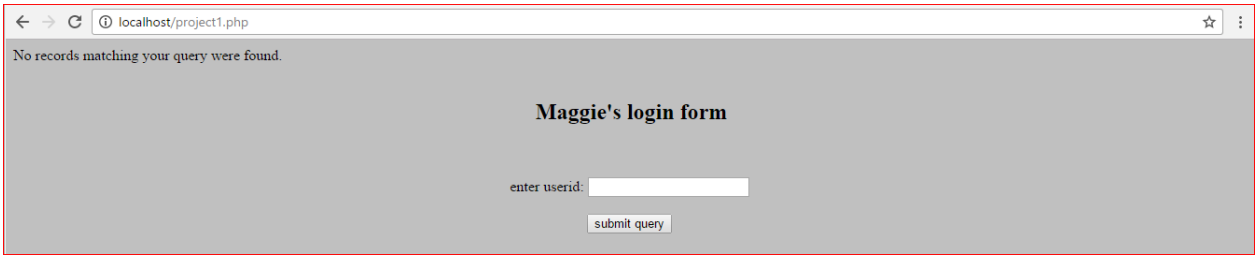
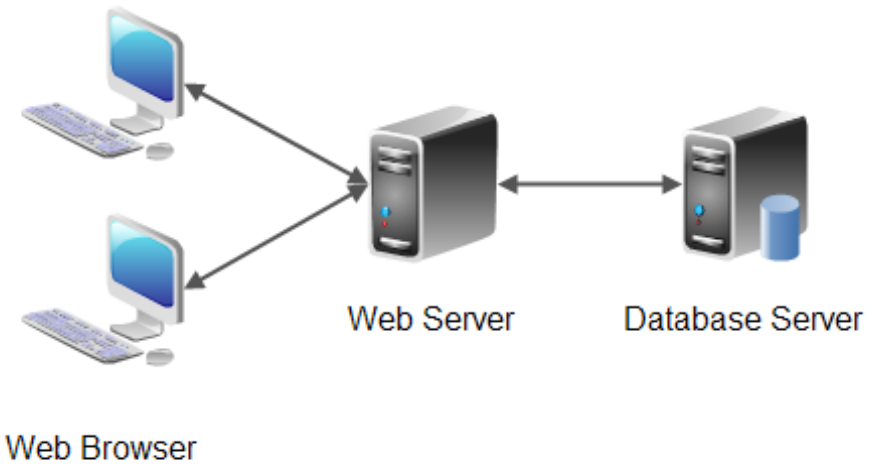
SQL Injection Lab

Overview: In this exercise, you will learn more vulnerability testing: SQL Injection and Sqlninja. Your objective is to employ SQL Injection and Sqlninja to identify the vulnerabilities of a given web application:



This is a simple web application that doesn't contain anything but the usernames and passwords. Please disregard the default error, "No records matching your query were found".

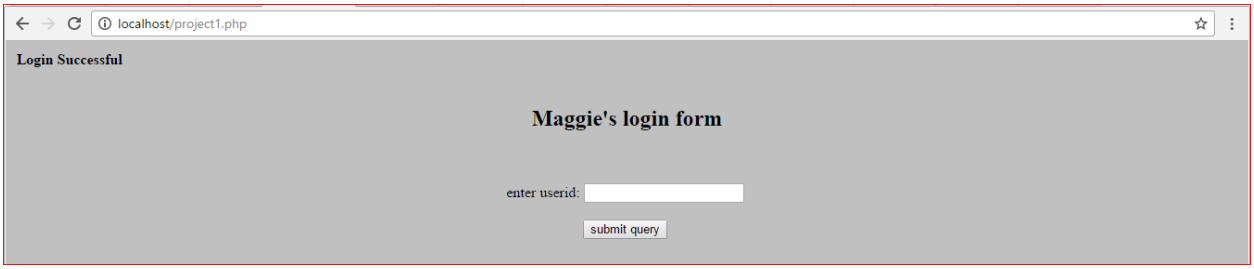
Note: You will need to download a PHP Development environment. Try XAMPP as it has Apache, MariaDB, PHP and Perl.



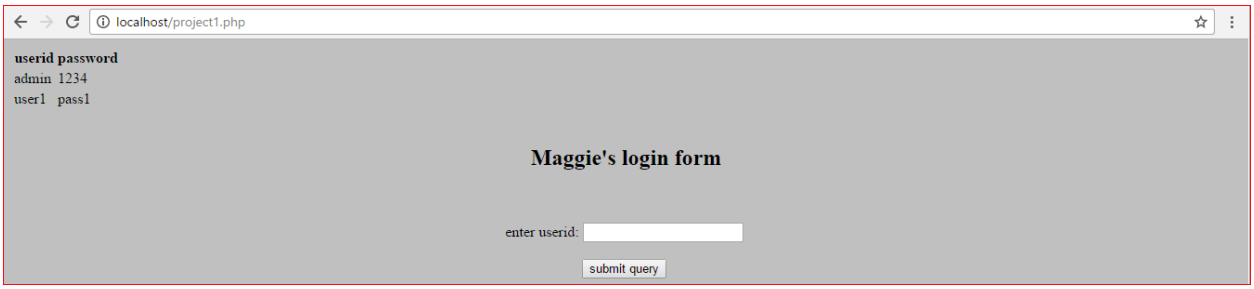
A little bit of background information about the typical modern web application architecture. Typically, a web application communicates with the web server then the web server communicates with the database server and then back from the database service to the web server and finally, to the web browser or web application. It is important to keep in mind in the image above that the database is vital part of a web application as it stores and process the particular application's contents (database).

First, ensure that you have Apache started on XAMPP. Then, open your web browser and type in 127.0.0.1/project1.php or localhost/project1.php. This is the web application that we will use to test vulnerabilities using SQL injection.

Tautology-based SQL Injection



Notice that the given web address prompts you to the login form. It will ask for a user id. If you enter a user id that is present in the database, it will return this user's id and password. Otherwise, it will return a message, "No records matching your query were found."



In this lab, the goal is to penetrate the given web application by figuring out a combination of strings that will allow you to log-in to Maggie's form. This may take a while for new hackers but for veteran hackers, it should be easy. If you are able to have the application return an error other than the default error "No records matching your query were found," use it as a clue to figure out the combination of strings to input into the login page. As you can see on the figure above, the system was penetrated and the userlogin database is displayed revealing the userids and passwords.

Sqlninja is another SQL injection tool that is readily available in Kali Linux. To access it, type the following in the cmd line: ./sqlninja or simply select it in the dropdown menu under applications. Once you run sqlninja, it will prompt you to the various options shown below:

```
Usage: ./sqlninja
        -m <mode> : Required. Available modes are:
            t/test - test whether the injection is working
            f/fingerprint - fingerprint user, xp_cmdshell and more
            b/bruteforce - bruteforce sa account
            e/escalation - add user to sysadmin server role
            x/resurrectxp - try to recreate xp_cmdshell
            u/upload - upload a .scr file
            s/dirshell - starts a direct shell
            k/backscan - look for an open outbound port
            r/revshell - starts a reverse shell
            d/dnstunnel - attempt a dns tunneled shell
            c/sqlcmd - issue a 'blind' OS command
        -f <file> : configuration file (default: sqlninja.conf)
        -p <password> : sa password
        -w <wordlist> : wordlist to use in bruteforce mode (dictionary method
                        only)
        -v : verbose output
```

Use ./sqlninja -m test to check the configuration accuracy. It will prompt for various entries, follow the instructions and enter the necessary information. Once the configuration is completed, it will ask the user to enter exploit string. This is the part where you have to figure out a combination of characters that will allow for command injection.

Note: In order to fully execute SQL injection using SQLninja to its full potentiality, you need to have a Windows Server that uses Microsoft SQL Server for instance Microsoft SQL Server 2000.

Try to test out the other options. For instance, try fingerprinting the remote server (-m fingerprint), bruteforce the 'sa' password (-m bruteforce -w <wordlist to be used>), or how to upload executables and obtain a shell (-m upload).