

# [NT] XTREME ASP Photo Gallery SQL Injection (adminlogin.asp)

**Source:** <http://www.derkeiler.com/Mailing-Lists/Securiteam/2004-01/0059.html>

---

**From:** SecuriTeam (*support\_at\_securiteam.com*)

**Date:** 01/19/04

To: list@securiteam.com

Date: 19 Jan 2004 12:50:47 +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>

-----

XTREME ASP Photo Gallery SQL Injection (adminlogin.asp)

---

## SUMMARY

<<http://www.pensacolawebdesigns.com/>> XTREME ASP Photo Gallery is "a photo gallery that allows easy photo management and complete administration via a web based interface. This interface offers many more features than conventional web based photo gallery's do. With XTREME ASP Photo Gallery, you can configure everything including colors, text styles, amount of imaged displayed per page and much more".

Xtreme ASP Photo Gallery is prone to a common SQL injection vulnerability. The problem occurs when handling user-supplied username and password data supplied to authentication procedures.

## DETAILS

Vulnerable Systems:

\* Xtreme ASP Photo Gallery Version 2.0

The vulnerable page adminlogin.asp is located by default in the following path:

[http://\[host\]/photoalbum/admin/adminlogin.asp](http://[host]/photoalbum/admin/adminlogin.asp)

Securiteam: [NT] XTREME ASP Photo Gallery SQL Injection (adminlogin.asp)

If we type as the username: 'or' and as the password: 'or' we can gain administrative access to the password protected administrative pages.

Vulnerable Code:

The following lines perform the authentication according to the user input:

```
MM_rsUser.Source = "SELECT username, password"
If MM_fldUserAuthorization < > "" Then MM_rsUser.Source = MM_rsUser.Source
& "," & MM_fldUserAuthorization
MM_rsUser.Source = MM_rsUser.Source & " FROM users WHERE username="" &
MM_valUsername &"" AND password="" & CStr(Request.Form("password")) & ""
```

The code does not filter dangerous characters such as '. This allows a malicious user to inject arbitrary SQL commands into the query.

Workaround:

Input from user should never be trusted. Always filter dangerous characters. For additional information on SQL Injections see: <<http://www.securiteam.com/securityreviews/5DPON1P76E.html>> SQL Injection Walkthrough.

ADDITIONAL INFORMATION

The information has been provided by <<mailto:posidron@tripbit.org>> posidron.

The original article can be found at: <<http://www.tripbit.org/advisories/TA-150104.txt>> <http://www.tripbit.org/advisories/TA-150104.txt>.

=====

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

=====

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.