

# [UNIX] BasiliX Multiple Vulnerabilities (File Attachments, Privacy, SQL Injection)

*Source:* <http://www.derkeiler.com/Mailing-Lists/Securiteam/2002-06/0079.html>

---

*From:* [support@securiteam.com](mailto:support@securiteam.com)

*Date:* 06/19/02

From: [support@securiteam.com](mailto:support@securiteam.com)

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

Date: Wed, 19 Jun 2002 08:20:55 +0200 (CEST)

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

When was the last time you checked your server's security?

How about a monthly report?

<http://www.AutomatedScanning.com> -- Know that you're safe.

-----

BasiliX Multiple Vulnerabilities (File Attachments, Privacy, SQL Injection)

---

## SUMMARY

<<http://basilix.org/>> BasiliX is a webmail application based on PHP and IMAP, and powered with the MySQL database server. It supports simple mail actions, sending/receiving attachments, and an address book with group capability, settings utility, multiple languages, multiple folders and themes. Multiple security vulnerabilities have been found in the product. These vulnerabilities allow an attacker to send as attachment any type of content he wishes (even if he hasn't uploaded it), cause cross site scripting issues, view other people's emails, and inject SQL statements.

## DETAILS

Vulnerable systems:

\* BasiliX version 1.1.0 and all previous versions

Any File Attachment:

1) The attachment capability in Compose Mail can be fooled into treating any file on the web server as the uploaded file. This means that it is easy to steal sensitive information on that server (like the /etc/passwd file), and mail it off to someone.

## Securiteam: [UNIX] BasiliX Multiple Vulnerabilities (File Attachments, Privacy, SQL Injection)

When uploading files, PHP sets some global variables, one of which gives the temporary location where the uploaded file was stored. PHP usually also sets global variables with GET or POST form data. BasiliX doesn't check if the attachment really was uploaded by the user, or if it just was some POST data with the same format.

This issue can be fixed by using the `is_uploaded_file()` function, to see if a file was in fact uploaded.

### Cross Site Scripting Issues:

2) The program has got some cross-site scripting issues. In mail folders, in Find Mail and when you read a message, the Subject mail header is shown without removing any HTML tags. When a message is read, the mail body is also shown without removing any HTML tags. This means that an attacker can include JavaScript code in an e-mail message, and that it will be executed in the user's browser when he or she looks at that message.

This can be used for stealing a user's cookies, to allow the attacker to take over the user's session, by including JavaScript code like this:

```
<script>self.location.href="
```

It can also be used as a form of Denial of Service attack. If there is a message in your inbox folder that immediately redirects your browser to Slashdot as soon as you enter that folder, it gets rather hard to read your e-mail.

This can be fixed by always using the `htmlspecialchars()` function when printing variables that shouldn't contain HTML tags.

### Privacy Issue:

3) The attached files are saved in `/tmp/BasiliX`. They are readable by all users, and it seems like they never get deleted. This means that anyone who has got shell access to the server, or who can upload web scripts to it, can read all files any user has ever attached to an e-mail.

### SQL Injection:

4) BasiliX has got some SQL Injection holes. If you have an SQL statement where data from outside is not placed in apostrophes or quotes, like this:

```
DELETE FROM table WHERE id=$id
```

You can wipe all rows in the table by giving \$id the value "id". This will execute the statement:

```
DELETE FROM table WHERE id=id
```

The way to fix this is to put all outside data in apostrophes or quotes, like this:

```
DELETE FROM table WHERE id='$id'
```

Securiteam: [UNIX] BasiliX Multiple Vulnerabilities (File Attachments, Privacy, SQL Injection)

Or to use PHP's `is_numeric()` function.

Vendor status:

The vendor was contacted on the 19th of May. He replied, and we discussed these issues in a couple of mails. We haven't heard from him since the 26th of May. No fixed version has been released yet.

ADDITIONAL INFORMATION

The information has been provided by <mailto:[ulfh@update.uu.se](mailto:ulfh@update.uu.se)> Ulf Harnhammar.

=====

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](mailto:list-unsubscribe@securiteam.com)

In order to subscribe to the mailing list, simply forward this email to: [list-subscribe@securiteam.com](mailto:list-subscribe@securiteam.com)

=====

=====

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.

- 
- **Previous message:** [support@securiteam.com](mailto:support@securiteam.com): "[NEWS] <BODY>Builder SQL modification"
  - **Messages sorted by:** [\[ date \]](#) [\[ thread \]](#) [\[ subject \]](#) [\[ author \]](#) [\[ attachment \]](#)