TASK-1:

1. TITLE OF VULNERABILITY:

Cross-Site Scripting (XSS) Vulnerability

2. CVSS SCORE:

7.1 (High)

3. RELATE TO OWASP TOP 10:

A03:2021 - Injection

4. DESCRIPTION:

Cross-Site Scripting (XSS) is a vulnerability that allows attackers to inject malicious scripts into a trusted web application. These scripts can execute in a victim's browser, leading to various attacks such as data theft, session hijacking, or malware distribution. XSS can be classified into three main types:

Reflected XSS: The malicious payload is reflected off the server in the immediate response.

Stored XSS: The malicious payload is permanently stored on the server and delivered to victims whenever the stored data is viewed.

DOM-Based XSS: The vulnerability arises on the client side, where JavaScript processes unsanitized input to modify the DOM.

5. DETAILED EXPLANATION:

a) Reflected XSS:

Occurs when user-supplied data is included in the server's response without proper validation or encoding. For example:

http://example.com/search?q=<script>alert('XSS')</script>

b) Stored XSS:

Occurs when malicious input is saved on the server and presented to other users later. For example, posting a comment with the payload:

<script>alert('XSS');</script>

c) DOM-Based XSS:

Occurs when JavaScript dynamically updates the DOM using unsanitized user input. For example:

document.write(location.hash);

Visiting a URL like http://example.com#<script>alert('XSS')</script> can trigger the attack.

6. IMPACT:

Confidentiality: Theft of sensitive user data (e.g., cookies, credentials).

Integrity: Injection of malicious scripts that modify page behaviour.

Availability: Potential disruption of legitimate user activities.

Reputation Damage: Loss of user trust in the application.

7. RECOMMENDATIONS:

Input Validation: Validate and sanitize all user inputs. Reject unexpected characters and enforce strict input types.

Output Encoding: Encode user inputs before rendering them on the web page (e.g., HTML, JavaScript, or URL encoding).

Content Security Policy (CSP): Use CSP to restrict the execution of inline scripts.

Avoid Dangerous Functions: Refrain from using functions like eval(), document.write(), and innerHTML.

Testing and Monitoring: Regularly test the application for XSS vulnerabilities using automated scanners and manual reviews. Monitor for unusual activities that may indicate exploitation.

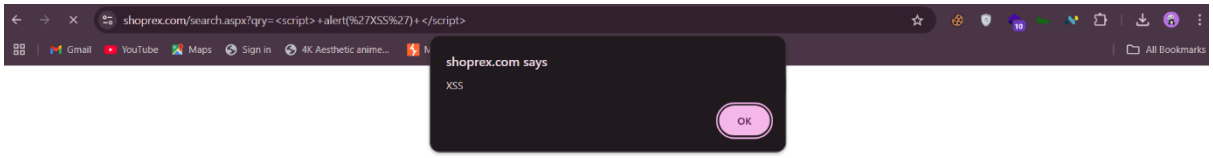
7. STEP BY STEP PROCEDURE:

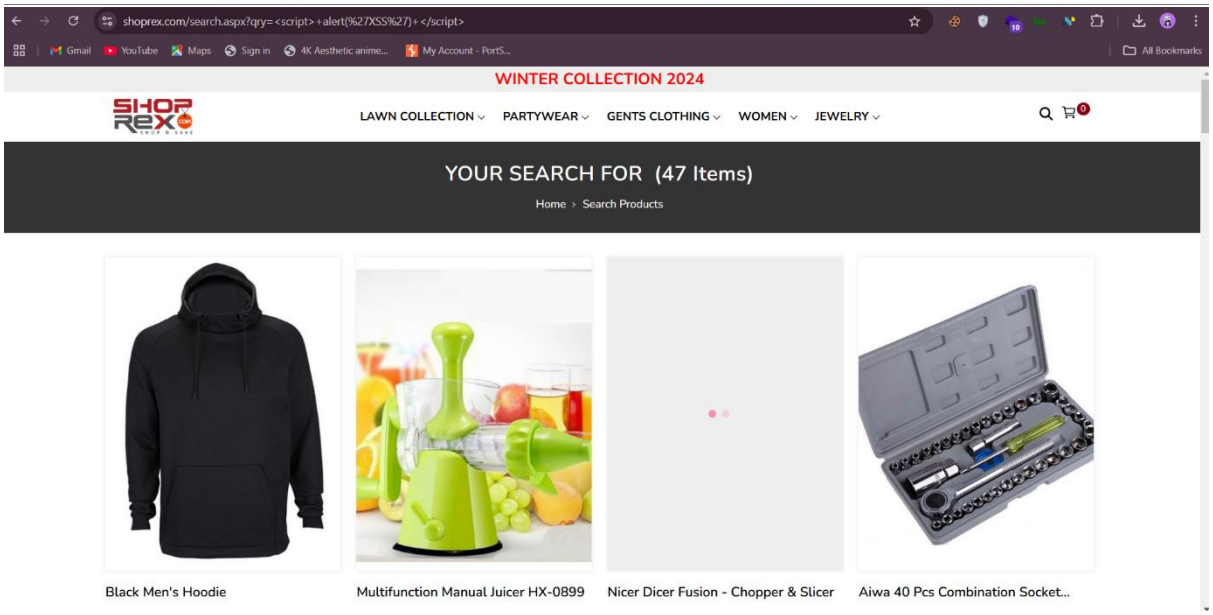
1. Now we get the website by the check it was XSS or not.

2. After injecting the malicious Script it can reveal the malicious.

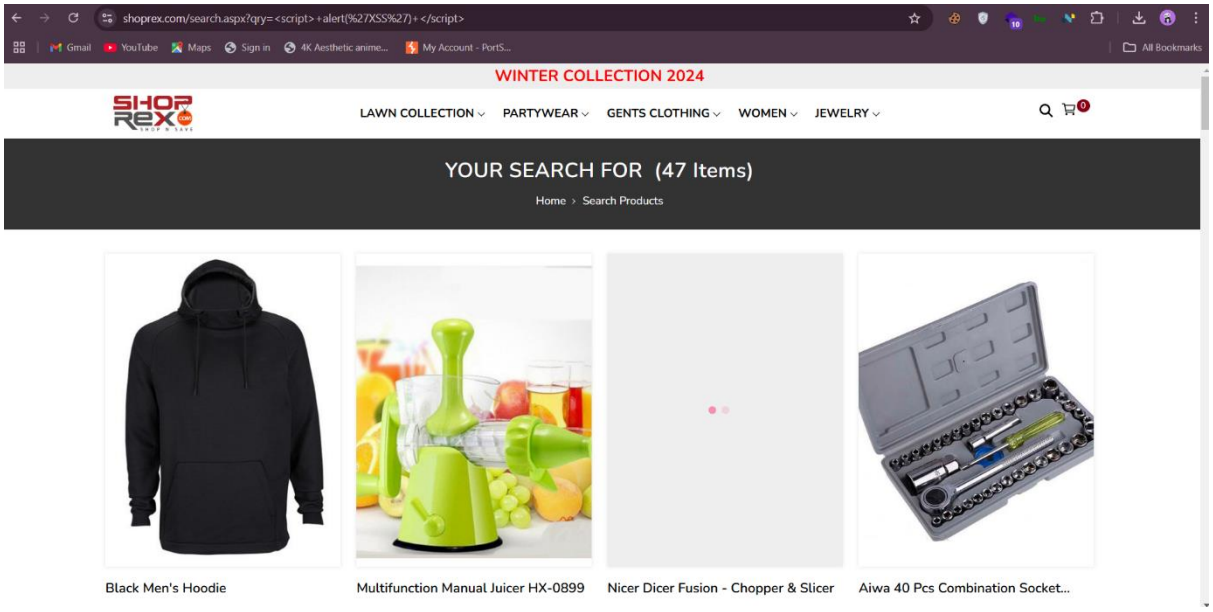
3. Now we done the procedure by pasting script in the input fields of the target website.

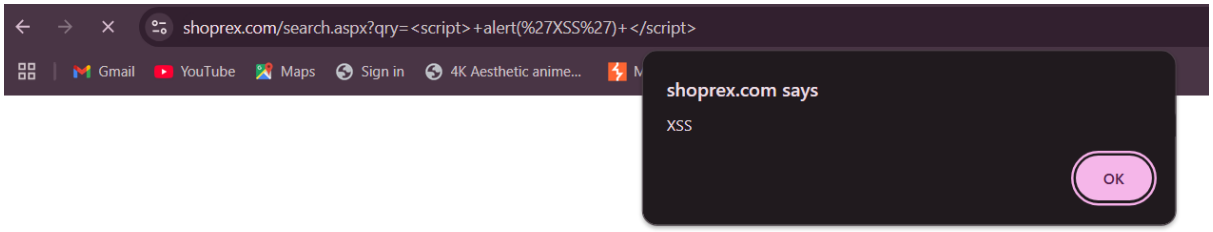
1. Reflected XSS:


shoprex.com/search.aspx?qry=<script>+alert(%27XSS%27)+</script>
Gmail
YouTube
Maps
Sign in
4K Aesthetic anime...
All Bookmarks
shoprex.com says
XSS
OK

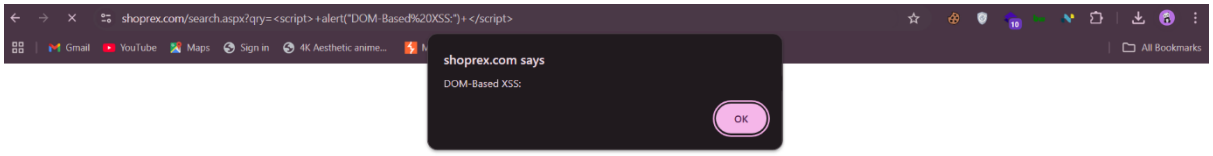

shoprex.com/search.aspx?qry=<script>+alert(%27XSS%27)+</script>
Gmail
YouTube
Maps
Sign in
4K Aesthetic anime...
My Account - PortS...
All Bookmarks
WINTER COLLECTION 2024
SHOP REX
LAWN COLLECTION
PARTYWEAR
GENTS CLOTHING
WOMEN
JEWELRY
YOUR SEARCH FOR (47 Items)
Home › Search Products
Black Men's Hoodie
Multifunction Manual Juicer HX-0899
Nicer Dicer Fusion - Chopper & Slicer
Aiwa 40 Pcs Combination Socket...

2. Stored XSS:





3. DOM BASED XSS:



TASK-2:

1. TITLE OF VULNERABILITY:

Insecure Direct Object Reference (IDOR) Vulnerability

2. CVSS SCORE:

7.5 (High)

3. RELATE TO OWASP TOP 10:

A01:2021 - Broken Access Control

4. DESCRIPTION:

IDOR (Insecure Direct Object Reference) occurs when an application exposes internal objects (e.g., database entries, files, or user records) directly in a URL or request parameter without adequate access controls. Attackers can manipulate these references to access unauthorized resources.

5. DETAILED EXPLANATION:

IDOR vulnerabilities often arise from improper authorization checks. Consider a URL that provides access to user profiles:

http://example.com/user/profile?user_id=123

If an attacker modifies user_id=123 to user_id=124, they may gain unauthorized access to another user's profile.

Other examples include:

Accessing or downloading another user’s files by altering a file ID.

Viewing order details or transaction history belonging to other users.

6. IMPACT:

Confidentiality: Unauthorized access to sensitive information (e.g., personal data, financial records).

Integrity: Potential for unauthorized modifications to data.

Reputation Damage: Loss of user trust due to data breaches.

7. RECOMMENDATIONS:

Implement Proper Access Controls: Validate the user's permissions before processing object references.

Use Indirect References: Replace direct object identifiers (e.g., user_id=123) with indirect references like tokens or hashed IDs.

Apply Role-Based Access Controls (RBAC): Ensure actions and resources are restricted based on user roles and permissions.

Secure APIs: Enforce strict authentication and authorization mechanisms in APIs.

Testing and Monitoring: Conduct regular security assessments to identify IDOR vulnerabilities and monitor for unusual activities that might indicate exploitation.

9. STEP BY STEP PROCEDURE:

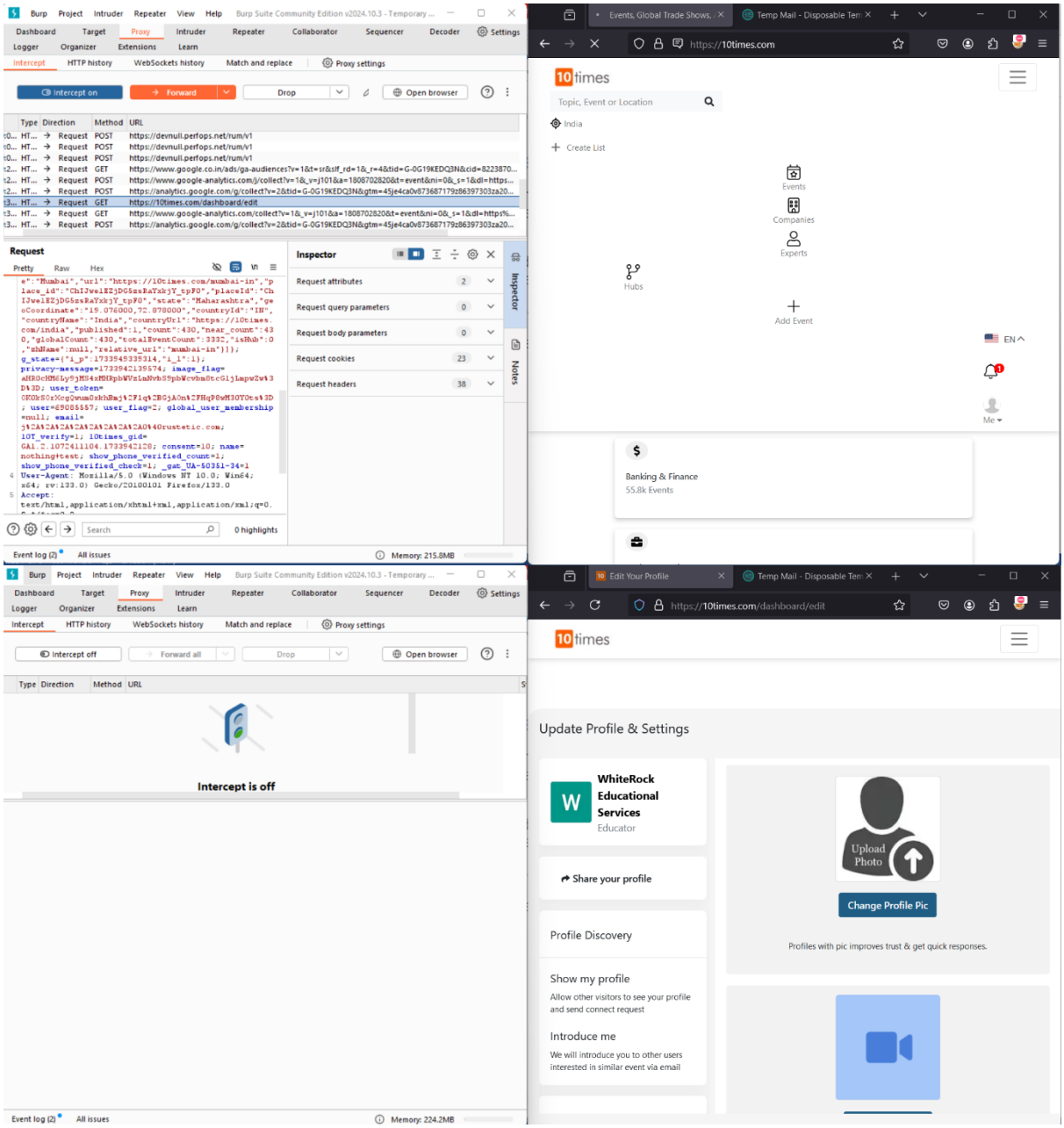
1. Here we done the IDOR vulnerability on the website here we done the test on Pakistan website.

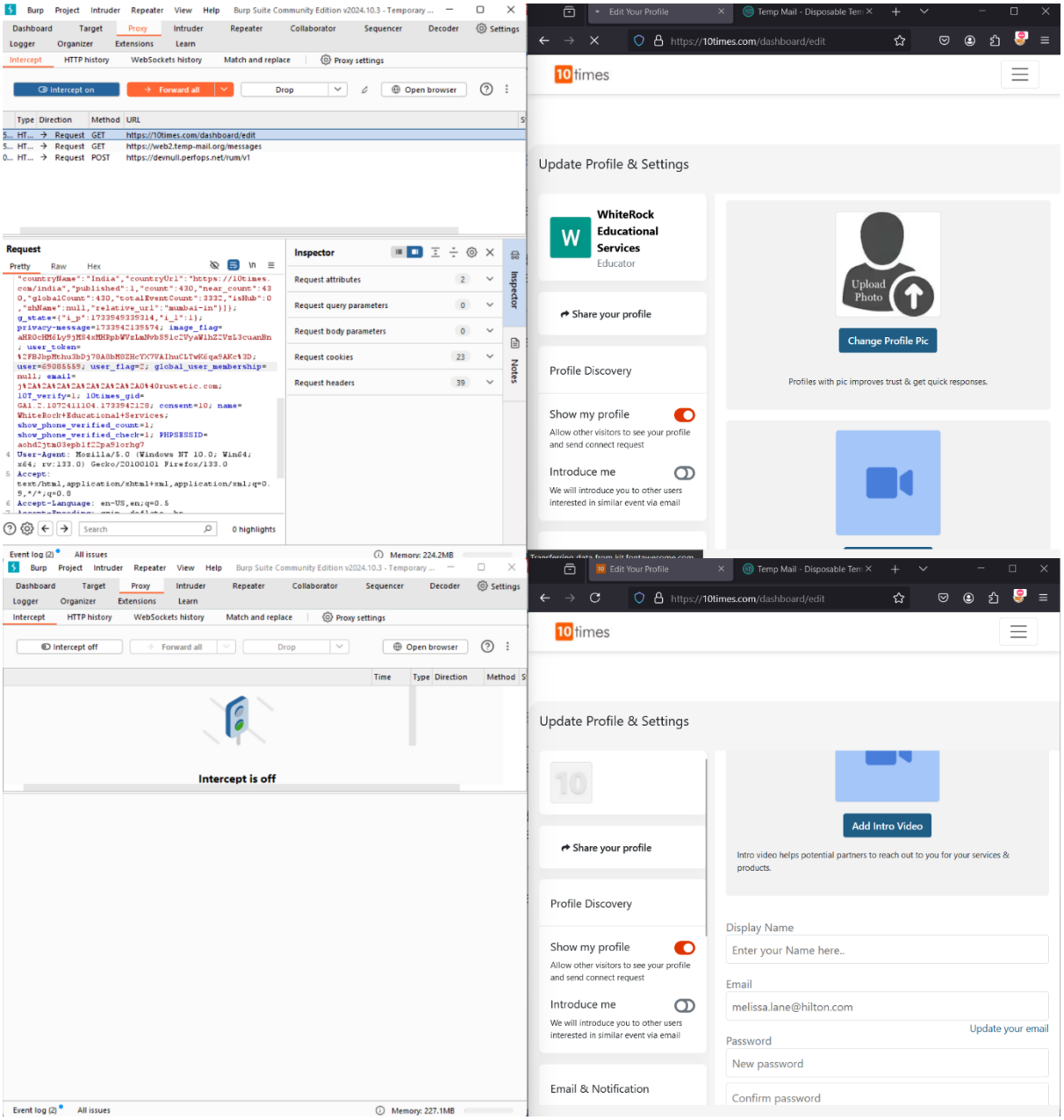
2. Here we go to the website and change the user id to another alternative number so that it will display the vulnerability in that which we can access another user details.

3. This can affect the once confidentiality.

4. Here we done the attack by the burp suite process.

10. Proof of content:


Burp Project Intruder Repeater View Help Burp Suite Community Edition v2024.10.3 - Temporary ...
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Settings
Logger Organizer Extensions Learn
Intercept HTTP history WebSockets history Match and replace Proxy settings
Intercept on Forward Drop Open browser
Type Direction Method URL
Request
Pretty Raw Hex
Inspector
Request attributes 2
Request query parameters 0
Request body parameters 0
Request cookies 23
Request headers 38
Inspector
Notes
Search 0 highlights
Event log (2) All issues Memory: 215.8MB
Events, Global Trade Shows, ... Temp Mail - Disposable Tem...
https://10times.com
10times
Topic, Event or Location
India
Create List
Events
Companies
Experts
Hubs
Add Event
EN
Me
Banking & Finance
55.8k Events
Burp Project Intruder Repeater View Help Burp Suite Community Edition v2024.10.3 - Temporary ...
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Settings
Logger Organizer Extensions Learn
Intercept HTTP history WebSockets history Match and replace Proxy settings
Intercept off Forward all Drop Open browser
Type Direction Method URL
Intercept is off
Event log (2) All issues Memory: 224.2MB
Edit Your Profile Temp Mail - Disposable Tem...
https://10times.com/dashboard/edit
10times
Update Profile & Settings
WhiteRock Educational Services
Educator
Share your profile
Profile Discovery
Show my profile
Allow other visitors to see your profile and send connect request
Introduce me
We will introduce you to other users interested in similar event via email
Upload Photo
Change Profile Pic
Profiles with pic improves trust & get quick responses.


Burp Project Intruder Repeater View Help Burp Suite Community Edition v2024.10.3 - Temporary ...
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Settings
Logger Organizer Extensions Learn
Intercept HTTP history WebSockets history Match and replace Proxy settings
Intercept on Forward all Drop Open browser
Type Direction Method URL
Request GET https://10times.com/dashboard/edit
Request GET https://web2.temp-mail.org/messages
Request POST https://devnull.perfops.net/rum/v1
Request
Pretty Raw Hex
Inspector
Request attributes
Request query parameters
Request body parameters
Request cookies
Request headers
Inspector
Notes
Search
0 highlights
Event log (2) All issues
Memory: 224.2MB
Intercept off Forward all Drop Open browser
Time Type Direction Method
Intercept is off
Event log (2) All issues
Memory: 227.1MB
Edit Your Profile
Temp Mail - Disposable Tem
https://10times.com/dashboard/edit
10times
Update Profile & Settings
WhiteRock Educational Services
Educator
Share your profile
Profile Discovery
Show my profile
Allow other visitors to see your profile and send connect request
Introduce me
We will introduce you to other users interested in similar event via email
Upload Photo
Change Profile Pic
Profiles with pic improves trust & get quick responses.
Add Intro Video
Intro video helps potential partners to reach out to you for your services & products.
Display Name
Enter your Name here..
Email
melissa.lane@hilton.com
Update your email
Password
New password
Confirm password
Email & Notification

TASK-3:

1. TITLE OF VULNERABILITY:

Broken Access Control

2. CVSS SCORE:

8.7 (High)

3. RELATE TO OWASP TOP 10:

A01:2024 - Broken Access Control

4. DESCRIPTION:

Broken Access Control occurs when an application fails to enforce proper restrictions on user access to resources or functions. This vulnerability allows attackers to bypass security mechanisms and access sensitive data, modify resources, or perform unauthorized actions.

5. DETAILED EXPLANATION:

Access control issues can arise in various forms, such as:

Horizontal Privilege Escalation: Accessing another user's data or actions (e.g., viewing another user's account details by modifying a user ID in the URL).

Vertical Privilege Escalation: Gaining higher privileges (e.g., a regular user performing admin-level actions).

Access to Restricted Endpoints: Directly accessing sensitive API endpoints or functions that should be restricted.

For example, if a regular user can access an admin-only page via:

http://example.com/admin/settings

Without proper authentication, it indicates broken access control.

6. IMPACT:

Confidentiality: Unauthorized access to sensitive data.

Integrity: Unauthorized modifications to critical resources.

Availability: Potential disruption of services or data deletion.

Reputation Damage: Loss of user trust due to compromised security.

7. RECOMMENDATIONS:

Enforce Least Privilege: Ensure users only have access to resources and actions necessary for their roles.

Implement Role-Based Access Control (RBAC): Restrict access based on defined roles and permissions.

Deny by Default: Restrict access to resources unless explicitly allowed.

Secure Sensitive Endpoints: Protect admin panels and critical API endpoints with strong authentication and authorization checks.

Regular Testing: Conduct security assessments to identify and remediate access control flaws.

Audit Logs: Monitor and log access attempts to detect unauthorized activities.

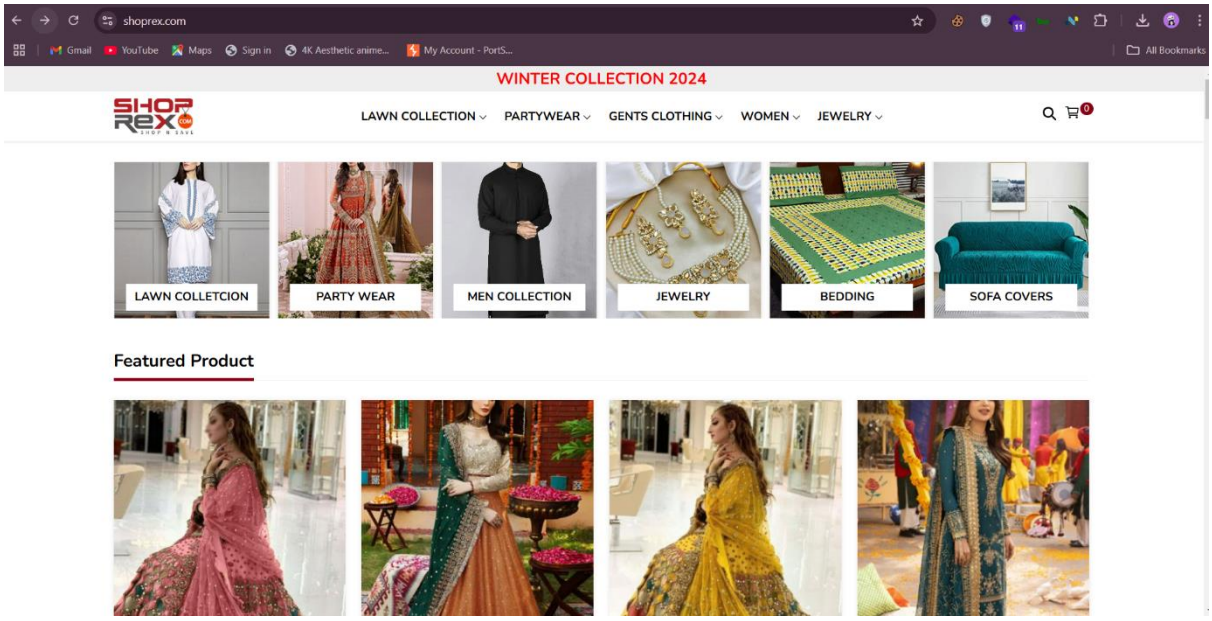
8. STEP BY STEP PROCEDURE:

1. Now we search website in the internet having broken access control vulnerability

2. Now we select the target and test for this vulnerability by placing malicious payloads to access the files which can private and can’t see by public users.

3. By this we can able to access the system configurations by add file names in the urls.

9. Proof of content:



Here by entering the file names in url it can deploy the details in the given website.

So it is broken access control vulnerability.