

Blackboard Academic Suite Multiple XSS Vulnerabilities

Source: <http://www.derkeiler.com/Mailing-Lists/securityfocus/bugtraq/2008-03/msg00335.html>

- *From:* knight4vn@xxxxxxxxxx
 - *Date:* 26 Mar 2008 04:13:44 -0000
-

////////////////////////////////////
//Note:
//The full version of this report (in pdf format) available at my blog:
//<http://www.secskill.wordpress.com>
// OR :
//<http://www.scribd.com/doc/2363025/Blackboard-Academic-Suite-Multiple-XSS-Vulnerabilities->
////////////////////////////////////

Blackboard Academic Suites Multiple Cross Site Scripting Vulnerabilities

Background:

Blackboard Academic Suite Blackboard is an enterprise software solution for providing interactive learning and management capabilities for educational institutions. Many institutions currently are using Blackboard such as: Princeton, Yale, Duke University of Pennsylvania, and University of Texas at Austin ?

Platforms Affected:

All versions (7.x and lower)

Description: Combining XSS and some conditions already exists in Blackboard system.

Attacker can login and do everything (change grades, edit online test?s content?) with instructors? identity.

Vulnerable paths:

1/

[http://site.edu/webapps/blackboard/execute/viewCatalog?type=Course&searchText=?><script>alert\(?xss?\)</script>](http://site.edu/webapps/blackboard/execute/viewCatalog?type=Course&searchText=?><script>alert(?xss?)</script>)

2/Add announcement page: (instructor access only)

http://site.edu/bin/common/announcement.pl?action=ADD&course_id=137839_1&render_type=EDITABLE&contex

<input type="text" name="data__announcements__pk1_pk2__subject" value=?><script>alert(?worm activated!?)</script>? />

Author: Duong Thanh – Knight4vn

(knightvn (at) gmail.com or knight4vn (at) yahoo.com)

Vulnerabilities discovered: 12/2007

Blackboard Academic Suite Multiple XSS Vulnerabilities

Vendor and Universities Contacted: 02/2008

Public disclosure: 03/2008

PART I – COMPROMISING USER'S ACCOUNT

Explanation:

When user already has session and he/she clicks on that link (from email), the exploit code will be automatically executed. User's email address is changed without his/her notice. At the same time, his/her current email address, first and last name, and current encrypted password (in User Information page) is logged by a remote server side script.

The attacker reads all these information in a log file.

After that, he gets a new user password sent to his email address by using Lost Password form.

With victim's username and password, the attacker has full permission on that account and does whatever he wants.

Upon finishing his works, he changes back user's initial email address and encrypted password.

Analysis:

Although we can change victim's password by using exploit code but we should not use it. The victim can not login to the system and he/she immediately realize there is something fishy.

Edit Personal Info page:

http://site.edu/webapps/blackboard/execute/editUser?context=self_modify

Blackboard stores encrypted user password in Edit Personal Info page:

```
<INPUT TYPE="hidden" NAME="password" VALUE="CE0BFD15059B68D67688884D7A3D3E8C">
```

Hence, we have no problem with grabbing current user's encrypted password.

On this page:

<http://site.edu/bin/common/user.pl?action=MODIFY&context=PASSWORD>

Blackboard calls a function in `../javascript/md5.js?` to encrypt password on client side before submitting this form to `../webapps/blackboard/execute/editUser?`. So it's possible to submit directly encrypted password to the server side script. Therefore, we take advantage of this to bring victim's encrypted password back to its initial state.

As a result, victim's account was compromised completely without his/her awareness.

Proof-of-concept:

Steal.js

PART II – MAKING A WEB-BASED WORM

Just imagine what would happen if someone took advantage of these holes to create a javascript-based worm? Think about this scenario for a second:

A black-hat guy wrote a worm and he send it to a person (for ex: an instructor).

