Penetration Testing Report

Full Name: Milind Patel
Program: HCS - Penetration Testing Internship Week-2
Date: 25-02-2025

Introduction

This report document hereby describes the proceedings and results of a Black Box security assessment conducted against the **Week 2 Labs**. The report hereby lists the findings and corresponding best practice mitigation actions and recommendations.

1. Objective

The objective of the assessment was to uncover vulnerabilities in the **Week 2 Labs** and provide a final security assessment report comprising vulnerabilities, remediation strategy and recommendation guidelines to help mitigate the identified vulnerabilities and risks during the activity.

2. Scope

This section defines the scope and boundaries of the project.

Application Name	**Insecure Direct Object Reference , SQL Injection**

3. Summary

Outlined is a Black Box Application Security assessment for the **Week 2 Labs**.

Total number of Sub-labs: {count} Sub-labs

High	Medium	Low
4	7	5

High - **Number of Sub-labs with hard difficulty level**

Medium - **Number of Sub-labs with Medium difficulty level**

Low - **Number of Sub-labs with Easy difficulty level**

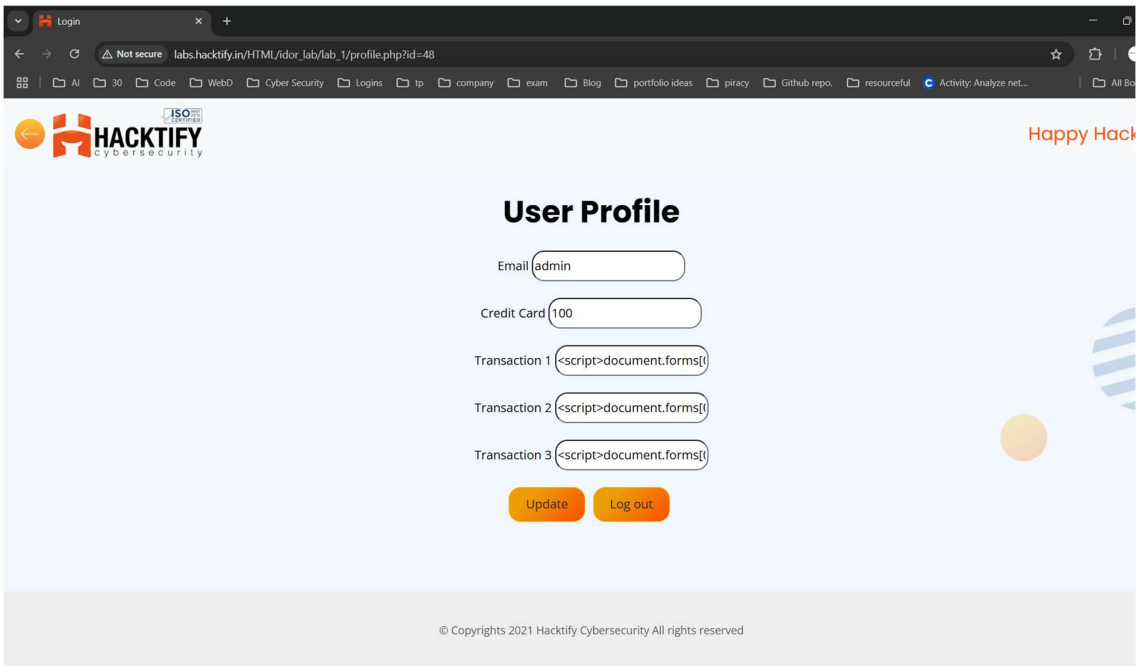
1. Insecure Direct Object Reference

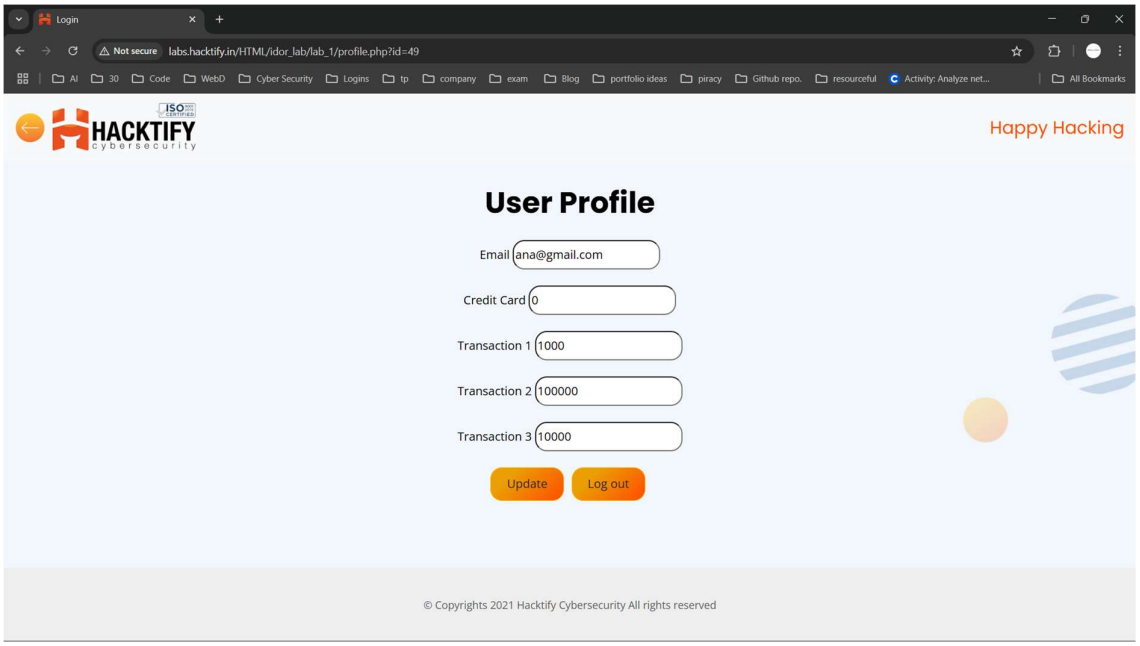
1.1. Give Me My Amount!!

Reference	Risk Rating
Give Me My Amount!!	**Low**
Tools Used	
burp suite	
Vulnerability Description	
The application is vulnerable to **IDOR**, allowing unauthorized access to other user accounts by modifying the **ID parameter** in the URL. The server does not validate whether the logged-in user has permission to access the requested resource.	
How It Was Discovered	
Manual Analysis: After registering and logging in, I observed the ID parameter in the URL, manually changed it to another user's ID, and successfully accessed their account without credentials.	
Vulnerable URLs	
http://labs.hacktify.in/HTML/idor_lab/lab_1/profile.php?id=48 http://labs.hacktify.in/HTML/idor_lab/lab_1/profile.php?id=49	
Consequences of not Fixing the Issue	
1. **Authentication Bypass:** Attackers can log in without valid credentials, gaining unauthorized access. 2. **Privilege Escalation:** If an admin account is compromised, the attacker can take full control of the system. 3. **Data Breach:** Sensitive user information may be exposed, leading to privacy violations.	
Suggested Countermeasures	
1. Use Prepared Statements (Parameterized Queries) 2. Implement Input Validation 3. Use Stored Procedures 4. Implement Multi-Factor Authentication (MFA)	
References	
https://portswigger.net/web-security/access-control/idor https://cheatsheetseries.owasp.org/cheatsheets/Insecure_Direct_Object_Reference_Prevention_Cheat_Sheet.html	

Proof of Concept

This section contains the proof of the above vulnerabilities as the screenshot of the vulnerability of the lab


Login
Not secure
labs.hacktify.in/HTML/idor_lab/lab_1/profile.php?id=48
HACKTIFY
cybersecurity
Happy Hack
User Profile
Email admin
Credit Card 100
Transaction 1 <script>document.forms[
Transaction 2 <script>document.forms[
Transaction 3 <script>document.forms[
Update
Log out
© Copyrights 2021 Hacktify Cybersecurity All rights reserved

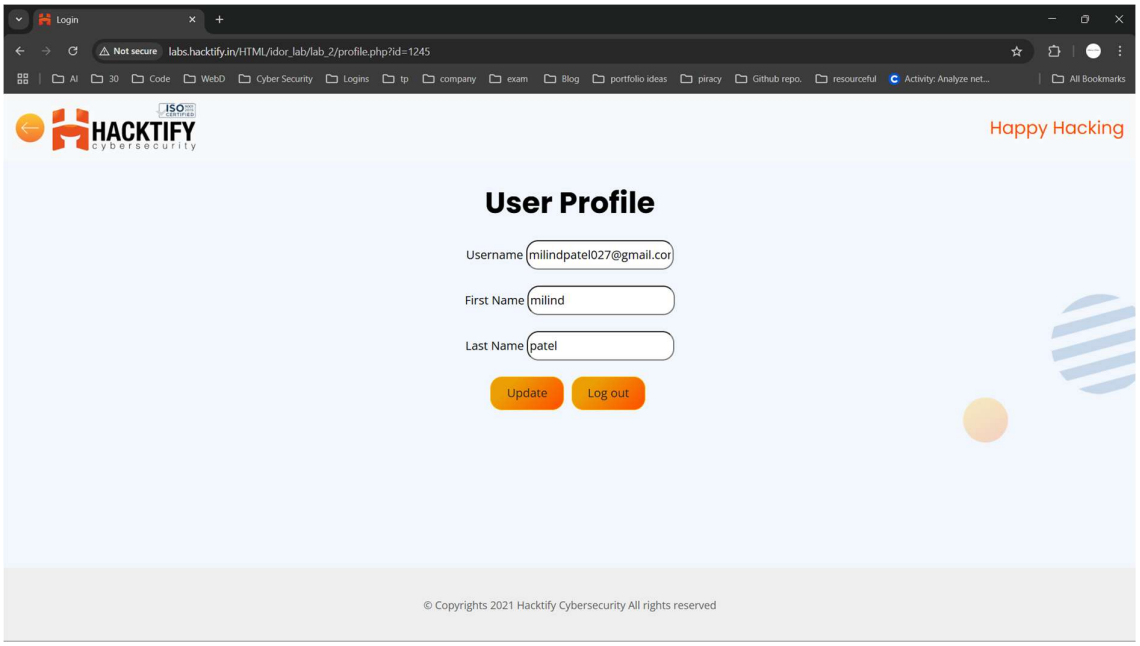

Login
Not secure
labs.hacktify.in/HTML/idor_lab/lab_1/profile.php?id=49
HACKTIFY
cybersecurity
Happy Hacking
User Profile
Email ana@gmail.com
Credit Card 0
Transaction 1 1000
Transaction 2 100000
Transaction 3 10000
Update
Log out
© Copyrights 2021 Hacktify Cybersecurity All rights reserved

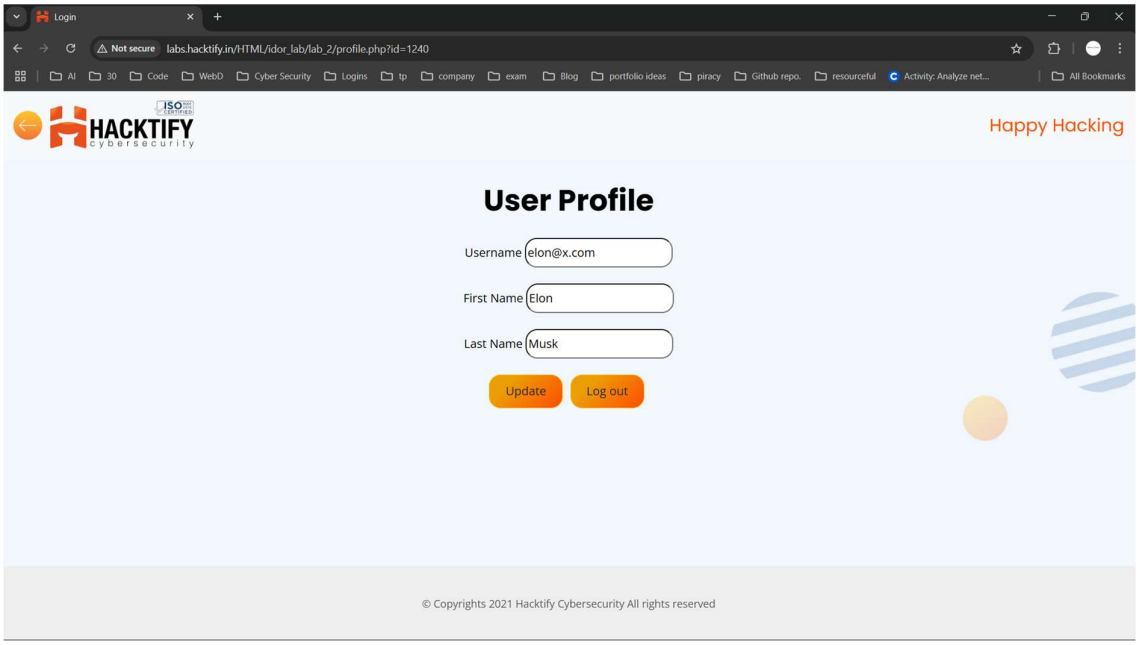
1.2. Stop pulling my params!

Reference	Risk Rating
Stop pulling my params!	**Medium**
Tools Used	
Burp Suite	
Vulnerability Description	
The application is vulnerable to **IDOR**, allowing unauthorized access to other user accounts by modifying the **ID parameter** in the URL. The server does not validate whether the logged-in user has permission to access the requested resource.	
How It Was Discovered	
Manual Analysis : After registering and logging in, I observed the ID parameter in the URL, manually changed it to another user's ID, and successfully accessed their account without credentials.	
Vulnerable URLs	
http://labs.hacktify.in/HTML/idor_lab/lab_2/profile.php?id=1245	
Consequences of not Fixing the Issue	
1. Account Takeover 2. Data Breach 3. Privilege Escalation 4. Compliance Violations	
Suggested Countermeasures	
1. Implement Proper Authorization Checks 2. Use Session-Based Authentication 3. Validate User Access Server-Side	
References	
https://portswigger.net/web-security/access-control/idor https://cheatsheetseries.owasp.org/cheatsheets/Insecure_Direct_Object_Reference_Prevention_Cheat_Sheet.html	

Proof of Concept

This section contains the proof of the above vulnerabilities as the screenshot of the vulnerability of the lab


Login
Not secure
labs.hacktify.in/HTML/idor_lab/lab_2/profile.php?id=1245
HACKTIFY
cybersecurity
Happy Hacking
User Profile
Username milindpatel027@gmail.cor
First Name milind
Last Name patel
Update
Log out
© Copyrights 2021 Hacktify Cybersecurity All rights reserved

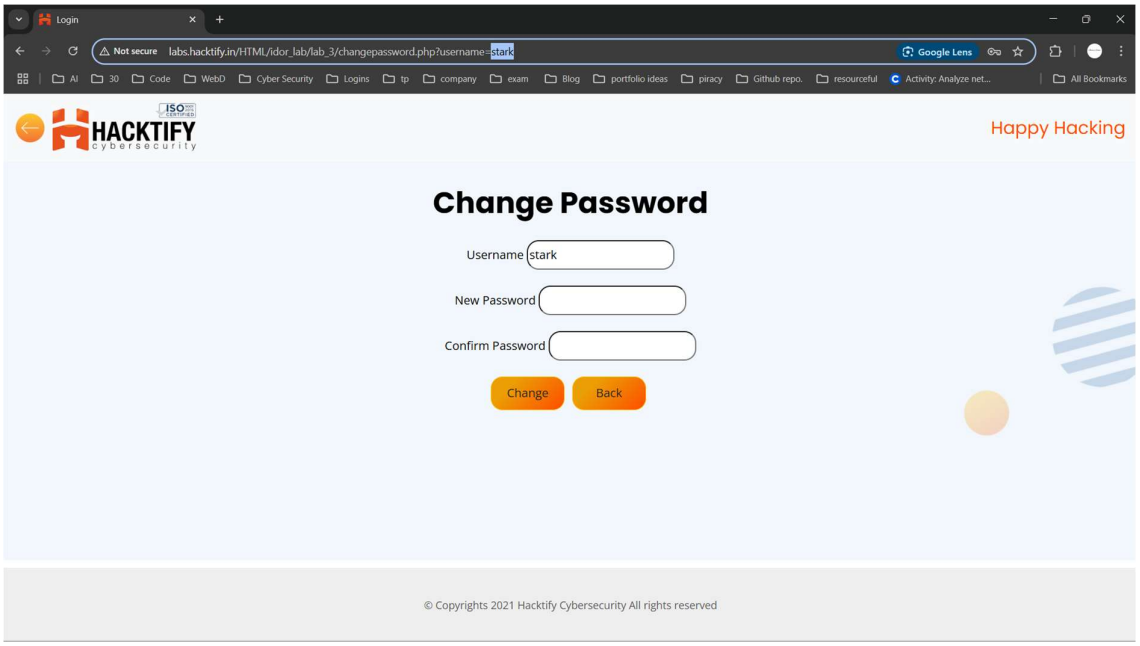

Login
Not secure
labs.hacktify.in/HTML/idor_lab/lab_2/profile.php?id=1240
HACKTIFY
cybersecurity
Happy Hacking
User Profile
Username elon@x.com
First Name Elon
Last Name Musk
Update
Log out
© Copyrights 2021 Hacktify Cybersecurity All rights reserved

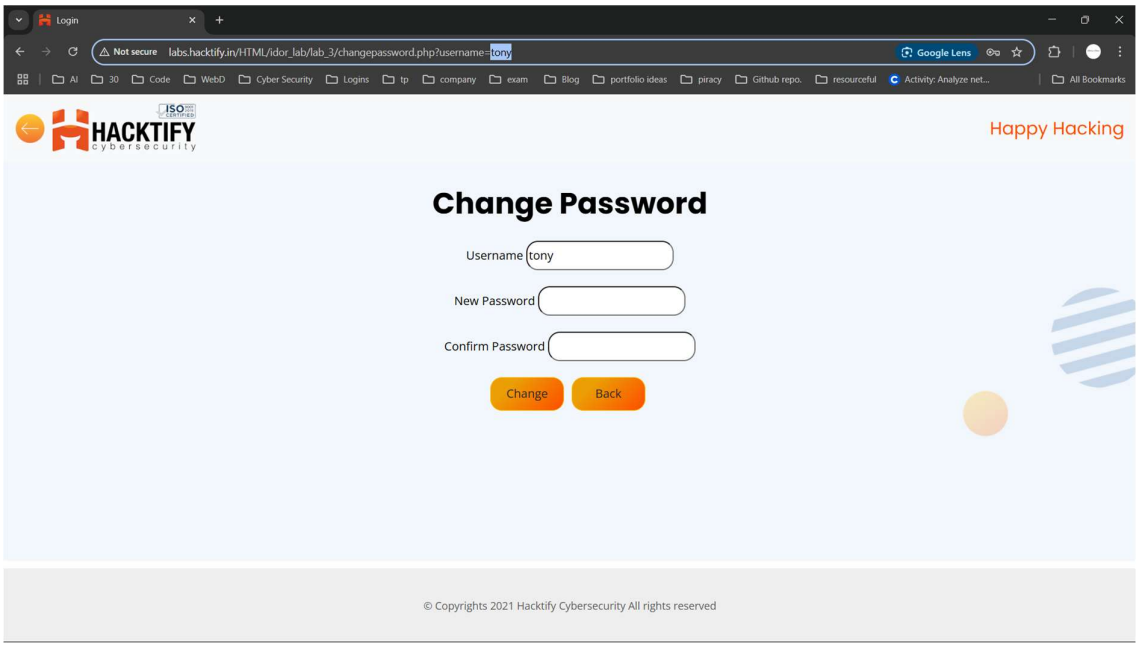
1.3. Someone changed my password!

Reference	Risk Rating
Someone changed my password	**Medium**
Tools Used	
Burp Suite	
Vulnerability Description	
The application is vulnerable to **IDOR**, allowing unauthorized access to other user accounts by modifying the **username parameter** in the URL. The server does not properly validate whether the logged-in user has permission to access the requested account	
How It Was Discovered	
Manual Analysis : after logging in, I observed the username parameter in the URL, changed it to another user's username, and successfully accessed their account without credentials.	
Vulnerable URLs	
http://labs.hacktify.in/HTML/idor_lab/lab_3/changepassword.php?username=stark http://labs.hacktify.in/HTML/idor_lab/lab_3/changepassword.php?username=tony	
Consequences of not Fixing the Issue	
1. Account Takeover 2. Data Breach 3. Privilege Escalation 4. Compliance Violations	
Suggested Countermeasures	
1. Implement Proper Authorization Checks 2. Use Session-Based Authentication 3. Validate User Access Server-Side	
References	
https://portswigger.net/web-security/access-control/idor https://cheatsheetseries.owasp.org/cheatsheets/Insecure_Direct_Object_Reference_Prevention_Cheat_Sheet.html	

Proof of Concept

This section contains the proof of the above vulnerabilities as the screenshot of the vulnerability of the lab


Login
Not secure
labs.hacktify.in/HTML/idor_lab/lab_3/changepassword.php?username=stark
Google Lens
HACKTIFY
cybersecurity
Happy Hacking
Change Password
Username stark
New Password
Confirm Password
Change
Back
© Copyrights 2021 Hacktify Cybersecurity All rights reserved

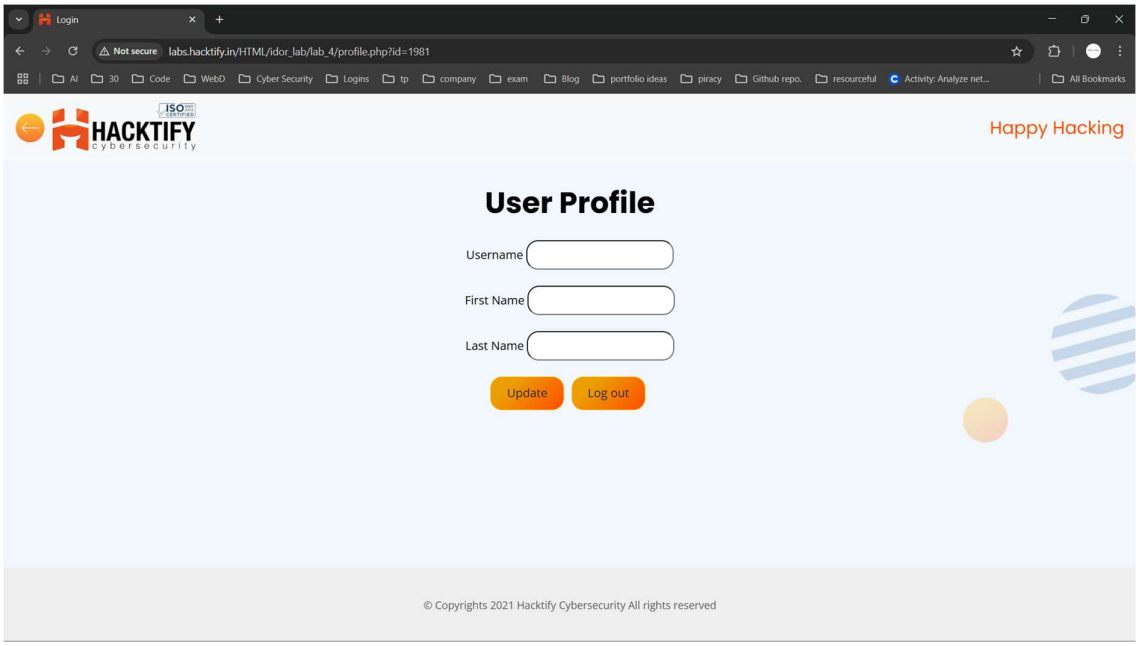

Login
Not secure
labs.hacktify.in/HTML/idor_lab/lab_3/changepassword.php?username=tony
Google Lens
HACKTIFY
cybersecurity
Happy Hacking
Change Password
Username tony
New Password
Confirm Password
Change
Back
© Copyrights 2021 Hacktify Cybersecurity All rights reserved

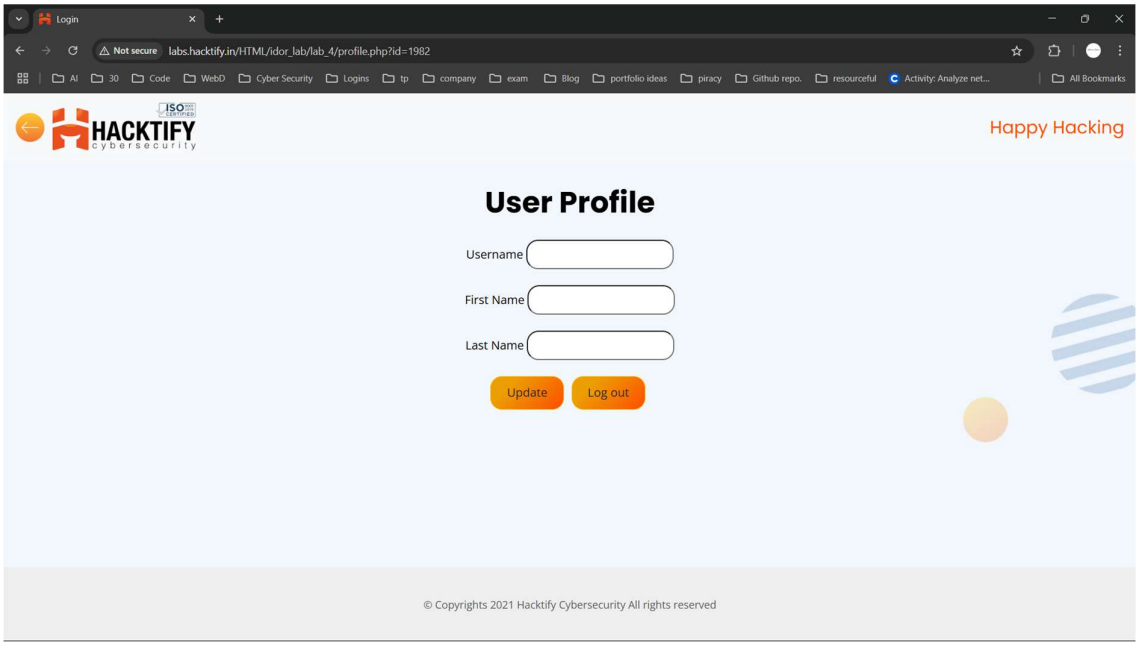
1.4. Change your methods.

Reference	Risk Rating
Change your methods	**Medium**
Tools Used	
Burp Suite	
Vulnerability Description	
The application is vulnerable to **IDOR**, allowing unauthorized access to and modification of other users' data by manipulating the **numerical ID parameter** in the URL. The server does not verify whether the logged-in user has permission to update another user's information.	
How It Was Discovered	
Manual Analysis : After registering two users, I observed numerical ID parameters in the URL, changed one to another user's ID, and was able to access and update their account data without authentication.	
Vulnerable URLs	
http://labs.hacktify.in/HTML/idor_lab/lab_4/profile.php?id=1981 http://labs.hacktify.in/HTML/idor_lab/lab_4/profile.php?id=1982	
Consequences of not Fixing the Issue	
1. Unauthorized Access 2. Data Tampering 3. Privilege Escalation 4. Identity Theft & Fraud	
Suggested Countermeasures	
1. Implement Proper Authorization Checks 2. Use Session-Based Authentication 3. Enforce Server-Side Validation	
References	
https://portswigger.net/web-security/access-control/idor https://cheatsheetseries.owasp.org/cheatsheets/Insecure_Direct_Object_Reference_Prevention_Cheat_Sheet.html	

Proof of Concept

This section contains the proof of the above vulnerabilities as the screenshot of the vulnerability of the lab


Login
Not secure
labs.hacktify.in/HTML/idor_lab/lab_4/profile.php?id=1981
HACKTIFY
cybersecurity
Happy Hacking
User Profile
Username
First Name
Last Name
Update
Log out
© Copyrights 2021 Hacktify Cybersecurity All rights reserved


Login
Not secure
labs.hacktify.in/HTML/idor_lab/lab_4/profile.php?id=1982
HACKTIFY
cybersecurity
Happy Hacking
User Profile
Username
First Name
Last Name
Update
Log out
© Copyrights 2021 Hacktify Cybersecurity All rights reserved

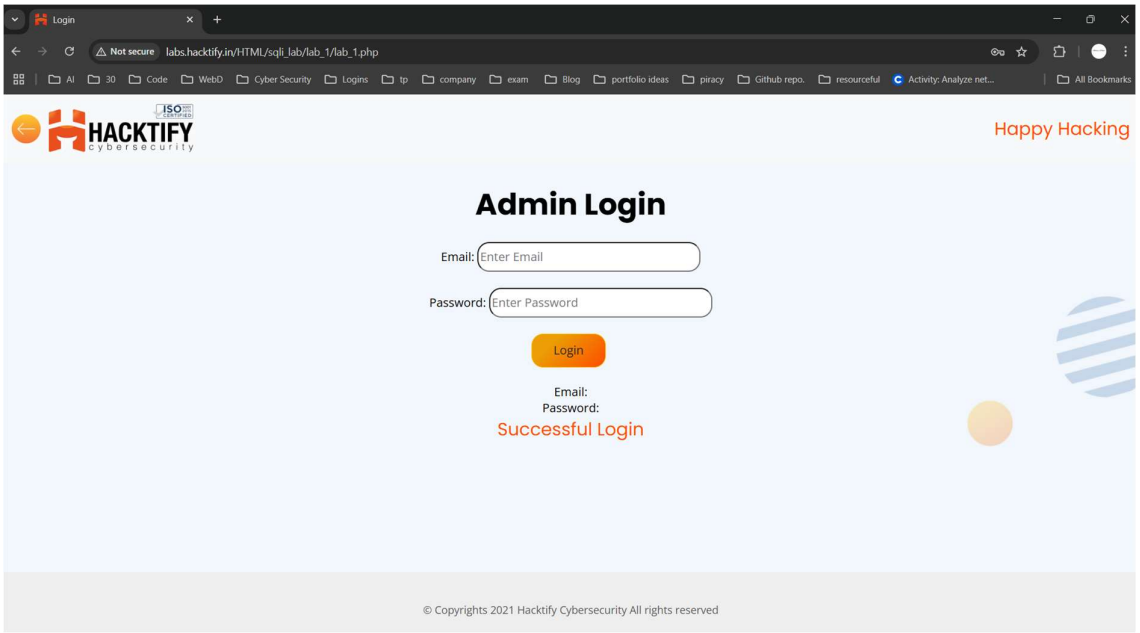
2. SQL Injection

2.1. Strings & Errors Part 1!

Reference	Risk Rating
Strings & Errors Part 1!	**Low**
Tools Used	
Burp Suite	
Vulnerability Description	
The application is vulnerable to **SQL Injection**, allowing an attacker to bypass authentication by injecting malicious SQL queries. The login form does not properly validate user inputs before executing database queries.	
How It Was Discovered	
Manual Analysis: By injecting SQL payloads in username and password.	
Vulnerable URLs	
http://labs.hacktify.in/HTML/sqli_lab/lab_1/lab_1.php	
Consequences of not Fixing the Issue.	
1. Unauthorized Access 2. Data Exposure 3. Privilege Escalation 4. Data Manipulation 5. System Compromise	
Suggested Countermeasures	
1. Use Prepared Statements (Parameterized Queries) 2. Implement Input Validation 3. Use Stored Procedures 4. Apply Least Privilege Principle 5. Enable Web Application Firewall (WAF)	
References	
https://portswigger.net/web-security/sql-injection https://owasp.org/www-community/attacks/SQL_Injection	

Proof of Concept

This section contains the proof of the above vulnerabilities as the screenshot of the vulnerability of the lab

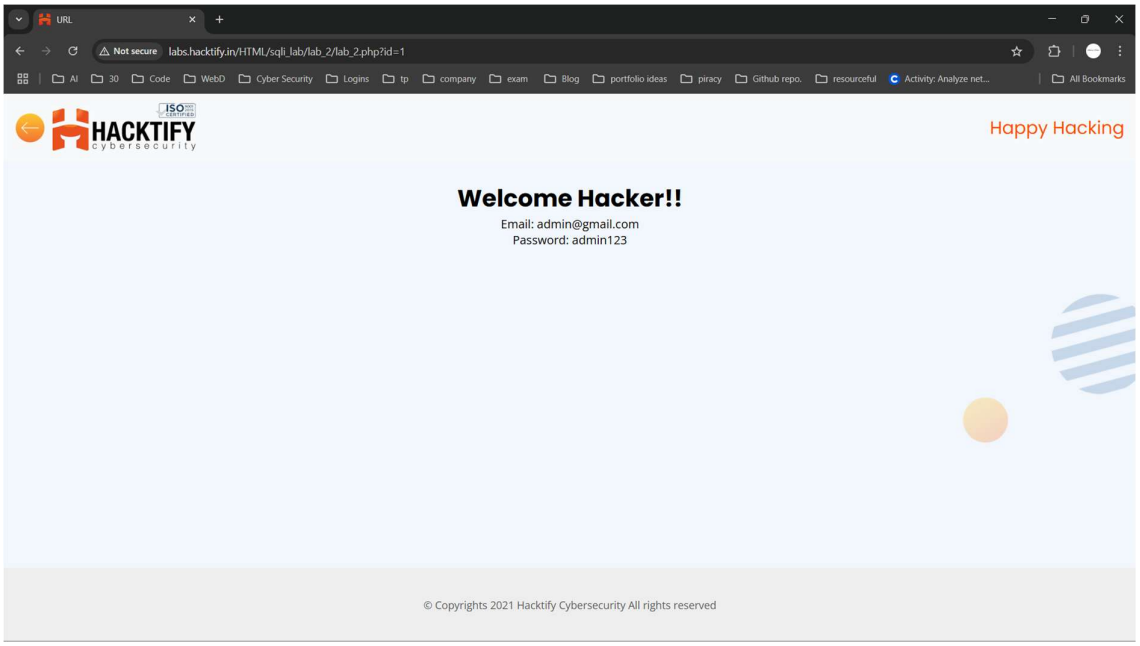

Login
Not secure
labs.hacktify.in/HTML/sqli_lab/lab_1/lab_1.php
AI
30
Code
WebD
Cyber Security
Logins
tp
company
exam
Blog
portfolio ideas
piracy
Github repo.
resourceful
Activity: Analyze net...
All Bookmarks
HACKTIFY
cybersecurity
Happy Hacking
Admin Login
Email:
Enter Email
Password:
Enter Password
Login
Email:
Password:
Successful Login
© Copyrights 2021 Hacktify Cybersecurity All rights reserved

2.2. Strings & Errors Part 2!

Reference	Risk Rating
Strings & Errors Part 2!	**Low**
Tools Used	
Burp Suite	
Vulnerability Description	
The application is vulnerable to **SQL Injection**, allowing an attacker to bypass authentication by injecting malicious SQL queries. The login form does not properly validate user inputs before executing database queries.	
How It Was Discovered	
Manual Analysis: I directly entered numeric values in the ID parameter of the login URL and successfully accessed different user accounts without authentication.	
Vulnerable URLs	
http://labs.hacktify.in/HTML/sqli_lab/lab_2/lab_2.php?id=1	
Consequences of not Fixing the Issue	
1. Unauthorized Access 2. Data Exposure 3. Privilege Escalation 4. Data Manipulation 5. System Compromise	
Suggested Countermeasures	
1. Use Prepared Statements (Parameterized Queries) 2. Implement Input Validation 3. Use Stored Procedures 4. Apply Least Privilege Principle 5. Enable Web Application Firewall (WAF)	
References	
https://portswigger.net/web-security/sql-injection https://owasp.org/www-community/attacks/SQL_Injection	

Proof of Concept

This section contains the proof of the above vulnerabilities as the screenshot of the vulnerability of the lab

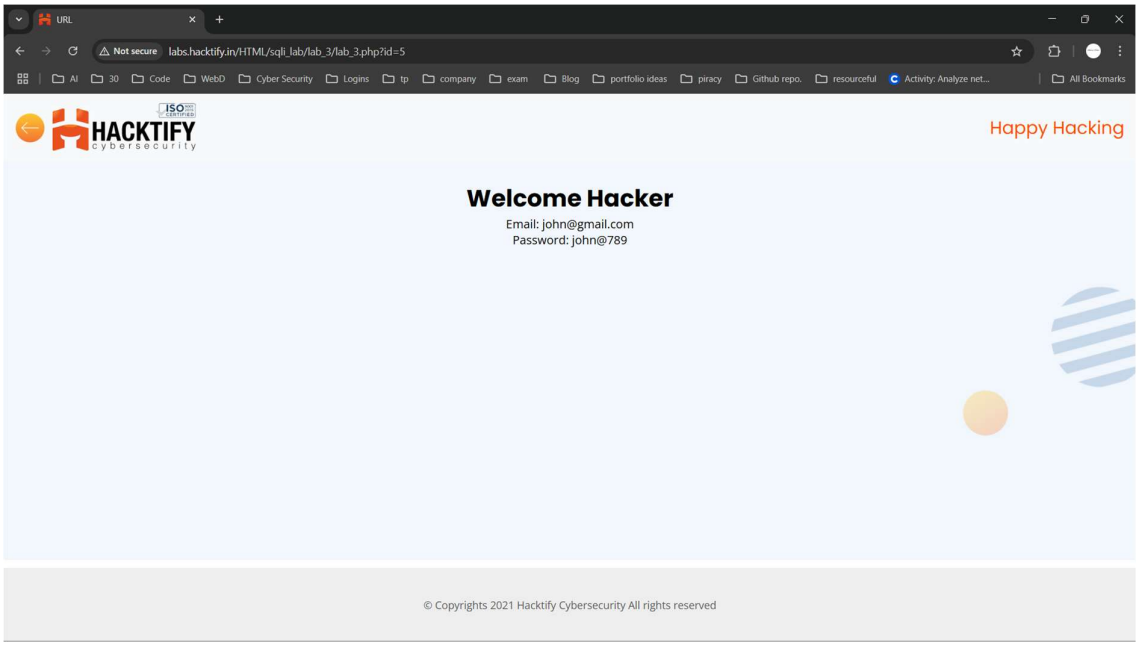

URL
Not secure
labs.hacktify.in/HTML/sqli_lab/lab_2/lab_2.php?id=1
AI
30
Code
WebD
Cyber Security
Logins
tp
company
exam
Blog
portfolio ideas
piracy
Github repo.
resourceful
Activity: Analyze net...
All Bookmarks
HACKTIFY
cybersecurity
Happy Hacking
Welcome Hacker!!
Email: admin@gmail.com
Password: admin123
© Copyrights 2021 Hacktify Cybersecurity All rights reserved

2.3. Strings & Errors Part 3!

Reference	Risk Rating
Strings & Errors Part 3!	**Low**
Tools Used	
Burp Suite	
Vulnerability Description	
The application is vulnerable to **SQL Injection**, allowing an attacker to bypass authentication by injecting malicious SQL queries. The login form does not properly validate user inputs before executing database queries.	
How It Was Discovered	
Manual Analysis: I directly entered numeric values in the ID parameter of the login URL and successfully accessed different user accounts without authentication.	
Vulnerable URLs	
http://labs.hacktify.in/HTML/sqli_lab/lab_3/lab_3.php?id=5	
Consequences of not Fixing the Issue	
1. Unauthorized Access 2. Data Exposure 3. Privilege Escalation 4. Data Manipulation 5. System Compromise	
Suggested Countermeasures	
1. Use Prepared Statements (Parameterized Queries) 2. Implement Input Validation 3. Use Stored Procedures 4. Apply Least Privilege Principle 5. Enable Web Application Firewall (WAF)	
References	
https://portswigger.net/web-security/sql-injection https://owasp.org/www-community/attacks/SQL_Injection	

Proof of Concept

This section contains the proof of the above vulnerabilities as the screenshot of the vulnerability of the lab

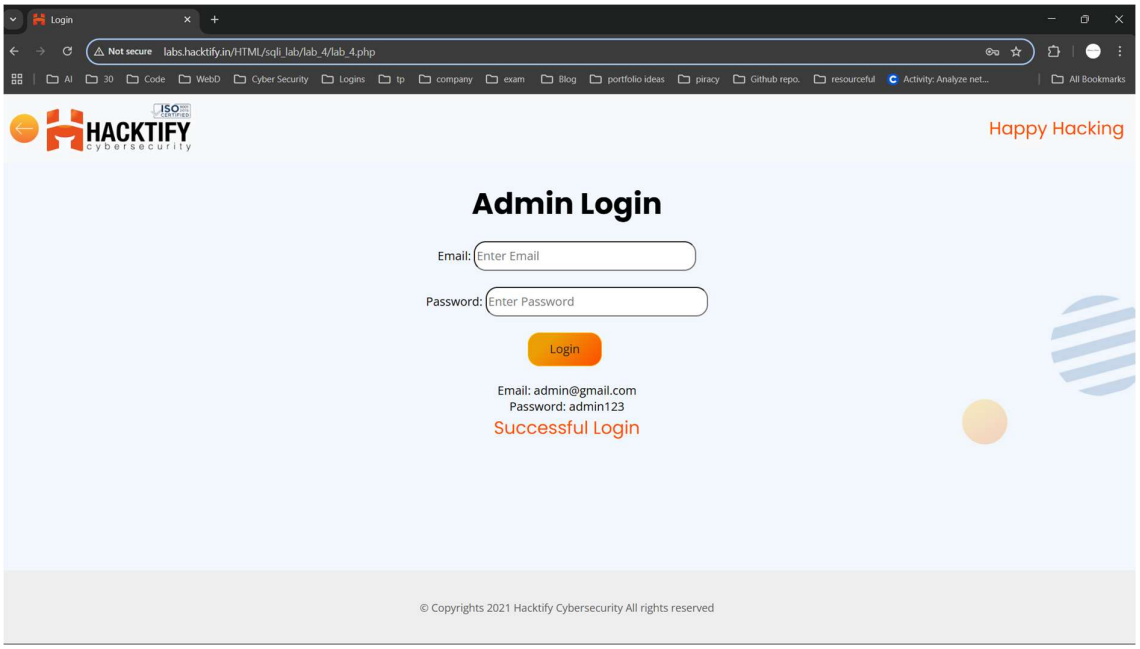


2.4. Let's trick 'em!

Reference	Risk Rating
Let's trick 'em!	**Medium**
Tools Used	
Burp Suite	
Vulnerability Description	
The application is vulnerable to **SQL Injection**, allowing an attacker to bypass authentication by injecting malicious SQL queries. The login form does not properly validate user inputs before executing database queries.	
How It Was Discovered	
Manual Analysis: By injecting SQL payloads in username and password.	
Vulnerable URLs	
http://labs.hacktify.in/HTML/sqli_lab/lab_4/lab_4.php	
Consequences of not Fixing the Issue	
1. Unauthorized Access 2. Data Exposure 3. Privilege Escalation 4. Data Manipulation 5. System Compromise	
Suggested Countermeasures	
1. Use Prepared Statements (Parameterized Queries) 2. Implement Input Validation 3. Use Stored Procedures 4. Apply Least Privilege Principle 5. Enable Web Application Firewall (WAF)	
References	
https://portswigger.net/web-security/sql-injection https://owasp.org/www-community/attacks/SQL_Injection	

Proof of Concept

This section contains the proof of the above vulnerabilities as the screenshot of the vulnerability of the lab

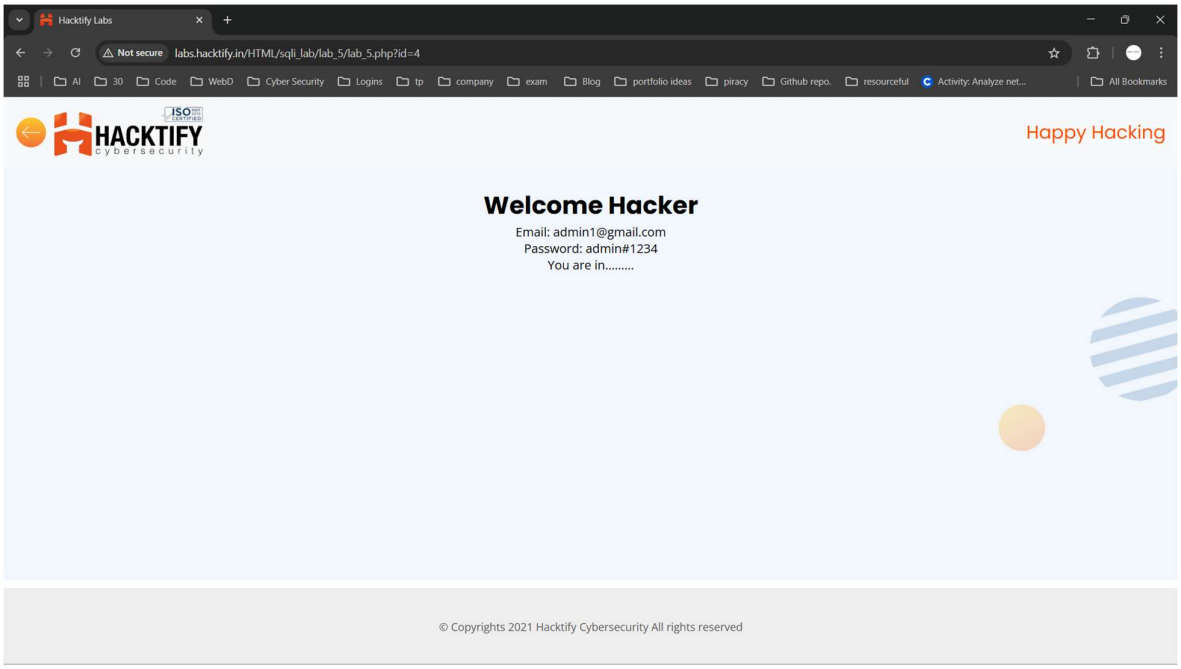

Login
Not secure
labs.hacktify.in/HTML/sqli_lab/lab_4/lab_4.php
HACKTIFY
cybersecurity
Happy Hacking
Admin Login
Email:
Enter Email
Password:
Enter Password
Login
Email: admin@gmail.com
Password: admin123
Successful Login
© Copyrights 2021 Hacktify Cybersecurity All rights reserved

2.5. Booleans and Blind!

Reference	Risk Rating
Booleans and Blind!	**High**
Tools Used	
Burp Suite	
Vulnerability Description	
The application is vulnerable to **SQL Injection**, allowing an attacker to bypass authentication by injecting malicious SQL queries. The login form does not properly validate user inputs before executing database queries.	
How It Was Discovered	
Manual Analysis: I directly entered numeric values in the ID parameter of the login URL and successfully accessed different user accounts without authentication	
Vulnerable URLs	
http://labs.hacktify.in/HTML/sqli_lab/lab_5/lab_5.php?id=4	
Consequences of not Fixing the Issue	
1. Unauthorized Access 2. Data Exposure 3. Privilege Escalation 4. Data Manipulation 5. System Compromise	
Suggested Countermeasures	
1. Use Prepared Statements (Parameterized Queries) 2. Implement Input Validation 3. Use Stored Procedures 4. Apply Least Privilege Principle 5. Enable Web Application Firewall (WAF)	
References	
https://portswigger.net/web-security/sql-injection https://owasp.org/www-community/attacks/SQL_Injection	

Proof of Concept

This section contains the proof of the above vulnerabilities as the screenshot of the vulnerability of the lab

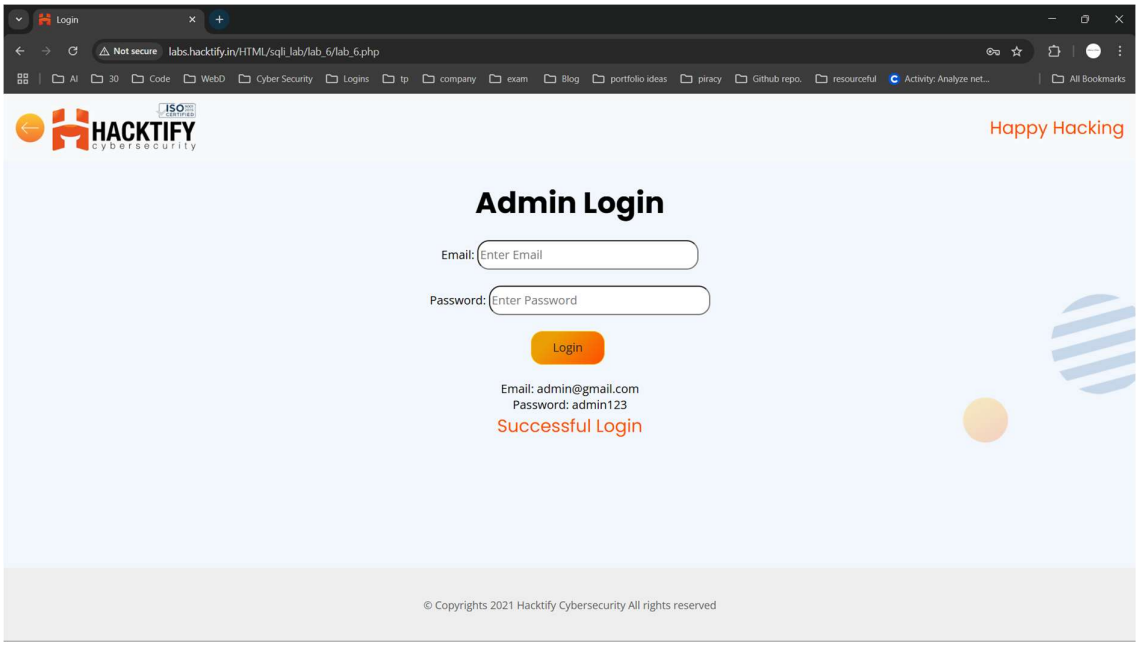

Hacktify Labs
Not secure
labs.hacktify.in/HTML/sqli_lab/lab_5/lab_5.php?id=4
AI
30
Code
WebD
Cyber Security
Logins
tp
company
exam
Blog
portfolio ideas
piracy
Github repo.
resourceful
Activity: Analyze net...
All Bookmarks
HACKTIFY
cybersecurity
Happy Hacking
Welcome Hacker
Email: admin1@gmail.com
Password: admin#1234
You are in.....
© Copyrights 2021 Hacktify Cybersecurity All rights reserved

2.6. Error based : Tricked !

Reference	Risk Rating
Error based : Tricked !	**Medium**
Tools Used	
Burp Suite	
Vulnerability Description	
The application is vulnerable to **SQL Injection**, allowing an attacker to bypass authentication by injecting malicious SQL queries. The login form does not properly validate user inputs before executing database queries.	
How It Was Discovered	
Manual Analysis: By injecting SQL payloads in username and password.	
Vulnerable URLs	
http://labs.hacktify.in/HTML/sqli_lab/lab_6/lab_6.php	
Consequences of not Fixing the Issue	
1. Unauthorized Access 2. Data Exposure 3. Privilege Escalation 4. Data Manipulation 5. System Compromise	
Suggested Countermeasures	
1. Use Prepared Statements (Parameterized Queries) 2. Implement Input Validation 3. Use Stored Procedures 4. Apply Least Privilege Principle 5. Enable Web Application Firewall (WAF)	
References	
https://portswigger.net/web-security/sql-injection https://owasp.org/www-community/attacks/SQL_Injection	

Proof of Concept

This section contains the proof of the above vulnerabilities as the screenshot of the vulnerability of the lab

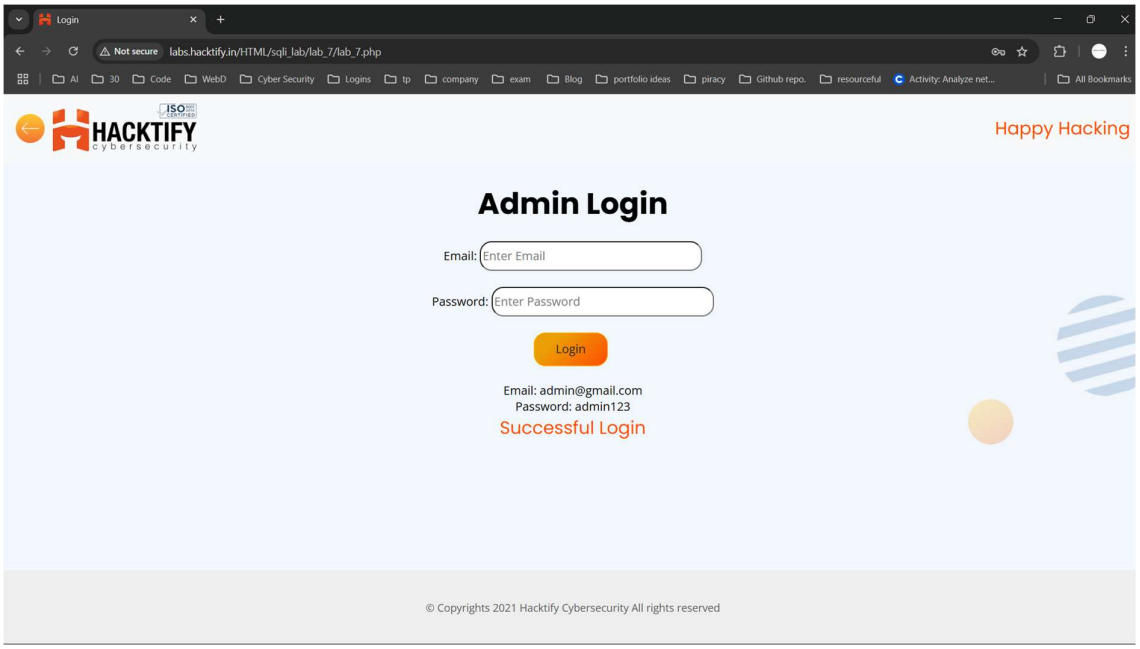

Login
Not secure
labs.hacktify.in/HTML/sqli_lab/lab_6/lab_6.php
AI
30
Code
WebD
Cyber Security
Logins
tp
company
exam
Blog
portfolio ideas
piracy
Github repo.
resourceful
Activity: Analyze net...
All Bookmarks
HACKTIFY
cybersecurity
Happy Hacking
Admin Login
Email:
Enter Email
Password:
Enter Password
Login
Email: admin@gmail.com
Password: admin123
Successful Login
© Copyrights 2021 Hacktify Cybersecurity All rights reserved

2.7. Errors and Post!

Reference	Risk Rating
Errors and Post !	**Low**
Tools Used	
Burp Suite	
Vulnerability Description	
The application is vulnerable to **SQL Injection**, allowing an attacker to bypass authentication by injecting malicious SQL queries. The login form does not properly validate user inputs before executing database queries.	
How It Was Discovered	
Manual Analysis: By injecting SQL payloads in username and password.	
Vulnerable URLs	
http://labs.hacktify.in/HTML/sqli_lab/lab_7/lab_7.php	
Consequences of not Fixing the Issue	
1. Unauthorized Access 2. Data Exposure 3. Privilege Escalation 4. Data Manipulation 5. System Compromise	
Suggested Countermeasures	
1. Use Prepared Statements (Parameterized Queries) 2. Implement Input Validation 3. Use Stored Procedures 4. Apply Least Privilege Principle 5. Enable Web Application Firewall (WAF)	
References	
https://portswigger.net/web-security/sql-injection https://owasp.org/www-community/attacks/SQL_Injection	

Proof of Concept

This section contains the proof of the above vulnerabilities as the screenshot of the vulnerability of the lab

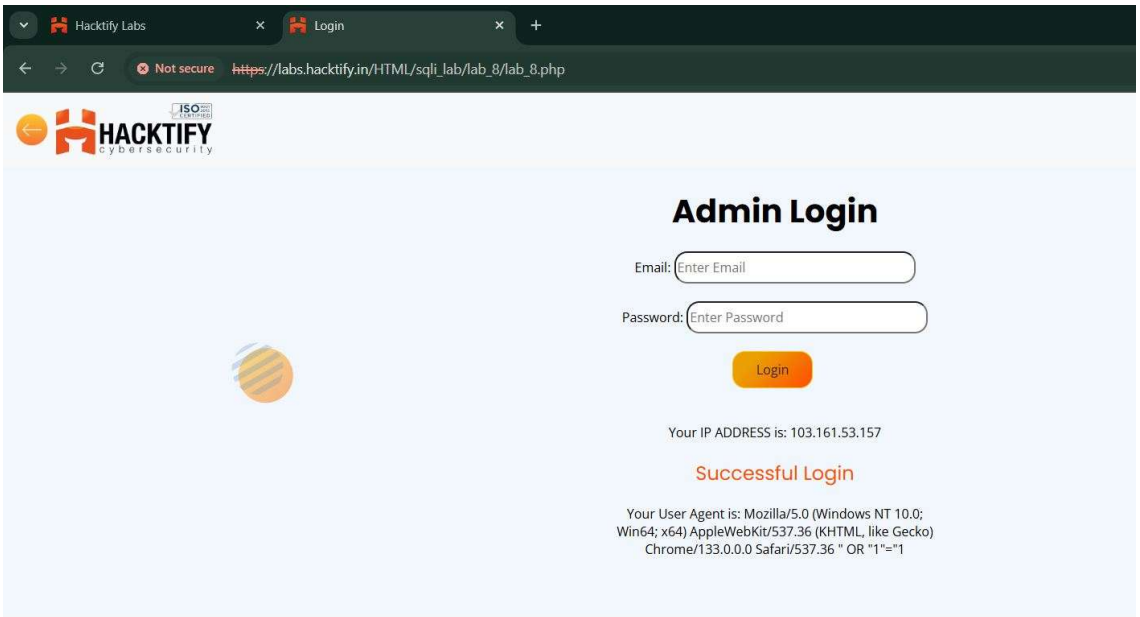

Login
Not secure
labs.hacktify.in/HTML/sqli_lab/lab_7/lab_7.php
AI
30
Code
WebD
Cyber Security
Logins
tp
company
exam
Blog
portfolio ideas
piracy
Github repo.
resourceful
Activity: Analyze net...
All Bookmarks
HACKTIFY
cybersecurity
Happy Hacking
Admin Login
Email:
Enter Email
Password:
Enter Password
Login
Email: admin@gmail.com
Password: admin123
Successful Login
© Copyrights 2021 Hacktify Cybersecurity All rights reserved

2.8. User Agents lead us!

Reference	Risk Rating
User agents lead us!	**High**
Tools Used	
Burp Suite	
Vulnerability Description	
The application is vulnerable to **SQL Injection**, allowing an attacker to bypass authentication by injecting malicious SQL queries. The login form does not properly validate user inputs before executing database queries.	
How It Was Discovered	
Manual Analysis: By injecting SQL payloads in username and password.	
Vulnerable URLs	
http://labs.hacktify.in/HTML/sqli_lab/lab_8/lab_8.php	
Consequences of not Fixing the Issue	
1. Unauthorized Access 2. Data Exposure 3. Privilege Escalation 4. Data Manipulation 5. System Compromise	
Suggested Countermeasures	
1. Use Prepared Statements (Parameterized Queries) 2. Implement Input Validation 3. Use Stored Procedures 4. Apply Least Privilege Principle 5. Enable Web Application Firewall (WAF)	
References	
https://portswigger.net/web-security/sql-injection https://owasp.org/www-community/attacks/SQL_Injection	

Proof of Concept

This section contains the proof of the above vulnerabilities as the screenshot of the vulnerability of the lab

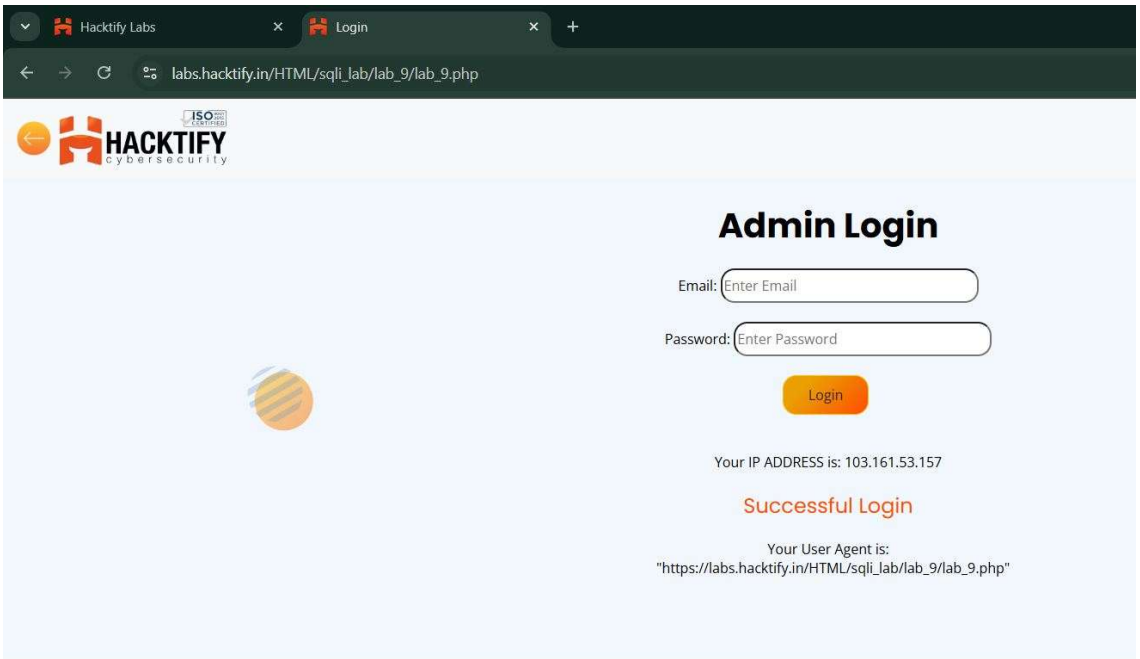

Hacktify Labs
Login
Not secure
https://labs.hacktify.in/HTML/sqli_lab/lab_8/lab_8.php
HACKTIFY
cybersecurity
Admin Login
Email: Enter Email
Password: Enter Password
Login
Your IP ADDRESS is: 103.161.53.157
Successful Login
Your User Agent is: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/133.0.0.0 Safari/537.36 " OR "1"="1

2.9. Referer lead u!

Reference	Risk Rating
Referer lead us!	**Medium**
Tools Used	
Burp Suite	
Vulnerability Description	
The application is vulnerable to **SQL Injection**, allowing an attacker to bypass authentication by injecting malicious SQL queries. The login form does not properly validate user inputs before executing database queries.	
How It Was Discovered	
Manual Analysis: By injecting SQL payloads in username and password.	
Vulnerable URLs	
http://labs.hacktify.in/HTML/sqli_lab/lab_9/lab_9.php	
Consequences of not Fixing the Issue	
1. Unauthorized Access 2. Data Exposure 3. Privilege Escalation 4. Data Manipulation 5. System Compromise	
Suggested Countermeasures	
1. Use Prepared Statements (Parameterized Queries) 2. Implement Input Validation 3. Use Stored Procedures 4. Apply Least Privilege Principle 5. Enable Web Application Firewall (WAF)	
References	
https://portswigger.net/web-security/sql-injection https://owasp.org/www-community/attacks/SQL_Injection	

Proof of Concept

This section contains the proof of the above vulnerabilities as the screenshot of the vulnerability of the lab

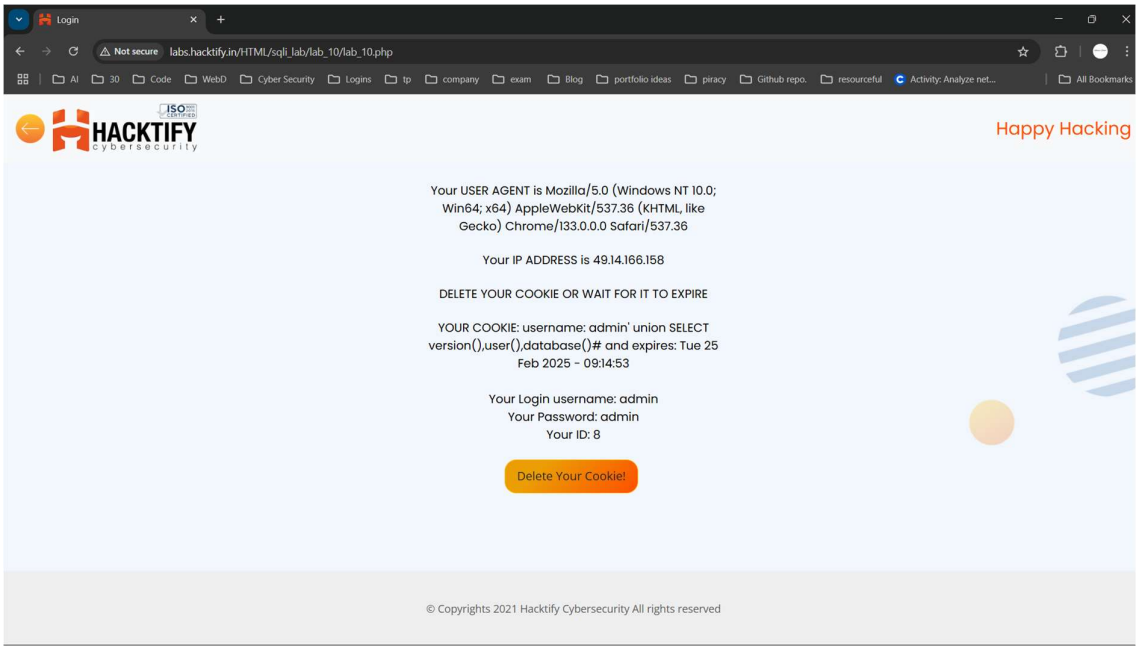

Hacktify Labs
Login
labs.hacktify.in/HTML/sqli_lab/lab_9/lab_9.php
HACKTIFY
cybersecurity
Admin Login
Email: Enter Email
Password: Enter Password
Login
Your IP ADDRESS is: 103.161.53.157
Successful Login
Your User Agent is:
"https://labs.hacktify.in/HTML/sqli_lab/lab_9/lab_9.php"

2.10. Oh Cookies!

Reference	Risk Rating
Oh Cookies!	**High**
Tools Used	
Burp Suite	
Vulnerability Description	
The application is vulnerable to **SQL Injection**, allowing an attacker to bypass authentication by injecting malicious SQL queries. The login form does not properly validate user inputs before executing database queries.	
How It Was Discovered	
Manual Analysis: By injecting SQL payloads in username and password.	
Vulnerable URLs	
http://labs.hacktify.in/HTML/sqli_lab/lab_10/lab_10.php	
Consequences of not Fixing the Issue	
1. Unauthorized Access 2. Data Exposure 3. Privilege Escalation 4. Data Manipulation 5. System Compromise	
Suggested Countermeasures	
1. Use Prepared Statements (Parameterized Queries) 2. Implement Input Validation 3. Use Stored Procedures 4. Apply Least Privilege Principle 5. Enable Web Application Firewall (WAF)	
References	
https://portswigger.net/web-security/sql-injection https://owasp.org/www-community/attacks/SQL_Injection	

Proof of Concept

This section contains the proof of the above vulnerabilities as the screenshot of the vulnerability of the lab

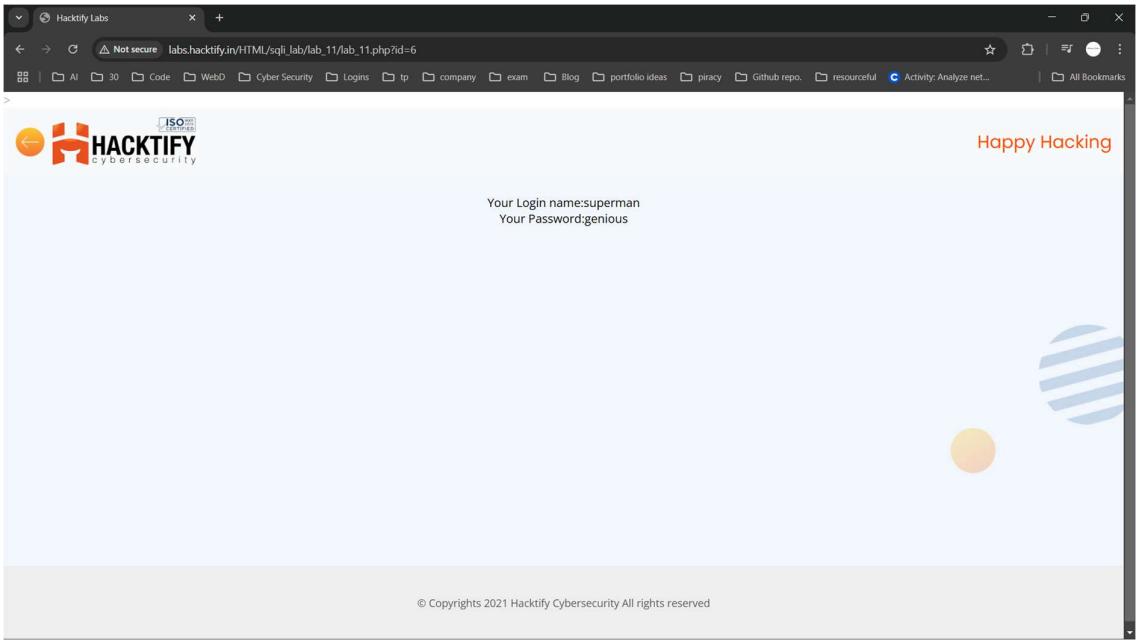

Login
Not secure
labs.hacktify.in/HTML/sqli_lab/lab_10/lab_10.php
AI
30
Code
WebD
Cyber Security
Logins
tp
company
exam
Blog
portfolio ideas
piracy
Github repo.
resourceful
Activity: Analyze net...
All Bookmarks
HACKTIFY
cybersecurity
Happy Hacking
Your USER AGENT is Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/133.0.0.0 Safari/537.36
Your IP ADDRESS is 49.14.166.158
DELETE YOUR COOKIE OR WAIT FOR IT TO EXPIRE
YOUR COOKIE: username: admin' union SELECT version(),user(),database()# and expires: Tue 25 Feb 2025 - 09:14:53
Your Login username: admin
Your Password: admin
Your ID: 8
Delete Your Cookie!
© Copyrights 2021 Hacktify Cybersecurity All rights reserved

2.11. WAF's are injected!

Reference	Risk Rating
WAF's are injected!	**High**
Tools Used	
Burp Suite	
Vulnerability Description	
The application is vulnerable to **SQL Injection**, allowing an attacker to bypass authentication by injecting malicious SQL queries. The login form does not properly validate user inputs before executing database queries.	
How It Was Discovered	
Manual Analysis: I directly entered numeric values in the ID parameter of the login URL and successfully accessed different user accounts without authentication.	
Vulnerable URLs	
http://labs.hacktify.in/HTML/sqli_lab/lab_11/lab_11.php?id=6	
Consequences of not Fixing the Issue	
1. Unauthorized Access 2. Data Exposure 3. Privilege Escalation 4. Data Manipulation 5. System Compromise	
Suggested Countermeasures	
1. Use Prepared Statements (Parameterized Queries) 2. Implement Input Validation 3. Use Stored Procedures 4. Apply Least Privilege Principle 5. Enable Web Application Firewall (WAF)	
References	
https://portswigger.net/web-security/sql-injection https://owasp.org/www-community/attacks/SQL_Injection	

Proof of Concept

This section contains the proof of the above vulnerabilities as the screenshot of the vulnerability of the lab

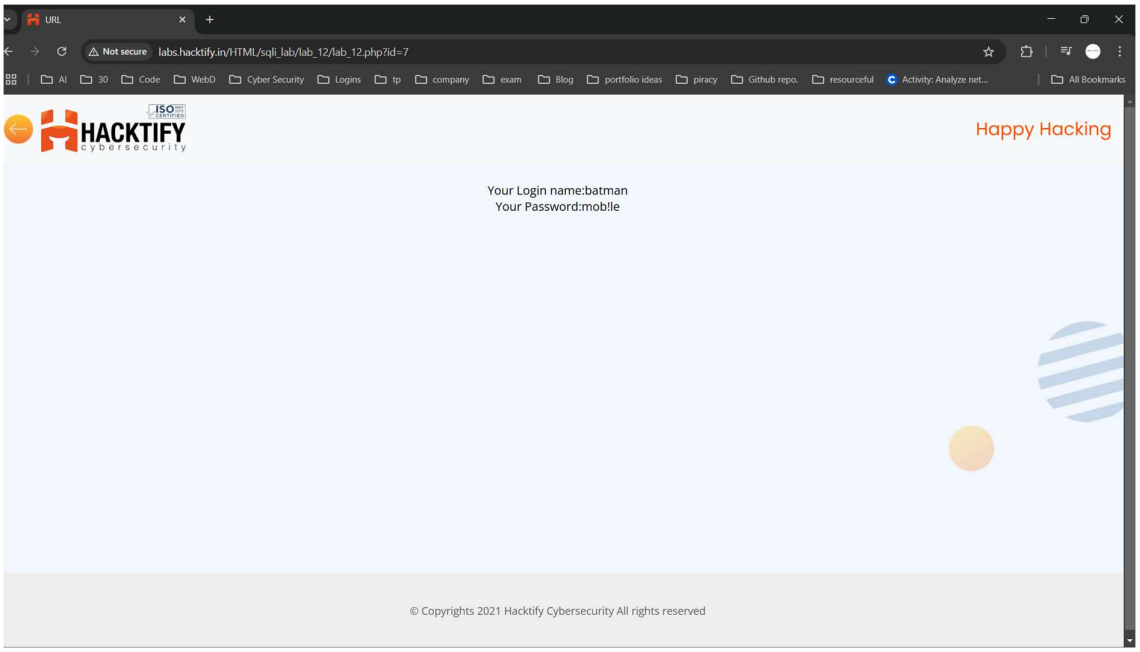

Hacktify Labs
Not secure
labs.hacktify.in/HTML/sqli_lab/lab_11/lab_11.php?id=6
AI
30
Code
WebD
Cyber Security
Logins
tp
company
exam
Blog
portfolio ideas
piracy
Github repo.
resourceful
Activity: Analyze net...
All Bookmarks
HACKTIFY
cybersecurity
Happy Hacking
Your Login name:superman
Your Password:genious
© Copyrights 2021 Hacktify Cybersecurity All rights reserved

2.12. WAF's are injected Part 2!

Reference	Risk Rating
WAF's are injected Part 2!	**Medium**
Tools Used	
Burp Suite	
Vulnerability Description	
The application is vulnerable to **SQL Injection**, allowing an attacker to bypass authentication by injecting malicious SQL queries. The login form does not properly validate user inputs before executing database queries.	
How It Was Discovered	
Manual Analysis: I directly entered numeric values in the ID parameter of the login URL and successfully accessed different user accounts without authentication.	
Vulnerable URLs	
http://labs.hacktify.in/HTML/sqli_lab/lab_12/lab_12.php?id=7	
Consequences of not Fixing the Issue	
1. Unauthorized Access 2. Data Exposure 3. Privilege Escalation 4. Data Manipulation 5. System Compromise	
Suggested Countermeasures	
1. Use Prepared Statements (Parameterized Queries) 2. Implement Input Validation 3. Use Stored Procedures 4. Apply Least Privilege Principle 5. Enable Web Application Firewall (WAF)	
References	
https://portswigger.net/web-security/sql-injection https://owasp.org/www-community/attacks/SQL_Injection	

Proof of Concept

This section contains the proof of the above vulnerabilities as the screenshot of the vulnerability of the lab


URL
Not secure
labs.hacktify.in/HTML/sqli_lab/lab_12/lab_12.php?id=7
AI
30
Code
WebD
Cyber Security
Logins
tp
company
exam
Blog
portfolio ideas
piracy
Github repo.
resourceful
Activity: Analyze net...
All Bookmarks
HACKTIFY
cybersecurity
Happy Hacking
Your Login name:batman
Your Password:mob!le
© Copyrights 2021 Hacktify Cybersecurity All rights reserved