

# [NT] MySQL UDF Multiple Vulnerabilities (Directory Traversal, DoS, Arbitrary Library Including, Buffer Overflow)

*Source:* <http://www.derkeiler.com/Mailing-Lists/Securiteam/2005-08/0035.html>

---

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

*Date:* 08/10/05

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

Date: 10 Aug 2005 15:04:51 +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>

-----  
MySQL UDF Multiple Vulnerabilities (Directory Traversal, DoS, Arbitrary Library Including, Buffer Overflow)

---

## SUMMARY

User-defined functions in MySQL allow a user in the database to call binary libraries on the operating system. Creating a user-defined function requires insert privileges on the mysql.func table. The lack of proper length validation allow attackers to execute arbitrary code using MySQL UDL. Improper directory separator checking, allow attacker to perform directory traversal using MySQL UDL. The lack of proper checking allow attackers to cause a denial of service or load arbitrary library with MySQL UDL.

## DETAILS

Vulnerable Systems:

- \* MySQL version 4.0.25
- \* MySQL version 4.1.13
- \* MySQL version 5.0.7-beta

#### Buffer Overflow:

The `init_syms()` function uses an unsafe string function to copy a user specified string into a stack based buffer. Due to improper sanitation this buffer is able to be overflowed, overwriting portions of the stack. This allows an attacker to write 14 bytes of arbitrary data and 8 bytes of hard coded data <beyond the end of the buffer> beyond the end of the buffer.

The format of the `CREATE FUNCTION` statement is as follows:  
`CREATE FUNCTION function_name RETURNS type SONAME "library_name"`

User specified input to the "function\_name" field is limited to 64 characters. If this library can be successfully loaded by the operating system, control is then passed to `init_syms()`. This will attempt to copy the user string into a buffer 50 bytes in length. Hard coded strings are then copied onto the end of this string. In some older versions of MySQL this can be used to gain complete control over the EIP or copy attacker specified data to an arbitrary location.

One issue of concern is because this buffer is owned by the calling function, in an environment with a stack that grows upwards, it may be possible to overwrite the EIP return or other sensitive values.

Exploiting this vulnerability would require the ability to create user-defined functions. This is not typically granted to untrusted users, however given this vulnerability you should understand the ramifications of granting the ability to create user-defined functions.

#### Arbitrary Library Including:

MySQL attempts to filter execution of arbitrary libraries by requiring any UDF libraries to have either `XXX_deinit()` or `XXX_init()` functions defined. This is intended to prevent an attacker from including any libraries that were not specifically programmed to work with MySQL. Unfortunately this function naming convention is relatively common and default libraries may have these functions defined.

For instance, the "jpeg1x32.dll" and "jpeg2x32.dll" libraries, included by default with Windows 2000 have these functions defined.

This allows an attacker to load the `jpeg_cmp()` function from "jpeg1x32.dll" and the `jpeg_decmp()` function from "jpeg2x32.dll". When either of these functions is called, the MySQL daemon will crash due to improper argument passing.

Both the `jpeg_cmp_init()` and `jpeg_decmp_init()` functions assumes there are 6 arguments waiting for it on the stack. One of these, Arg 6 (EBP+0x1C) is assumed to be a pointer to a memory location. Areas of the memory past this pointer are later overwritten by other arguments passed to this function. Due to the fact that Arg 4 (EBP+0x14) through Arg 6 (EBP+0x1C) are not used prior to this call, it may be possible to pollute the stack and overwrite arbitrary memory locations with attacker supplied values.

Although this is a Windows specific example, it is possible that other operating systems are affected.

Exploiting this vulnerability would require the ability to create user-defined functions. This is not typically granted to untrusted users, however given this vulnerability you should understand the ramifications of granting the ability to create user-defined functions.

#### DoS:

If an attacker asks a Windows based MySQL server to load an invalid library file the application will hang until a dialog box is acknowledged on the server. By requesting one of the many non-library files included in the PATH by default on Windows installations a server will be effectively halted. This is due to the fact that the Windows function LoadLibraryEx() will block when loading an invalid library file with the following message:

"The application or DLL XXXX is not a valid Windows image. Please check this against your installation diskette."

It should be noted that this is a Windows specific issue; other operating systems are not likely to be affected.

An attacker attempting to exploit this issue must have insert privileges on the mysql.func table. This is a high level of privilege that is not normally given to untrusted users.

#### Directory Traversal:

The problem exists in the fact that MySQL only uses the forward slash (/) as a path separator. Windows machines use the backslash (\) character to separate directories in paths. This allows an attacker to bypass directory traversal checks and include arbitrary files.

This may allow an attacker to execute arbitrary code if they are able to drop a file either through FILE privileges, or other attacks on MySQL or other programs.

It should be noted that this is a Windows specific issue; other operating systems are not likely to be affected.

Exploiting this vulnerability would require the ability to create user-defined functions. This is not typically granted to untrusted users, however given this vulnerability you should understand the ramifications of granting the ability to create user-defined functions.

#### Vendor Status:

The vendor has released patches for MySQL versions 4.0.25, 4.1.13 and 5.0.7-beta: <<http://dev.mysql.com/downloads/>>  
<http://dev.mysql.com/downloads/>

#### ADDITIONAL INFORMATION

Securiteam: [NT] MySQL UDF Multiple Vulnerabilities (Directory Traversal, DoS, Arbitrary Library Including, Buffer Over

The information has been provided by <mailto:shatter@appsecinc.com> Team SHATTER.

The original article can be found at:

- <<http://www.appsecinc.com/resources/alerts/mysql/2005-001.html>>
- <http://www.appsecinc.com/resources/alerts/mysql/2005-001.html>,
- <<http://www.appsecinc.com/resources/alerts/mysql/2005-003.html>>
- <http://www.appsecinc.com/resources/alerts/mysql/2005-003.html>,
- <<http://www.appsecinc.com/resources/alerts/mysql/2005-002.html>>
- <http://www.appsecinc.com/resources/alerts/mysql/2005-002.html>

=====

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.