

# [UNIX] vBulletin SQL Injection While Verifying Subscription Information

**Source:** <http://www.derkeiler.com/Mailing-Lists/Securiteam/2004-09/0028.html>

---

**From:** SecuriTeam ([support\\_at\\_securiteam.com](mailto:support_at_securiteam.com))

**Date:** 09/14/04

To: [list@securiteam.com](mailto:list@securiteam.com)

Date: 14 Sep 2004 14:06:38 +0200

The following security advisory is sent to the securiteam mailing list, and can be found at the SecuriTeam web site: <http://www.securiteam.com>

-- promotion

The SecuriTeam alerts list – Free, Accurate, Independent.

Get your security news from a reliable source.

<http://www.securiteam.com/maillinglist.html>

-----

vBulletin SQL Injection While Verifying Subscription Information

---

## SUMMARY

" <<http://www.vbulletin.com/>> vBulletin is a powerful, scalable and fully customizable forums package for your web site. It has been written using the Web's quickest-growing scripting language; PHP, and is complimented with a highly efficient and ultra fast back-end database engine built using MySQL."

vBulletin is prone to an SQL injection vulnerability due to improper use of POST variables when attempting to verify a user's subscription.

## DETAILS

Vulnerable Systems:

\* vBulletin version 3.0 up to and including version 3.0.3

In a typical SQL injection scenario, variables from HTTP requests such as GET and POST are directly passed to an SQL query, allowing the malicious attacker to insert additional SQL commands, thereby concatenating additional commands to the original. A code snippet from vBulletin's code clearly shows the problem:

## Securiteam: [UNIX] vBulletin SQL Injection While Verifying Subscription Information

```
error_reporting(E_ALL & ~E_NOTICE);

define('NO_REGISTER_GLOBALS', 1);
define('SESSION_BYPASS', 1);

$phrasegroups = array();
$specialtemplates = array();

chdir('./../');
require('./includes/init.php');
require('./includes/functions.php');
require('./includes/adminfunctions.php');
require('./includes/functions_subscriptions.php');

$check_hash = strtoupper(md5($vboptions['authorize_loginid'] .
$_POST['x_trans_id'] . $_POST['x_amount']));

if ($check_hash == $_POST['x_MD5_Hash'] AND $_POST['x_response_code'] ==
1)
{
$item_number = explode('_', $_POST['x_invoice_num']);
$subscriptionid = intval($item_number[0]);

if (empty($item_number[1]) OR empty($item_number[2]))
{ // non vBulletin subscription
exit;
}

$userid = $DB_site->query_first("SELECT userid, languageid, styleid FROM
". TABLE_PREFIX . "user WHERE userid = " . $item_number[1]);
```

It is easily seen that the \$item\_number[1] variable is take from the POST request and passed directly into the SQL query, thus facilitating SQL injection.

### ADDITIONAL INFORMATION

The information has been provided by <mailto:al3ndaleeb@uk2.net>  
al3ndaleeb.

=====

This bulletin is sent to members of the SecuriTeam mailing list.

To unsubscribe from the list, send mail with an empty subject line and body to:

list-unsubscribe@securiteam.com

In order to subscribe to the mailing list, simply forward this email to: list-subscribe@securiteam.com

=====  
=====

## Securiteam: [UNIX] vBulletin SQL Injection While Verifying Subscription Information

### DISCLAIMER:

The information in this bulletin is provided "AS IS" without warranty of any kind.

In no event shall we be liable for any damages whatsoever including direct, indirect, incidental, consequential, loss of business profits or special damages.