In this tutorial, I will give you a step-by-step guide on how to set up and install DVWA on your Kali Linux system.

When starting as a penetration tester, you will need a pentesting-lab to test out your penetration skills. One such system is the Damn Vulnerable Web Application (DVWA). DVWA is a vulnerable web application developed using PHP and MySQL that allows ethical hackers to test out their hacking skills and security tools.

It’s also an excellent guide for professional web developers with security in mind. They can use it to learn which features of a web application are easy to exploit. Some of the most common web vulnerabilities demonstrated by this application include Cross-Site Request Forgery (CSRF), File Inclusion, SQL injection, Bruteforce attacks, and much more.

**Pre-requisites to install DVWA**

This tutorial assumes that you already have a Kali Linux Server Up and Running.

**Step 1: Download Damn Vulnerable Web Application (DVWA)**

//ss

//ss

Once in this directory, we will clone the DVWA GitHub repository with the command below.

//ss

//ss

After cloning, we can rename the DVWA folder to dvwa. That is not mandatory, but it makes work easier when executing multiple commands.

//ss

**Step 2: Configure DVWA**

After downloading cloning DVWA in our /var/www/html directory, we still need to do some minor configurations. To get started, let’s set read, write, and execute permissions to the DVWA directory. Execute the command below.

//ss

//ss

After successfully executing the command, we need to set up the user and password required to access the database. Change directory to point to the config directory with the command below.

//ss

When you run the ls command to view the files inside the directory, you will see the config.inc.php.dist file. That is the original file containing the default configurations. We won’t edit it. Instead, we will create a copy of this file called config.inc.php and the original config.inc.php.dist file will act as our backup in case things go wrong.

//ss

//ss

Run the command below to open the newly created file with nano editor and make the necessary changes, as shown in the image below. We will set db\_user as user and db\_password as pass. Feel free to use a different username or password.

//ss

//ss

Save the file (Ctrl + O, then Enter) and Exit (Ctrl + X). That’s it! We are done configuring the DVWA Web application. Let’s move on and configure the database (MySQL).

**Step 3: Install MySQL on Kali Linux**

By default, MySQL comes pre-installed on Kali Linux. If that's not the case for you or maybe you messed up with MySQL, we can go ahead and install it manually. If you have worked with Debian-based distributions, MySQL comes in two packages:

* mysql-server
* mysql-client

In our case, we will need to install the mysql-server. However, there is a catch. If you try using the command apt install mysql-server you will most likely get the error "Package mysql-server is not available, but is referred to by another package. E: Package 'mysql-server' has no installation candidate." That's because the package mysql-server is referred to default-mysql-server in Kali Linux and also in the latest release of Debian (Debian 10). Therefore, use the command below:

//ss

//ss

**Step 4: Configure MySQL Database**

Start the Mysql service with the command below:

//ss

You can check whether the service is running using the systemctl status command below.

//ss

//ss

Login to the MySQL database using the command below as root. If you have another name set for the superuser in your system, use it instead of root.

//ss

Login to the MySQL database using the command below as root. If you have another name set for the superuser in your system, use it instead of root.

//ss

We will create a new user with the username and password set in our DVWA application configuration file. In my case, the username was ‘user,' and the password was ‘pass.’ The server we are using is Localhost (127.0.0.1). Use the command below.

//ss

//ss

We need to grant this new user privilege over the dvwa database. Execute the command below.

//ss

Up to this point, we are through with configuring both the DVWA application and the MySQL database. Type exit to close the database.

//ss

**Step 5: Install PHP**

PHP comes installed in Kali Linux. However, if you want to install a particular version, you can do it manually from the Terminal. In this post, we will install PHP 7.4 which is the latest release as of writing this post. Follow the steps below.

First, update your system and add the SURY PHP PPA repository by executing the commands below.

//ss

//ss

After successfully adding the repository, use the command below to install PHP 7.4

//ss

//ss

To install additional PHP extensions, use the syntax below where xxx stands for the extension name.

//ss

//ss

**Step 6: Configure Apache Server**

Now, we need to configure the server. Use the command below to change your location on the Terminal to point to /etc/php/7.3/apache2 directory.

//ss

In the /etc/php/7.4/apache2, when you execute the ls command, you will see a file called php.ini. That is the file we will edit to configure our localhost server. Use the command below to open it using the nano editor.

//ss

Scroll down and look for these two lines: allow\_url\_fopen and allow\_url\_include. Set them both as On. Save the file (Ctrl + O, then Enter) and Exit (Ctrl + X).

//ss

Start Apache server using the command below:

//ss

To check whether the service started successfully, use the status command.

//ss

//ss

**Step 7: Access DVWA on Your Browser**

That’s it! We now have everything configured, and we can proceed to launch DVWA. Open your browser and enter the URL:

//ss

That will open the setup.php web page as shown in the image:

//ss

You might see the errors colored in red as in the image above. Don’t panic! Scroll down and click the Create / Reset Database button.

//ss

That will create and configure the database. After some time, you will be redirected to the DVWA login page. Log in with these credentials:

* Username - admin
* Password - password

//ss

Once logged in, you will see the DVWA main page. On the left panel, we have the different types of attacks you can exploit and the DVWA Security button that allows you to choose the desired security level - Low, Medium, High, or Impossible.

//ss

**What is SQL injection (SQLi)?**

SQL injection is one of the most common attacks used by hackers to exploit any SQL database-driven web application. It’s a technique where SQL code/statements are inserted in the execution field with an aim of either altering the database contents, dumping useful database contents to the hacker, cause repudiation issues, spoof identity, and much more. Let’s take a simple scenario where we have a web application with a login form with username and password fields. If the developer used PHP for development, the code would look like this:

//ss

If a user Karen with the password ‘12345’ wanted to log in, after clicking the Submit or the Log in button, the query that would be sent to the database would look like this:

//ss

If an attacker knew the username and wanted to bypass the login window, they would put something like Karen;-- in the username field. The resulting SQL query would look like this:

//ss

What the attacker has done, is adding the -- (double-dash) which comments the rest of the SQL statement. The above query will return the information entered in the password field making it easier for the attacker to bypass the login screen.

**How to prevent SQL injection**

The main reason that makes websites vulnerable to SQL injection attacks can be traced back to the web development stage. Some of the techniques that can be implemented to prevent SQL injection include: Input validation: If the website allows user input, this input should be verified whether it’s allowed or not.

* Parametrized queries: This is a technique where the SQL statements are precompiled and all you have to do is supply the parameters for the SQL statement to be executed.
* Use Stored procedures
* Use character-escaping functions
* Avoid administrative privileges - Don't connect your application to the database using an account with root access
* Implement a Web application firewall (WAF)

Any penetration tester who wants to get started or advance their skills in SQL injection will need a vulnerable platform to practice. There are many vulnerable applications available both for offline and online use.

In this particular tutorial, we will focus on the Damn Vulnerable Web Application (DVWA).

**Pre-requisites**

This tutorial expects that you have an up and running DVWA setup. If you have not yet installed DVWA on your Kali Linux system, please check out the article which gives a step-by-step guide.

**Step 1: Setup DVWA for SQL Injection**

After successfully installing DVWA, open your browser and enter the required URL 127.0.0.1/dvwa/login.php Log in using the username “admin” and password as “password”. These are the default DVWA login credentials. After a successful login, set the DVWA security to LOW then click on SQL Injection on the left-side menu.

//ss

**Step 2: Basic Injection**

On the User ID field, enter “1” and click Submit. That is supposed to print the ID, First\_name, and Surname on the screen as you can see below. The SQL syntax being exploited here is:

//ss

Interestingly, when you check the URL, you will see there is an injectable parameter which is the ID. Currently, my URL looks like this:

//ss

Let’s change the ID parameter of the URL to a number like 1,2,3,4 etc. That will also return the First\_name and Surname of all users as follows:

//ss

If you were executing this command directly on the DVWA database, the query for User ID 3 would look like this:

//ss

//ss

**Step 3: Always True Scenario**

An advanced method to extract all the First\_names and Surnames from the database would be to use the input: %' or '1'='1'

//ss

The percentage % sign does not equal anything and will be false. The '1'='1' query is registered as True since 1 will always equal 1. If you were executing that on a database, the query would look like this:

//ss

//ss

**Step 4: Display Database Version**

To know the database version the DVWA application is running on, enter the text below in the User ID field.

//ss

The database version will be listed under surname in the last line as shown in the image below.

//ss

**Step 5: Display Database User**

To display the Database user who executed the PHP code powering the database, enter the text below in the USER ID field.

//ss

The Database user is listed next to the surname field in the last line as in the image below.

//ss

**Step 6: Display Database Name**

To display the database name, we will inject the SQL code below in the User ID field.

//ss

The database name is listed next to the surname field in the last line.

//ss

**Step 7: Display all tables in information\_schema**

The Information Schema stores information about tables, columns, and all the other databases maintained by MySQL. To display all the tables present in the information\_schema, use the text below.

//ss

//ss

**Step 8: Display all the user tables in information\_schema**

For this step, we will print all the tables that start with the prefix user as stored in the information\_schema. Enter the SQL code below in the User ID.

//ss

//ss

**Step 9: Display all the columns fields in the information\_schema user table**

We will print all the columns present in the users’ table. This information will include column information like User\_ID, first\_name, last\_name, user, and password. Enter the input in the User\_ID field.

//ss

//ss

**Step 10: Display Column field contents**

To display all the necessary authentication information present in the columns as stored in the information\_schema, use the SQL syntax below:

//ss

//ss

From the image above, you can see the password was returned in its hashed format. To extract the password, copy the MD5 hash and use applications like John the Ripper to crack it. There are also sites available on the internet where you can paste the hash and if lucky, you will be able to extract the password.

**Conclusion**

From the various examples listed in this article, SQL injection proves to be a critical vulnerability that can exist in a system. Not only can attackers exploit it to reveal user or customer information, but it can also be used to corrupt the entire database thus bringing the whole system down. As of writing this post (2021), Injection is listed as the number one vulnerability in the OWASP Top 10 Vulnerabilities summary. The DVWA acts as a reliable resource for both penetration testers who want to improve their skills and web developers who want to develop systems with security in mind.