Finding Name: Information disclosure vulnerability

Name	Team	Role	Project	Quality Assurance	Is this a re-tested Finding?
ABDULMAJEED HUSSAIN A ALZAHRANI	PT	Senior	Company Website	Mankaran Singh Saggu	

Was this Finding Successful?
Yes

Finding Description

An Information Disclosure vulnerability was identified in the application that exposes sensitive information to unauthorized users. This issue can potentially lead to exploitation by malicious actors, compromising the confidentiality of critical data. This information includes parts of the source code, PATH, server software, server version…. etc.

Risk Rating

Impact: Major

Likelihood: High

Impact values				
Very Minor	**Minor**	**Significant**	**Major**	**Severe**
Risk that holds little to no impact. Will not cause damage and regular activity can continue.	Risk that holds minor form of impact, but not significant enough to be of threat. Can cause some damage but not enough to impede regular activity.	Risk that holds enough impact to be somewhat of a threat. Will cause damage that can impede regular activity but will be able to run normally.	Risk that holds major impact to be of threat. Will cause damage that will impede regular activity and will not be able to run normally.	Risk that holds severe impact and is a threat. Will cause critical damage that can cease activity to be run.

Likelihood				
Rare	**Unlikely**	**Moderate**	**High**	**Certain**
Event may occur and/or if it did, it happens in specific circumstances.	Event could occur occasionally and/or could happen (at some point)	Event may occur and/or happens.	Event occurs at times and/or probably happens a lot.	Event is occurring now and/or happens frequently.

Business Impact

Unauthorized Access:

- Sensitive information such as API keys, credentials, or configuration details could enable attackers to gain unauthorized access to systems or data.

Privilege Escalation:

- Exposed internal details may allow attackers to escalate their privileges and gain deeper control over the application or infrastructure.

Increased Attack Surface:

- Information disclosure provides insights into system architecture, making it easier for attackers to exploit other vulnerabilities

Affected Assets

Application Components:

- Web Applications: Exposed sensitive data could compromise frontend and backend services.

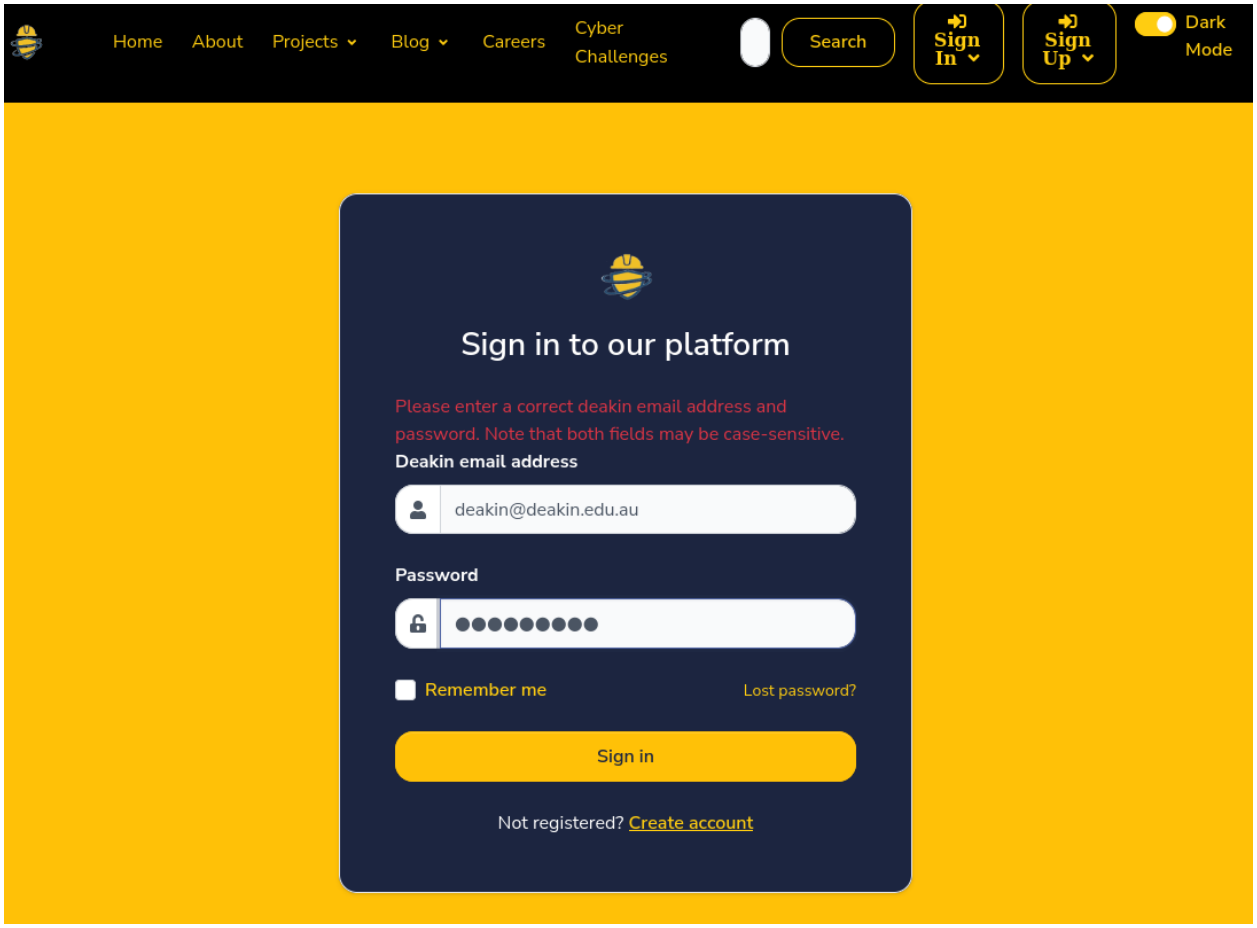
System Infrastructure:

- Servers: Exposure of server software, software version, or source code facilitate server attacks

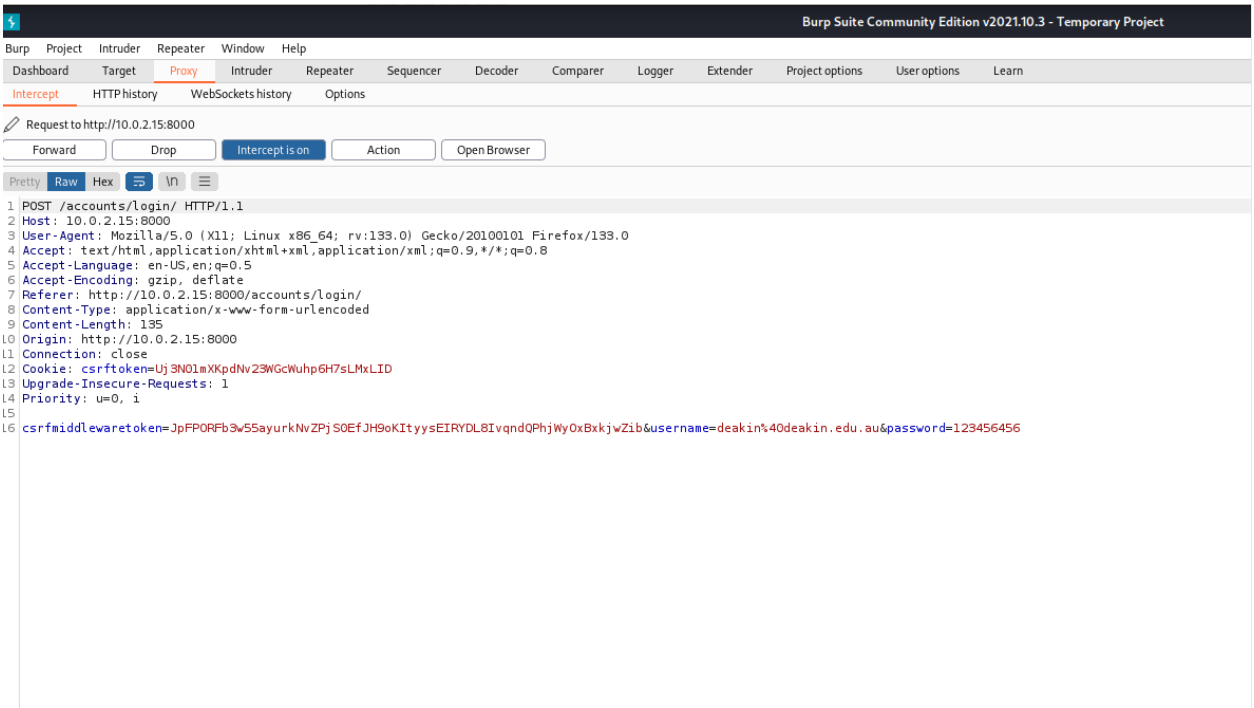
Evidence

Provide a step-by-step guide on how to reproduce vulnerability with screenshots

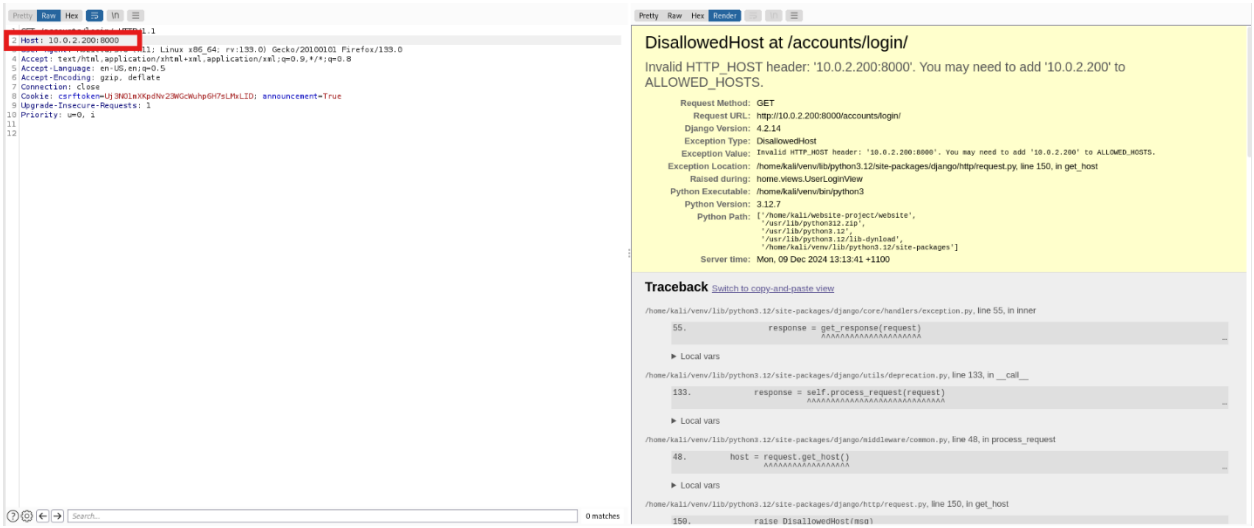
Step 1. Visit the login page and enter random data



Step 2. Intercept the request using burp suite

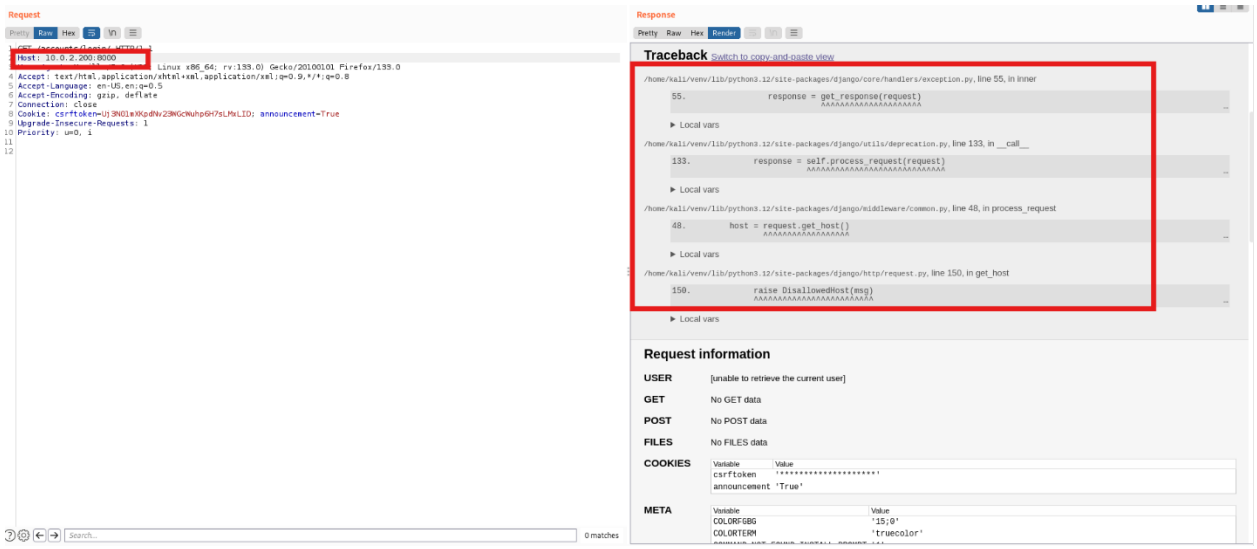


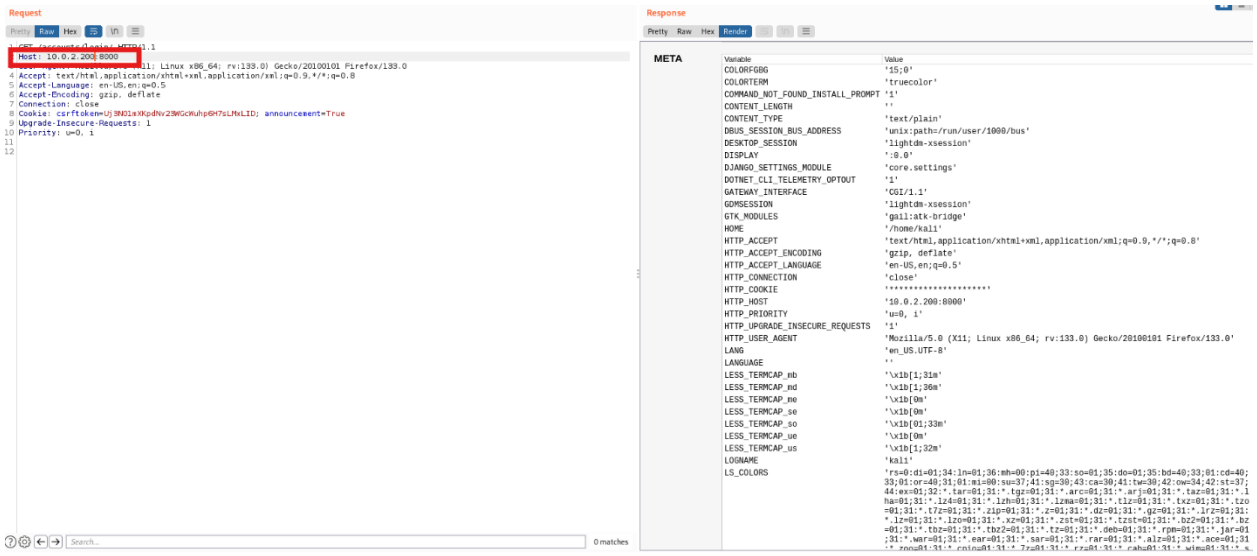
Step 3. Send request to repeater and change the host header to external ip address (e.g 10.0.2.200) to see if the server validates the host header or not

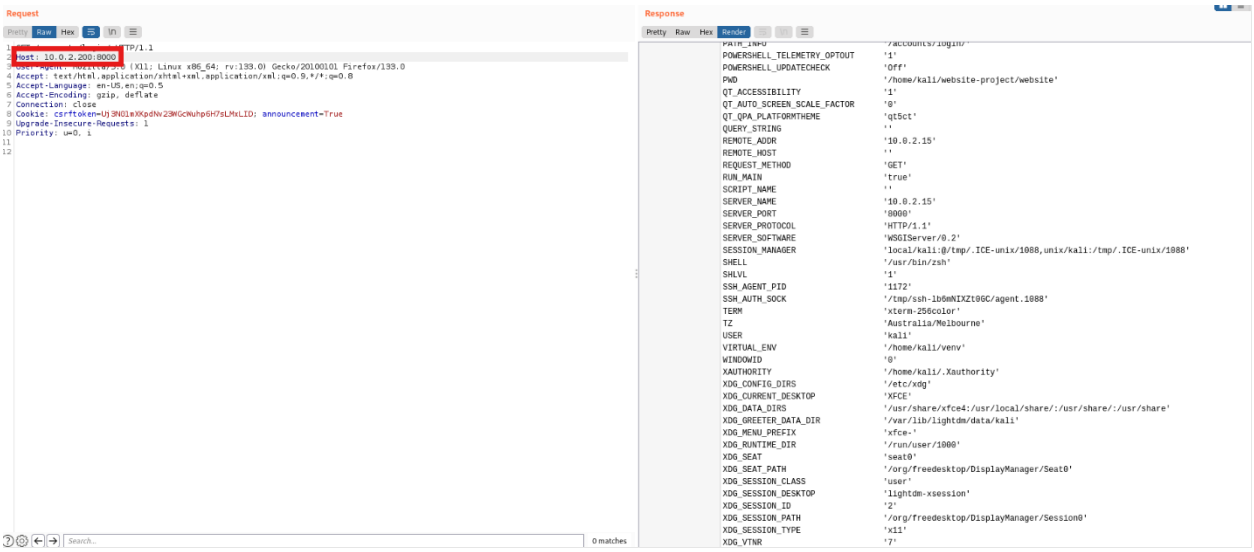


Step 4. We send a request and, despite encountering an error, the server discloses sensitive information such as the PATH environment, server version, and other details. This can lead to potential Remote Code Execution (RCE) if the server is running an outdated or vulnerable version. Additionally, the source code revealed by the error indicates the specific part of the application that triggered the issue. The application's current directory (PWD) is also exposed, providing attackers with insights into the directory structure and hosting details. This information could be leveraged for further exploitation, the current user and operating system

running the application are also exposed







Remediation Advice

The reason we could see the source code always because the debug mode is enabled.

1) Disable Debugging in Production

```
# SECURITY WARNING: keep the secret key used in production secret!
SECRET_KEY = os.environ.get('SECRET_KEY')
if not SECRET_KEY:
    SECRET_KEY = ''.join(random.choice( string.ascii_lowercase  ) for i in range( 32 ))

# Render Deployment Code
#DEBUG = False
#original:
# DEBUG = 'RENDER' not in os.environ
PRODUCTION = 'RUN_MAIN' not in os.environ
# Set DEBUG based on the environment. TO test 404 locally, set Debug = False.
DEBUG = not PRODUCTION
```

2) Restrict Access:

- Ensure that sensitive information is not accessible to unauthorized users. Use proper access control mechanisms

3) Secure Configuration Files:

- Store sensitive information like credentials, API keys, and system configurations securely using environment variables or secret management tools

References

Information Disclosure Vulnerabilities – PortSwigger Web Security Academy

What Is an Information Disclosure Vulnerability? – HackerOne

Vulnerability Disclosure Cheat Sheet – OWASP