Database Security and Blockchain

1 Overview

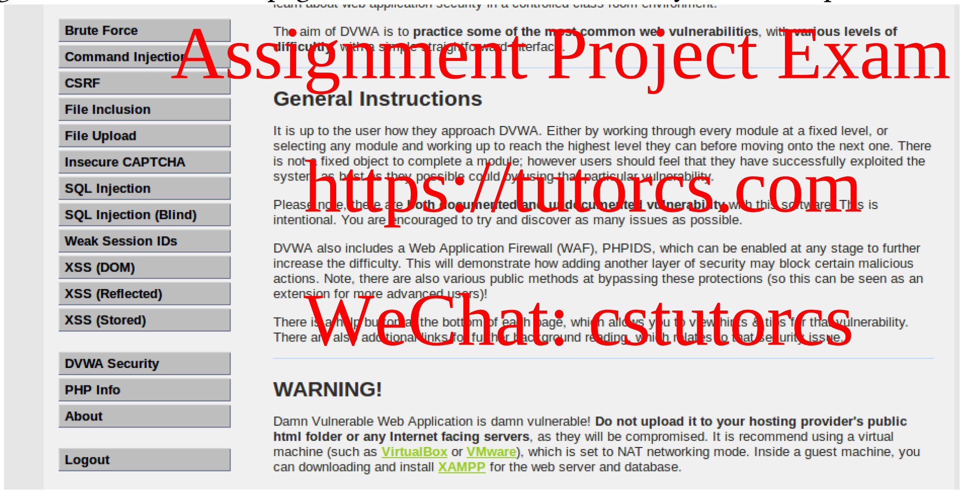
The learning objective of this lab is for students to get familiar with blockchain, and vulnerabilities of and attacks targeting Web applications and back-end database.

2 Lab Environment

In this lab we will exploit SQL injection vulnerabilities on an intentionally vulnerable web server using the cloud VM.

3 Lab Task: SQL Injection Attack

Open the Firefox web browser in the VM and type `http://127.0.0.1/DVWA` in the address bar. Please note that the url is case sensitive. This should open the login page. Use `admin` as username and `password` as password. Once logged in scroll down the page and find the **DVWA Security** on the left pane.



The aim of DVWA is to **practice some of the most common web vulnerabilities**, with **various levels of difficulty**, with a simple straightforward interface.

General Instructions

It is up to the user how they approach DVWA. Either by working through every module at a fixed level, or selecting any module and working up to reach the highest level they can before moving onto the next one. There is not a fixed object to complete a module; however users should feel that they have successfully exploited the system as best as they possible could by using that particular vulnerability.

Please note, there are **both documented and undocumented vulnerability** with this software. This is intentional. You are encouraged to try and discover as many issues as possible.

DVWA also includes a Web Application Firewall (WAF), PHPIDS, which can be enabled at any stage to further increase the difficulty. This will demonstrate how adding another layer of security may block certain malicious actions. Note, there are also various public methods at bypassing these protections (so this can be seen as an extension for more advanced users)!

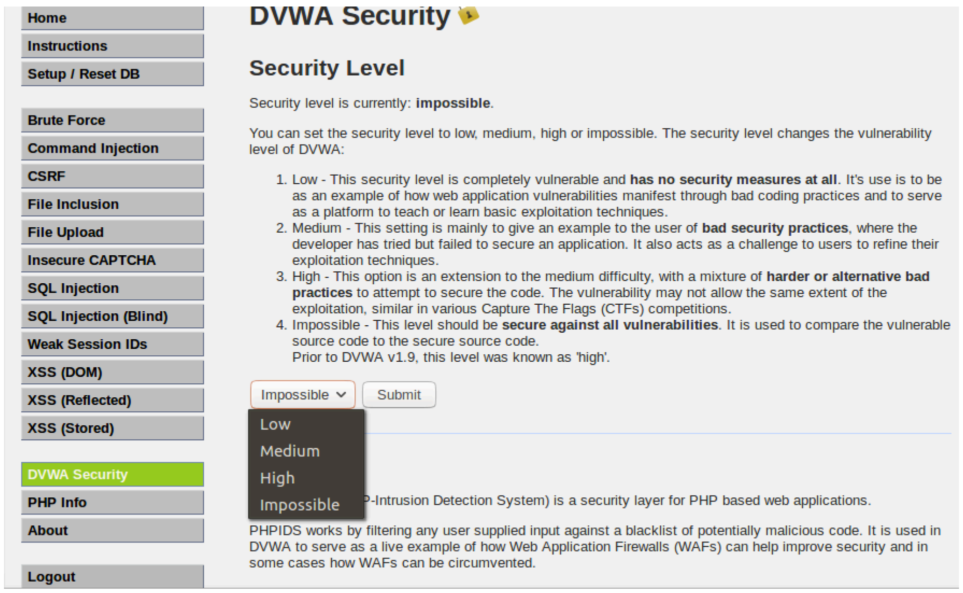
There is a help button at the bottom of each page, which allows you to view hints & tips for that vulnerability. There are also additional links for further background reading, which relates to that security issue.

WARNING!

Damn Vulnerable Web Application is damn vulnerable! **Do not upload it to your hosting provider's public html folder or any Internet facing servers**, as they will be compromised. It is recommend using a virtual machine (such as VirtualBox or VMware), which is set to NAT networking mode. Inside a guest machine, you can downloading and install XAMPP for the web server and database.

Assignment Project Exam Help
https://tutorcs.com
WeChat: cstutorcs

Click on the **DVWA Security** and within the opened page change the security level to low and then click **Submit**.



DVWA Security

Security Level

Security level is currently: **impossible**.

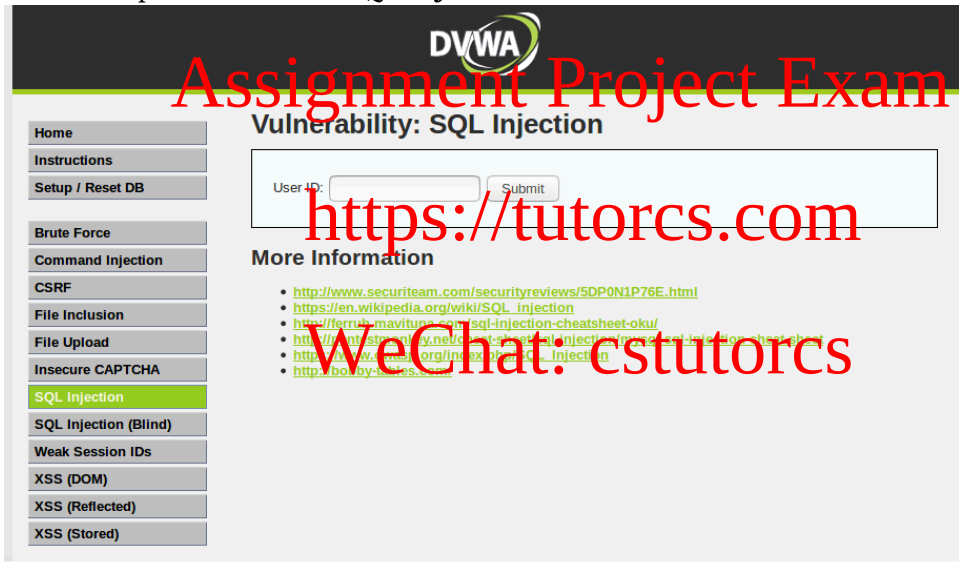
You can set the security level to low, medium, high or impossible. The security level changes the vulnerability level of DVWA:

1. Low - This security level is completely vulnerable and **has no security measures at all**. It's use is to be as an example of how web application vulnerabilities manifest through bad coding practices and to serve as a platform to teach or learn basic exploitation techniques.
2. Medium - This setting is mainly to give an example to the user of **bad security practices**, where the developer has tried but failed to secure an application. It also acts as a challenge to users to refine their exploitation techniques.
3. High - This option is an extension to the medium difficulty, with a mixture of **harder or alternative bad practices** to attempt to secure the code. The vulnerability may not allow the same extent of the exploitation, similar in various Capture The Flags (CTFs) competitions.
4. Impossible - This level should be **secure against all vulnerabilities**. It is used to compare the vulnerable source code to the secure source code.
   Prior to DVWA v1.9, this level was known as 'high'.

P-Intrusion Detection System) is a security layer for PHP based web applications.

PHPIDS works by filtering any user supplied input against a blacklist of potentially malicious code. It is used in DVWA to serve as a live example of how Web Application Firewalls (WAFs) can help improve security and in some cases how WAFs can be circumvented.

On the left pane click on the **SQL Injection**.



Vulnerability: SQL Injection

More Information

- http://www.securiteam.com/securityreviews/5DP0N1P76E.html
- https://en.wikipedia.org/wiki/SQL_injection
- http://ferruh.mavituna.com/sql-injection-cheatsheet-oku/
- http://pentestmonkey.net/cheat-sheet/sql-injection/mysql-sql-injection-cheat-sheet
- https://www.owasp.org/index.php/SQL_Injection
- http://bobby-tables.com/

Assignment Project Exam Help
https://tutorcs.com
WeChat: cstutorcs

1. Use the provided links to learn more about this type of attack.
2. Enter a single quote in the provided text box and submit. What is the effect? What do we learn from the output?
3. Enter "1" (without the quotes) in the provided text box and submit. The code within the page is supposed to output the first and last name of a user based on the provided user ID. Who has the user ID=1?
4. We know that the text entered in this field is not properly sanitised. We also know that a SQL query conditioned on the provided ID will be sent to the SQL server. We need to modify this query so that the condition will always be true. Enter `%' or '0'='0` in the field and submit. What is the result? Explain why the provided input has this effect.
5. Additional SQL commands can be added to this query to gain more information. Enter `%' or 0=0 union select null, version() #` in the text box and submit. Explain the result. What is the purpose of the added SQL statement?

6. Enter `%' or 0=0 union select null, user()` # and submit. Explain the result.
7. In the above statement replace `user()` with `database()`. Explain the output.
8. Enter `%' and 1=0 union select null, table_name from information_schema.tables` # and submit. Explain the statement and the result.
9. Enter `%' and 1=0 union select null, table_name from information_schema.tables where table_name like 'user%'`# and submit. Explain the statement and the result.
10. Enter `%' and 1=0 union select null, concat(table_name,0x0a,column_name) from information_schema.columns where table_name = 'users'` # and submit. Explain the statement and the result.
11. Enter `%' and 1=0 union select null, concat(first_name,0x0a, last_name,0x0a,user,0x0a,password) from users` # and submit. Explain the statement and the result.

4 Optional Task: Blockchain

Use the Bitcoin Explorer `https://www.blockchain.com/explorer`

1. What information can you get from the block information page?
2. Click on a specific address under the transaction list. What are the transactions related to that address?

Assignment Project Exam Help
https://tutorcs.com
WeChat: cstutorcs