";
15     var safe_sql = "select_*_from_SFLIGHT.SNVOICE_where_CustomID_=" +
16                 + sqlSanitize(userInput) + "'";
17
18     var db_object = $.db;
19     var conn = db_object.getConnection();
20
21     var pstmt00 = $.db.getConnection().prepareStatement(sql); // SQL injection
22     var pstmt01 = $.db.getConnection().prepareStatement(safe_sql); // secure
```

History Repeats: SQL Injection

```
1 var sql = "select_*_from_SFLIGHT.SNVOICE_where_CustomID_=" +
2         + userInput + "'";
3 var safe_sql = "select_*_from_SFLIGHT.SNVOICE_where_CustomID_=" +
4             + sqlSanitize(userInput) + "'";
5
6 var db_object = $.db;
7 var conn = db_object.getConnection();
8
9 var pstmt00 = $.db.getConnection().prepareStatement(sql); // SQL injection
10 var pstmt01 = $.db.getConnection().prepareStatement(safe_sql); // secure
11
12 var pstmt02 = db_object.getConnection().prepareStatement(sql); // SQL injection
13 var pstmt03 = db_object.getConnection().prepareStatement(safe_sql); // secure
14
15 var pstmt04 = conn.prepareStatement(sql); // SQL injection
16 var pstmt05 = conn.prepareStatement(safe_sql); // secure
17
18 var pstmt06 = conn.prepareStatement("...where_ID_='&#39;$1'&#39;,userInput); // secure
19 var pstmt07 = myconn.prepareStatement("...where_ID_='&#39;$1'&#39;,userInput); // SQL injection
20
21 var pstmt08 = $.mydb.getConnection().prepareStatement(sql); // SQL injection
22 var pstmt09 = $.mydb.getConnection().prepareStatement(safe_sql); // SQL injection
```

© 2014 SAP AG. All rights reserved

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG. The information contained herein may be changed without prior notice.

Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors.

Microsoft, Windows, Excel, Outlook, and PowerPoint are registered trademarks of Microsoft Corporation.
IBM, DB2, DB2 Universal Database, System i, System i5, System p, System p5, System x, System z, System z10, System z9, z10, z9, iSeries, pSeries, xSeries, zSeries, eServer, zVM, z/OS, i5/OS, S/390, OS/390, OS/400, AS/400, S/390 Parallel Enterprise Server, PowerVM, Power Architecture, POWER6+, POWER6, POWER5+, POWER5, POWER, OpenPower, PowerPC, BatchPipes, BladeCenter, System Storage, GPFS, HACMP, RETAIN, DB2 Connect, RAG, Redbooks, OS/2, Parallel Sysplex, MVS/ESA, AIX, Intelligent Miner, WebSphere, Netfinity, Tivoli and Informix are trademarks or registered trademarks of IBM Corporation.

Linux is the registered trademark of Linus Torvalds in the U.S. and other countries.
Adobe, the Adobe logo, Acrobat, PostScript, and Reader are either trademarks or registered trademarks of Adobe Systems Incorporated in the United States and/or other countries.

Oracle is a registered trademark of Oracle Corporation.
UNIX, X/Open, OSF/1, and Motif are registered trademarks of the Open Group.
Citrix, ICA, Program Neighborhood, MetaFrame, WinFrame, VideoFrame, and MultiWin are trademarks or registered trademarks of Citrix Systems, Inc.

HTML, XML, XHTML and W3C are trademarks or registered trademarks of W3C®. World Wide Web Consortium, Massachusetts Institute of Technology.

Java is a registered trademark of Sun Microsystems, Inc.
JavaScript is a registered trademark of Sun Microsystems, Inc., used under license for technology invented and implemented by Netscape.
SAP, R/3, SAP NetWeaver, Duet, PartnerEdge, ByDesign, SAP BusinessObjects Explorer, StreamWork, and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and other countries.

Business Objects and the Business Objects logo, BusinessObjects, Crystal Reports, Crystal Decisions, Web Intelligence, Xcelsius, and other Business Objects products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of Business Objects Software Ltd.

Business Objects is an SAP company.
Sybase and Adaptive Server, iAnywhere, Sybase 365, SQL Anywhere, and other Sybase products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of Sybase, Inc. Sybase is an SAP company.

All other product and service names mentioned are the trademarks of their respective companies. Data contained in this document serves informational purposes only. National product specifications may vary.

The information in this document is proprietary to SAP. No part of this document may be reproduced, copied, or transmitted in any form or for any purpose without the express prior written permission of SAP AG.

This document is a preliminary version and not subject to your license agreement or any other agreement with SAP. This document contains only intended strategies, developments, and functionalities of the SAP® product and is not intended to be binding upon SAP to any particular course of business, product strategy, and/or development. Please note that this document is subject to change and may be changed by SAP at any time without notice.

SAP assumes no responsibility for errors or omissions in this document. SAP does not warrant the accuracy or completeness of the information, text, graphics, links, or other items contained within this material. This document is provided without a warranty of any kind, either express or implied, including but not limited to the implied warranties of merchantability, fitness for a particular purpose, or non-infringement.

SAP shall have no liability for damages of any kind including without limitation direct, special, indirect, or consequential damages that may result from the use of these materials. This limitation shall not apply in cases of intent or gross negligence.

The statutory liability for personal injury and defective products is not affected. SAP has no control over the information that you may access through the use of hot links contained in these materials and does not endorse your use of third-party Web pages nor provide any warranty whatsoever relating to third-party Web pages.