Assignment 3 - SQL Injection

$< Simon > < Rommer >, < 1225253 >$

November 29, 2016

A friend of yours has asked you to perform a security audit on the administration of members in her golf club. She asks you to test if there are any security issues concerning the MySQL database implementation. She gives you a short introduction to the system, but leaves you to find out other information you need for your test:

1 SQL Injection

Assignment

Be aware that SQL injection can be a cumbersome task and it may take a while until you find the right query. Therefore it is advisable to start the assignment early and get back to it after a while when you find yourself stuck. You might want to read a bit into the syntax of SQL and search for information on SQL injection. In this assignment you will exercise what is called a "Blind SQL injection" which means you will not get any error messages from the server if the query you passed is wrong or doesn't yield any results. Note that in this exercise no output can mean you are on a good way.

Level 1

Assignment

Try to login without having any user data.

Description

Username asdf

PIN ' or '1' = '1

I used the escape character $'$ to escape the input query and be able to type in something that was interpreted as actual code.
The most simple SQL-Injection is $'or'1' =' 1$ which means in the context of the login

prompt that the login was successful if the right user was found or 1 = 1 is true [1]. This is always the case so the login was successful. I tried out different combination, but it seemed that the Username field was propperly escaped hence the PIN-field was vulnerable.
After that I was logged in as the first user in the database [2]. After logging in I looked through the "logininfo" and it said that I was logged in as exactly what I typed in and not as a regular user. I really want to know if there was a regular user set up in the first place. After finishing the first assignment i switched to KaliLinux in a VM for convenience reasons.

Level 2

Assignment

Find out which of the members has the highest balance on his/her account. You will not be able to see the balance on the website, you must find it out by passing an appropriate SQL query to the server.

Description

Name Daniel Davis

Balance 295000

For the second part of the assignment I started to dig deeper into tutorials about SQL in general[3] and SQL-Injections in particular[4][5][6][7]. Also watched a some tutorials on how to SQLi with SQLmap [8][9][10][11] . Giving SQLmap just the url as parameter didn't work ,so I had to use a HTTP POST request instead. Weapon of choice was BurpSuite in intercept mode with a configured browser for localhost:8080 which is the default configuration for BurpSuit proxy [12]. With the request from the second and third task recorded and saved to request2.txt and request3.txt I openend 3 terminals and started analysing the database. The SQLmap output[13] was really helpful with showing what the databases and tables looked like

[1] https://www.youtube.com/watch?v=FwIUkAwKzG8
[2] https://www.youtube.com/watch?v=h-9rHTLHJTY
[3] http://www.w3schools.com/sql/
[4] http://www.kalitutorials.net/2014/03/sql-injection-how-it-works.html
[5] http://www.kalitutorials.net/2014/03/hacking-websites-using-sql-injection.html
[6] http://www.kalitutorials.net/2015/02/blind-sql-injection.html
[7] http://www.kalitutorials.net/2014/03/hacking-website-with-sqlmap-in-kali.html
[8] https://www.youtube.com/watch?v=clczL7x1T4Y
[9] https://www.youtube.com/watch?v=jSPr3MPPLLM
[10] https://www.youtube.com/watch?v=yPMbb38pwVI
[11] https://www.youtube.com/watch?v=y4nMgoY5fpY
[12] https://www.youtube.com/watch?v=qsEO4AhlJrc
[13] https://github.com/Acrasy/IntroSec-SQLi/blob/master/sqlmap-files/log

I did so with 3 terminals open seperately. The first for sqlmap[14] with request2.txt the second one for sqlmap with request3.txt and the third one for looking at the dumps and figuring out where I was and what i got so far.
The database dumps are shown in the tables below or can be looked up as the original csv-files at my github[15].

```
1        11  sqlmap -r /request.txt  -D inject_2 --tables
   --threads=5
2        14  sqlmap -r /request.txt  -D inject_2 --threads=5
   -T accounts --dump
```

```
1        8  sqlmap -r /request.txt  -D inject_3  --threads 5
   --tables
2        11  sqlmap -r /request.txt  -D inject_3  --threads 5
   -T vip --dump
3        12  sqlmap -r /request.txt  -D inject_3  --threads 5
   -T regular --dump
```

```
1        49  cd dump/inject_2/
2        52  cat accounts.csv
3        53  cd ..
4        56  cd inject_3/
5        57  cat vip.csv
6        58  cat regular.csv
```

Above I gave an excerpt of the commands that I used. Below I will explain some flags that you can see.

- -r Allowed me to use HTTP POST request instead of the full URL
- -D Name the Database to work with
- –threads 5 To make the process faster
- -T "*name*" Gives the name of the table to work with
- –dump Saves the output to a textfile

[14] https://github.com/sqlmapproject/sqlmap/wiki/Usage
[15] https://github.com/Acrasy/IntroSec-SQLi/tree/master/sqlmap-files/dump

Accounts-DB

ID	Name	Balance	AccountNumber
1	Morgaine DeHavellandt	5000	85632415
2	Hendrik van Haar	260000	23498723
3	Charles Suhr	3000	14269583
4	Daniel Davis	295000	75395146
5	Peter Reed Smith	21	24563985

Level 3

Assignment

There is a members database which consists of two tables `regular` and `vip`. Find out the `memberno` of the member who had the highest balance in step two. The `name` of every member has a suffix `(reg)` or `(VIP)` - this way you will recognize which table you are operating on. Again, you will not be able to see the `memberno` on the website but you must try to find it by using an appropriate SQL query.

Description

MemberNo 13213

Step 3 was just compare the two dump files that i got from the second exercise and write down the membernumber of the person with the highest balance. To no surprise Daniel Davis belonged to the vip-members and not the regulars.

Regulars

ID	Name	MemberNumber
1	Charles Shaughnessy (reg)	11235
2	Nicholle Tom (reg)	12358
3	Benjamin Salisbury (reg)	23581
4	Madeline Zima (reg)	35813
5	Lauren Lane (reg)	58132

VIP's

ID	Name	MemberNumber
1	Fran Drescher (VIP)	81321
2	Daniel Davis (VIP)	13213
3	Ann Guilbert (VIP)	32134

Comparing the VIP-table and the Accounts-DB one can clearly see that Daniel Davis'es membernumber is 13213. Also you see that the names differ greatly, I assume to make the manual approach via changing the URL easier.