EXPLOIT WEB APPLICATION SECURITY USING DVWA

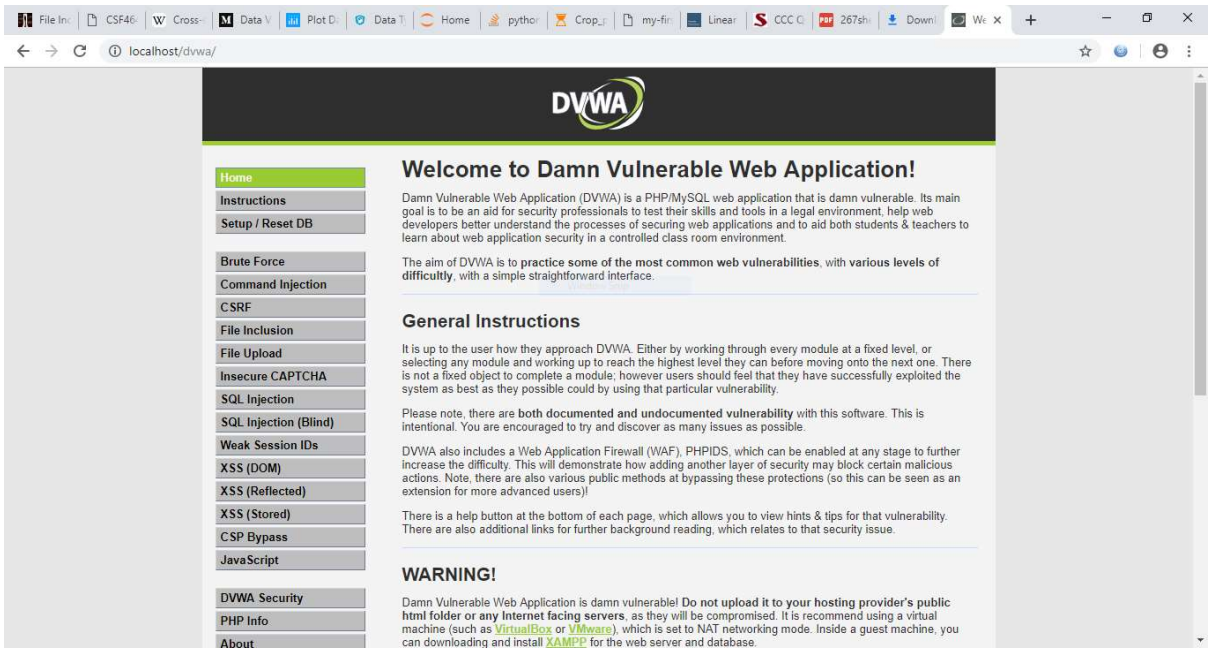
WHAT IS DVWA?

- Damn Vulnerable Web App (DVWA) is a PHP/MySQL web application.
- Its main goals are to be an aid for security professionals to test their skills and tools in a legal environment,
- help web developers better understand the processes of securing web applications and aid teachers/students to teach/learn web application security in a class room environment.

HOW TO INSTALL DVWA?

- Go to official site: http://www.dvwa.co.uk/
- Click on download.
- Put downloaded file in localhost folder (i.e. xampp\htdocs)
- Extract the file
- Start the server
- Run the local site in browser.
- Login with below credentials
- Username: ‘admin’ password: ‘password’

SCREENSHOT:



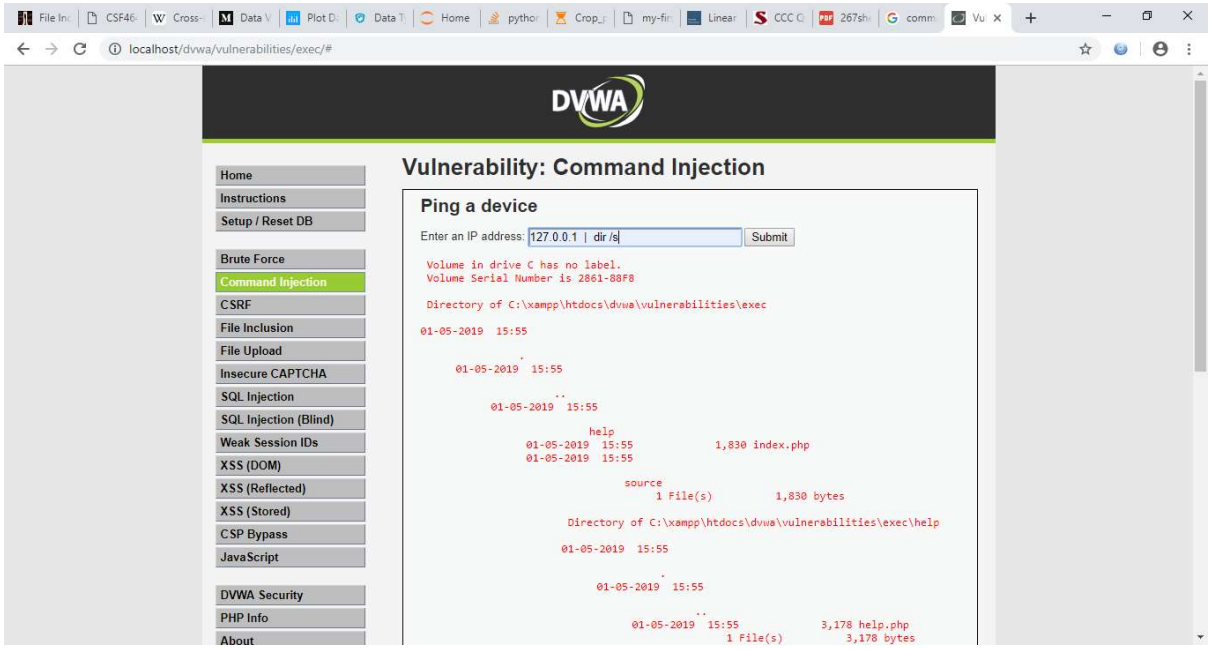
COMMAND EXECUTION/INJECTION

- **DVWA** - **Command Injection**. **Command injection** is an attack in which the goal is **execution** of arbitrary **commands** on the host operating system via a vulnerable application.
- **Command injection** attacks are possible when an application passes unsafe user supplied data (forms, cookies, HTTP headers etc.) to a system shell.

PURPOSE:

- The purpose of the command injection attack is to inject and execute commands specified by the attacker in the vulnerable application.
- In situations like this, the application, which executes unwanted system commands, is like a pseudo system shell, and the attacker may use it as an authorized system user.
- Note, the commands are executed with the same privileges as the application and/or web server.
- Command injection attacks are possible in most cases because of lack of correct input data validation, which can be manipulated by the attacker (forms, cookies, HTTP headers etc.

SCREENSHOT:



SQL INJECTION

- SQL injection (also known as SQL fishing) is a technique often used to attack data driven applications.
- This is done by including portions of SQL statements in an entry field in an attempt to get the website to pass a newly formed rogue SQL command to the database (e.g., dump the database contents to the attacker).
- SQL injection is a code injection technique that exploits a security vulnerability in an application's software.
- The vulnerability happens when user input is either incorrectly filtered for string literal escape characters embedded in SQL statements or user input is not strongly typed and unexpectedly executed.
- SQL injection is mostly known as an attack vector for websites but can be used to attack any type of SQL database.

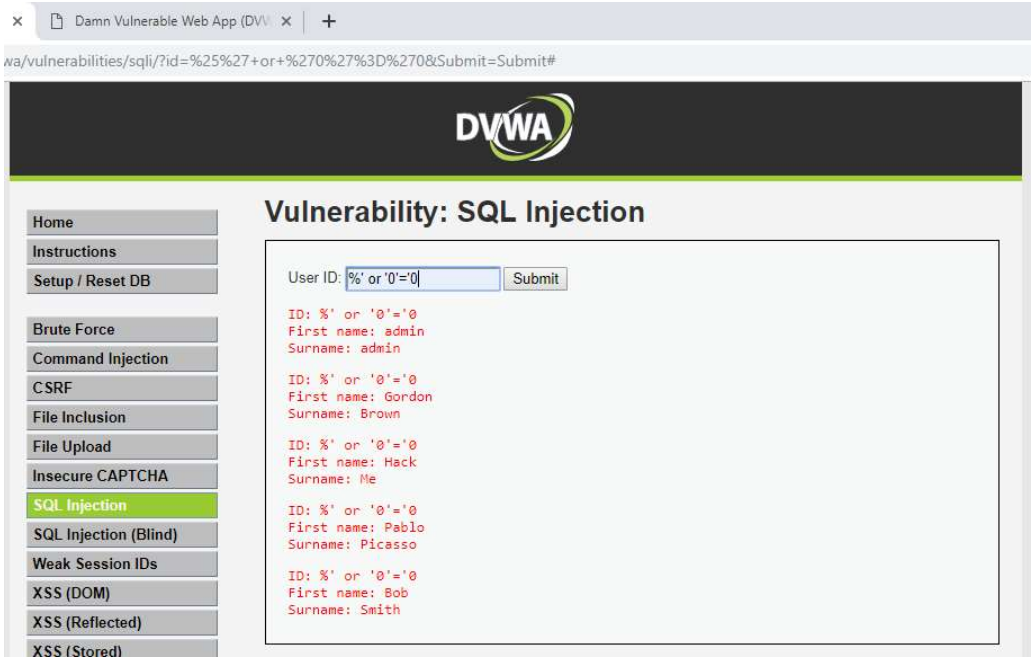
INSTRUCTIONS:

1. Input the below text into the User ID Textbox (See Picture).
2. %' or '0'='0
3. Click Submit

NOTES:

- In this scenario, we are saying display all record that are **false** and all records that are **true**.
  - %' - Will probably not be equal to anything, and will be false.
  - '0'='0' - Is equal to true, because 0 will always equal 0.
- **Database Statement**
  - mysql> SELECT first_name, last_name FROM users WHERE user_id = '**%' or '0'='0**';
- **Know the Database name**
  - %' or 0=0 union select null, database()
- **Display all tables in information_schema**
  - %' and 1=0 union select null, table_name from information_schema.tables #
- **Display all the user tables in information_schema**
  - %' and 1=0 union select null, table_name from information_schema.tables where table_name like 'user%'#
- **For more information visit:** http://www.computersecuritystudent.com/SECURITY_TOOLS/DVWA/DVWAv107/lesson6/

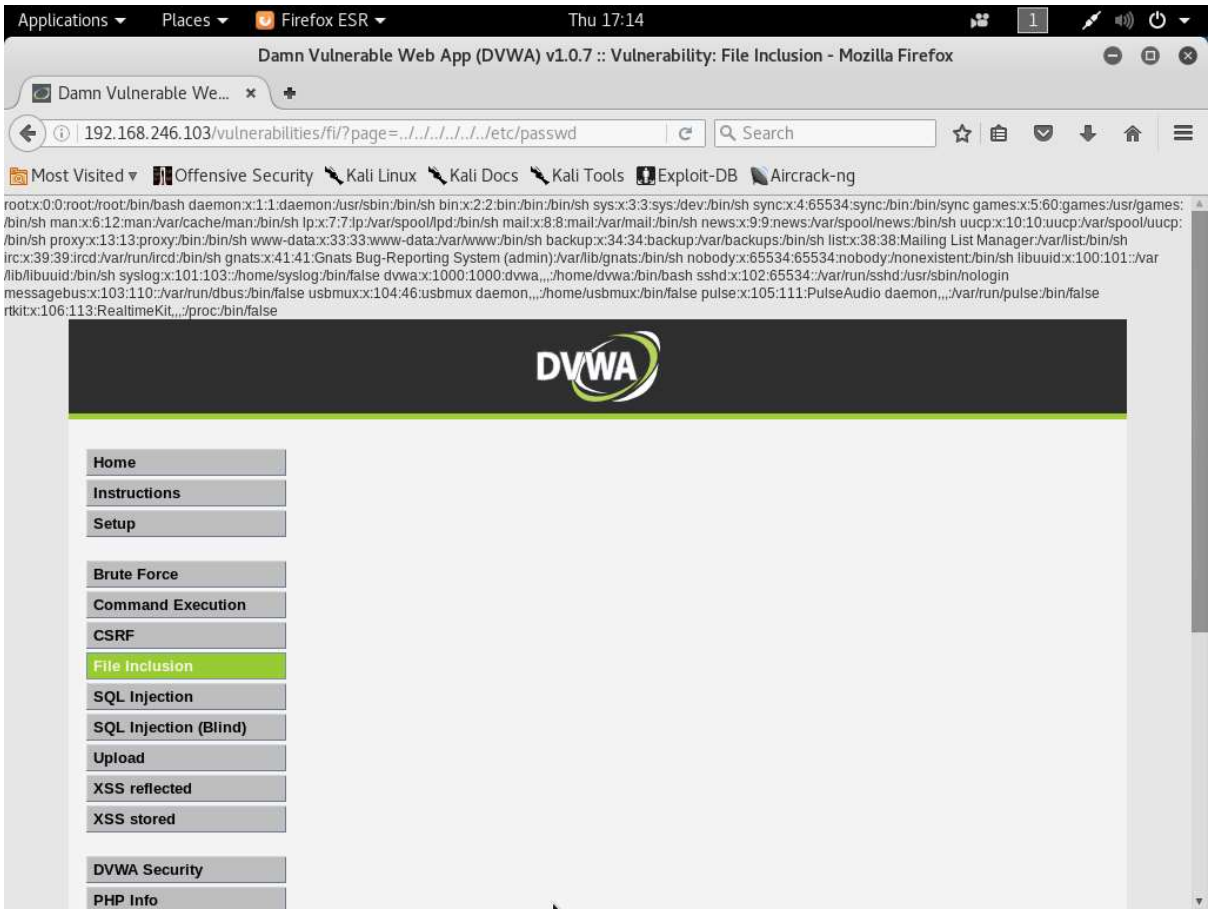
SCREENSHOT:



FILE INCLUSION

- Allows an 'attacker' to include remote/local files into the web application.
- Remote File Inclusion (RFI) and Local File Inclusion (LFI) are vulnerabilities that are often found in poorly-written web applications.
- These vulnerabilities occur when a web application allows the user to submit input into files or upload files to the server.
- LFI vulnerabilities allow an attacker to read (and sometimes execute) files on the victim machine.
- This can be very dangerous because if the web server is misconfigured and running with high privileges, the attacker may gain access to sensitive information.
- If the attacker is able to place code on the web server through other means, then they may be able to execute arbitrary commands.
- RFI vulnerabilities are easier to exploit but less common. Instead of accessing a file on the local machine, the attacker is able to execute code hosted on their own machine.

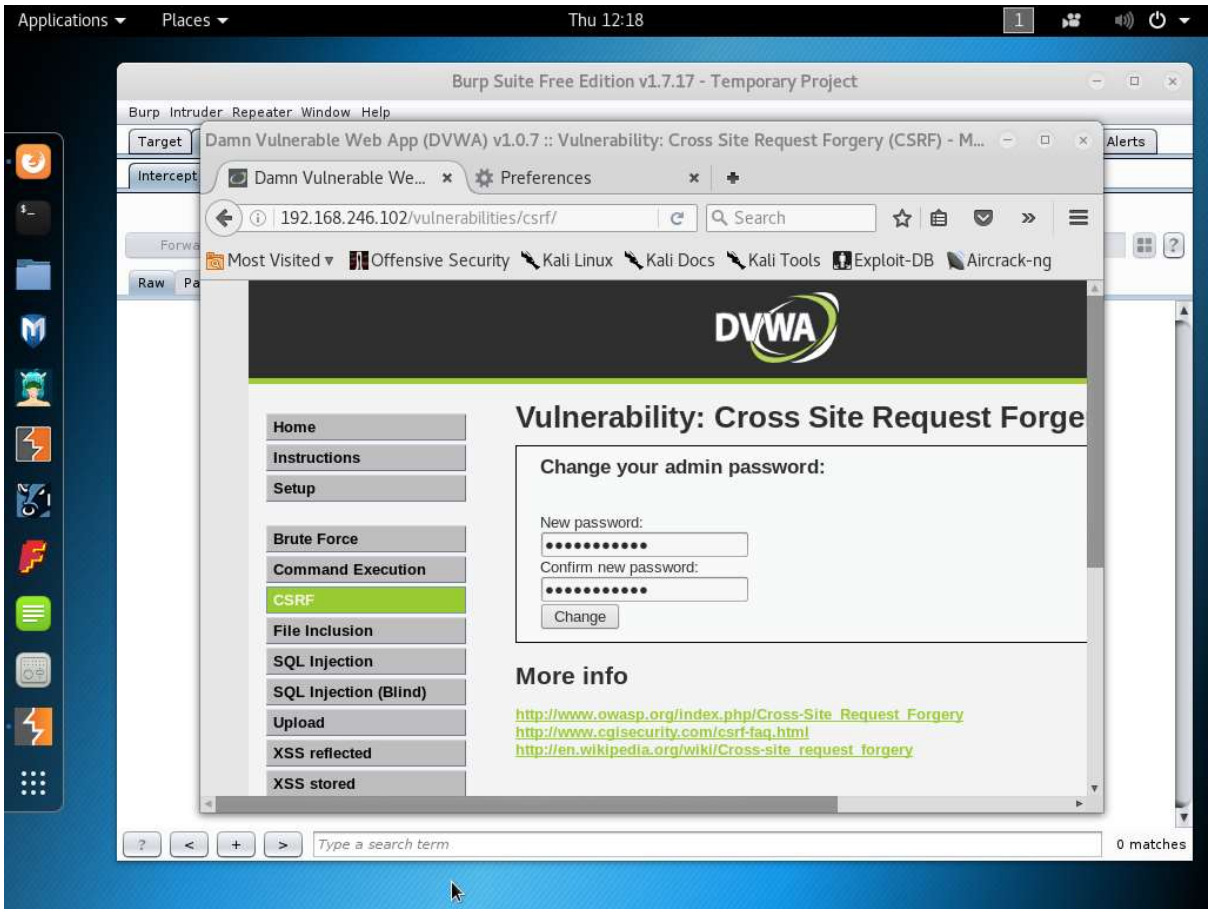
SCREENSHOT:



CSRF/XXS

- **Cross-site request forgery**, also known as **one-click attack** or **session riding** and abbreviated as **CSRF** (sometimes pronounced *sea-surf*) or **XSRF**,
- is a type of malicious exploit of a website where unauthorized commands are transmitted from a user that the web application trusts.
- There are many ways in which a malicious website can transmit such commands; specially-crafted image tags, hidden forms, and JavaScript XMLHttpRequests,
- for example, can all work without the user's interaction or even knowledge. Unlike cross-site scripting (XSS), which exploits the trust a user has for a particular site, CSRF exploits the trust that a site has in a user's browser.

- Firstly, User will try to Enter New password and Hits on Change Button.

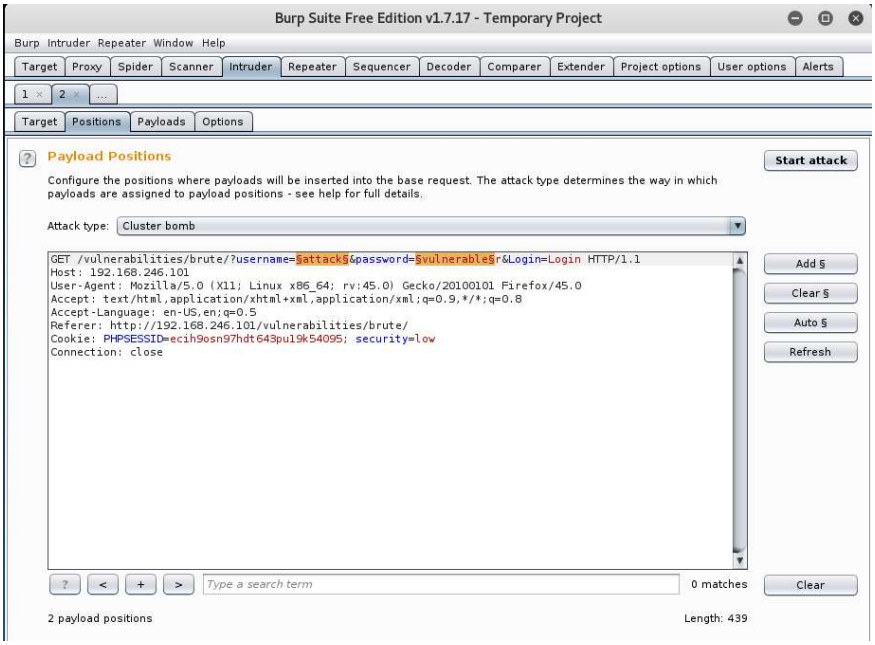


- Secondly, Burp suite will Capture the Request and Alters it as shown in image below : User enters Password as validuser:validuser but we edited as hacker:hacker.

- **Query** Submitted by Middle-Man will executed Successfully.
- when the Authenticated User tries to Change Password again it throws Error message because data stored in database will be the password entered by Middle-man not by the Valid User.
- Acknowledgement show to User as "Password Changed", shown below

BRUTE FORCE

- A **brute force attack** is a term in cryptanalysis. It means trying to break a coded cyphertext by trying a lot of possibilities with fast computers. For example, a large number of the possible keys are tried in the key space. If successful , this decrypts the encrypted message.
- The theoretical possibility of a brute force attack is recognized by the cryptographic system designers. They work to make the cryptographic system very difficult for computers to break using brute force attack. For that reason, one of the definitions of "breaking" a cryptographic scheme is to find a method faster than a brute force attack.
- The selection of an appropriate key length depends on how difficult it will be to break it using a brute force attack. By obfuscating the data before encryption, brute force attacks are less effective and more difficult to determine.
- In this challenge we have to find out whether it has been vulnerable to Brute Force Attack and IF yes, then Find out the Username and Password.
- We required a Tool here to Capture the Request-Response transactions of Web Application, We used Burpsuite Here to exploit this Vulnerability.
- Open Burpsuite and Set Proxy to **127.0.0.1:8080**
- Open Browser Setting and set proxy to **localhost:8080**
- Enter any credentials and capture those Packets in Burpsuite under Proxy tab, just forward it to **Intruder** (note down error reply which has to be entered in **Grep-Match field**)
- Analyze the Request and find Username and Password text filed enclosed with **$___$**
- Mark username and Password Field with **$** symbol.



- Select Attack Type = **Cluster Bomb**, Upload Payload 1 and 2
- Under Option tab Set Grep-Match as **Incorrect**
- Hit Button => Start Attack
- After Matching the Every Payload Entries, Burpsuite gives Username & password (Observes **untick** on Grep Field)
- Obtained Username and Password is "**admin**"& "**password**"