

AI FOR CYBERSECURITY WITH IBM QRADAR

ASSIGNMENT 1

SUBMITTED BY : GAURI SHARMA

DATE: 30/08/23

1— 1.1 . **Vulnerability Name:** Cross-Site Scripting (Stored)

CWE : CWE-79

OWASP Category: A03:2021 – Injection

Description: Untrusted data enters a web application, typically from a web request

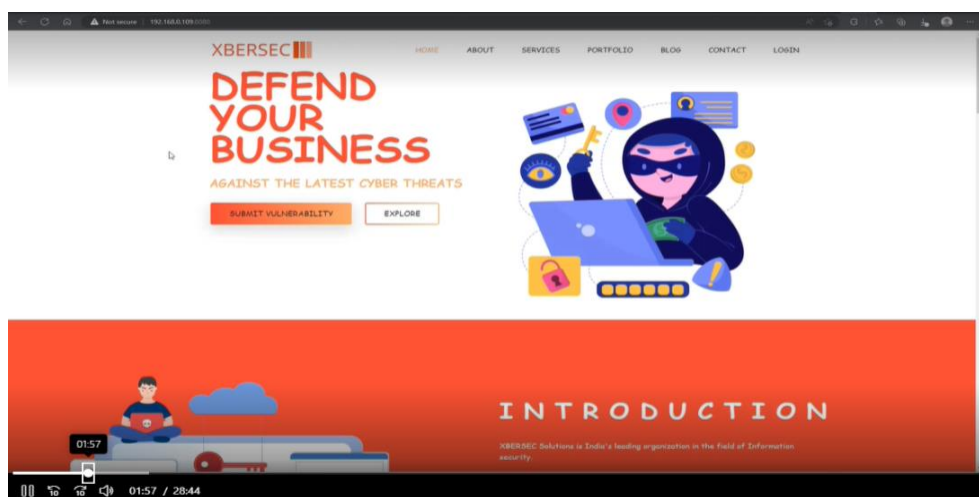
Business Impact: The application stores dangerous data in a database, message forum, visitor log, or other trusted data store. At a later time, the dangerous data is subsequently read back into the application and included in dynamic content. From an attacker's perspective, the optimal place to inject malicious content is in an area that is displayed to either many users or particularly interesting users. Interesting users typically have elevated privileges in the application or interact with sensitive data that is valuable to the attacker. If one of these users executes malicious content, the attacker may be able to perform privileged operations on behalf of the user or gain access to sensitive data belonging to the user. For example, the attacker might inject XSS into a log message, which might not be handled properly when an administrator views the logs.

Vulnerability Path : <http://192.168.0.109:8080/>

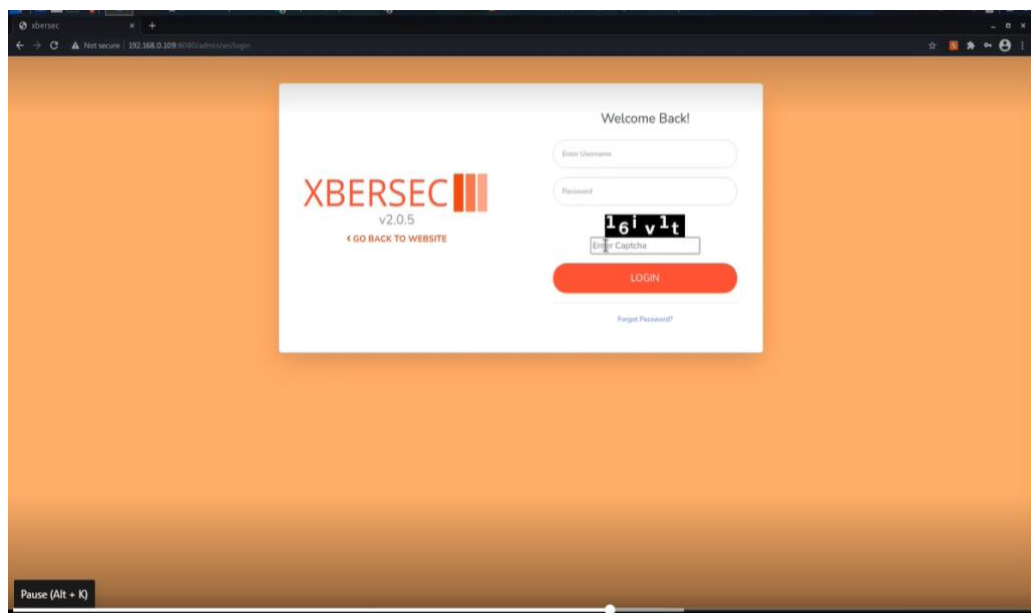
Vulnerability Parameter: <http://192.168.0.109:8080/admin/en/blog>

Steps to Reproduce :

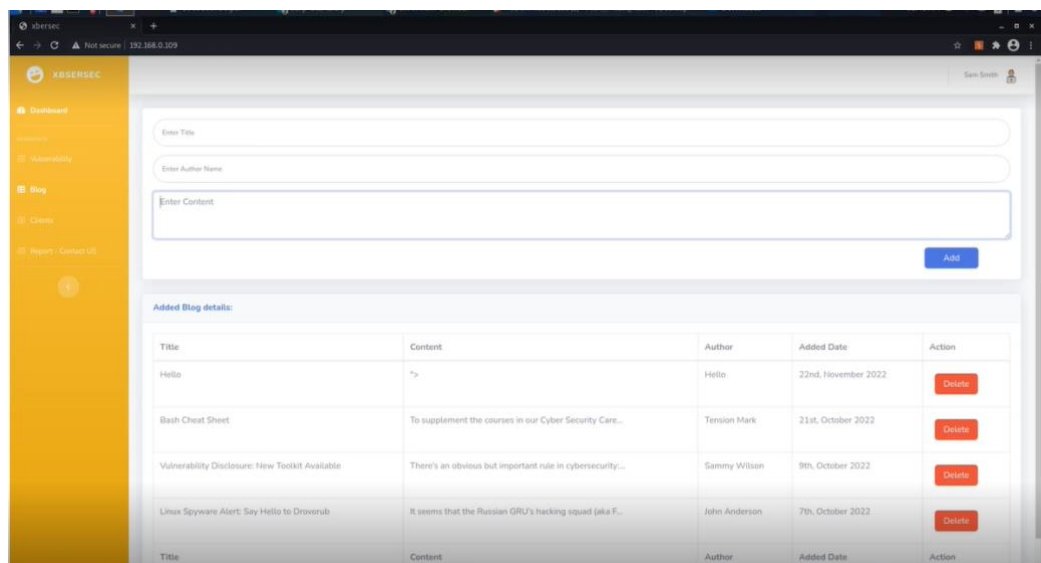
Step 1. Access the URL



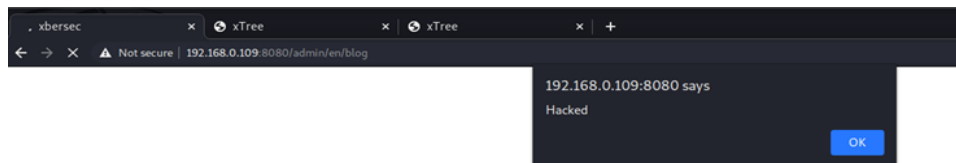
Step 2: Go to the login page and enter credentials



Step 3: Now you will be redirected to the dashboard where we will enter the script.



Step 3:-after entering the script content like" hacked" u will find the dialogue box as shown below.

**Recommendation:**

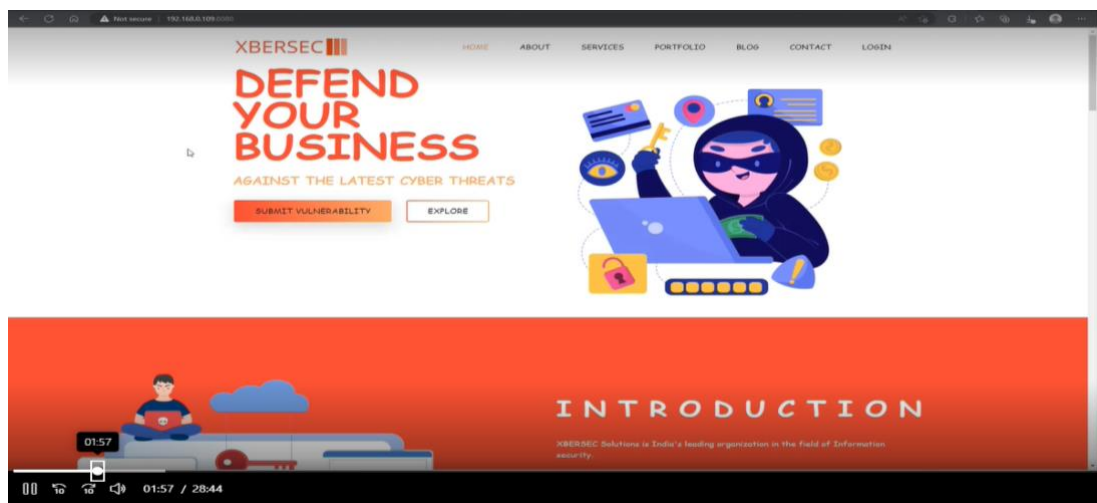
- Note that proper output encoding, escaping, and quoting is the most effective solution for preventing XSS, although input validation may provide some defense-in-depth.

1— 1.2 . Vulnerability Name: Cross-Site Scripting (Stored)**CWE :** CWE-79**OWASP Category:** A03:2021 – Injection**Description:** Untrusted data enters a web application, typically from a web request

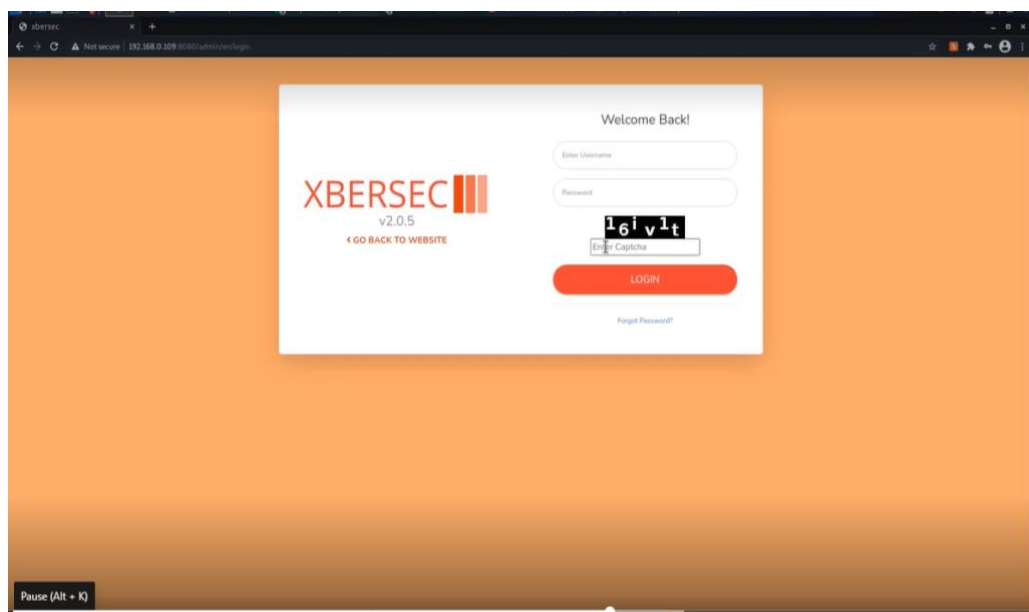
Business Impact: The application stores dangerous data in a database, message forum, visitor log, or other trusted data store. At a later time, the dangerous data is subsequently read back into the application and included in dynamic content. From an attacker's perspective, the optimal place to inject malicious content is in an area that is displayed to either many users or particularly interesting users. Interesting users typically have elevated privileges in the application or interact with sensitive data that is valuable to the attacker. If one of these users executes malicious content, the attacker may be able to perform privileged operations on behalf of the user or gain access to sensitive data belonging to the user. For example, the attacker might inject XSS into a log message, which might not be handled properly when an administrator views the logs.

Vulnerability Path : <http://192.168.0.109:8080/>**Vulnerability Parameter:** http://192.168.0.109:8080/admin/en/report_contact_us**Steps to Reproduce :**

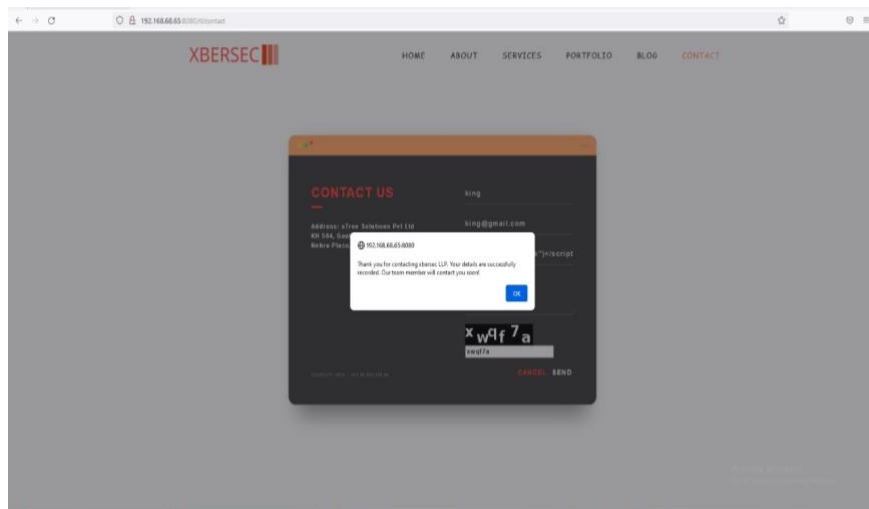
Step 1. Access the URL



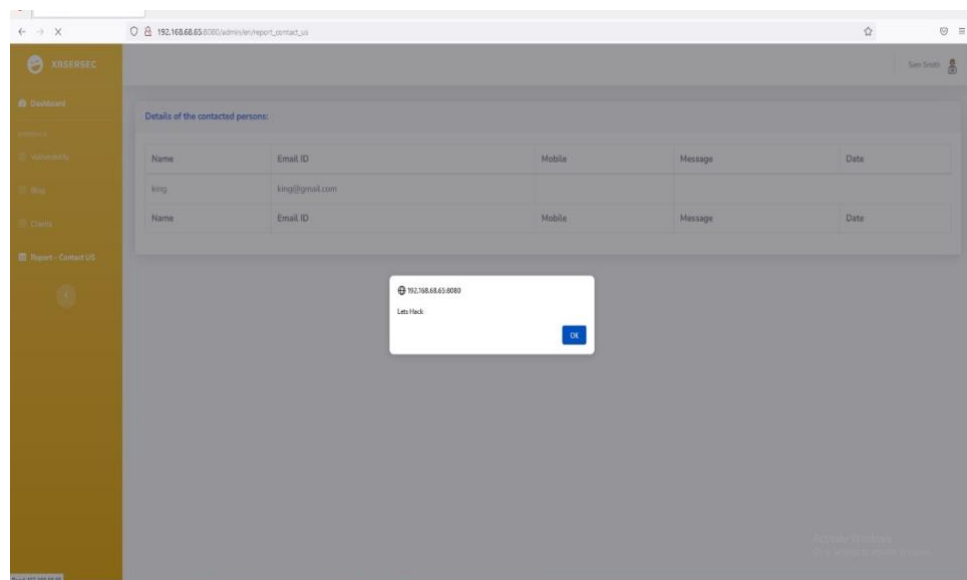
Step 2: Go to the login page and enter credentials



Step 3: Now you will be redirected to the dashboard where we enter the script in the contact_us page.



Step 4:- this is the pop up you get after you successfully inject the script in the contact page.



Recommendation:

- Note that proper output encoding, escaping, and quoting is the most effective solution for preventing XSS, although input validation may provide some defense-in-depth

2 .Vulnerability Name:No Session Management

CWE : CWE-384

OWASP Category:A07:2021 –Identification and Authentication Failures

Description:An attacker is able to force a known session identifier on a user so that, once the user authenticates, the attacker has access to the authenticated session.

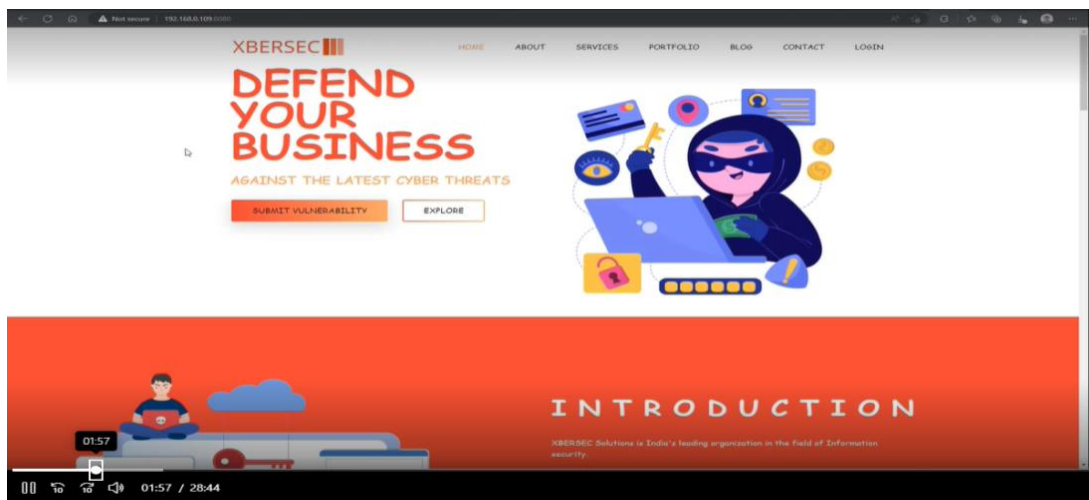
Business Impact: Without appropriate session management, you can run into several security problems, putting your users at risk. Common vulnerabilities caused by a lack of or poorly implemented session management include: Session hijacking

Vulnerability Path : <http://192.168.0.109:8080/>

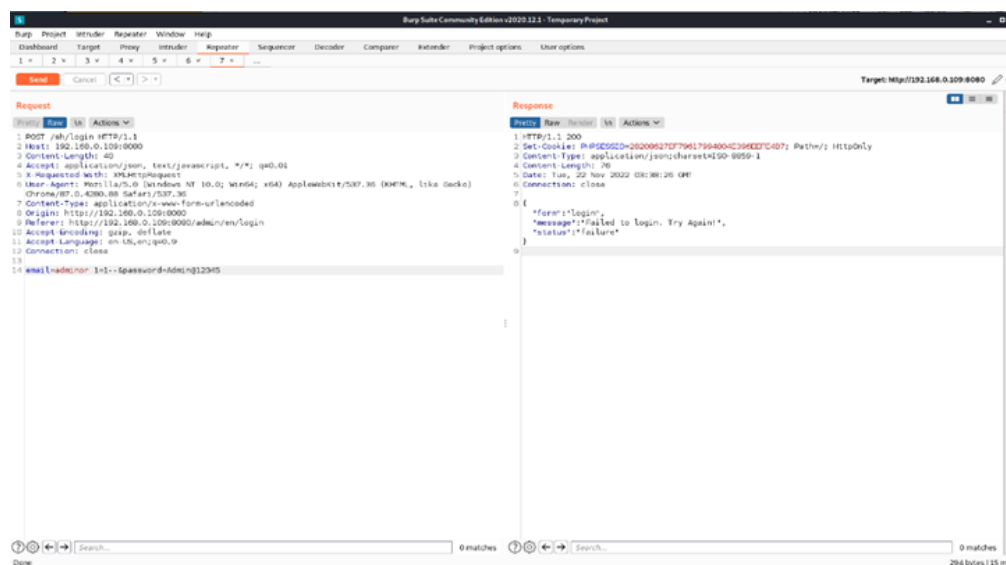
Vulnerability Parameter: <http://192.168.0.100:8080/sh/admclient>

Steps to Reproduce :

Step 1. Access the URL



Step 2. without the proper session management the burp can still access the request of session as shown.



Recommendation:

- Invalidate any existing session identifiers prior to authorizing a new user session.

3 . Vulnerability Name:Login Captcha Bypass

CWE : CWE-804

OWASP Category:A06:2021-Vulnerable and Outdated Components

Description:An automated attacker could bypass the intended protection of the CAPTCHA challenge and perform actions at a higher frequency than humanly possible, such as launching spam attacks.

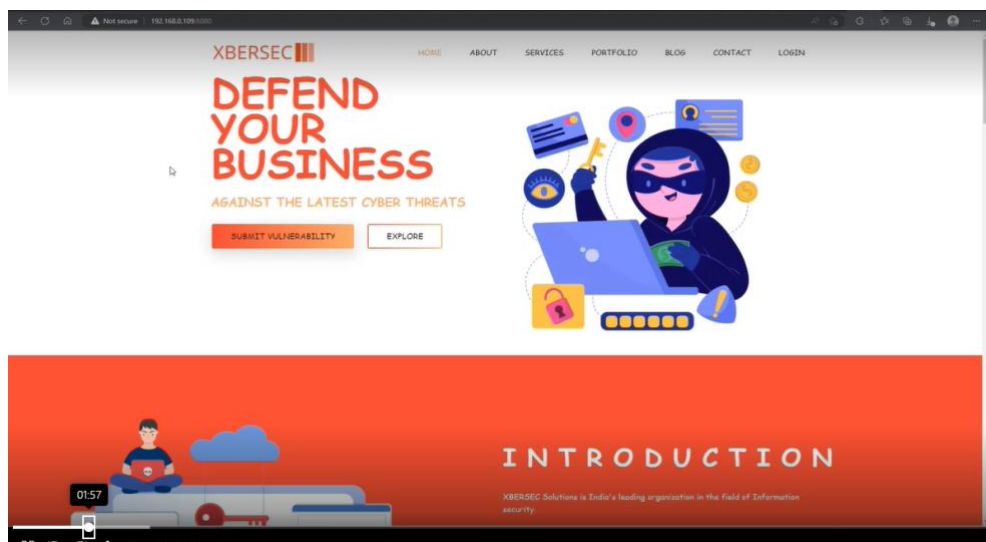
Business Impact:When authorization, authentication, or another protection mechanism relies on CAPTCHA entities to ensure that only human actors can access certain functionality, then an automated attacker such as a bot may access the restricted functionality by guessing the CAPTCHA.

Vulnerability Path :<http://192.168.0.109:8080/>

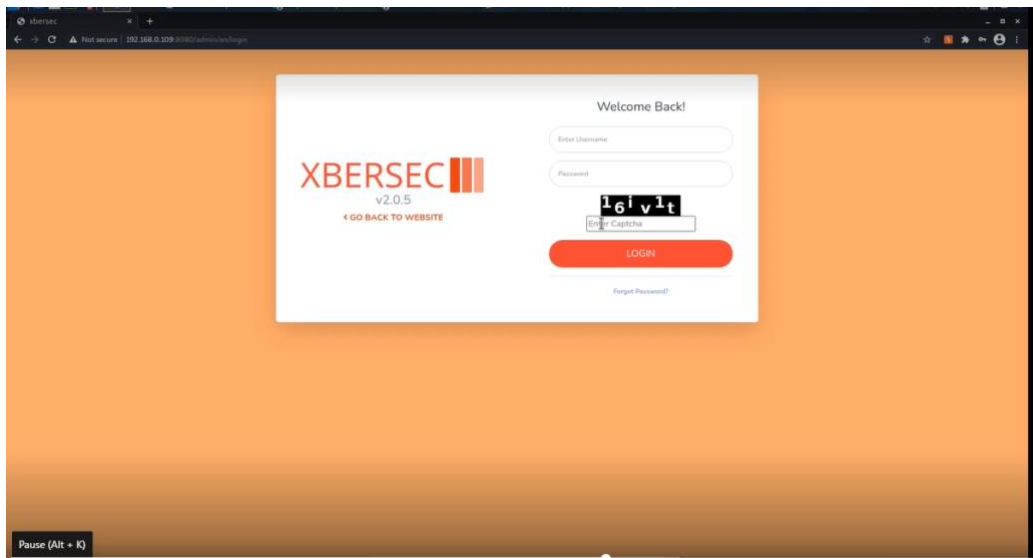
Vulnerability Parameter:<http://192.168.0.109:8080/sh/login>

Steps to Reproduce :

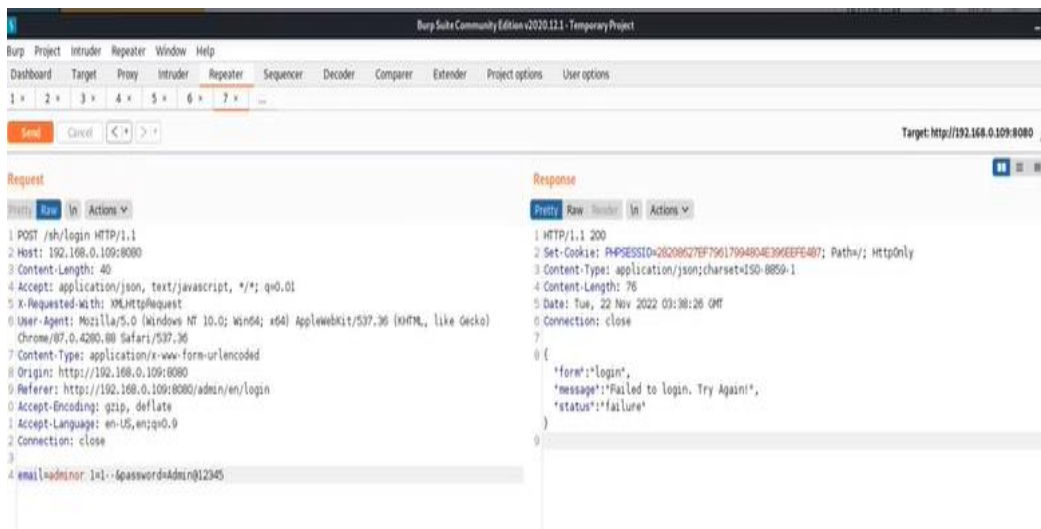
Step 1. Access the URL



Step :- in the backend this login page is by default giving the access without the captcha.



Step 3:- these the backend in the burp where you can see that without the captcha validation it is given the access to session.



Recommendation:

Ensure that the CAPTCHA value stored in the session is verified against the user's input and is removed when the request is submitted.

4 . Vulnerability Name:Admin Login SQL Injection

CWE : CWE-284

OWASP Category:A03:2021-Injections

Description: Without sufficient removal or quoting of SQL syntax in user-controllable inputs, the generated SQL query can cause those inputs to be interpreted as SQL instead of ordinary user data. This can be used to alter query logic to bypass security checks, or to insert additional statements that modify the back-end database, possibly including execution of system commands.

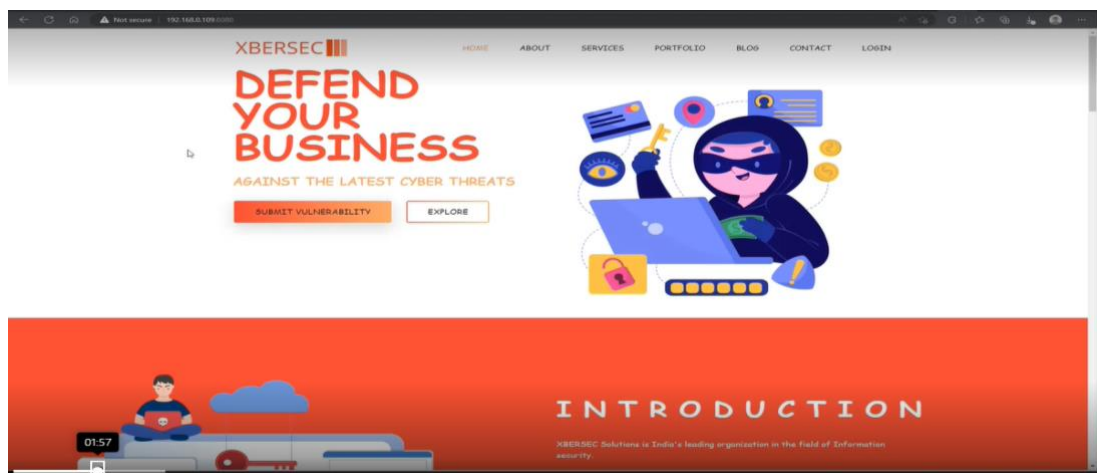
Business Impact: If authorization information is held in a SQL database, it may be possible to change this information through the successful exploitation of a SQL injection vulnerability

Vulnerability Path : <http://192.168.0.109:8080/>

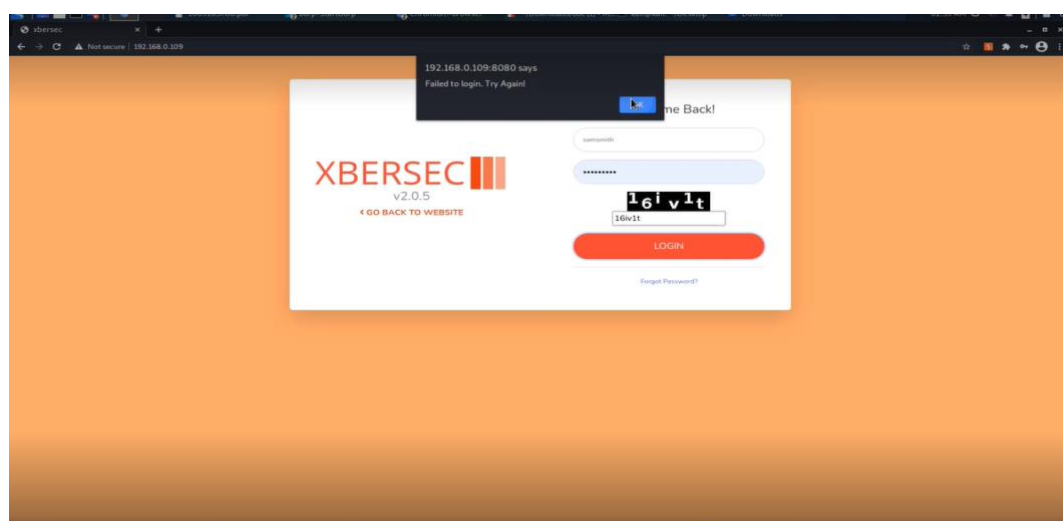
Vulnerability Parameter: <http://192.168.0.109:8080/sh/login>

Steps to Reproduce :

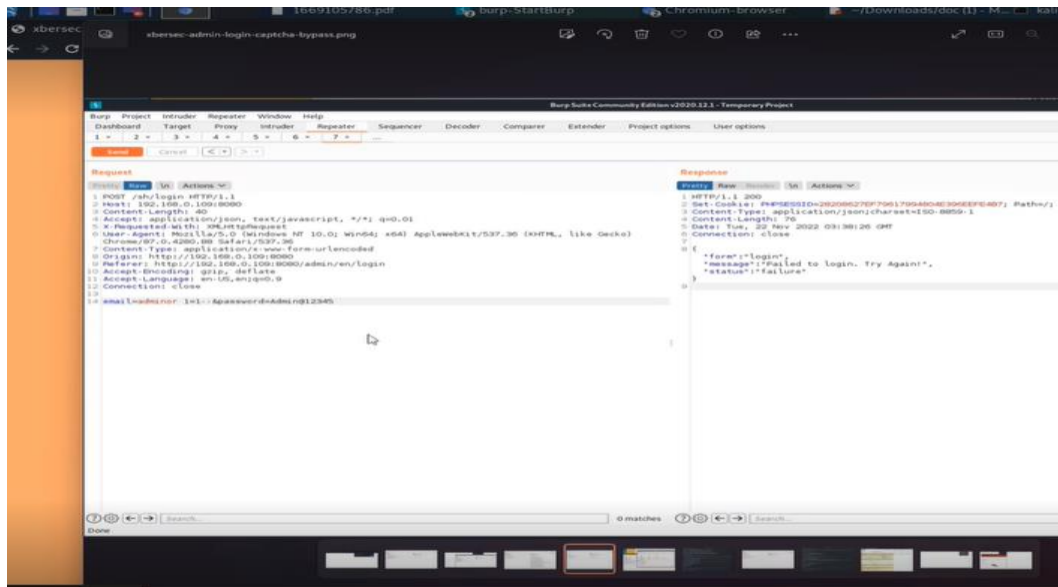
Step 1. Access the URL



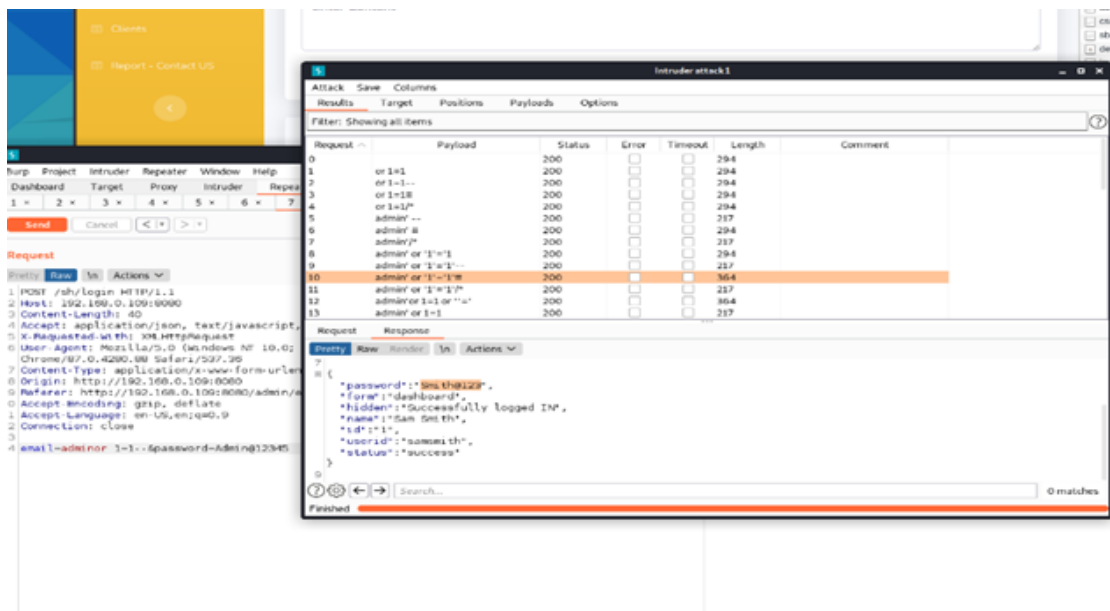
Step 2:- Enter the login credentials in and try to validate as shown below.



Step 3:- in the backend we need to use the burp to configure the user with sql injection.



Step 4: - this is the place where we upload injection then you will gain access to one of the user accounts.



Recommendation:

Employ a layered approach to security that includes utilizing parameterized queries when accepting user input, ensuring that only expected data (white listing) is accepted by an application, and harden the database server to prevent data from being accessed inappropriately.

5 . Vulnerability Name:Improper access control

CWE : CWE-284

OWASP Category:A01:2021 – Broken Access Control

Description:The SQL Injection occurs when user-controllable input is interpreted as a SQL command, rather than as normal data, by the backend database.Exploitation of this vulnerability can have critical implications, including creation, modification, or exfiltration of database content.

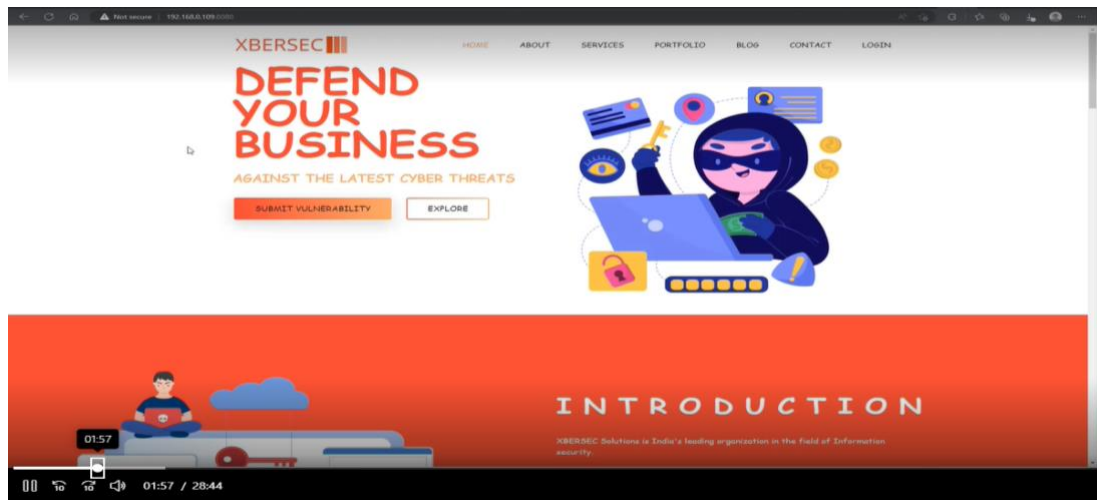
Business Impact:It may be possible to read sensitive information, it is also possible to make changes or even delete this information with a SQL injection attack.

Vulnerability Path :<http://192.168.0.109:8080/>

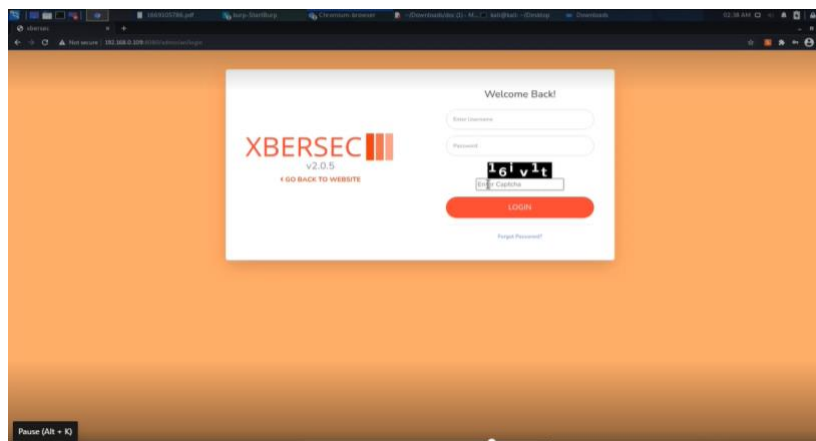
Vulnerability Parameter:<http://192.168.0.109:8080/admin/en/vulnerability>

Steps to Reproduce :

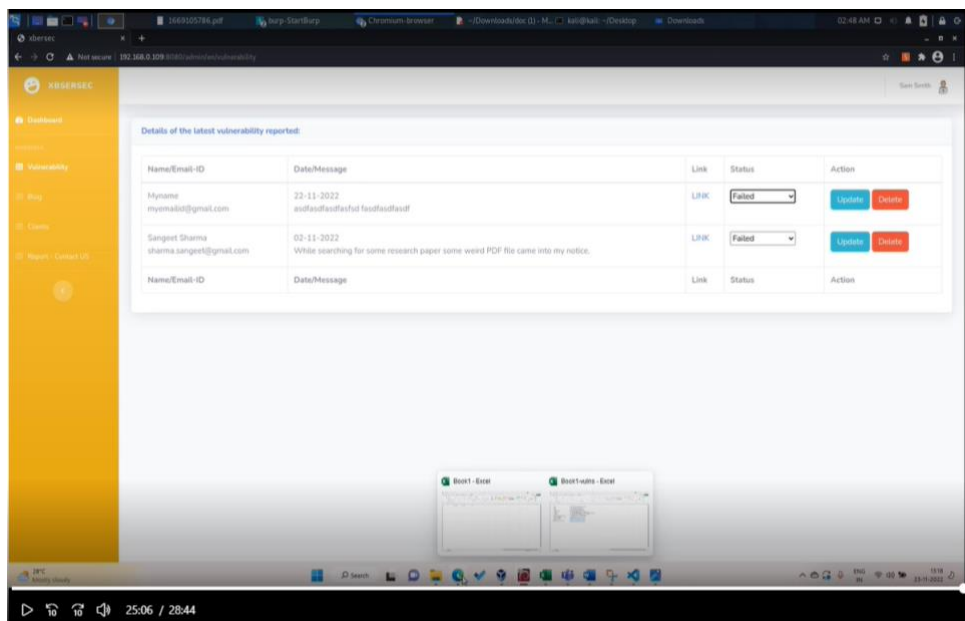
Step 1. Access the URL



Step 2:- Try to enter the login credentials



Step 3 :- After that with the default credentials assessed we can gain access to the database with entering the default credentials.



Step 4:- this is the map where status url remain unchanged

```
kali@kali: ~/Desktop
File Actions Edit View Help
[V...] [V...] http://sqlmap.org
Target: http://192.168.0.109:8080/

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility
to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage c
aused by this program

[*] starting @ 23:54:09 /2022-11-21/

[23:54:09] [INFO] parsing HTTP request from 'xbersec-request'
[23:54:09] [INFO] resuming back-end DBMS 'mysql'
[23:54:09] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: status (POST)
Type: time-based blind
Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
Payload: id=3&type=update&status=3' AND (SELECT 8544 FROM (SELECT(SLEEP(5)))zvJU) AND 'FNPv'='FNPv
---
[23:54:09] [INFO] the back-end DBMS is MySQL
back-end DBMS: MySQL >= 5.0.12
[23:54:09] [INFO] fetching database names
[23:54:09] [INFO] fetching number of databases
[23:54:09] [WARNING] time-based comparison requires larger statistical model, please wait..... (done)
do you want sqlmap to try to optimize value(s) for DBMS delay responses (option '--time-sec')? [Y/n]
[23:54:17] [WARNING] it is very important to not stress the network connection during usage of time-based payloads to prevent potential
disruptions
5
[23:54:22] [INFO] retrieved:
[23:54:27] [INFO] adjusting time delay to 1 second due to good response times
mysql
[23:54:43] [INFO] retrieved: information_schema
[23:55:42] [INFO] retrieved: performance_schema
[23:56:39] [INFO] retrieved: sys
[23:56:58] [INFO] retrieved: xtree
available databases [5]:
[*] information_schema
[*] mysql
[*] performance_schema
[*] sys
[*] xtree
[23:57:08] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/192.168.0.109'
[23:57:08] [WARNING] your sqlmap version is outdated

[*] ending @ 23:57:08 /2022-11-21/

(kali@kali)~[~/Desktop]
$
```

Recommendation:

- Applications should not incorporate any user-controllable data directly into SQL queries.
- Parameterized queries (also known as prepared statements) should be used to safely insert data into predefined queries.

6 . Vulnerability Name:Default Credentials

CWE : CWE-1392

OWASP Category:A07:2021-Identification and Authentication Failures

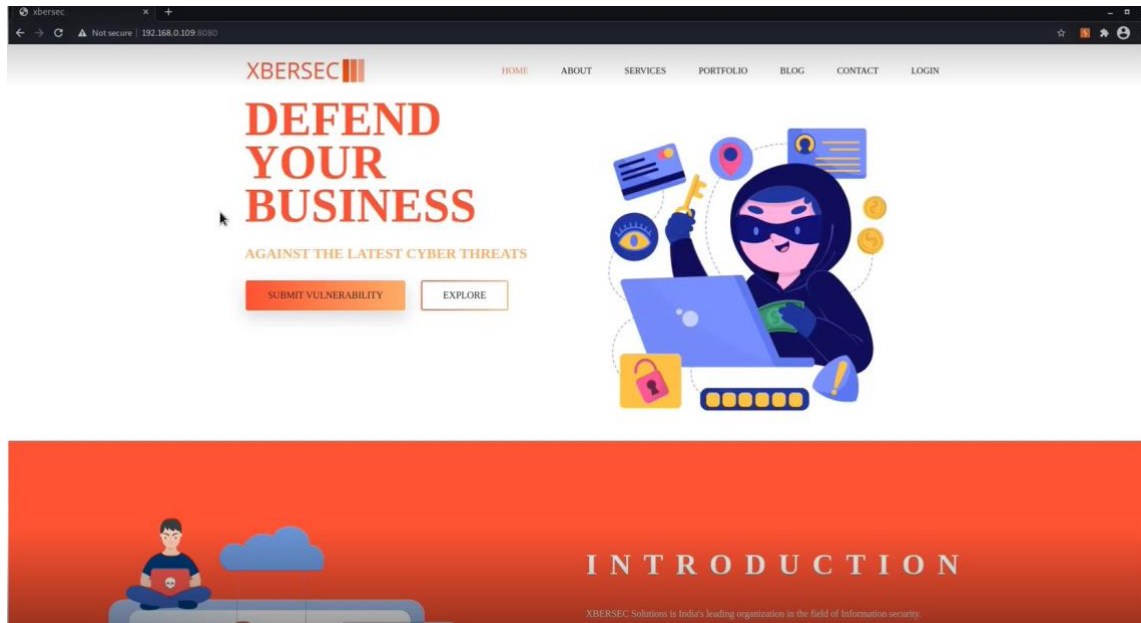
Description:It is common practice for products to be designed to use default keys, passwords, or other mechanisms for authentication. The rationale is to simplify the manufacturing process or the system administrator's task of installation and deployment into an enterprise. However, if admins do not change the defaults, it is easier for attackers to bypass authentication quickly across multiple organizations.

Business Impact:Attackers can easily obtain default passwords and identify internet-connected target systems. Passwords can be found in product documentation and compiled lists available on the internet.

Vulnerability Path :<http://192.168.0.109:8080/>

Vulnerability Parameter: <http://192.168.0.109:8080/manager/html>

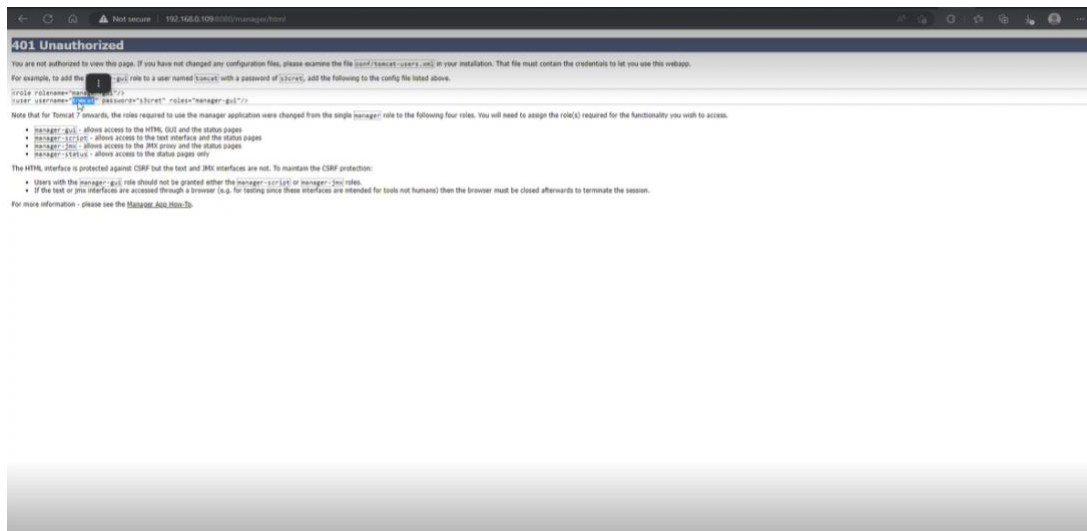
Steps to Reproduce : Step 1. Access the URL



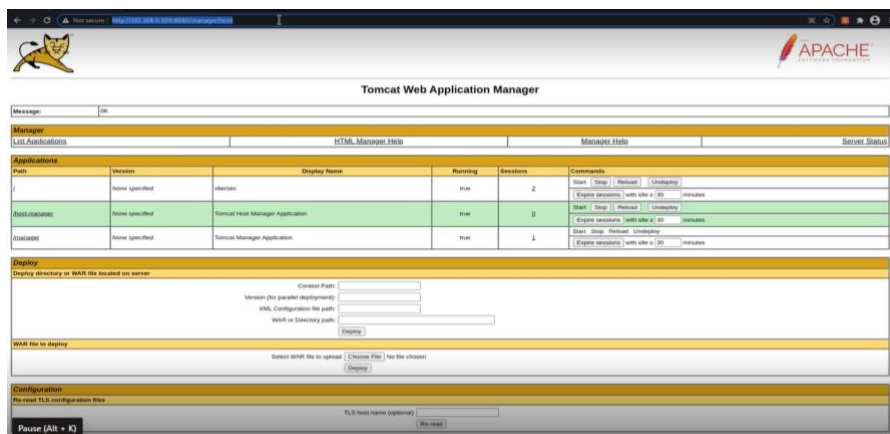
Step 2: By changing the URL parameters with tomcat configurations we find the below page.



Step 3:- By closing the dialog box without canceling it this will give you the access to the default credentials



Step4 :- Here in this step the default credentials are printed on the screen which helps to gain access to the DB.



7—7.1. Vulnerability Name:File Upload

CWE : CWE-434

OWASP Category:A03:2021 – Injections

Description:The software allows the attacker to upload or transfer files of dangerous types that can be automatically processed within the product's environment.

Business Impact:It includes complete system takeover, an overloaded file system or database, forwarding attacks to back-end systems, client-side attacks, or simple defacement. It depends on what the application does with the uploaded file and especially where it is stored.

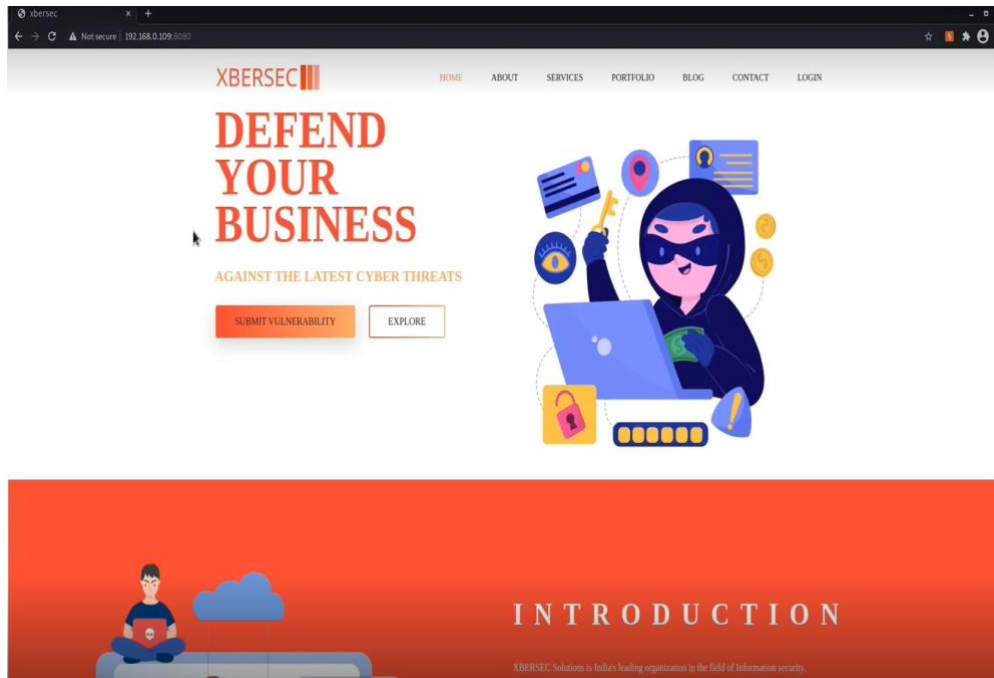
Vulnerability Path :<http://192.168.0.109:8080/>

Vulnerability Parameter:1 :-<http://192.168.0.109:8080/tl/vulnerability>

2 :-<http://192.168.0.109:8080/uploads/shell.jsp?cmd=ls>

Steps to Reproduce :

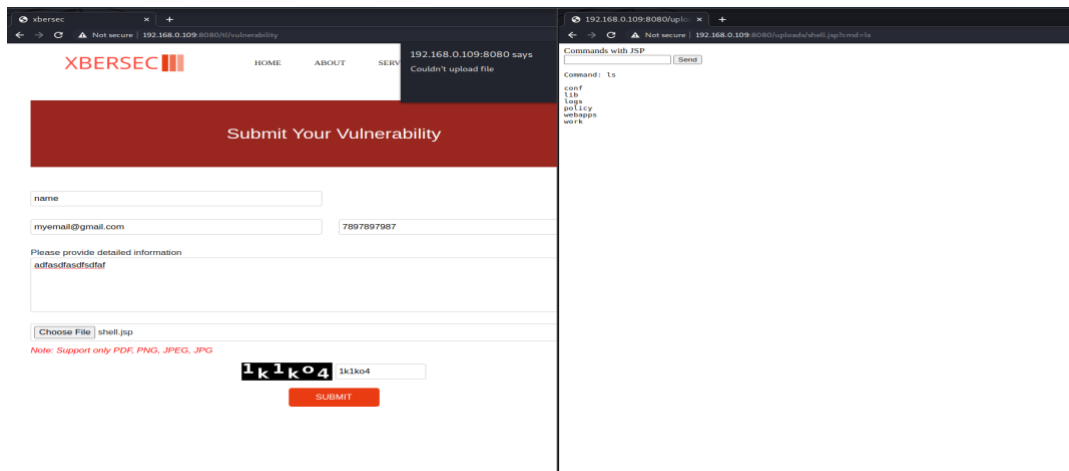
Step 1. Access the URL



Step2:- here it gives the access to upload unwanted malicious files which does not restrict .

A screenshot of the 'Submit Your Vulnerability' form on the XBERSEC website. The browser's address bar shows '192.168.0.109:8080/vulnerability'. The form is titled 'Submit Your Vulnerability' in white text on a dark red background. Below the title, there are input fields for 'Your Name', 'Enter Email ID', and 'Enter Mobile Number'. A section titled 'Please provide detailed information' contains a text area for 'Enter information regarding the issue'. Below this is a file upload section with a 'Choose File' button and the text 'No file chosen'. A note below the file upload section states 'Note: Support only PDF, PNG, JPEG, JPG'. At the bottom, there is a captcha image showing the characters '1 m f h s 1' and a text input field for 'Enter Captcha'. A red 'SUBMIT' button is located at the bottom right of the form.

Step3: this how you upload the file parameters in the URL and get the configurations as shown below:-



Recommendation:

- Segregate Your Uploads.
- Ensure Upload Files Cannot Be Executed.
- Validate File Formats and Extensions.

7-7 . 2 Vulnerability Name:File Upload

CWE : CWE-434

OWASP Category:A03:2021 – Injections

Description:The software allows the attacker to upload or transfer files of dangerous types that can be automatically processed within the product's environment.

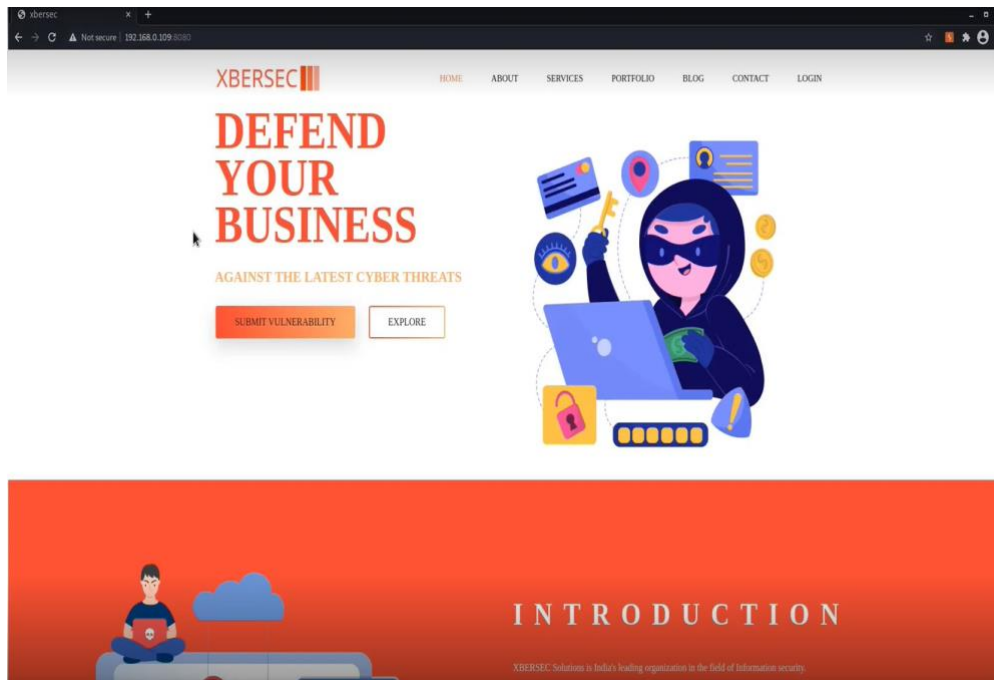
Business Impact:It includes complete system takeover, an overloaded file system or database, forwarding attacks to back-end systems, client-side attacks, or simple defacement. It depends on what the application does with the uploaded file and especially where it is stored.

Vulnerability Path :<http://192.168.0.109:8080/>

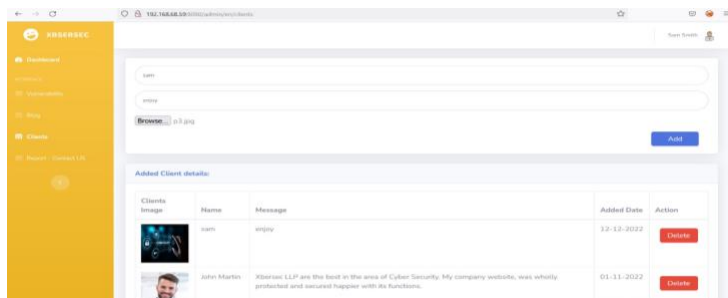
Vulnerability Parameter:1 :-<http://192.168.0.109:8080/admin/en/clients>

Steps to Reproduce :

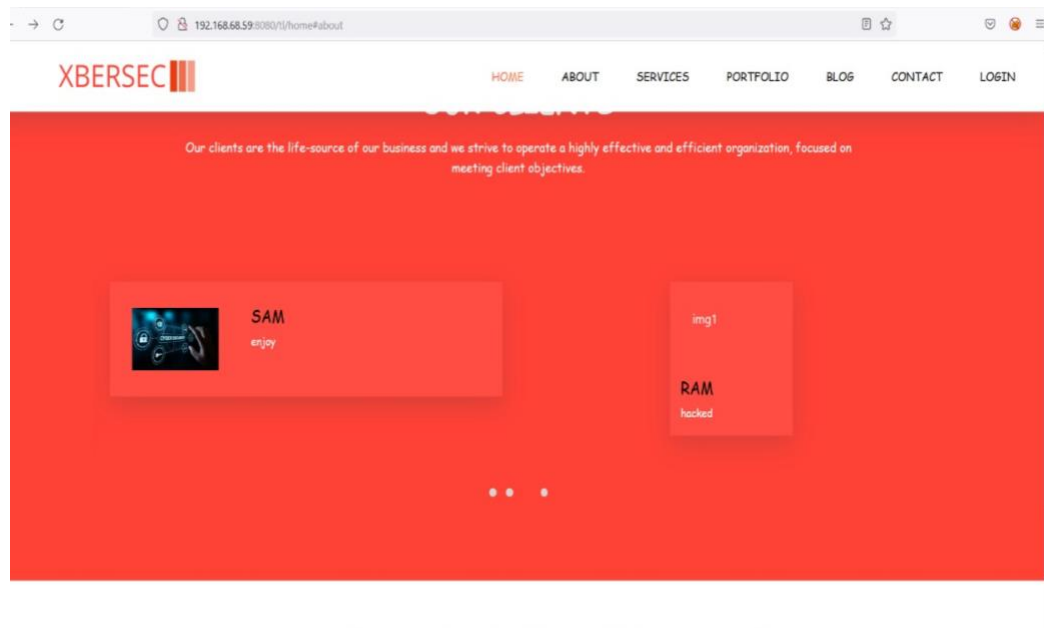
Step 1. Access the URL



Step2:- here it gives the access to upload unwanted malicious files which does not restrict inside the admin panel .



Step3: this how you upload the file parameters in the URL and get the configurations as shown below:-



7-7 . 3 Vulnerability Name:SQL injection

CWE : CWE-564

OWASP Category:A03:2021 – Injections

Description:The software allows the attacker to upload or transfer files of dangerous types that can be automatically processed within the product's environment.

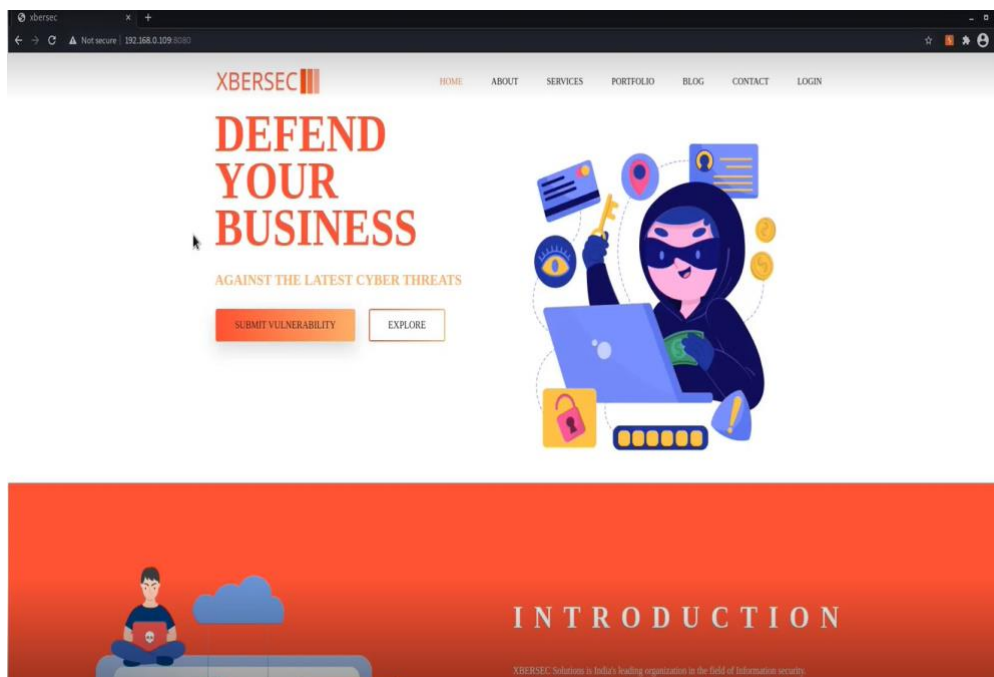
Business Impact:It includes complete system takeover, an overloaded file system or database, forwarding attacks to back-end systems, client-side attacks, or simple defacement. It depends on what the application does with the uploaded file and especially where it is stored.

Vulnerability Path :<http://192.168.0.109:8080/>

Vulnerability Parameter:1 :http://192.168.0.109:8080/admin/en/profile_settings

Steps to Reproduce :

Step 1. Access the URL



Step2:- here it gives the access to upload unwanted malicious files which does not restrict inside the admin panel .

Profile Settings

Update your personal details

Sam Smith

7659986939

king@gmail.com

Enter Confirm Password

Update Go to DASHBOARD

Step3: this how you upload the file parameters in the URL and get the configurations as shown below:-

```

kali@kali: ~/Desktop
File Actions Edit View Help
[00:18:04] [INFO] testing 'MySQL UNION query (NULL) - 21 to 40 columns'
[00:18:04] [INFO] testing 'MySQL UNION query (random number) - 21 to 40 columns'
[00:18:04] [INFO] testing 'MySQL UNION query (NULL) - 41 to 60 columns'
[00:18:05] [INFO] testing 'MySQL UNION query (random number) - 41 to 60 columns'
[00:18:05] [INFO] testing 'MySQL UNION query (NULL) - 61 to 80 columns'
[00:18:06] [INFO] testing 'MySQL UNION query (random number) - 61 to 80 columns'
[00:18:06] [INFO] testing 'MySQL UNION query (NULL) - 81 to 100 columns'
[00:18:06] [INFO] testing 'MySQL UNION query (random number) - 81 to 100 columns'
[00:18:07] [INFO] checking if the injection point on POST parameter 'userID' is a false positive
POST parameter 'userID' is vulnerable. Do you want to keep testing the others (if any)? [y/N]
sqlmap identified the following injection point(s) with a total of 599 HTTP(s) requests:
--
Parameter: userID (POST)
Type: boolean-based blind
Title: AND boolean-based blind - WHERE or HAVING clause
Payload: inputEmail=smith@g.co&inputName=Sam Smith&inputMobile=9966586523&inputPassword=Smith@123&inputCPassword=Smith@123&userID=1
AND 5610=5610
Type: time-based blind
Title: MySQL > 5.0.12 AND time-based blind (query SLEEP)
Payload: inputEmail=smith@g.co&inputName=Sam Smith&inputMobile=9966586523&inputPassword=Smith@123&inputCPassword=Smith@123&userID=1
AND (SELECT 9544 FROM (SELECT(SLEEP(5)))HPBq)
[00:18:09] [INFO] the back-end DBMS is MySQL
back-end DBMS: MySQL > 5.0.12
[00:18:09] [INFO] fetching database names
[00:18:09] [INFO] fetching number of databases
[00:18:09] [WARNING] running in a single-thread mode. Please consider usage of option '--threads' for faster data retrieval
[00:18:09] [INFO] retrieved: 5
[00:18:09] [INFO] retrieved: mysql
[00:18:09] [INFO] retrieved: information_schema
[00:18:11] [INFO] retrieved: performance_schema
[00:18:13] [INFO] retrieved: sys
[00:18:14] [INFO] retrieved: xtree
available databases [5]:
[*] information_schema
[*] mysql
[*] performance_schema
[*] sys
[*] xtree
[00:18:14] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/192.168.0.109'
[00:18:14] [WARNING] your sqlmap version is outdated
[*] ending @ 00:18:14 /2022-11-22/
kali@kali:~/Desktop

```

8 . Vulnerability Name:Anti-CSRF Token missing

CWE : CWE-352

OWASP Category:A05:2021 – Security Misconfiguration

Description:The web application does not, or cannot, sufficiently verify whether a well-formed, valid, consistent request was intentionally provided by the user who submitted the request.

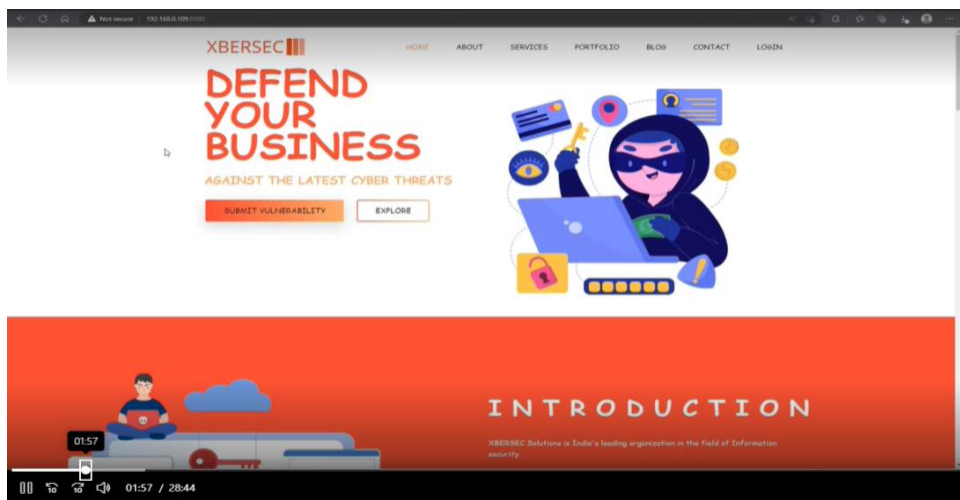
Business Impact:It can result in damaged client relationships, unauthorized fund transfers, changed passwords and data theft—including stolen session cookies.

Vulnerability Path :<http://192.168.0.109:8080/>

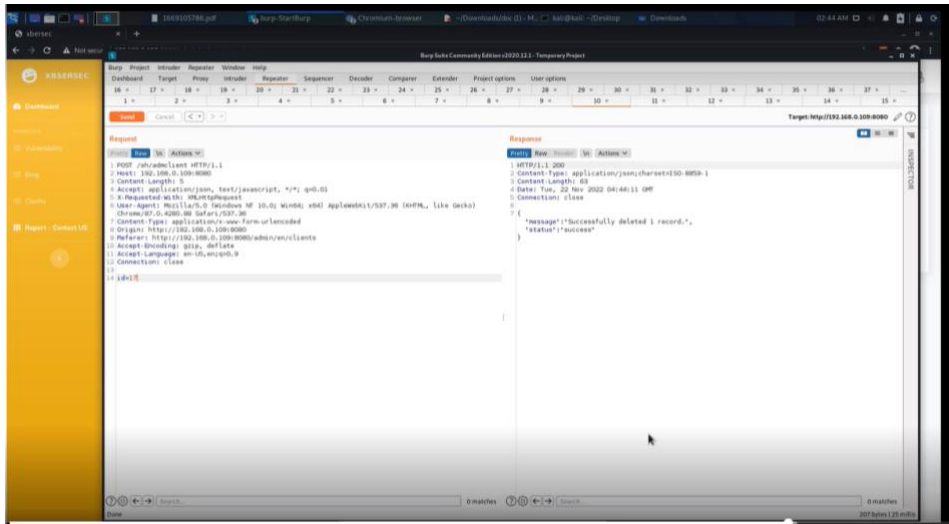
Vulnerability Parameter:<http://192.168.0.109:8080/>

Steps to Reproduce :

Step 1. Access the URL



Step 2: Now intercept with the burp proxy and send request in the repeater



9 . Vulnerability Name: Apache Tomcat Enabled

CWE : 284

OWASP Category: A02:2021 – Cryptographic Failures

Description: The product exposes sensitive information to an actor that is not explicitly authorized to have access to that information

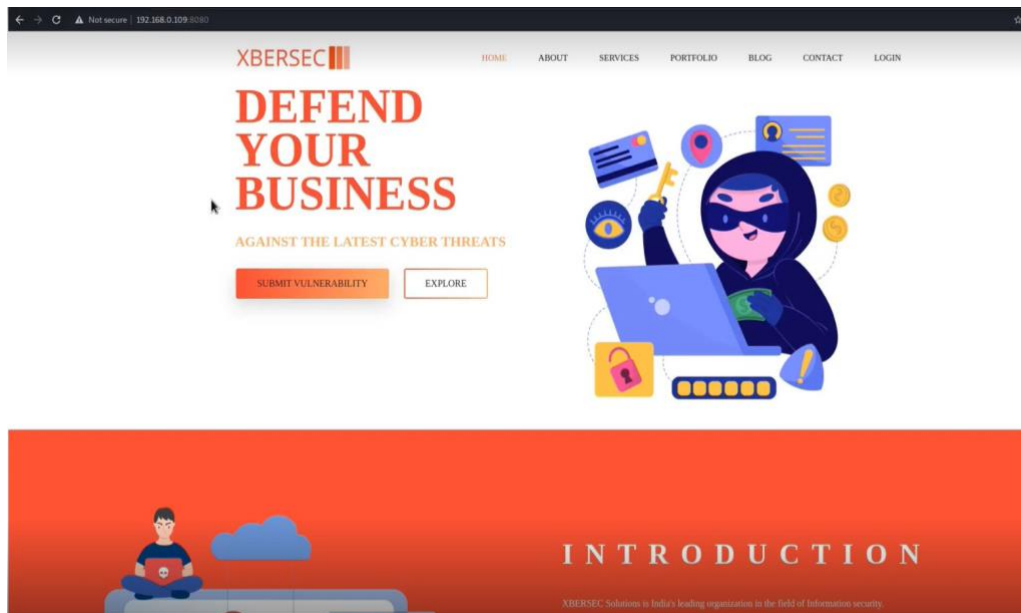
Business Impact: Security incident leads to the accidental or unlawful destruction, loss, alteration, or unauthorized disclosure of, or access to sensitive data. Such Data exposure may occur as a result of inadequate protection of a database, misconfigurations when bringing up new instances of datastores, inappropriate usage of data systems, and more

Vulnerability Path : <http://192.168.0.109:8080/>

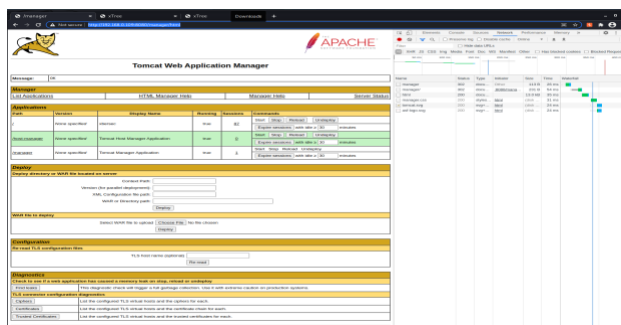
Vulnerability Parameter: <http://192.168.0.109:8080/manager>

Steps to Reproduce :

Step 1. Access the URL



Step 2: change the url parameter by tomcat server manager then we will cancel it as shown in the below page.



10. Vulnerability Name: System Information Exposure

CWE : CWE-497

OWASP Category: A05:2021 – Security Misconfiguration

Description: The application does not properly prevent sensitive system-level information from being accessed by unauthorized actors who do not have the same level of access to the underlying system as the application does.

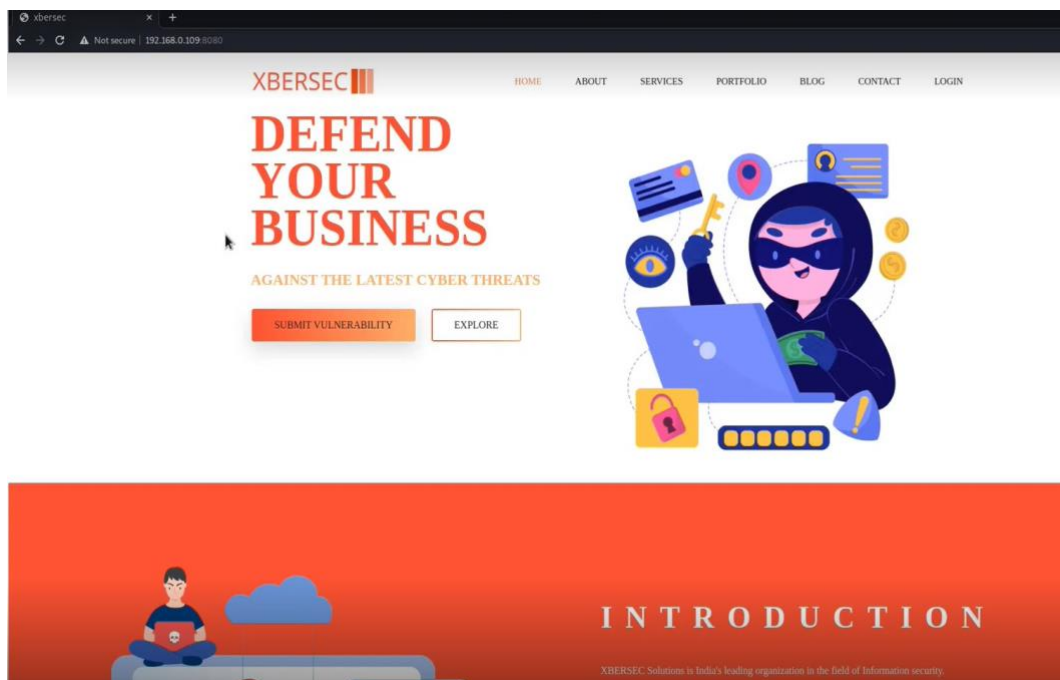
Business Impact: This may lead to expose of admin interfaces and allow the adversary to get privileged access, configuration details or business logic, and even add, remove, or modify application functionality

Vulnerability Path : <http://192.168.0.109:8080/>

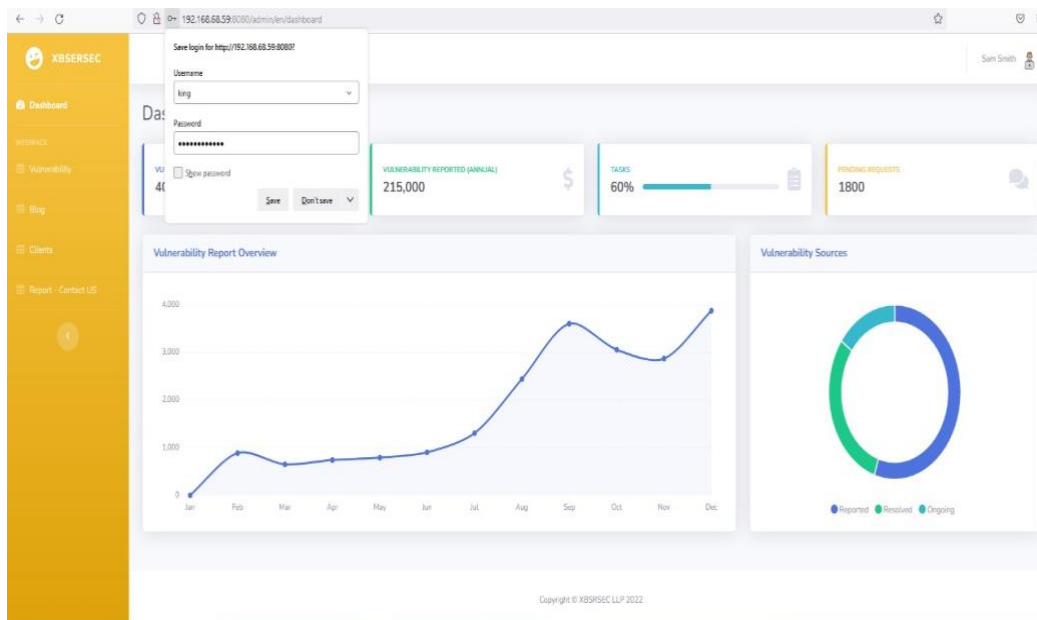
Vulnerability Parameter: <http://192.168.0.109:8080/>

Steps to Reproduce :

Step 1. Access the URL



Step 2:- here we see that the modification of the data in the profile is possible and it also successful in this.



11 . Vulnerability Name: Secure Http flag

CWE : CWE-497

OWASP Category: A05:2021 – Security Misconfiguration

Description: The software uses a cookie to store sensitive information, but the cookie is not marked with the HttpOnly flag.

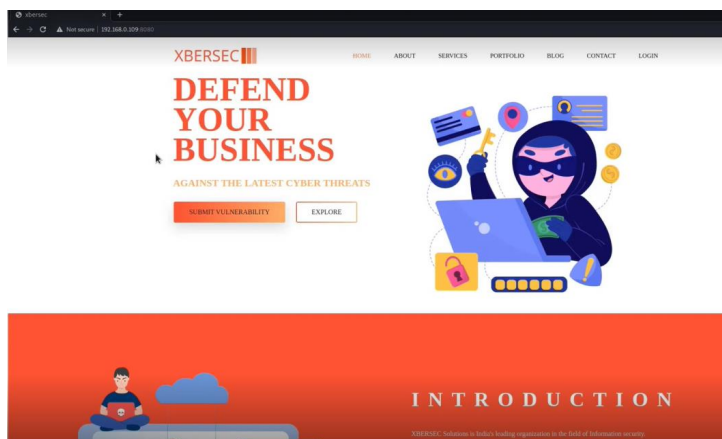
Business Impact: This measure makes certain client-side attacks, such as cross-site scripting, slightly harder to exploit by preventing them from trivially capturing the cookie's value via an injected script.

Vulnerability Path : <http://192.168.0.109:8080/>

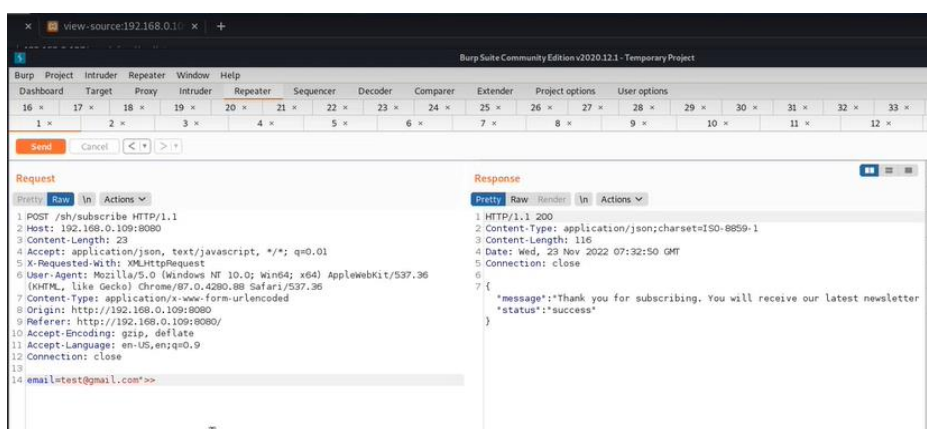
Vulnerability Parameter: <http://192.168.0.109:8080/>

Steps to Reproduce :

Step 1. Access the URL



Step 2: Now intercept with burp proxy and run the request in the repeater.



12 . Vulnerability Name: SSL Encryption is not enabled

CWE : 319

OWASP Category: A05:2021 – Security Misconfiguration

Description:: User credentials are transmitted over an unencrypted channel. This information should always be transferred via an encrypted channel(HTTPS) to avoid being intercepted by a malicious user.

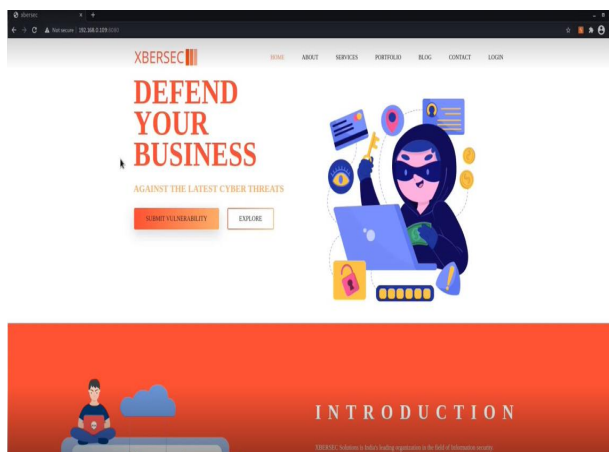
Business Impact : In the absence of an SSL certificate, all your communications are not encrypted at all. Meaning attackers have access to all the data. This includes simple login or registration details, feedback data to most important credit card details as well

Vulnerability Path : <http://192.168.0.109:8080/>

Vulnerability Parameter: <http://192.168.0.109:8080/>

Steps to Reproduce :

Step 1. Access the URL



Step2:On the address bar type the following link <http://192.168.0.109/8080/>

13 . Vulnerability Name: Local file Inclusion

CWE : 98

OWASP Category: A05:2021 – Security Misconfiguration

Description: Local File Inclusion is an attack technique in which attackers trick a web application into either running or exposing files on a web server. LFI attacks can expose sensitive information, and in severe cases, they can lead to cross-site scripting (XSS) and remote code execution

Business Impact : A local file inclusion vulnerability can lead to Directory Traversal attacks, where an attacker will try to find and access files on the web server to gain more useful

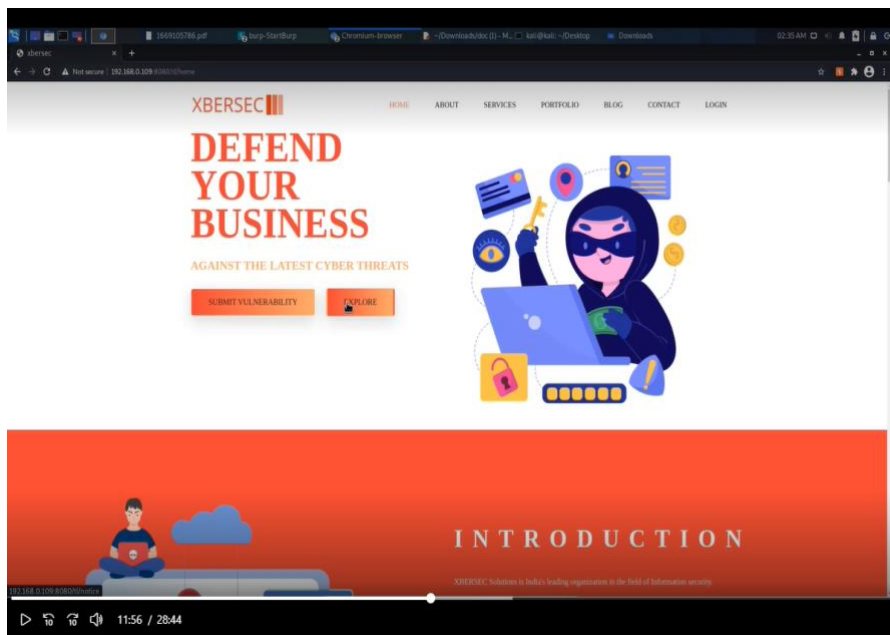
information, such as log files. Log files can reveal the structure of the application or expose paths to sensitive files.

Vulnerability Path : <http://192.168.0.109:8080/>

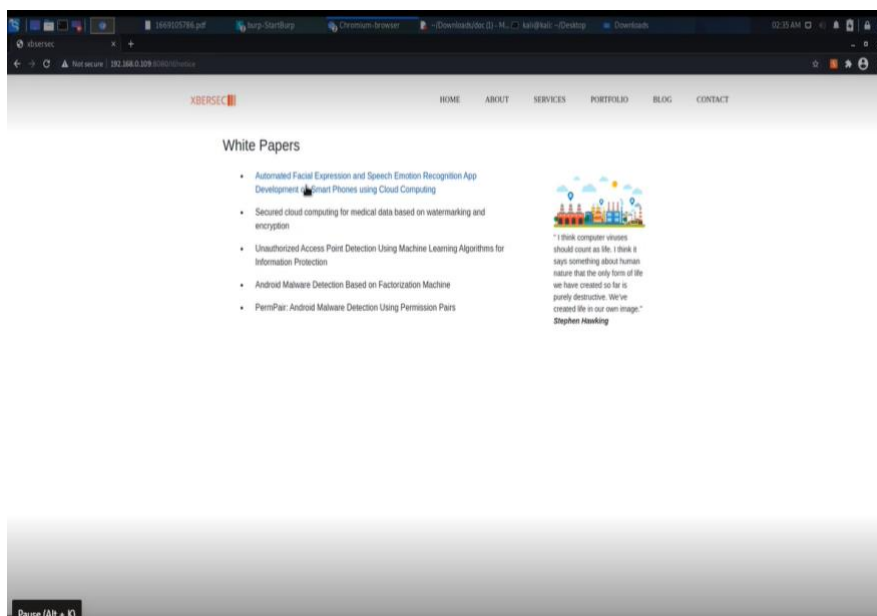
Vulnerability Parameter: <http://192.168.0.109:8080/>

Steps to Reproduce :

Step 1: Access to the url



Step2: Click on the blog we will get whitepapers title page



[illegible][illegible]

OWASP Category: A03:2021 –Injections

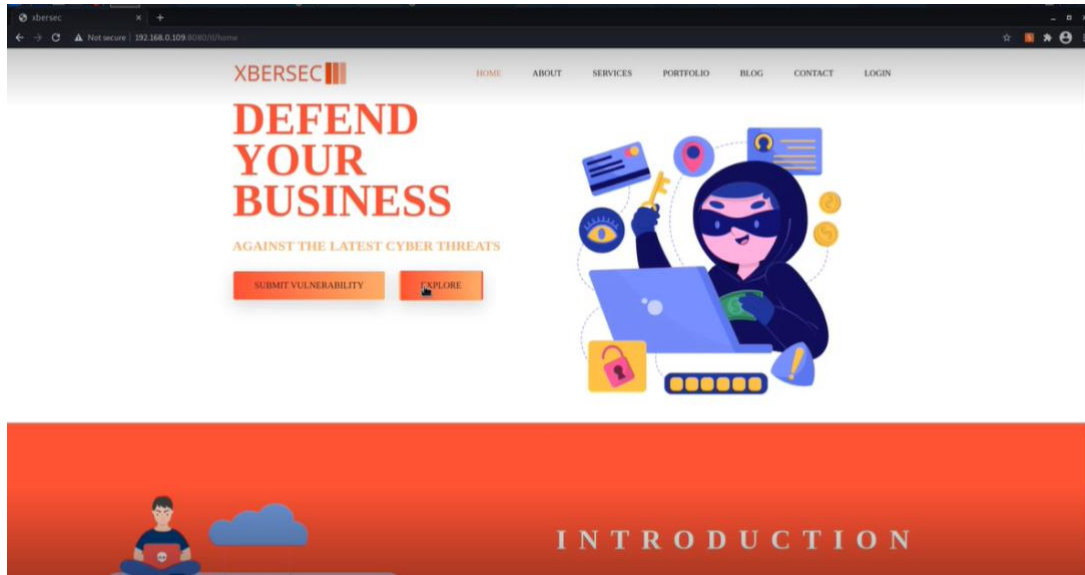
Business Impact: The server reads data directly from the HTTP request and reflects it back in the HTTP response. Reflected XSS exploits occur when an attacker causes a victim to supply dangerous content to a vulnerable web application, which is then reflected back to the victim and executed by the web browser. The most common mechanism for delivering malicious content is to include it as a parameter in a URL that is posted publicly or e-mailed directly to the victim.

Vulnerability Path : <http://192.168.0.109:8080/>

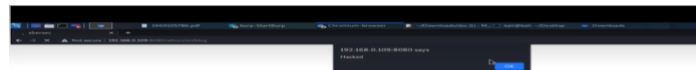
Vulnerability Parameter: <http://192.168.0.109:8080/admin/en/blog>

Steps to Reproduce :

Step 1. Access the URL



Step2: After clicking on the blog site we will get a pop which is vulnerable to cross site scripting.



14–14.2 .Vulnerability Name: Cross-Site Scripting (Reflected)

CWE : CWE-79

OWASP Category: A03:2021 –Injections

Description: It occurs when a malicious script is reflected off of a web application to the victim's browser. The script is activated through a link, which sends a request to a website with a vulnerability that enables

execution of malicious scripts.

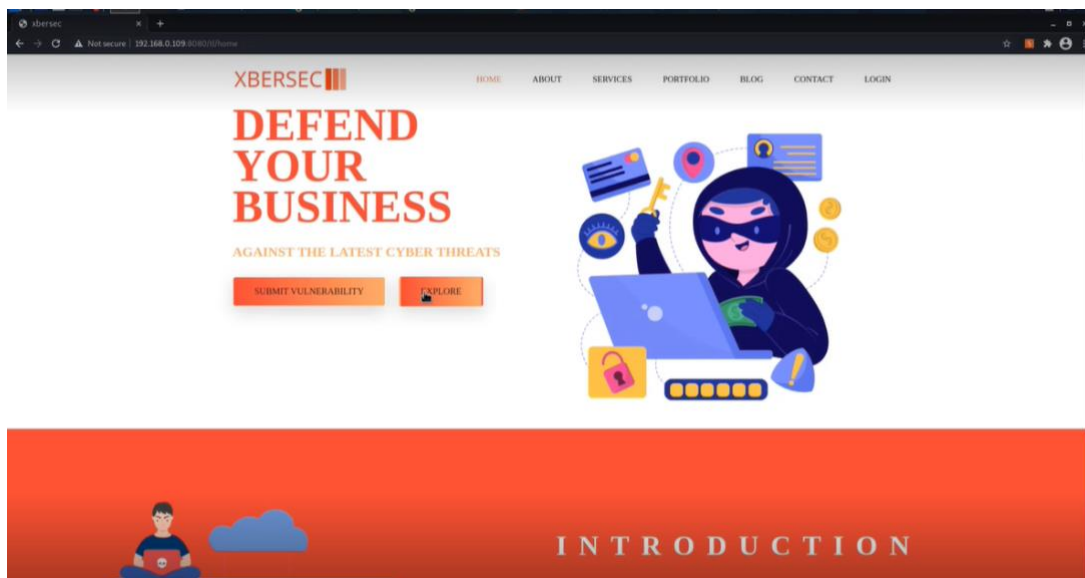
Business Impact: The server reads data directly from the HTTP request and reflects it back in the HTTP response. Reflected XSS exploits occur when an attacker causes a victim to supply dangerous content to a vulnerable web application, which is then reflected back to the victim and executed by the web browser. The most common mechanism for delivering malicious content is to include it as a parameter in a URL that is posted publicly or e-mailed directly to the victim.

Vulnerability Path : <http://192.168.0.109:8080/>

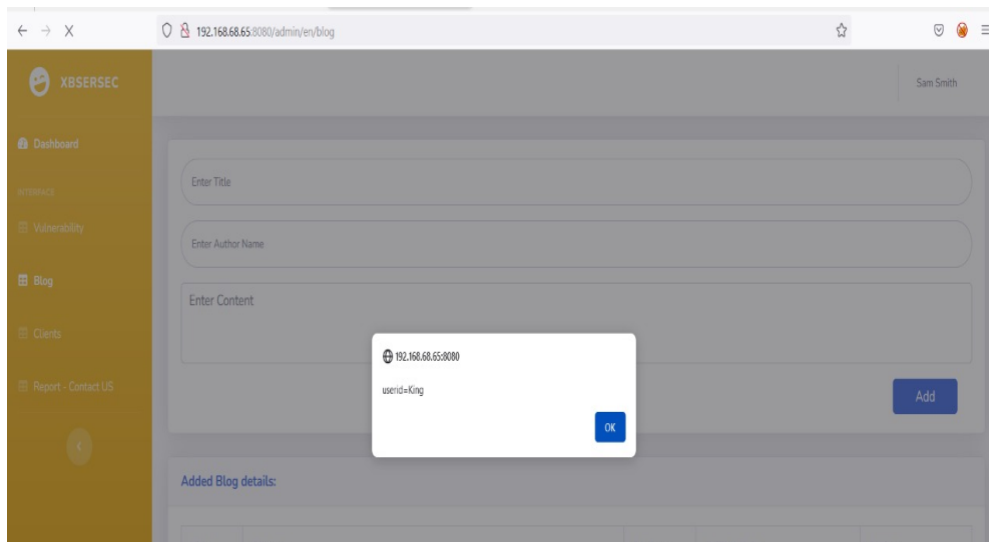
Vulnerability Parameter: <http://192.168.0.109:8080/admin/en/blog>

Steps to Reproduce :

Step 1. Access the URL



Step2: After clicking on the blog site we will get a pop which is vulnerable to cross site scripting.



15.Vulnerability Name: : Violation of Secure Design Principles

CWE :CWE-657

OWASP Category: A04:2021 – Insecure Design

Description: insecure design is a broad category representing different weaknesses, expressed as “missing or ineffective control design.” Insecure design is not the source for all other Top 10 risk categories. There is a difference between insecure design and insecure implementation. We differentiate between design flaws and implementation defects for a reason, they have different root causes and remediation. A secure design can still have implementation defects leading to vulnerabilities that may be exploited. An insecure design cannot be fixed by a perfect implementation as by definition, needed security controls were never created to defend against specific attacks. One of the factors that contribute to insecure design is the lack of business risk profiling inherent in the software or system being developed, and thus the failure to determine what level of security design is required.

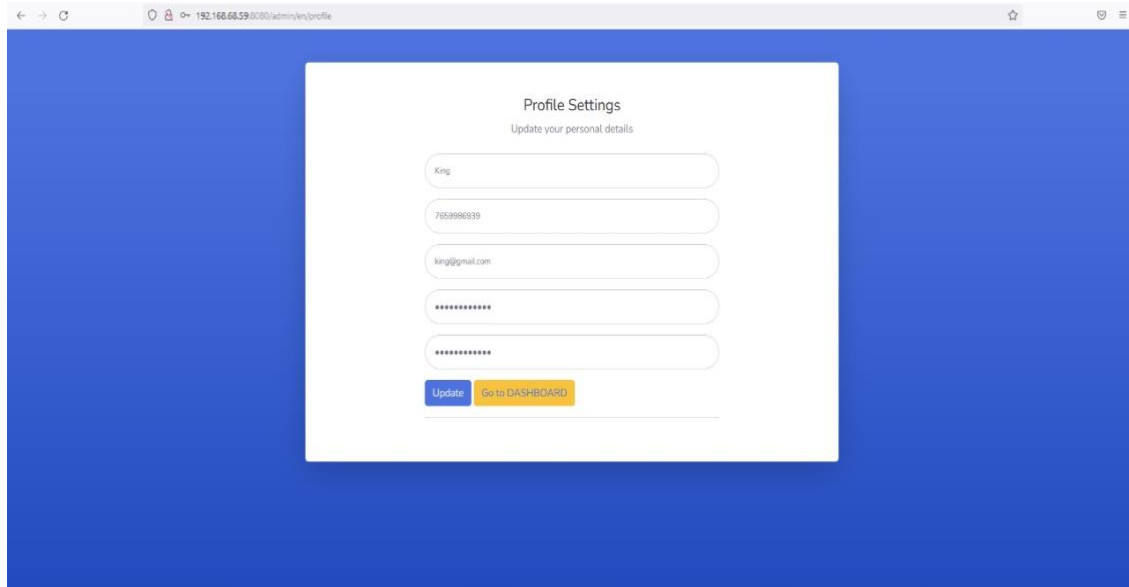
Business Impact: This mainly focuses on risks related to design and architectural flaws, with a call for more use of threat modeling, secure design patterns, and reference architectures. As a community we need to move beyond "shift-left" in the coding space to pre-code activities that are critical for the principles of Secure by Design. Notable Common Weakness Enumerations (CWEs) include *CWE-209: Generation of Error Message Containing Sensitive Information*,

Vulnerability Path : <http://192.168.0.109:8080/>

Vulnerability Parameter: <http://192.168.68.59:8080/admin/en/profile>

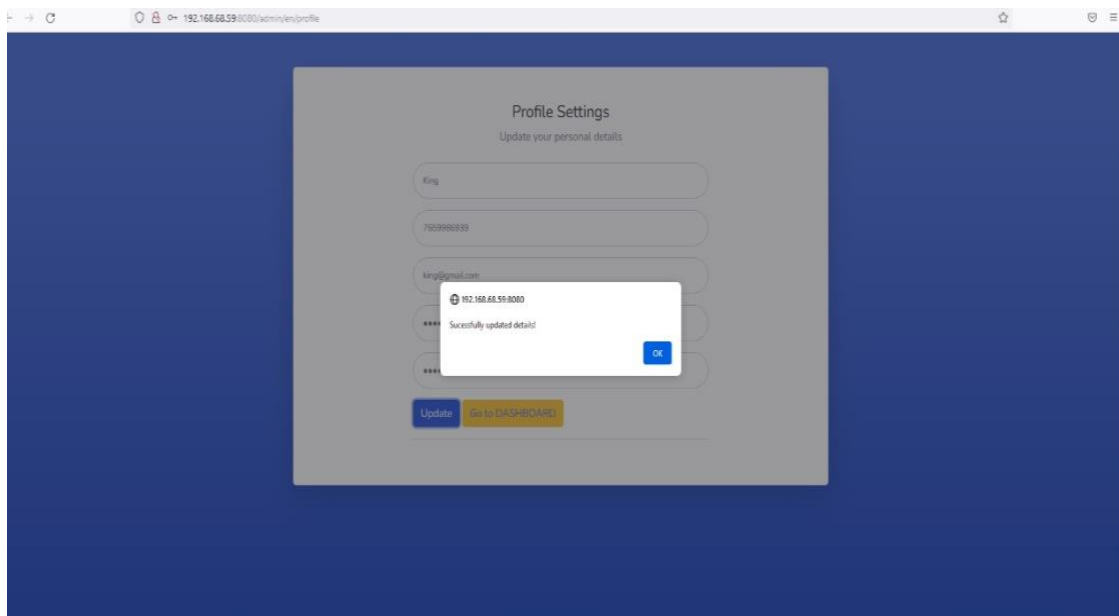
Steps to Reproduce :

Step 1. Access the URL (The actual name here is sam smith but here it allows you edit the name)



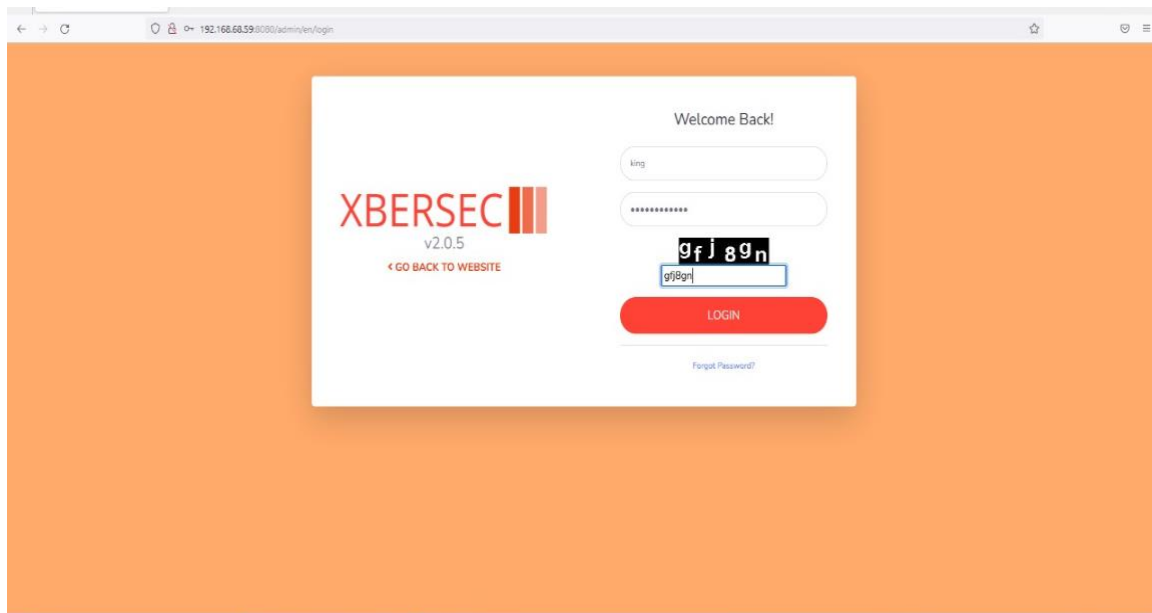
A screenshot of a web browser showing a 'Profile Settings' page. The page has a blue background and a white central form. The form is titled 'Profile Settings' with the subtitle 'Update your personal details'. It contains several input fields: a text field with 'King', a text field with '765998039', a text field with 'king@gmail.com', and two password fields represented by asterisks. At the bottom of the form are two buttons: a blue 'Update' button and a yellow 'Go to DASHBOARD' button.

Step 2: After this you can save the details in the next step which will successfully store these details .

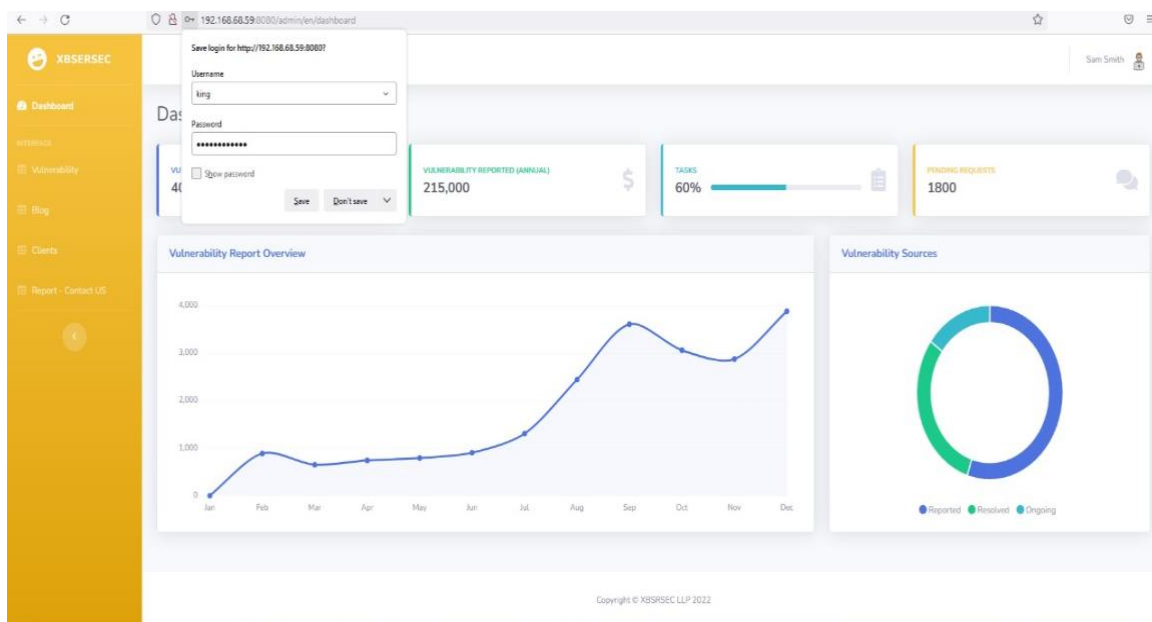


A screenshot of the same 'Profile Settings' page, but with a semi-transparent grey overlay. A white modal box is centered on the screen, displaying a success message: 'Successfully updated detail' with a green checkmark icon. The modal has an 'OK' button. The background form is visible but dimmed.

Step 3 :Now you can login with an updated name and it allows you with the same old credentials.



Step 4 : As you can see, we updated the name but still it shows the same old name .



15–15.2 Vulnerability Name: Clickjacking (Improper Restriction of Rendered UI Layers or Frames)

CWE :CWE-1021

OWASP Category: A04-2021- insecure design

Description:it occurs whenever a web application is fetching a remote resource without validating the user-supplied URL. It allows an attacker to coerce the application to send a crafted request to an unexpected destination, even when protected by a firewall, VPN, or another type of network access control list (ACL).

Business Impact: The data shows a relatively low incidence rate with above average testing coverage and above-average Exploit and Impact potential ratings. As new entries are likely to be a single or small cluster of Common Weakness Enumerations (CWEs) for attention and awareness, the hope is that they are subject to focus and can be rolled into a larger category in a future edition.

Vulnerability Path : <http://192.168.0.109:8080/>

Vulnerability Parameter: <http://192.168.68.59:8080>

Steps to Reproduce :

Step 1:- Access the URI

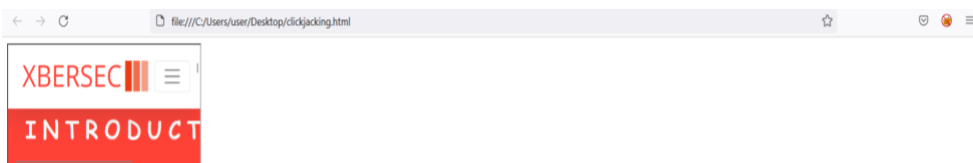


Step 2:- we write a html code as shown below when we run with the ip address with sight the we get

```
<html>
<head>

<title>Hacked</title>
</head>
<body>
<iframe src="http://192.168.68.65:8080/"></iframe>
</body>
</html>
```

Step 3:- this will be the output of clickjacking the website with html code .



16. Vulnerability Name: Reliance on Cookies without Validation and Integrity Checking in a Security Decision

CWE :CWE-784:

OWASP Category:A08:2021 – Software and Data Integrity Failures

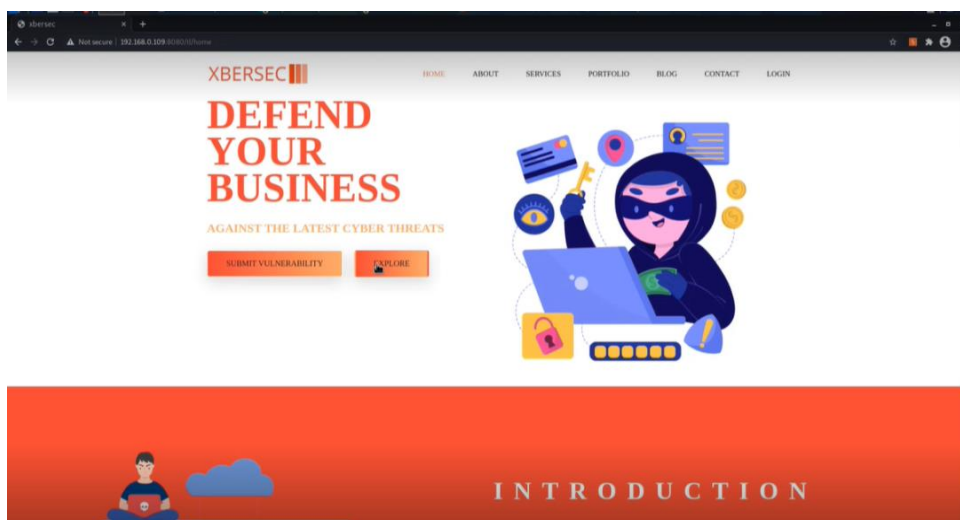
Description: Software and data integrity failures relate to code and infrastructure that does not protect against integrity violations. An example of this is where an application relies upon plugins, libraries, or modules from untrusted sources, repositories, and content delivery networks (CDNs). An insecure CI/CD pipeline can introduce the potential for unauthorized access, malicious code, or system compromise. Lastly, many applications now include auto-update functionality, where updates are downloaded without sufficient integrity verification and applied to the previously trusted application. Attackers could potentially upload their own updates to be distributed and run on all installations. Another example is where objects or data are encoded or serialized into a structure that an attacker can see and modify is vulnerable to insecure deserialization.

Business Impact: This mainly focuses on making assumptions related to software updates, critical data, and CI/CD pipelines without verifying integrity. One of the highest weighted impacts from Common Vulnerability and Exposures/Common Vulnerability Scoring System (CVE/CVSS) data. Notable Common Weakness Enumerations (CWEs) include *CWE-829: Inclusion of Functionality* .

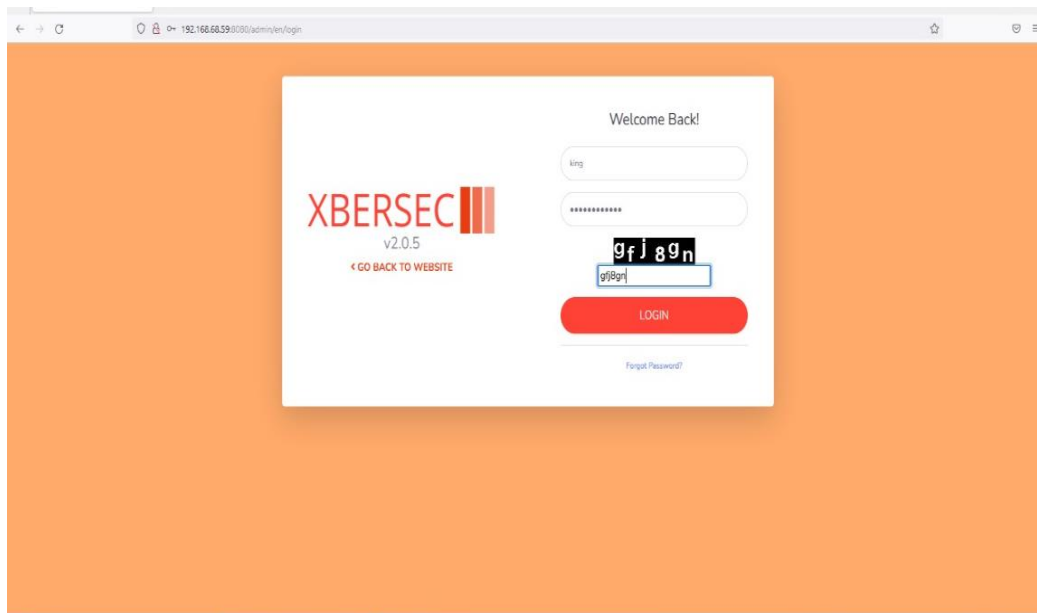
Vulnerability Path : <http://192.168.0.109:8080/>

Vulnerability Parameter: <http://192.168.0.109:8080/admin/en/blog>

Steps to Reproduce : **Step 1.** Access the URL



Step 2 :Now you can login with an update credentials and session id would be the same .



Step 3:- your session id should not give leakage to any information but here it shows the recently updated name last login .

17.Vulnerability Name: Sensitive Cookie without 'Httponly' Flag 1021 - Improper design

CWE : CWE -1004

OWASP Category: A09:2021 – Security Logging and Monitoring Failures

Description: basically this category is to help detect, escalate, and respond to active breaches. Without logging and monitoring, breaches cannot be detected. Insufficient logging, detection, monitoring, and active response occurs any time:

- Logs of applications and APIs are not monitored for suspicious activity.
- Logs are only stored locally.
- Appropriate alerting thresholds and response escalation processes are not in place or effective.
- Penetration testing and scans by dynamic application security testing (DAST) tools (such as OWASP ZAP) do not trigger alerts.

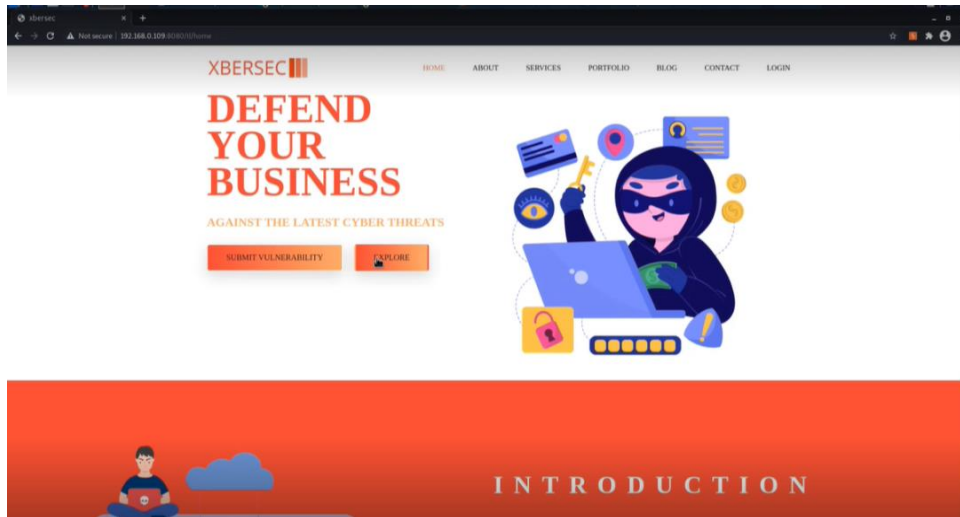
Business Impact: Logging and monitoring can be challenging to test, often involving interviews or asking if attacks were detected during a penetration test. There isn't much CVE/CVSS data for this category, but detecting and responding to breaches is critical. Still, it can be very impactful for accountability, visibility, incident alerting, and forensics.

Vulnerability Path : <http://192.168.0.109:8080/>

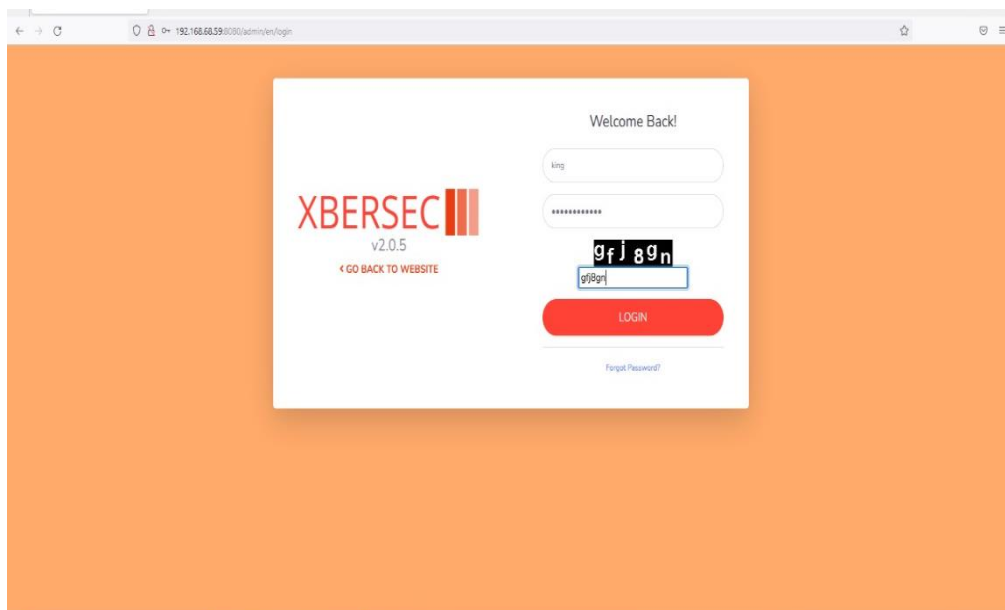
Vulnerability Parameter: <http://192.168.0.109:8080/admin/en/blog>

Steps to Reproduce :

Step 1. Access the URL

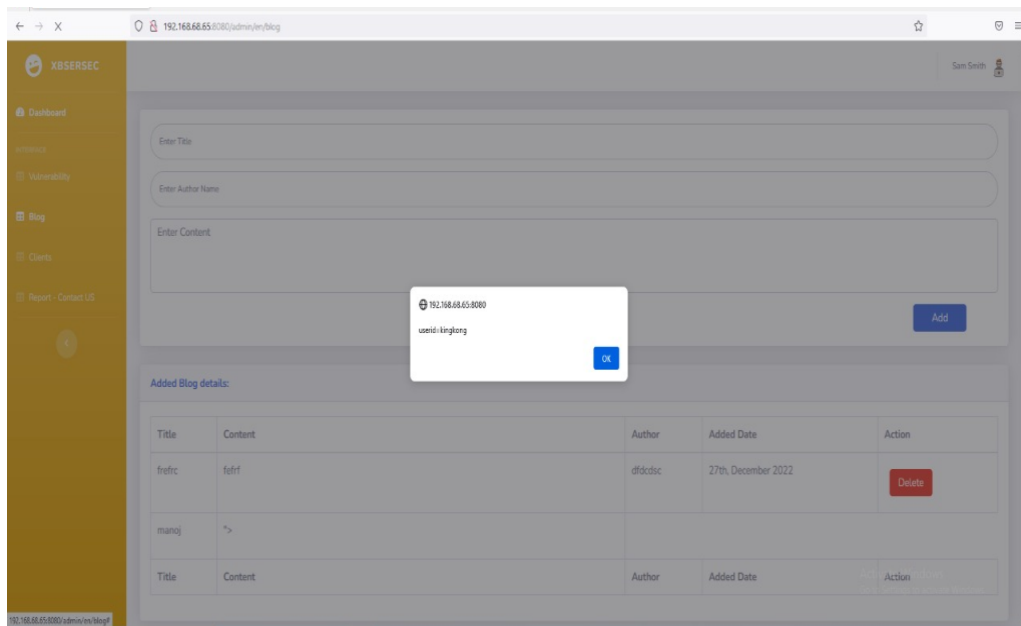


Step 2 : Now you can login with an update credentials and session id would be the same .

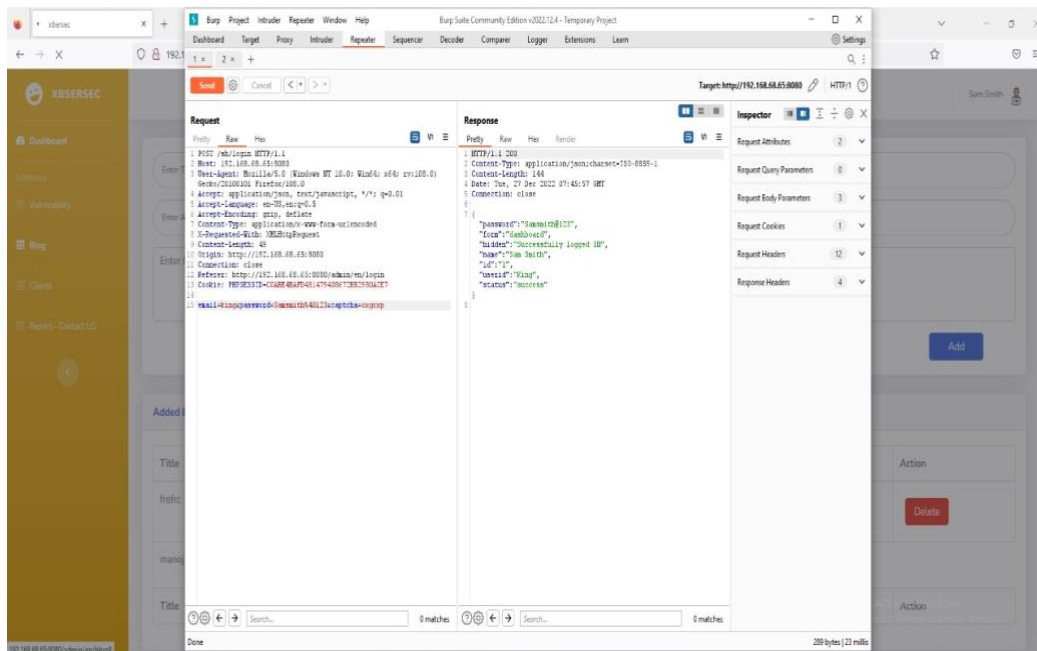


Step 3:- your session id should not give leakage to any information but here it shows the recently

updated name last login .



Step 4:- here you can see that both are different browsers having same session ids and same cookies thrown when the cookies are accessed .



18. Vulnerability Name: Exposure of Sensitive Information to an Unauthorized Actor

CWE :CWE-200

OWASP Category: A10:2021-Server-Side Request Forgery (SSRF)

Description:it occurs whenever a web application is fetching a remote resource without validating the user-supplied URL. It allows an attacker to coerce the application to send a crafted request to an unexpected destination, even when protected by a firewall, VPN, or another type of network access control list (ACL).

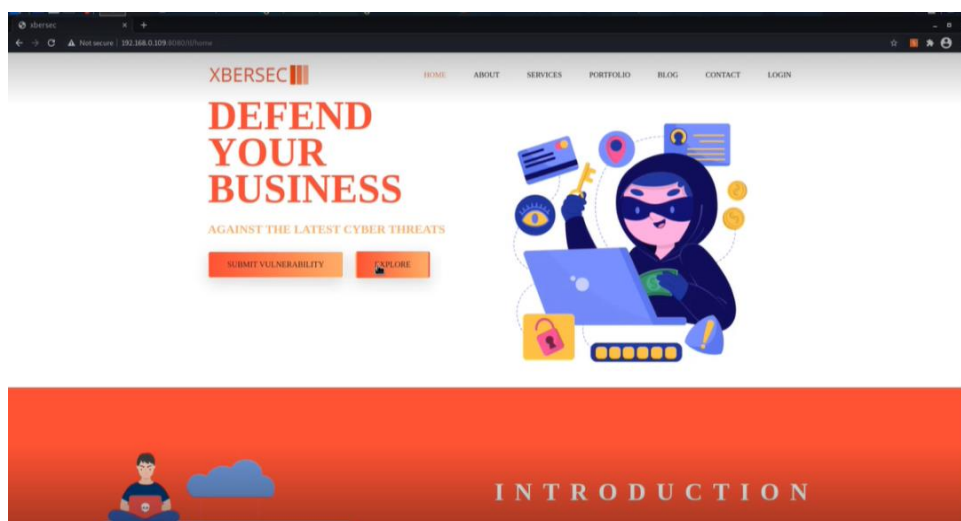
Business Impact: The data shows a relatively low incidence rate with above average testing coverage and above-average Exploit and Impact potential ratings. As new entries are likely to be a single or small cluster of Common Weakness Enumerations (CWEs) for attention and awareness, the hope is that they are subject to focus and can be rolled into a larger category in a future edition.

Vulnerability Path : <http://192.168.0.109:8080/>

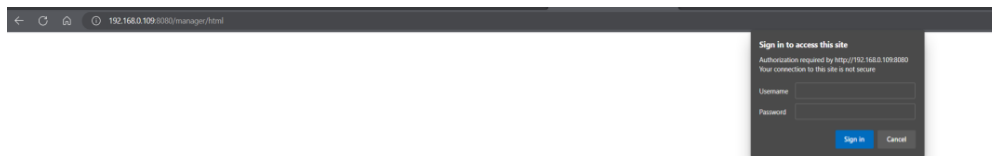
Vulnerability Parameter: <http://192.168.68.59:8080/Manger>

Steps to Reproduce :

Step 1:- Access the URI



Step 2: By changing the URL parameters with tomcat configurations we find the below page.



Step 3:- By closing the dialog box without canceling it this will give you the access to the default credentials

