

ALTORO-MUTUAL-VULNERABILITY-ANALYSIS-REPORT

KRISHNA SATHVIKA GANNI
4TH YEAR,B.TECH
RCEE

STEPS:

Step 1: OWASP Top 10 Vulnerabilities Overview:

1. **Broken Access Control:** Flaws in authentication and access control can allow unauthorized access to sensitive data or systems. Using Infrastructure as Code (IaC) tools can help detect configuration errors leading to access control failures.
2. **Cryptographic Failures:** Mistakes like hardcoded passwords or weak encryption methods can expose sensitive data. Scanning for hardcoded secrets and ensuring proper encryption can mitigate these risks.
3. **Injection:** Attackers exploit vulnerabilities in web applications to inject malicious code, like SQL injection or Cross-Site Scripting (XSS). Application security testing can help detect these flaws.
4. **Insecure Design:** Focuses on fundamental design flaws rather than implementation issues. Secure design practices, developer training, and threat modeling are essential to prevent such vulnerabilities.
5. **Security Misconfiguration:** Errors in server, framework, or cloud infrastructure configurations can lead to breaches. Regularly hardening configurations and scanning for misconfigurations are crucial for mitigation.
6. **Vulnerable and Outdated Components:** Third-party libraries and components can introduce vulnerabilities. Building a Software Bill of Materials (SBOM) and using tools for vulnerability management can help track and mitigate risks.
7. **Identification and Authentication Failures:** Weaknesses in user identification and authorization processes can be exploited. Secure coding practices and tools for detecting credential stuffing and brute force attacks are essential for protection.
8. **Software and Data Integrity Failures:** Attackers can exploit the

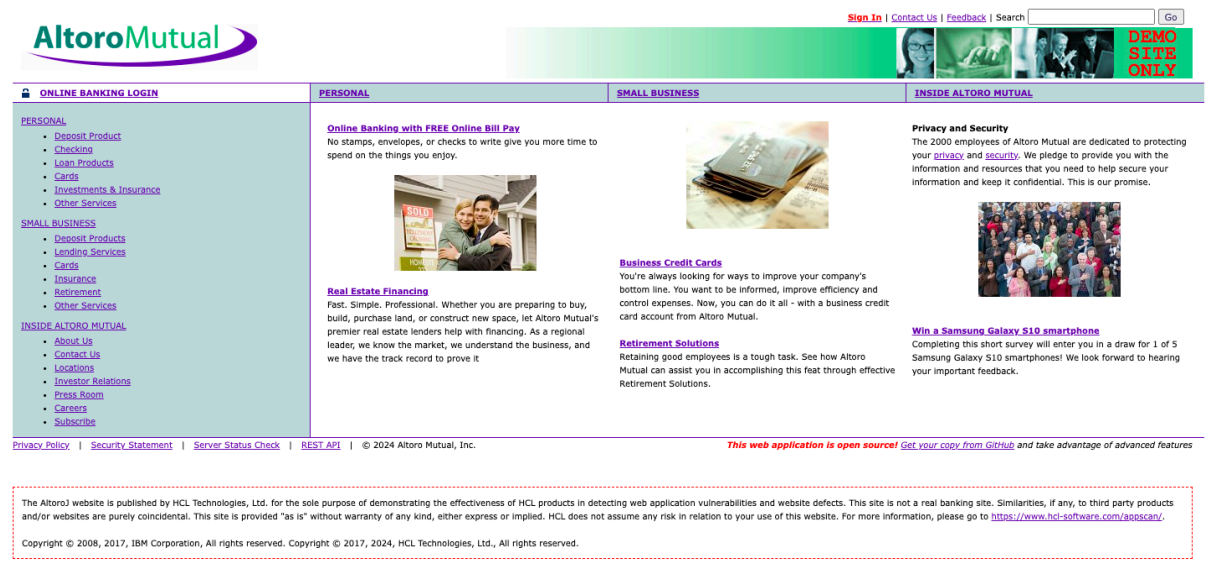
software build process to inject malicious code or steal secrets. Ensuring the security of the build process and components used is crucial for mitigating this threat.

9. **Security Logging and Monitoring Failures:** Adequate logging and monitoring are essential for detecting and responding to security breaches. Regular verification of logging and alerting processes is necessary for effective incident response.
10. **Server-Side Request Forgery (SSRF):** Attackers can manipulate web applications to send requests to unintended destinations. Mitigation involves sanitizing user input and inspecting request responses to prevent SSRF attacks.

The potential impact of these vulnerabilities on web application security and the importance of addressing them to prevent exploitation by attackers:

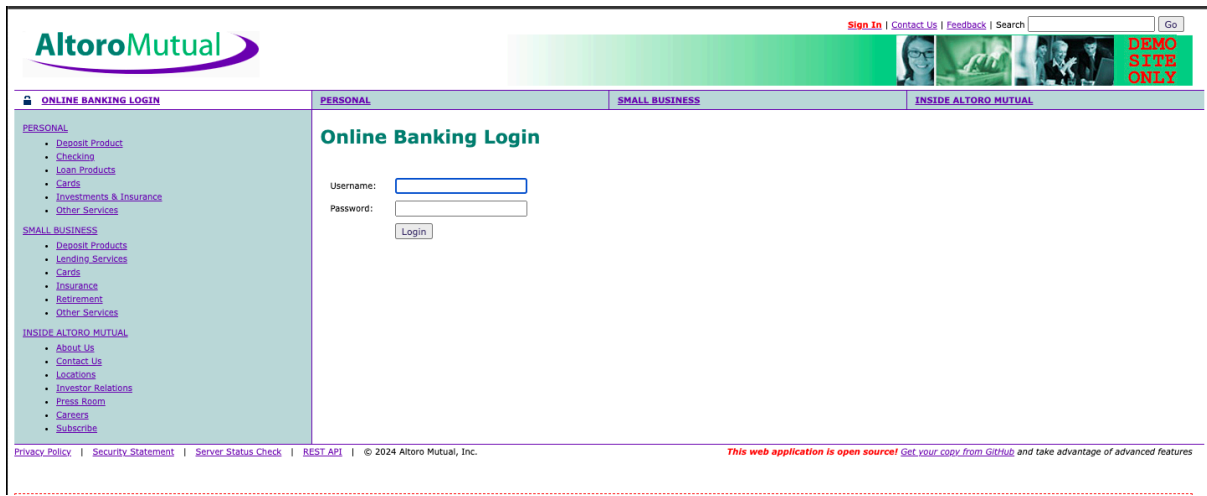
1. **Broken Access Control:** Unauthorized access can lead to data breaches and financial loss.
2. **Cryptographic Failures:** Weak encryption can expose sensitive data, leading to theft and reputational damage.
3. **Injection:** Attacks can manipulate data, steal information, and compromise systems.
4. **Insecure Design:** Fundamental flaws can result in widespread vulnerabilities, causing data breaches and system downtime.
5. **Security Misconfiguration:** Misconfigurations can lead to unauthorized access and compliance violations.
6. **Vulnerable and Outdated Components:** Exploiting vulnerabilities can lead to system compromise and reputational damage.
7. **Identification and Authentication Failures:** Weak authentication can result in unauthorized access and fraud.
8. **Software and Data Integrity Failures:** Compromised software integrity can lead to malware distribution and financial losses.
9. **Security Logging and Monitoring Failures:** Inadequate monitoring can delay detection of security incidents, allowing attackers to cause further damage.
10. **Server-Side Request Forgery (SSRF):** Exploitation can lead to unauthorized access and data exfiltration.

Step 2: Altro Mutual Website Analysis:



It's the main page of the Altro Mutual website. It features "feedback," "contact," and "sign in" options. The search feature should be added after logging in and deleted in order to facilitate component searches on the website. An attack using "cross-site scripting" could be possible. There are several categories, including personal, small company, and inside Alotoro Mutual, as seen in the graphic below. The left side column lists the sub-divisions for each category.


login page:



Contact page:

AltoroMutual

[Sign In](#) | [Contact Us](#) | [Feedback](#) | Search



Go

ONLINE BANKING LOGIN

PERSONAL

SMALL BUSINESS

INSIDE ALTORO MUTUAL

PERSONAL

- Deposit Product
- Checking
- Loan Products
- Cards
- Investments & Insurance
- Other Services

SMALL BUSINESS

- Deposit Products
- Lending Services
- Cards
- Insurance
- Retirement
- Other Services

INSIDE ALTORO MUTUAL

- About Us
- Contact Us
- Locations
- Investor Relations
- Press Room
- Careers
- Subscribe

Contact Us

Have a question? It's easy to reach us.

E-mail

Filling out the [online form](#) is the most efficient method of contact. If you are requesting a change to your account, please call the phone number listed below.

Phone

To open a new account, please call:
1.800.555.0001
8:00 a.m. - 6:00 p.m., Eastern Time, Monday - Friday


For assistance with your account, please call:
1.800.555.0002
24-hour touch-tone banking information is available 7 days a week.

Mail

To contact us by mail, you may send inquiries to:
Altoro Mutual
Altoro Mutual Tower
Anywhere, MA

E-mail Security

Any inquiry you send to Altoro Mutual via our Contact Us page uses Secure Socket Layer (SSL) encryption. SSL helps to ensure that your personal information remains confidential.




Altoro Mutual is headquartered at the Altoro Mutual Tower in Anywhere, MA.

DEMO SITE ONLY

Feedback form:

AltoroMutual

[Sign In](#) | [Contact Us](#) | [Feedback](#) | Search



Go

ONLINE BANKING LOGIN

PERSONAL

SMALL BUSINESS

INSIDE ALTORO MUTUAL

PERSONAL

- Deposit Product
- Checking
- Loan Products
- Cards
- Investments & Insurance
- Other Services

SMALL BUSINESS

- Deposit Products
- Lending Services
- Cards
- Insurance
- Retirement
- Other Services

INSIDE ALTORO MUTUAL

- About Us
- Contact Us
- Locations
- Investor Relations
- Press Room
- Careers
- Subscribe

Feedback

Our Frequently Asked Questions area will help you with many of your inquiries. If you can't find your question, return to this page and use the e-mail form below.

IMPORTANT! This feedback facility is not secure. Please do not send any account information in a message sent from here.

To: **Online Banking**

Your Name:

Your Email Address:

Subject:

Question/Comment:

DEMO SITE ONLY

[Privacy Policy](#) | [Security Statement](#) | [Server Status Check](#) | [REST API](#) | © 2024 Altoro Mutual, Inc.

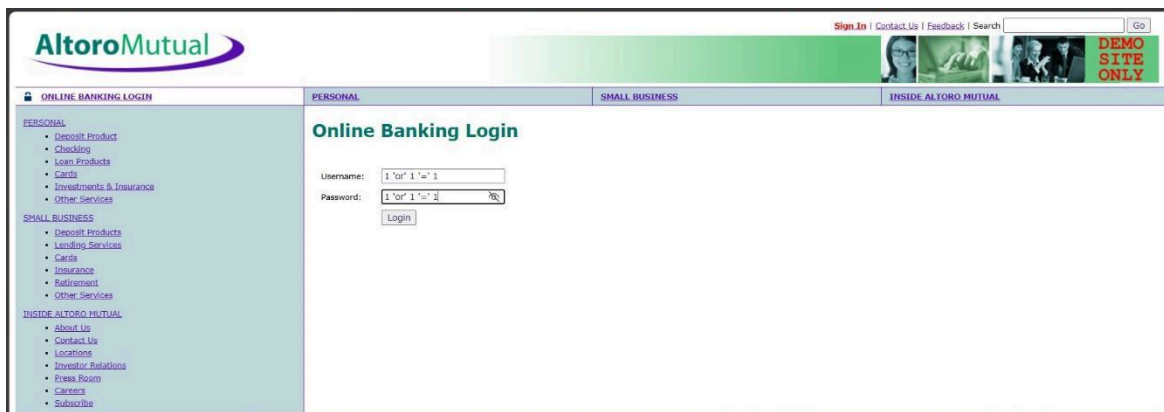
This web application is open source! [Get your copy from Github](#) and take advantage of advanced features

Step 3: Vulnerability Identification Report:

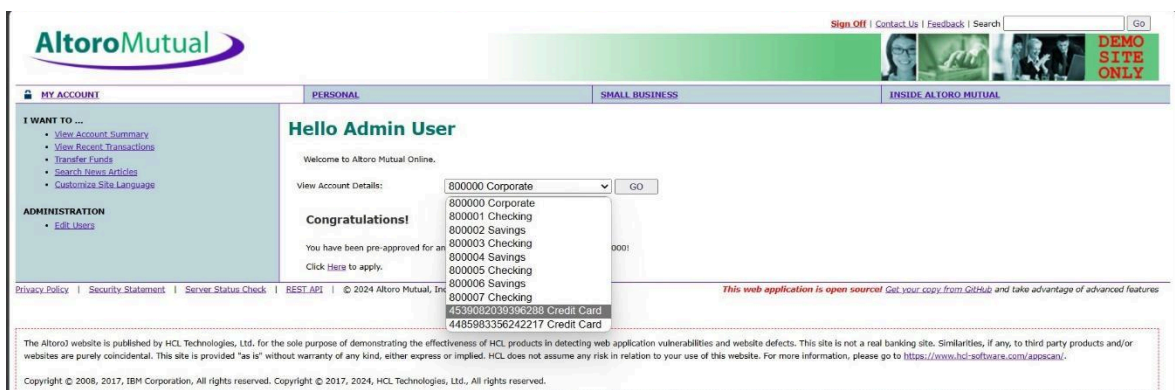
➔ The report includes a detailed description of Altro Mutual's website structure and functionality, including potential areas of vulnerability.

Step 4: Vulnerability Exploitation Demonstration:

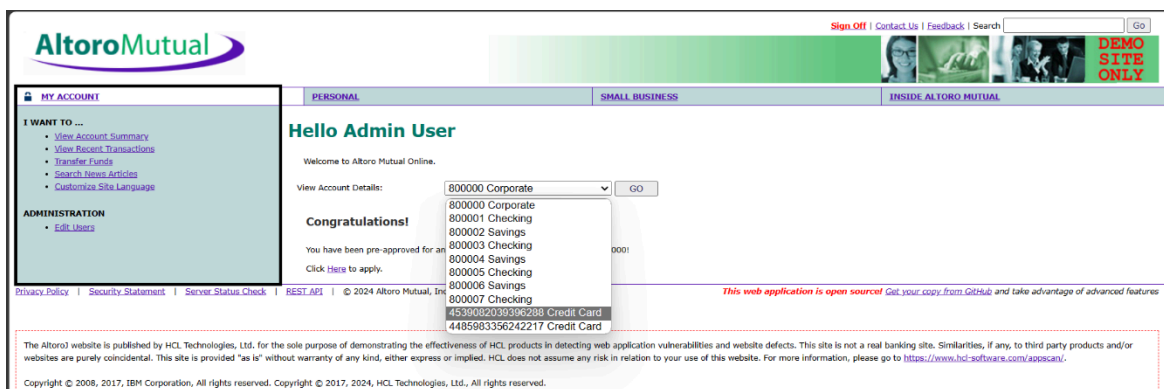
Here, we'll see if we can log in to the website. If so, we have a vulnerability known as "Broken Authorization," which can lead to other vulnerabilities like Broken Access Control and Sensitive Data Exposure.



Here, we logged in as admin and used the payload 1 "or" 1 "=" 1 as the login and password. Other users are visible to us. We have the ability to add users, modify their passwords, and manage their accounts.



Users of the bank can view their account details here. The content pages for fund transfers, account summaries, and recent transactions are shown in the image below in the highlighted column.



We are able to edit users and their details, as seen in the image above. These are depicted in the graphic below.

AltoroMutual

Sign Off | Contact Us | Feedback | Search [Go]

DEMO SITE ONLY

MY ACCOUNT

PERSONAL | **SMALL BUSINESS** | **INSIDE ALTORO MUTUAL**

I WANT TO ...

- View Account Summary
- View Recent Transactions
- Transfer Funds
- Search News Articles
- Customize Site Language

ADMINISTRATION

- Edit Users

Edit User Information

Add an account to an existing user

Users: Account Types:

Change user's password

Users: Password: Confirm:

Add a new user

First Name: Last Name: Username: Password: Confirm:

It is highly recommended that you leave the username as first initial last name.

Privacy Policy | Security Statement | Server Status Check | REST API | © 2024 Altoro Mutual, Inc. *This web application is open source! Get your copy from GitHub and take advantage of advanced features*

The Altoro website is published by HCL Technologies, Ltd. for the sole purpose of demonstrating the effectiveness of HCL products in detecting web application vulnerabilities and website defects. This site is not a real banking site. Similarities, if any, to third party products and/or websites are purely coincidental. This site is provided "as is" without warranty of any kind, either express or implied. HCL does not assume any risk in relation to your use of this website. For more information, please go to <https://www.hcl-software.com/aopscan/>.

Copyright © 2008, 2017, IBM Corporation. All rights reserved. Copyright © 2017, 2024, HCL Technologies, Ltd., All rights reserved.

We obtained users from above. We have access to users' data and can reset their password.

We will now verify the user's authorization.

AltoroMutual

Sign Off | Contact Us | Feedback | Search [Go]

DEMO SITE ONLY

MY ACCOUNT

PERSONAL | **SMALL BUSINESS** | **INSIDE ALTORO MUTUAL**

Online Banking Login

Username: Password:

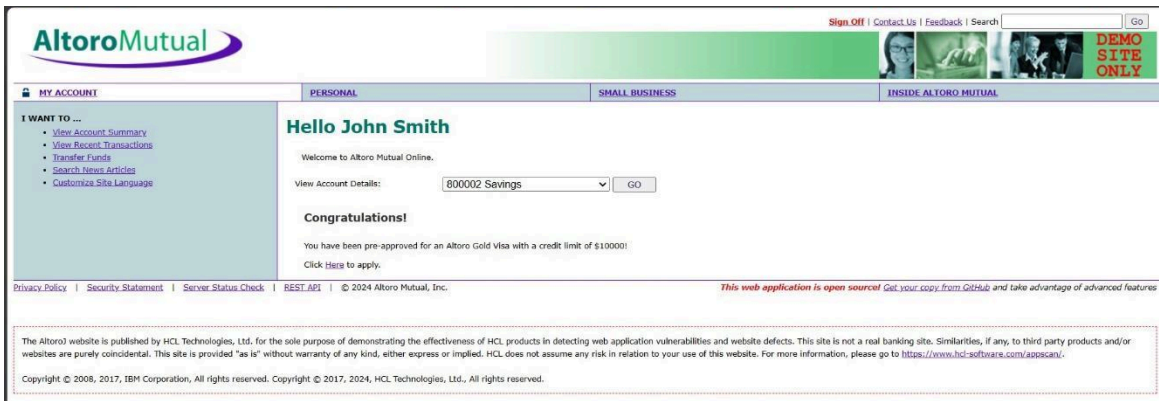
Privacy Policy | Security Statement | Server Status Check | REST API | © 2024 Altoro Mutual, Inc. *This web application is open source! Get your copy from GitHub and take advantage of advanced features*

The Altoro website is published by HCL Technologies, Ltd. for the sole purpose of demonstrating the effectiveness of HCL products in detecting web application vulnerabilities and website defects. This site is not a real banking site. Similarities, if any, to third party products and/or websites are purely coincidental. This site is provided "as is" without warranty of any kind, either express or implied. HCL does not assume any risk in relation to your use of this website. For more information, please go to <https://www.hcl-software.com/aopscan/>.

Copyright © 2008, 2017, IBM Corporation. All rights reserved. Copyright © 2017, 2024, HCL Technologies, Ltd., All rights reserved.

"Jsmith" is among the users that we discovered. However, we are utilising a payload to conflate the permission code and are using the password "1234" because we do not know the password.

The real username, "jsmith," is attached with '--' as the payload.



We have access to user transactions as well as their transaction history.

The screenshot shows the 'Recent Transactions' page on the AltoroMutual website. It features a table with columns for Transaction ID, Transaction Time, Account ID, Action, and Amount. The table lists various transactions, including deposits and withdrawals, with amounts ranging from \$10,000.00 to -\$12,34.00. The page also includes a search bar for filtering transactions by date range.

Transaction ID	Transaction Time	Account ID	Action	Amount
2418	2024-03-17 01:09	800003	Deposit	\$10000.00
2417	2024-03-17 01:09	800002	Withdrawal	-\$10000.00
2416	2024-03-17 01:03	800003	Deposit	\$1234.00
2415	2024-03-17 01:03	800003	Withdrawal	-\$1234.00
2414	2024-03-17 01:03	800003	Deposit	\$1234.00
2413	2024-03-17 01:03	800003	Withdrawal	-\$1234.00
2412	2024-03-17 01:03	800003	Deposit	\$1234.00
2411	2024-03-17 01:03	800003	Withdrawal	-\$1234.00
2410	2024-03-17 01:03	800003	Deposit	\$1234.00
2409	2024-03-17 01:03	800003	Withdrawal	-\$1234.00
2408	2024-03-17 01:03	800003	Deposit	\$1234.00
2407	2024-03-17 01:03	800003	Withdrawal	-\$1234.00
2406	2024-03-17 01:03	800003	Deposit	\$1234.00
2405	2024-03-17 01:03	800003	Withdrawal	-\$1234.00
2404	2024-03-17 01:03	800003	Deposit	\$1234.00
2403	2024-03-17 01:03	800003	Withdrawal	-\$1234.00
2402	2024-03-17 01:03	800003	Deposit	\$1234.00
2401	2024-03-17 01:03	800003	Withdrawal	-\$1234.00
2400	2024-03-17 01:03	800003	Deposit	\$1234.00
2399	2024-03-17 01:03	800003	Withdrawal	-\$1234.00
2398	2024-03-17 01:03	800003	Deposit	\$1234.00
2397	2024-03-17 01:03	800003	Withdrawal	-\$1234.00

Transferring funds:

AltoroMutual

Sign Off | Contact Us | Feedback | Search [] Go

MY ACCOUNT PERSONAL SMALL BUSINESS INSIDE ALTORO MUTUAL

I WANT TO ...

- View Account Summary
- View Recent Transactions
- Transfer Funds
- Search News Articles
- Customize Site Language

Transfer Funds

From Account: 800002 Savings

To Account: 800003 Checking

Amount to Transfer: 100000

Transfer Money

Privacy Policy | Security Statement | Server Status Check | REST API | © 2024 Altoro Mutual, Inc.

This web application is open source! Get your copy from GitHub and take advantage of advanced features

The Altoro website is published by HCL Technologies, Ltd. for the sole purpose of demonstrating the effectiveness of HCL products in detecting web application vulnerabilities and website defects. This site is not a real banking site. Similarities, if any, to third party products and/or websites are purely coincidental. This site is provided "as is" without warranty of any kind, either express or implied. HCL does not assume any risk in relation to your use of this website. For more information, please go to <https://www.hcl-software.com/privacy/>.

Copyright © 2008, 2017, IBM Corporation, All rights reserved. Copyright © 2017, 2024, HCL Technologies, Ltd., All rights reserved.

- Students can demonstrate how each identified vulnerability could be exploited using proof-of-concept attacks or simulation tools.
- For example, they could demonstrate how SQL injection attacks can be used to extract sensitive information from the database or how cross-site scripting (XSS) attacks can be used to execute malicious scripts in users' browsers.

Step 5: Mitigation Strategy Proposal:

1. Broken Access Control:

- Implement proper authentication and authorization mechanisms.
- Enforce least privilege access.
- Regularly audit and review access controls.

2. Cryptographic Failures:

- Use strong encryption algorithms and secure storage methods.
- Avoid hardcoded passwords and keys.
- Regularly update cryptographic libraries and protocols.

3. Injection:

- Use parameterized queries and input validation to prevent SQL injection.

- b. Employ output encoding to mitigate Cross-Site Scripting (XSS).
 - c. Implement strict OS command filtering to prevent command injection.
- 4. **Insecure Design:**
 - a. Conduct thorough threat modeling and security reviews during the design phase.
 - b. Follow secure coding practices and principles. Implement secure design patterns and controls.
- 5. **Security Misconfiguration:**
 - a. Follow secure configuration guidelines for servers, frameworks, and cloud services.
 - b. Regularly audit and update configurations.
 - c. Utilize automated tools for configuration scanning and validation.
- 6. **Vulnerable and Outdated Components:**
 - a. Maintain an inventory of third-party components and libraries. Regularly update and patch components to address known vulnerabilities.
 - b. Use Software Bill of Materials (SBOM) to track dependencies and vulnerabilities.
- 7. **Identification and Authentication Failures:**
 - a. Implement multi-factor authentication (MFA) where possible.
 - b. Enforce strong password policies and account lockout mechanisms.
 - c. Monitor for abnormal login activities and brute force attempts.
- 8. **Software and Data Integrity Failures:**
 - a. Implement code signing and verification mechanisms.
 - b. Employ secure build and deployment pipelines.
 - c. Regularly scan code for vulnerabilities and malicious code.
- 9. **Security Logging and Monitoring Failures:**
 - a. Implement comprehensive logging across systems and applications.
 - b. Set up alerting mechanisms for suspicious activities.
 - c. Regularly review logs and conduct incident response drills.
- 10. **Server-Side Request Forgery (SSRF):**
 - a. Validate and sanitize user-supplied URLs to prevent SSRF.
 - b. Use whitelists to restrict allowed destinations for outbound requests.
 - c. Implement rate limiting and request validation mechanisms.

Step 6: Documenting the Exploit Process:

- ➔ Document the exploit process, including the commands used, the output received, and any challenges encountered.