

Program: Time and Expense Management System

Version: 1.6.0 (latest)

Site: <http://www.tems-online.com>

Demo: <http://tems.sourceforge.net>

Date: April 27, 2011

Credit: Stephen Cartwright <sgcartwr@ucalgary.ca>

Description: Unsanitized GET variable *load* resulting in XSS vulnerability.

Impact Analysis

A reflected XSS attack requires an external link such as an e-mail message to inject a crafted payload covertly [1]. XSS attacks may utilize GET variables or POST variables [3]. There are several methods of attack that this kind of vulnerability facilitates. These include:

- Session Riding – session variables and authentication may be leveraged to harvest information from active sessions on the client or perform trusted actions [2].
- Information from cookies can be harvested via *document.cookie* [1].
- A fake form can be inserted to convince a user to input sensitive information. The information can then be sent to an external site and an un-altered request forwarded to make it transparent to the user [1].
- Arbitrary JavaScript can be executed. This allows for any attack that is possible via JavaScript. A keylogger for use in XSS attacks has even been proposed [4].

Resolution:

The developers have been notified. I am waiting for a response in order to coordinate a disclosure plan with them.

References

[1] The Open Web Application Security Project. Cross-Site Scripting (XSS).

https://www.owasp.org/index.php/Cross-site_Scripting_%28XSS%29.

[2] Digger's Blog. How to defeat digg.com ...an introduction to session riding.

<http://4diggers.blogspot.com>.

[3] ha.ckers.org. Exploiting Cross Site Scripting Through Post.

<http://ha.ckers.org/blog/20060814/exploiting-cross-site-scripting-through-post>.

[4] ClsHack.it. Attack with XSS [keylogger,Forms-autocomplete grabber] POST/GET.

<http://www.clsHack.it/en/attack-keylogger-form-autocomplete-xss-post-get-request.html>.