Advanced MySQL Exploitation

Muhaimin Dzulfakar

muhaimindz@gmail.com

Contents

1 Abstract

This document discusses in detail how SQL Injection vulnerabilities can be used to conduct remote code execution on the LAMP (Linux, Apache, MySQL and PHP) [1] or WAMP (Windows, Apache, MySQL and PHP) environments [2]. Attackers performing SQL injection on a MySQL platform must deal with several limitations and constraints. For example, the lack of multiple statements in one query makes MySQL an unpopular platform for remote code execution, compared to other platforms. A lot of research has been performed in the last few years, concentrating on arbitrary code execution by attacking SQL Injection vulnerabilities. However, these types of attacks concentrate on the DBMS connectors that support stacked queries [3]. This document will explain how different methods can be used to achieve the same result on a DBMS connector that does not support stacked queries.

2 Introduction

SQL injection is an attack where malicious code is injected into the SQL statement. SQL Injection attack allows a malicious user to retrieve the structure of the database and uncover information regarding the application's operating environment.

It has been proven that arbitrary code execution is the highest risk level a malicious user can achieve when conducting SQL injection attack.

MySQL is a popular database server and acts as the database component of the LAMP (Linux, Apache, MySQL and PHP) and WAMP (Windows, Apache, MySQL and PHP) software stacks. DBMS connectors for this technology do not support stacked queries by default. This makes the same technique used to conduct remote code execution on platforms that support stacked queries, cannot be applied on this platform.

3 Stacked Query

Stacked query is a term to define if a database connection layer can execute more than one query at a time. Each query is separated by semicolon.

The following example is an example of stacked queries in an SQL Injection attack:

```
SELECT name FROM record WHERE id = 1; DROP table record; DROP table
address--
```

In the example above, there are three separate queries requested at one time. The first query is the actual query from the application. These queries are allowed when called within the database server. However, when this stacked query is called within the MySQL-PHP application by default, an error message will be replayed back by the application and none of the queries will be executed.

4 Attacking MySQL on applications that do support stacked queries

It has been discovered that the MySQL UDF (User Define Function) library [4] creation technique, can be used to conduct remote code execution on applications which support stacked queries. This has been clearly explained by David Litchfield in The Database Hacker's Handbook [5].

At Blackhat Europe 2009, Bernando Damele explained the similar technique in detail. The steps below are taken from Bernando Damele's whitepaper [6].

1) Create a support table with one field, data-type longblob.
2) Encode the local file content to its corresponding hexadecimal string.
3) Split the hexadecimal encoded string into chunks long 1024 characters each
4) INSERT [7] the first chunk into the support table's field.
5) UPDATE [8] the support table's field by appending to the entry the chunks from the second to the last.
6) Export the hexadecimal encoded file content from the support table's entry to the destination file path by using SELECT's INTO DUMPFILE [9] clause. This is possible because on MySQL, a query like SELECT 0x41 returns the corresponding ASCII character A.

Here are some requirements to perform this action:

1) DBMS connector must support stacked queries.
2) Session user is required to have FILE [10], INSERT, CREATE [11] and UPDATE privileges.
3) On some MySQL versions below 5.1.19, shared library [12] path must be writable by a session user. Prior knowledge of this directory is also required.
4) On some MySQL versions 5.1.19 and above, system variable plugin_dir [13] must exist and be writable by a session user. Prior knowledge of this directory is also required.

It has been known that there are some limitations to this technique. By default, MySQL Linux runs as a mysql user and the shared library path is not writable by this user.

However, this behavior is not applied to MySQL on Windows. MySQL on Windows runs as a Local System user and by default, file created by this user can be overwritten onto any folder. For the recent versions of MySQL, the directory belonging to system variable plugin_dir does not exist. This directory is required to be created first and writable by a session user.

5 Attacking MySQL on applications that do not support stacked queries

Due to the unsupported stacked queries on MySQL-PHP platform, execution of another statement after the actual statement is not possible. However, execution of another SELECT statement is possible using UNION syntax [14]. UNION syntax is used to combine the result from multiple SELECT statements into a single result set.

The statement below taken from MySQL 5.1 Refence Manual:

```
SELECT
  [ALL | DISTINCT | DISTINCTROW ]
   [HIGH_PRIORITY]
   [STRAIGHT_JOIN]
   [SQL_SMALL_RESULT] [SQL_BIG_RESULT] [SQL_BUFFER_RESULT]
   [SQL_CACHE | SQL_NO_CACHE] [SQL_CALC_FOUND_ROWS]
  select_expr [, select_expr ...]
  [FROM table_references
  [WHERE where_condition]
  [GROUP BY {col_name | expr | position}
   [ASC | DESC], ... [WITH ROLLUP]]
  [HAVING where_condition]
  [ORDER BY {col_name | expr | position}
   [ASC | DESC], ...]
  [LIMIT {[offset,] row_count | row_count OFFSET offset}]
  [PROCEDURE procedure_name(argument_list)]
  [INTO OUTFILE 'file_name' export_options
    | INTO DUMPFILE 'file_name'
    | INTO var_name [, var_name]]
  [FOR UPDATE | LOCK IN SHARE MODE]]
```

It was discovered that only SELECT INTO DUMPFILE can be used to write arbitrary binary files onto the database server using UNION SELECT. When injecting this statement, only one statement can be used to create the whole file. Also, this file cannot be overwritten by calling another SELECT INTO DUMPFILE statement to write into the same file.

The statement below taken from MySQL 5.1 Reference Manual:

```
 Only the last SELECT statement can use INTO OUTFILE/DUMPFILE. (However,
the entire UNION result is written to the file.)
```

Due to the nature of the UNION statement, results retrieved from the first query will be the first set of data to write into our arbitrary file.

For example:

```
SELECT content FROM data WHERE id=21 UNION SELECT 0x8A789C....... INTO
DUMPFILE ‘file’
```

If the first query returns any data, this data will overwrite the file header. To prevent this, we can inject any non existing value in the WHERE clause so no data would be extracted from the first query. This is to ensure that only our chosen data is written into the file. The file is also required to be located in a directory where we can execute it. In the case of a DBMS connector that does not support stacked queries, the best directory to upload this file onto an Apache web server directory. However, this is only possible in an environment where the MySQL database and Apache web server are on the same machine. By uploading onto the Apache web server directory, a PHP script can be used to execute the file using the SYSTEM function [15]. Here are some requirements to perform this action:

1) Prior knowledge of the Apache web server directory.
2) Session user is required to have FILE privilege.
3) Session user is required to have access to write onto the web server directory.

6 Fingerprinting the web server directory

In some cases where the web server root directory is not located in the default folder during the installation process, fingerprinting can be used to achieve this information. There are two methods that can be used to fingerprint the web server directory. It can be fingerprinted by forcing the application to return an error message or by loading the default location of the Apache configuration file.

6.1 Fingerprint through error message method

By default, custom error message is turned off by PHP. There are many ways to force the application to return an error message that contains internal information including the web server directory. For example, this error message can be generated by injecting a single quote as part of the SQL data. Here is an example of the error message returned by the PHP application when a single quote is injected, as part of the SQL data.

Fatal error: Call to a member function execute() on a non-object in /var/www/output.php on line 15

6.2 Fingerprint through LOAD_FILE method

LOAD_FILE [16] can be used to read a file on the database server. To use this function, FILE privilege is required by the session user. The file must also be readable by all and its size must be less than max_allowed_packet bytes value in the Apache configuration file. A default apache configuration file meets this requirement. The DocumentRoot directive [17] in this file contains the path of the Apache web server root directory and can be used to disclose this information.

Here is an example of the query to load an Apache configuration file on Apache version 2.2, in a default installation of Ubuntu Linux.

```
SELECT LOAD_FILE('/etc/apache2/sites-available/default')
```

7 Maximum size of arbitrary code allowed

The LimitRequestLine directive [17] in the Apache configuration file allows the web server to reduce the size of an HTTP request . This includes all information passed in the query as part of the GET request. The default value for this directive is 8190 bytes. If the SQL Injection is discovered on the GET request and our query including the arbitrary code is larger than this value, Apache web server would response with the HTTP Status Code 414 and the request would not be processed.

By default, the Apache web server sets the LimitRequestBody directive to 2GB [17]. This is the allowed size of an HTTP request message body. If the SQL Injection is discovered in a POST request, 2GB will give us enough room to upload our arbitrary code.

Web application firewalls also have the ability to terminate a long request. This long request is normally detected as a buffer overflow attack.

8 Arbitrary file compression/decompression

PHP contains a module called Zlib [18] that can be used to read and write gzip compressed files. This module can be abused to compress the arbitrary file into a smaller file using the gzcompress function [19]. The Zlib module is able to compress our file from 9635 bytes to only 630 bytes. When the compressed file is successfully uploaded on the web server, gzuncompress function [20] can be used to decompress the file. However, this is only possible in an environment where the MySQL database and Apache web server are on the same machine.

9 Dealing with columns

A UNION clause will combine the result from multiple SELECT statements into a single result set. The two queries must have the same number of columns. Due to this reason, some unnecessary data will be added to our data which could potentially corrupt our file.

For example:

```
SELECT name, add, content FROM data WHERE id=21 UNION SELECT NULL,NULL,
0x8A789C....... INTO DUMPFILE ‘file’
```

The result from the first query will be added into our compressed file. As mentioned in the section 5, we can inject any non existing value into the WHERE clause so there would be no data extracted from the first query. However, due to the extra columns required in this statement, any bad character data added into the first two columns also could potentially corrupt our file. To prevent our file header from being overwritten by any bad character data, we can add our data in the first column and inject any random data into the other columns as seen in the following example.

```
SELECT name, add, content FROM data WHERE id=4444 UNION SELECT
0x8A789C.........,0x00,0x00 INTO DUMPFILE ‘file’
```

The first column in the example above is filled with our compressed arbitrary data. When Zlib library is used, the Adler32 checksum [21] will be found at the end of the compressed data in the first column . As mentioned in the RFC 1950 [22] , any data which may appear after the Adler32 checksum is not part of the Zlib stream. We can abuse this functionality by injecting any value into the other columns. This may help the compressed arbitrary file to be decompressed without having any issues.

Another method is to ensure that all the arbitrary data is filled in sequence into all columns in the second query. It would not be an issue if one of the columns contains more data than the other columns as seen in the following example.

```
SELECT name, add, content FROM data WHERE id=4444 UNION SELECT 0x8A,
0x78,0x9CED......9EEC....... INTO DUMPFILE ‘file’
```

10 Remote code execution on LAMP

Remote code execution on LAMP contains several limitations and constraints. By default, MySQL runs as a mysql user. Arbitrary files created through SELECT INTO DUMPFILE can only be uploaded onto the directory where the mysql user is allowed to write onto. By default, the uploaded file is not executable but readable. This file also is owned by mysql user. A PHP script can be used to read the file and write the same file content to a new file. This new file created is owned by the www-data user and the same PHP script can be used to change the permission of this file to be executable using the PHP SYSTEM function. To perform this action, this file needs to be uploaded onto the web server directory that is writable by mysql and www-data users. By default, any directory created by other users are not writable by these users. In some cases, these directories are set to be writable. An example of this can be found in an application where a user is allowed to upload content onto the web server directories, through the file upload feature.

If the writable directories can be discovered on the web server and the file is successfully uploaded, PHP script can be used to execute the malicious file using the PHP SYSTEM function.

11 Remote code execution on WAMP

Remote code execution on WAMP contains less limitations and constraints in order to function. By default, MySQL runs as a Local System user and files created through SELECT INTO DUMPFILE can be uploaded onto any of the web server directories. By default, this file is executable. Just like on the LAMP platform, PHP script can be used to execute the malicious file using the PHP SYSTEM function.

References

[1] LAMP
http://en.wikipedia.org/wiki/LAMP_(software_bundle)

[2] WAMP
http://en.wikipedia.org/wiki/WAMP

[3] Stacked Queries
http://www.sqlinjectionwiki.com/Default.aspx?Page=Stacked%20Query&AspxAutoDetectCookieSupport=1

[4] MySQL 5.1 Reference Manual: Adding New Functions
http://dev.mysql.com/doc/refman/5.1/en/adding-functions.html

[5] Database Hacker's Handbook
http://www.ngssoftware.com/press-releases/database-hackers-handbook-published/

[6] Advanced SQL Injection Exploitation to Operating System Full Control
http://www.blackhat.com/presentations/bh-europe-09/Guimaraes/Blackhat-europe-09-Damele-SQLInjection-slides.pdf

[7] MySQL 5.1 Reference Manual: INSERT Syntax
http://dev.mysql.com/doc/refman/5.1/en/insert.html

[8] MySQL 5.1 Reference Manual: UPDATE Syntax
http://dev.mysql.com/doc/refman/5.1/en/update.html

[9] MySQL 5.1 Reference Manual: SELECT Syntax
http://dev.mysql.com/doc/refman/5.1/en/select.html

[10] MySQL 5.1 Reference Manual: Privileges Provided by MySQL
http://dev.mysql.com/doc/refman/5.1/en/privileges-provided.html

[11] MySQL 5.1 Reference Manual: CREATE FUNCTION Syntax

http://dev.mysql.com/doc/refman/5.1/en/create-function-udf.html

[12] Shared Library
http://en.wikipedia.org/wiki/Library_(computing)#Shared_libraries

[14] MySQL 5.1 Reference Manual: UNION syntax
http://dev.mysql.com/doc/refman/5.1/en/union.html

[15] PHP System Function
http://ar.php.net/system

[16] MySQL 5.1 Reference Manual: String Functions
http://dev.mysql.com/doc/refman/5.1/en/string-functions.html

[17] Apache Core Features
http://httpd.apache.org/docs/2.1/mod/core.html#documentroot

[18] PHP Zlib Functions
http://www.php.net/manual/en/ref.zlib.php

[19] PHP gzcompress Function
http://www.php.net/manual/en/function.gzcompress.php

[20] PHP gzuncompress Function
http://www.php.net/manual/en/function.gzuncompress.php

[21] Adler32 Checksum
http://en.wikipedia.org/wiki/Adler-32

[22] RFC 1950
http://tools.ietf.org/html/rfc1950