Team 4.3
Vulnerabilities of practice website

Website: http://testphp.vulnweb.com

Tools used:

1. Kali Linus root terminal

2. SQL map

3. Burpsuite

1. Vulnerability name: Cross-Site Request Forgery (CSRF)

CWE: 352

OWASP Category: A05:2021 – Broken Session Management

Description:

Cross-Site Request Forgery (CSRF) is a web security vulnerability that allows an attacker to induce an authenticated user to perform an unwanted action on a website. CSRF attacks are often carried out by tricking the victim into clicking on a malicious link or opening a specially crafted web page.

CSRF attacks exploit the fact that web browsers automatically send authentication cookies with every request. When a user is authenticated on a website, their browser stores an authentication cookie on their computer. This cookie is used to identify the user to the website on subsequent requests.

An attacker can exploit this behaviour by constructing a malicious request that is designed to be executed by the victim's browser. The request will typically contain the victim's authentication cookie, so the website will believe that the request is coming from the victim.

If the request is successful, the attacker can cause the victim to perform any action that they are authorised to perform on the website. For example, the attacker could transfer money from the victim's bank account, change their password, or post a message on a social media account.

Business impact:

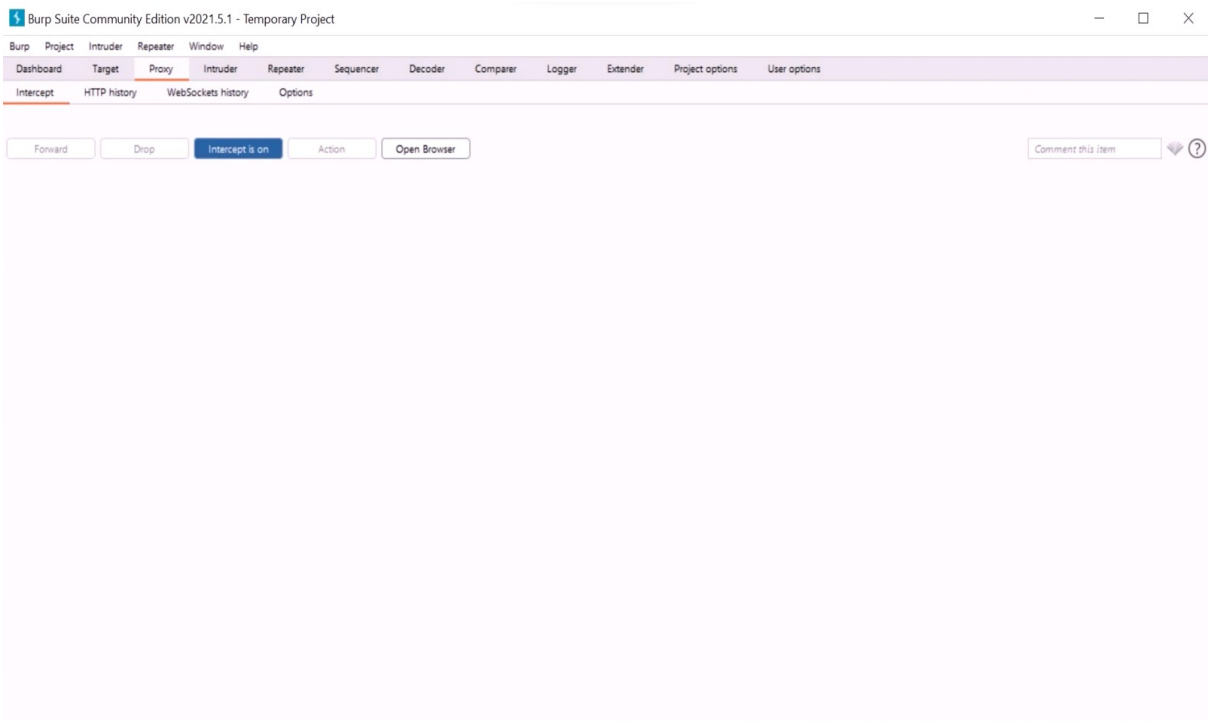
Cross-Site Request Forgery (CSRF) can have a significant business impact, leading to unauthorized actions on behalf of authenticated users. This can result in data breaches, financial losses, damage to reputation, and legal liabilities, affecting customer trust and overall business operations.

Vulnerability path: http://testphp.vulnweb.com/

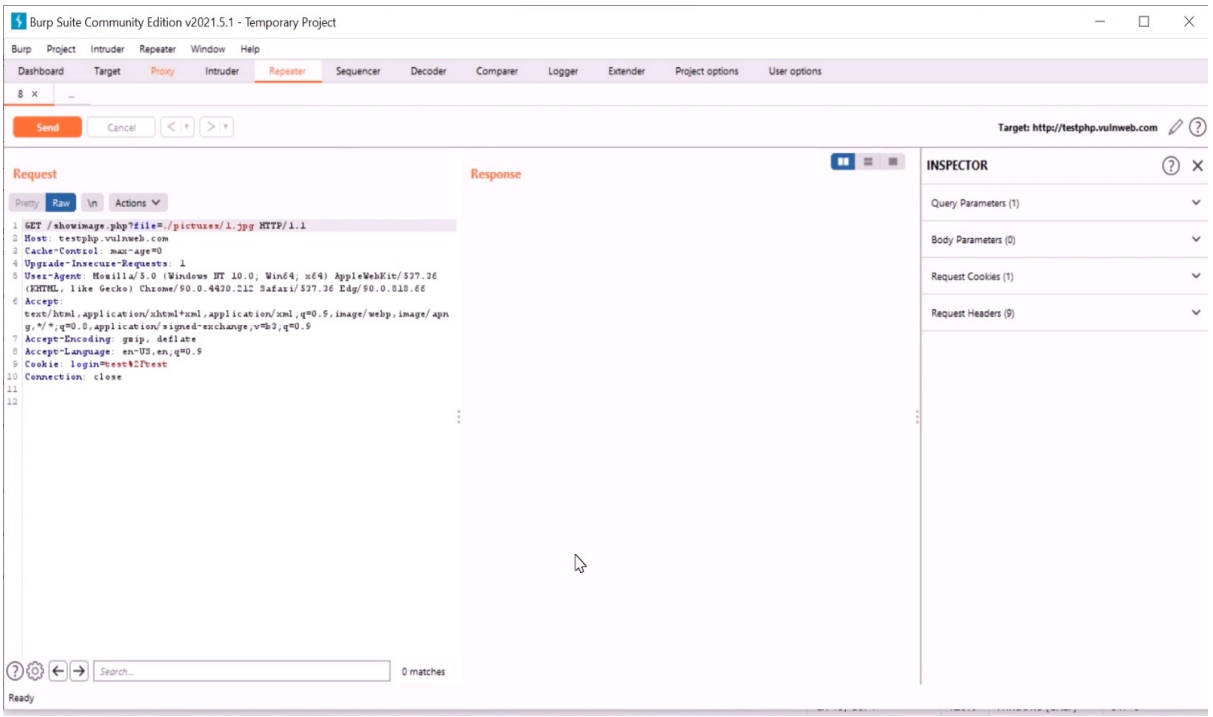
Vulnerable Parameter: http://testphp.vulnweb.com/showimage.php?file=./pictures/1.jpg

Steps to reproduce:

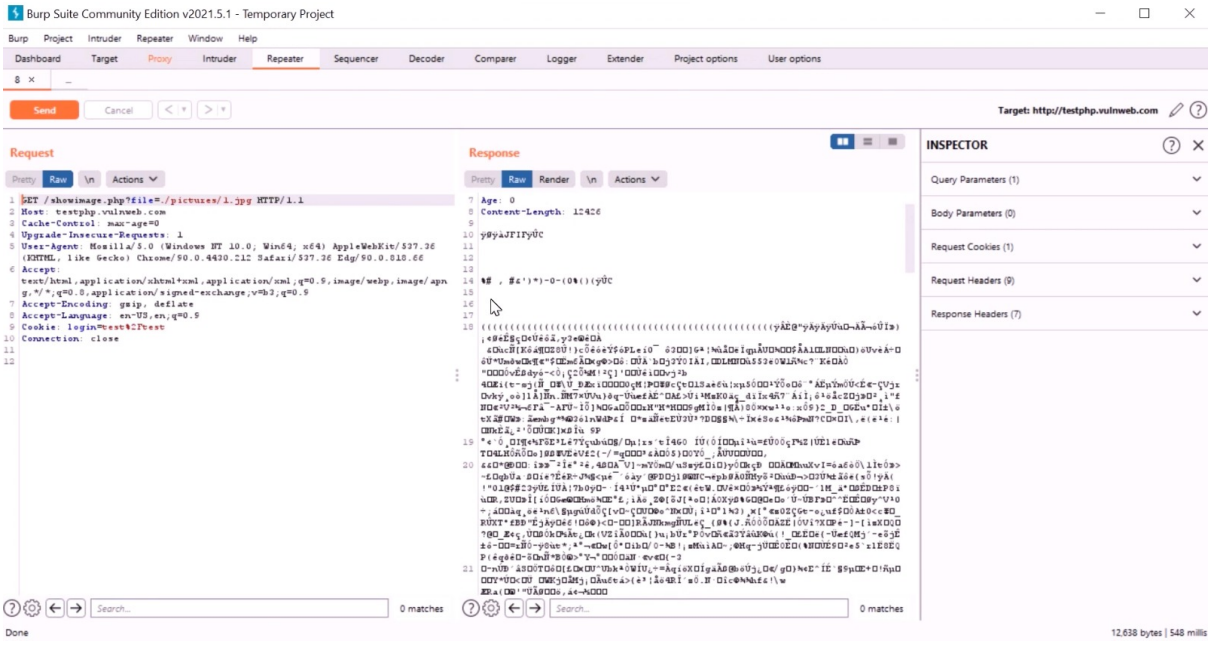
1. Intercept it with burp suite



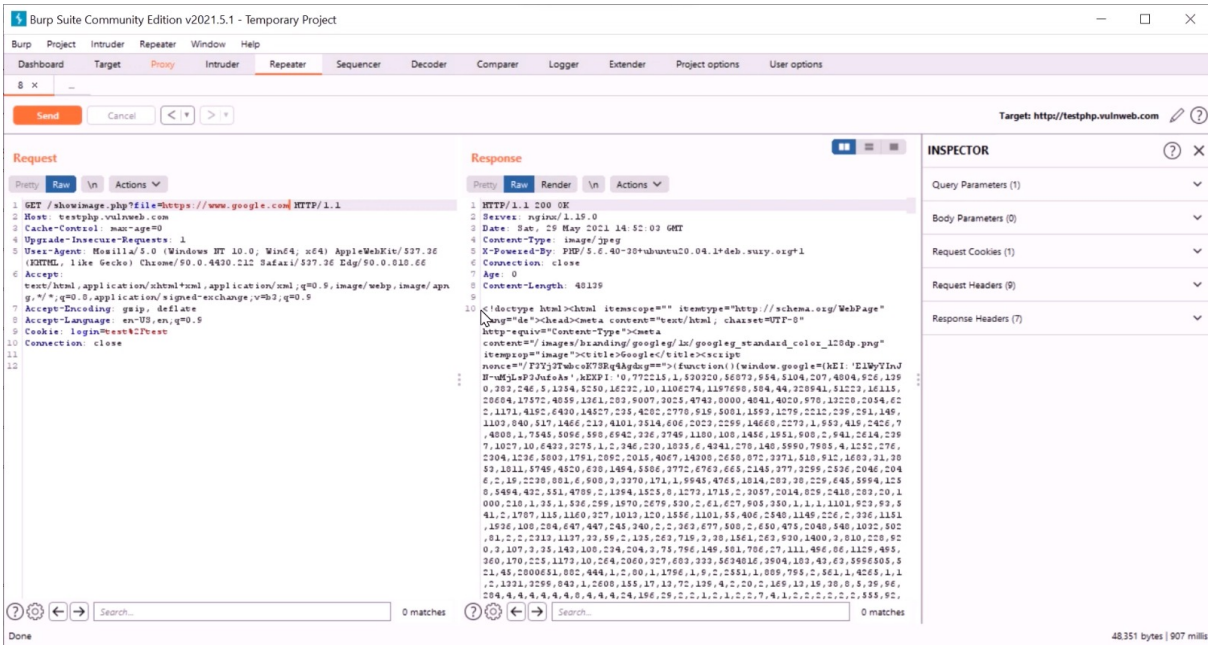
2. Send it to repeater



3. See the response of “http://testphp.vulnweb.com/showimage.php?file=./pictures/1.jpg”.



4. Change the parameter of the file from: ./pictures/1.jpg ,, to: https://www.google.com and see the response.

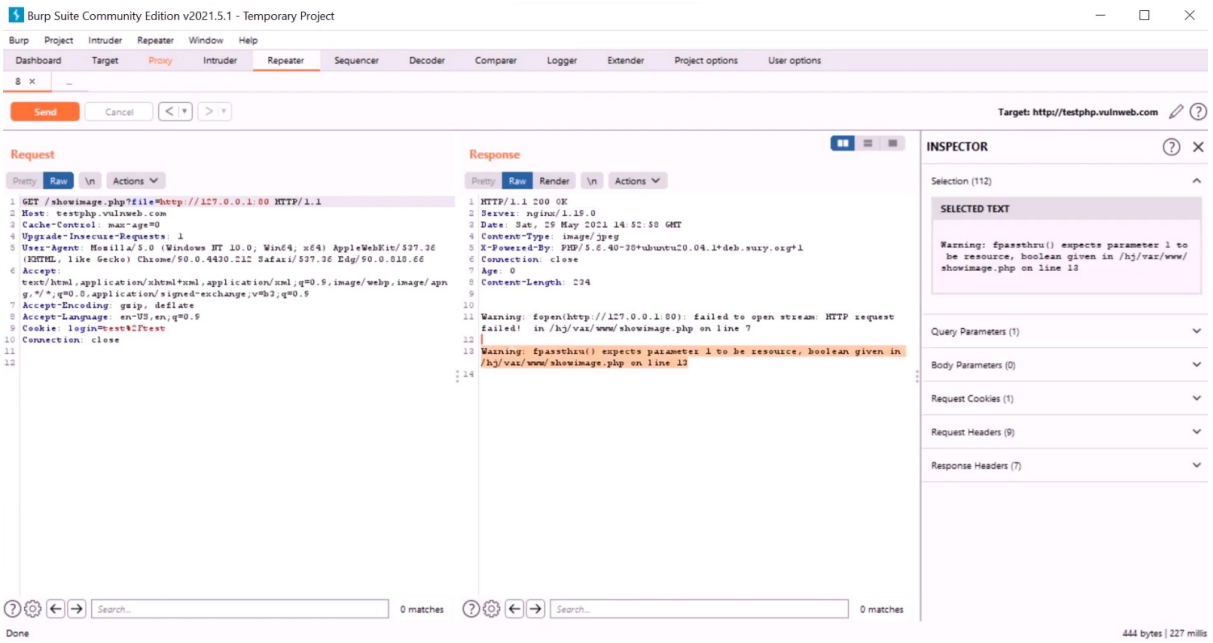


5. As we can see the response of http://testphp.vulnweb.com/showimage.php?file=https://www.google.com is different, the response changed to https://www.google.com ,, which means that it’s vuln against CSRF

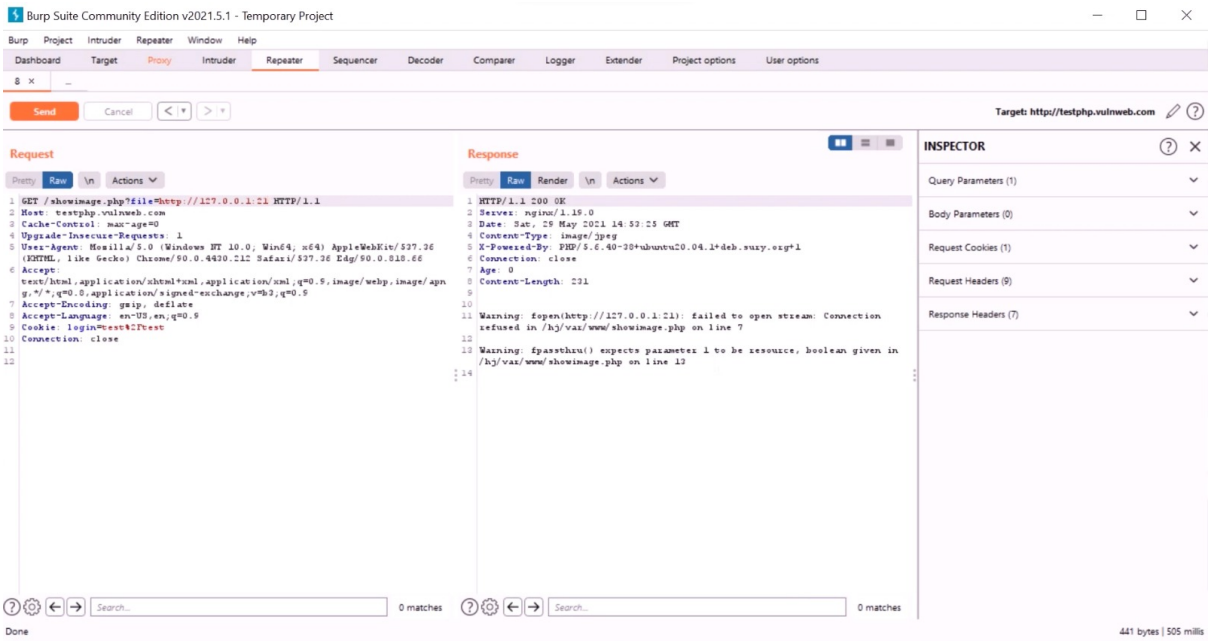
6. Now we want to scan the port of it

7. Change the parameter to http://127.0.0.1 :(port number)

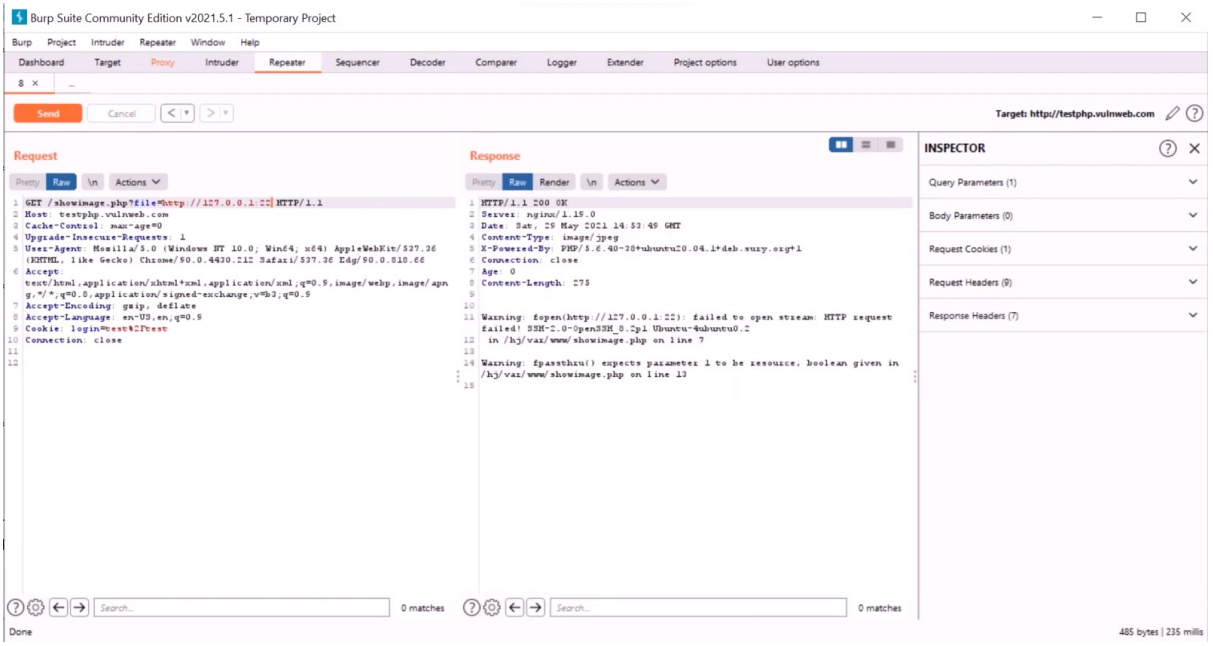
8. Check for port number 80



9. Now check for port number 21



10. Now check for port number 22



11. As we can see port 22 responds by showing the information that the backend service is running on SSH which means that port 22 is open.

Mitigation:

Use synchronizer tokens (also known as anti-CSRF tokens or nonce tokens). A synchronizer token is a unique value that is generated by the server and sent to the client with each form. When the client submits the form, they must include the synchronizer token in the request. The server will then verify that the synchronizer token is valid before processing the request.

Use HTTP headers such as SameSite and Origin. The SameSite and Origin headers can be used to restrict the websites that can send requests to your website. This can help to prevent CSRF attacks from being carried out from third-party websites.

Educate users about CSRF attacks. It is important to educate users about CSRF attacks so that they can be more aware of the risks. Users should be careful about clicking on links in emails and chat messages, and they should be careful about visiting unknown websites.

2. Vulnerability name : Information Exposure

CWE: 200

OWASP Category: A01:2021- Broken Access Control

Description: Information exposure is a security vulnerability that allows unauthorized access to sensitive information. Sensitive information can include things like personal data, financial data, health records, and trade secrets. Information exposure can occur in a variety of ways, including:

Misconfigured systems: Systems that are not properly configured can expose sensitive information to unauthorized users. For example, a web server that is configured to allow public access to its log files could expose sensitive information about the websites that are hosted on the server.

Software vulnerabilities: Software vulnerabilities can also allow attackers to expose sensitive information. For example, a vulnerability in a web application could allow an attacker to inject malicious code into the application, which could then be used to steal sensitive data from the application's database.

Human error: Human error can also lead to information exposure. For example, an employee may accidentally send an email containing sensitive information to the wrong person.

Business impact:

Information exposure can have a number of negative consequences, including:

Financial loss: Information exposure can lead to financial losses for individuals and organizations. For example, if an attacker steals credit card numbers, they can use them to make unauthorized purchases.

Identity theft: Information exposure can also lead to identity theft. For example, if an attacker steals someone's Social Security number, they can use it to open new credit accounts or file fraudulent tax returns.

Reputational damage: Information exposure can also damage an organization's reputation. For example, if a company exposes its customers' personal information, it could lose the trust of its customers.

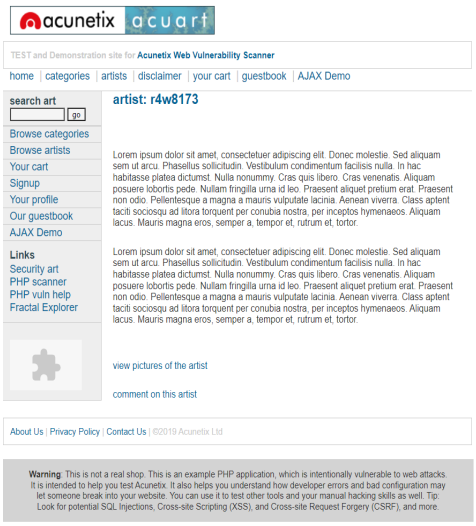
Vulnerable Path: http://testphp.vulnweb.com/index.php

Vulnerable Parameter: http://testphp.vulnweb.com/artists.php?artist=1

http://testphp.vulnweb.com/listproducts.php?cat=1

Steps to reproduce:

1. Access the URL and open up to the below page as shown in figure



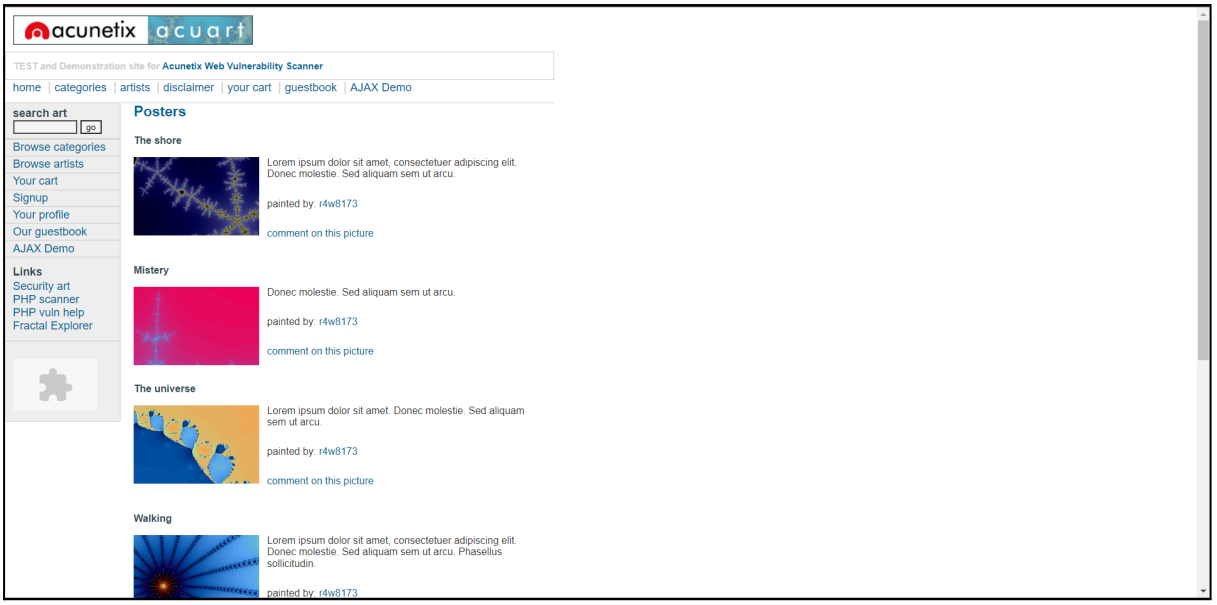
---

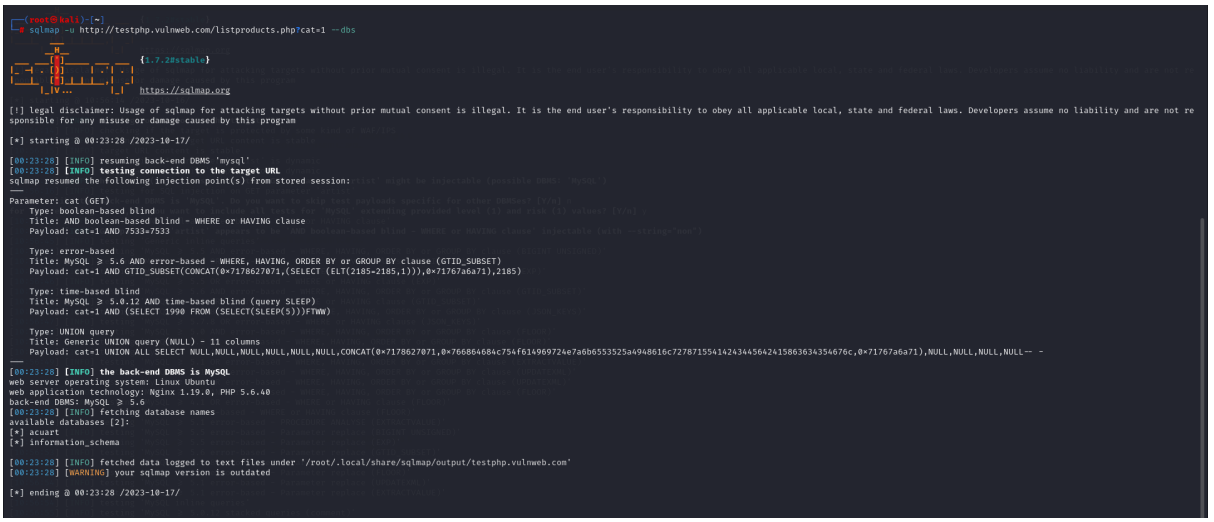
2. Copy the URL for the page

3. Open up the root terminal in Kali Linux

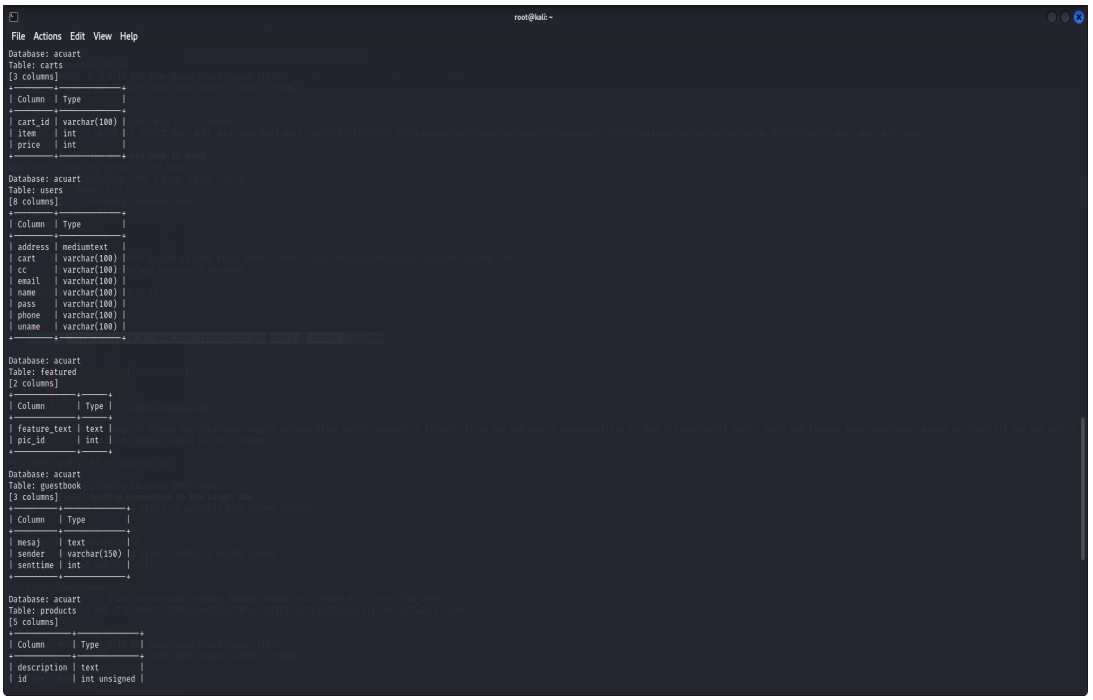
4. Now type the following in double quotes “sqlmap -u paste the URL here --dbs”, like the below figure.

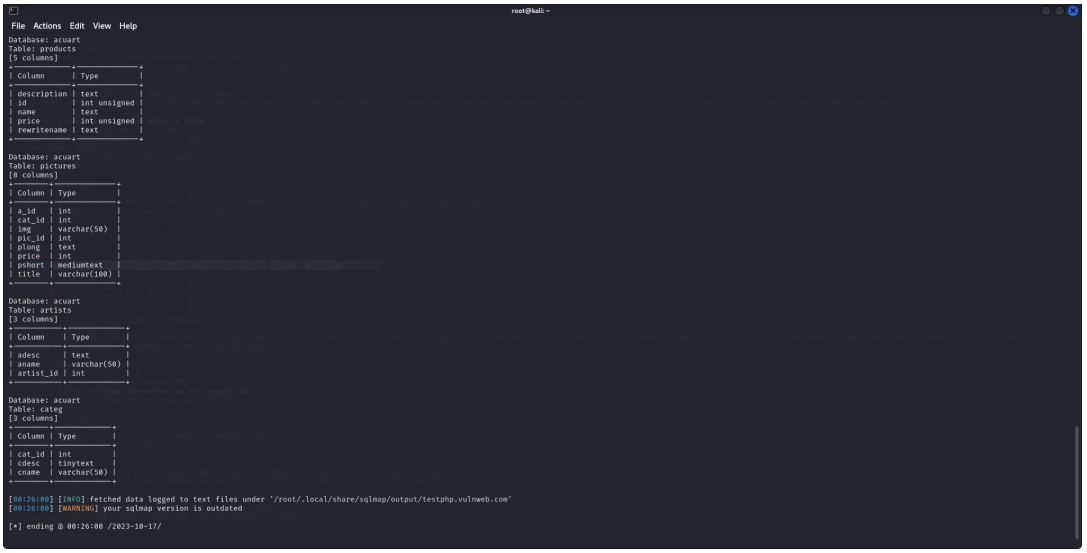
5. Now repeat the same for the 2nd parameter



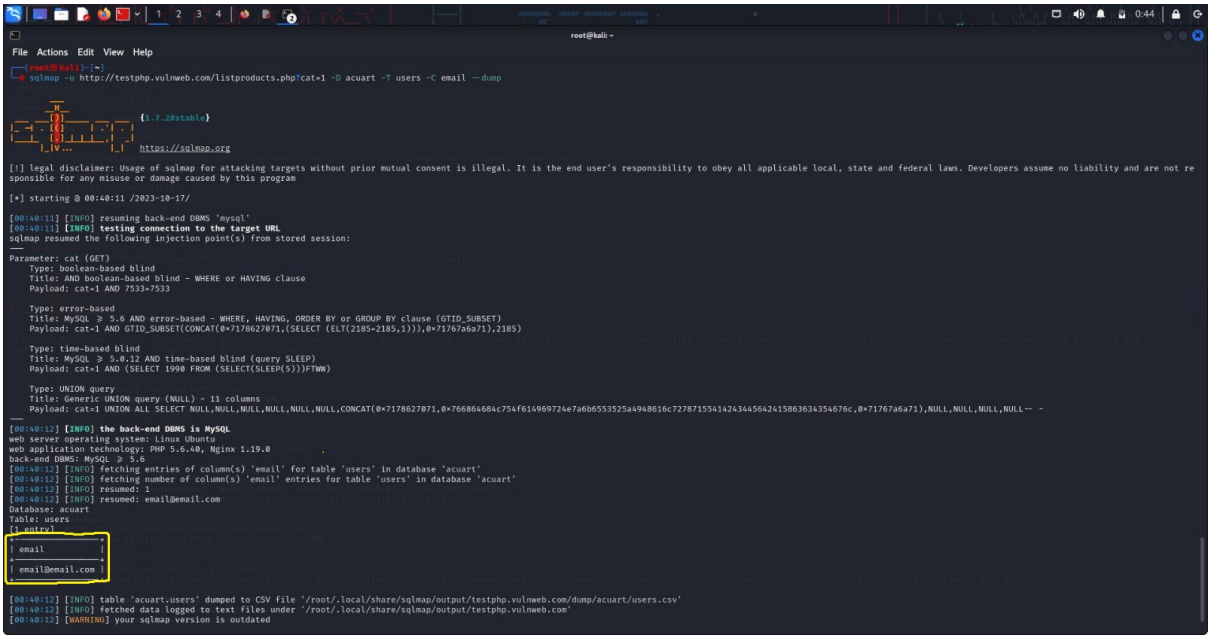


6. Now type this “sqlmap -u http://testphp.vulnweb.com/listproducts.php?cat=1 -D acuart --columns” This gives the information of everything in the database.





7. Now type this “sqlmap -u http://testphp.vulnweb.com/listproducts.php?cat=1 -D acuart -T users -C email --dump” This gives us the user email id by searching in the users table.



8. Here the the yellow highlighted box contains the email of a user
Note: I have registered to give a test email that is “email@email.com”

Mitigation:

There are a number of things that organizations can do to prevent information exposure, including:

Implementing strong security controls: Organizations should implement strong security controls, such as firewalls, intrusion detection systems, and access control lists, to protect their systems from unauthorized access.

Keeping software up to date: Organizations should keep their software up to date to patch known vulnerabilities.

Educating employees about security: Organizations should educate their employees about security best practices, such as how to create strong passwords and how to avoid phishing attacks.

3. Vulnerability Name: Exposure of Backup File to an Unauthorised Control Sphere

CWE: 530

OWASP Category: A03:2021 – Data Exposure

Description: This vulnerability occurs when backup files, which may contain sensitive data, are accessible to unauthorised entities or control spheres. Backup files often have less security measures in place compared to the live production data. If not properly protected, they can be exploited by attackers to gain access to sensitive information.

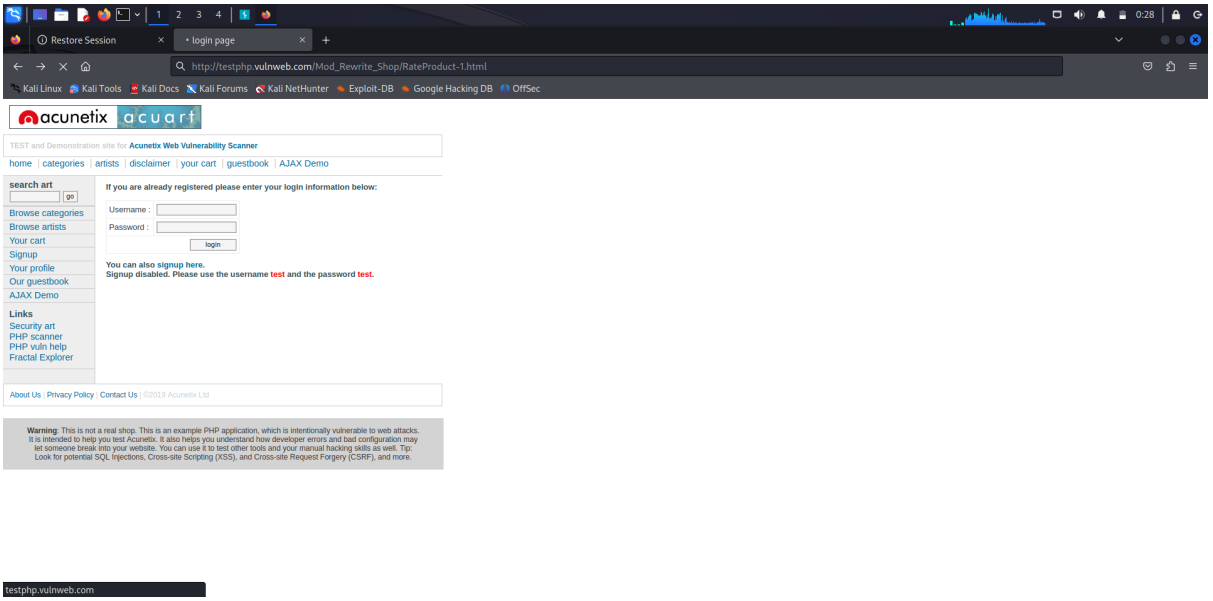
Business Impact: Exposure of backup files can lead to data breaches, data leakage, and unauthorised access to critical information. This can result in reputation damage, legal consequences, and financial losses for organisations.

Vulnerability path: http://testphp.vulnweb.com/

Vulnerability parameter:
http://testphp.vulnweb.com/Mod_Rewrite_Shop/RateProduct-1.html

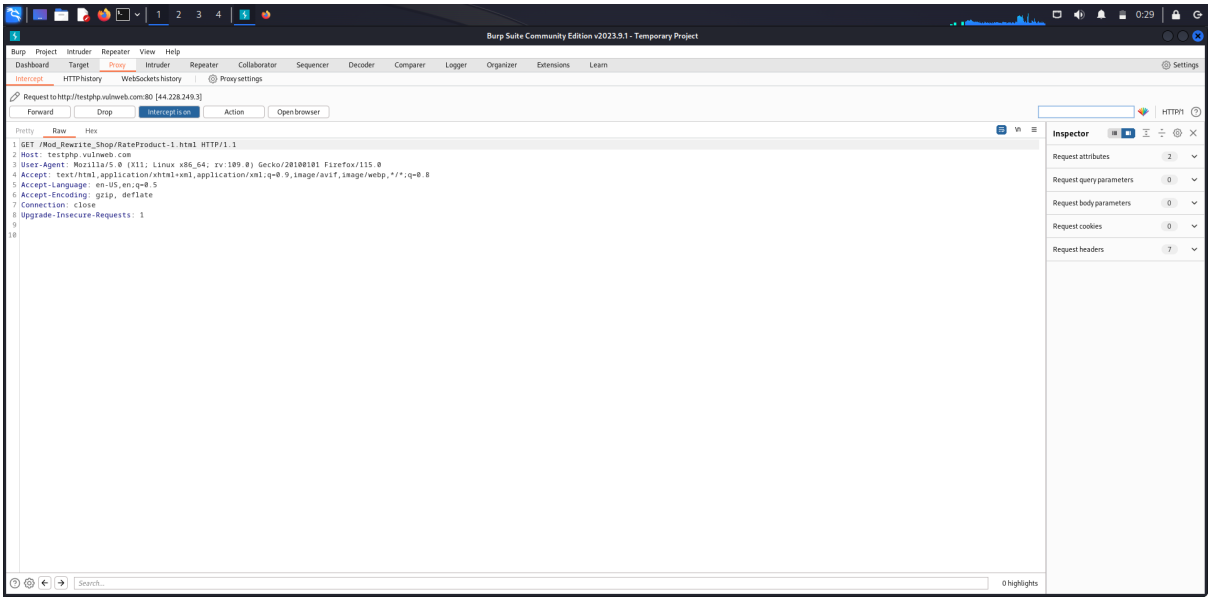
Steps to reproduce:

1. Access the website.



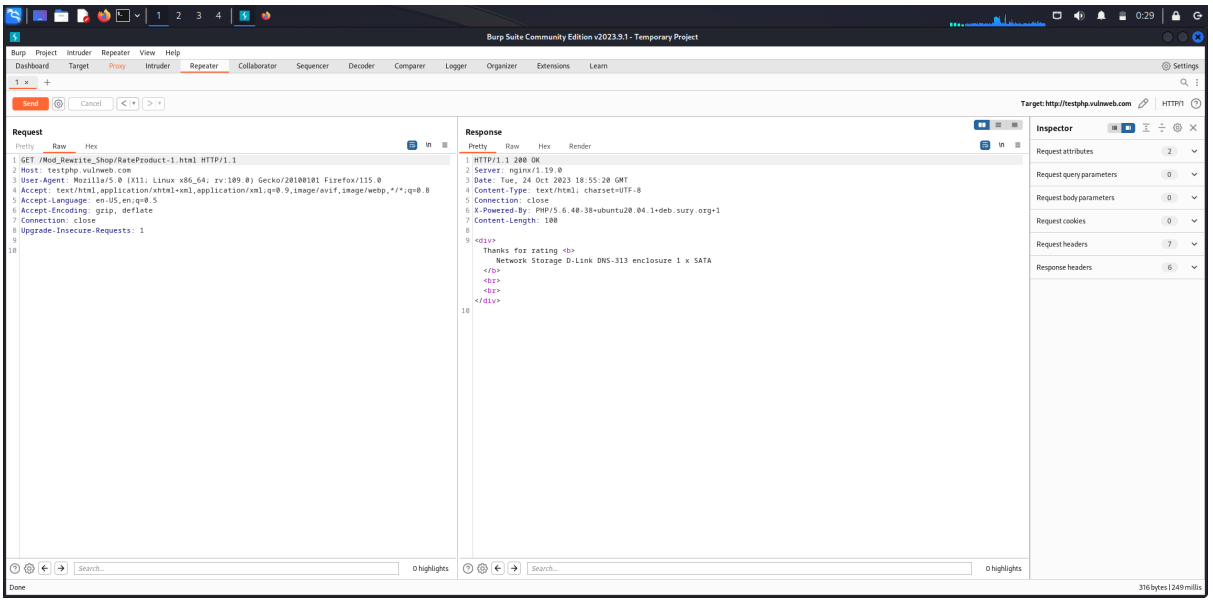
2. Use the parameter Mod_Rewrite_Shop/RateProduct-1.html

3. The information about the website will open in burpsuite.

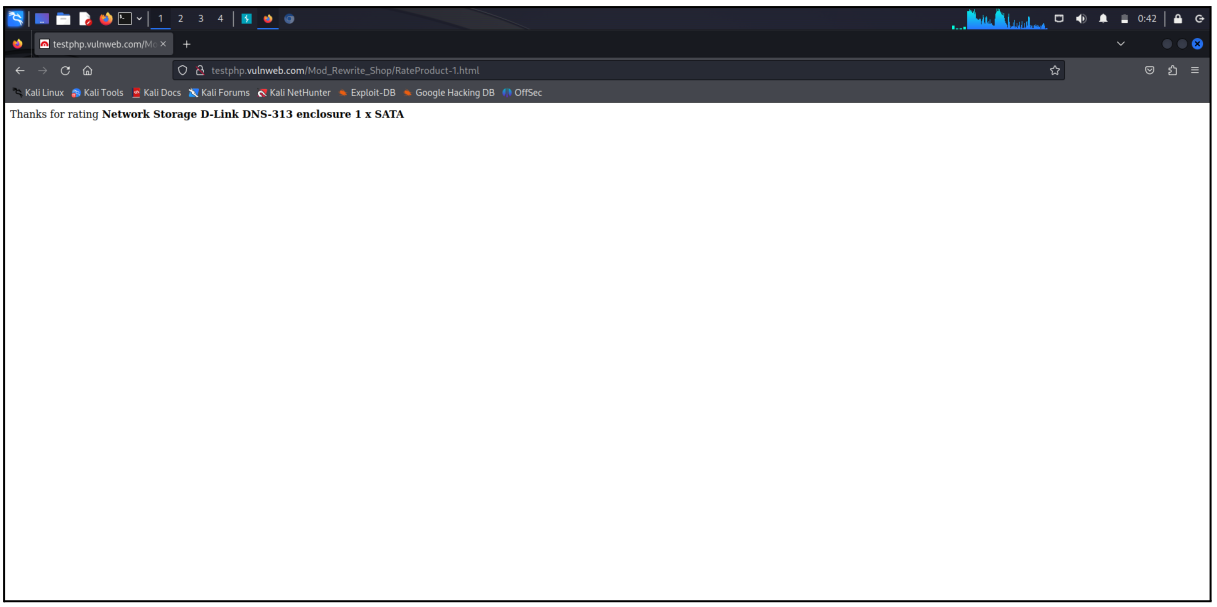


4. Send this to the repeater.

5. In the response, we can see the details of the website where we used the parameter..



6. Web interface of the link used



Mitigation: To mitigate this vulnerability, ensure that backup files are stored securely, implement access controls, encryption, and regular security assessments to identify and address any vulnerabilities in the backup process.

4. Vulnerability Name: Improper Input Validation

CWE: 20

OWASP Category: A01:2021 – Injection

Description: Improper input validation occurs when an application does not adequately validate user inputs, allowing attackers to inject malicious code or data. This can lead to various types of attacks, including SQL injection, cross-site scripting (XSS), and command injection, depending on the context.

Business Impact: Improper input validation can result in unauthorised data access, data manipulation, and even full compromise of the application or system. This can lead to data breaches, loss of user trust, and damage to an organisation's reputation.

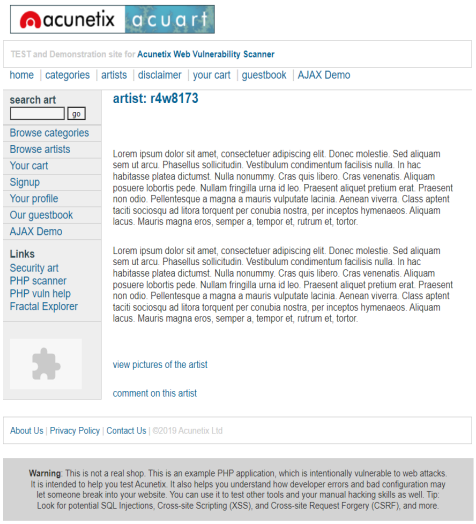
Vulnerable Path: http://testphp.vulnweb.com/index.php

Vulnerable Parameter: http://testphp.vulnweb.com/artists.php?artist=1

http://testphp.vulnweb.com/listproducts.php?cat=1

Steps to reproduce::

1. Access the URL and open up to the below page as shown in figure

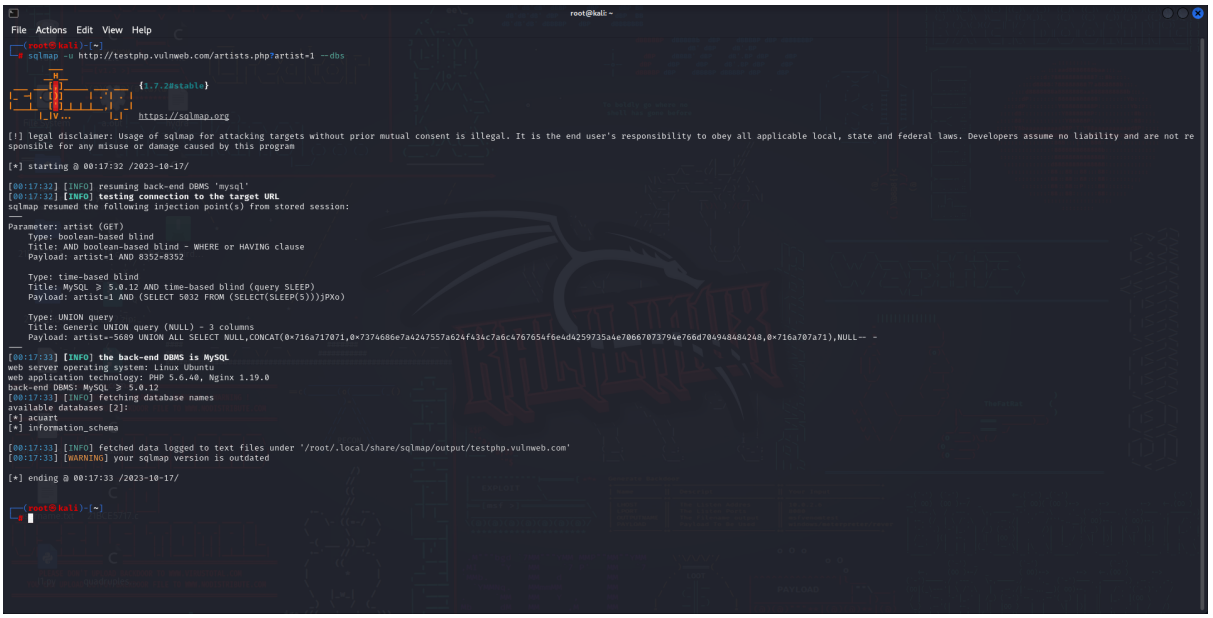


---

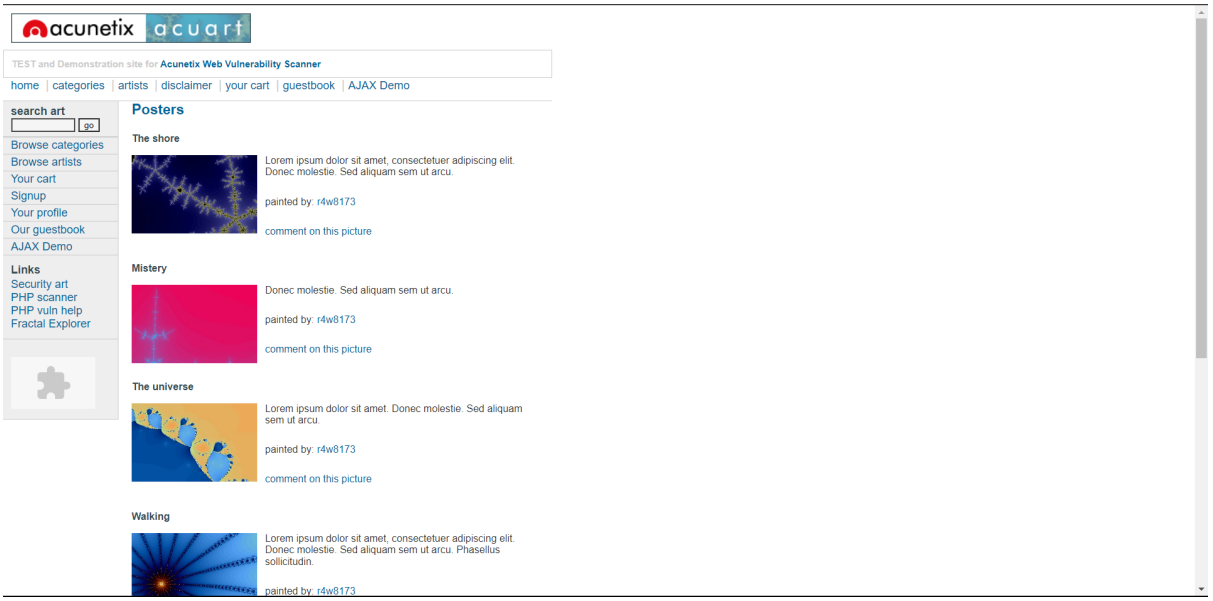
2. Copy the URL for the page

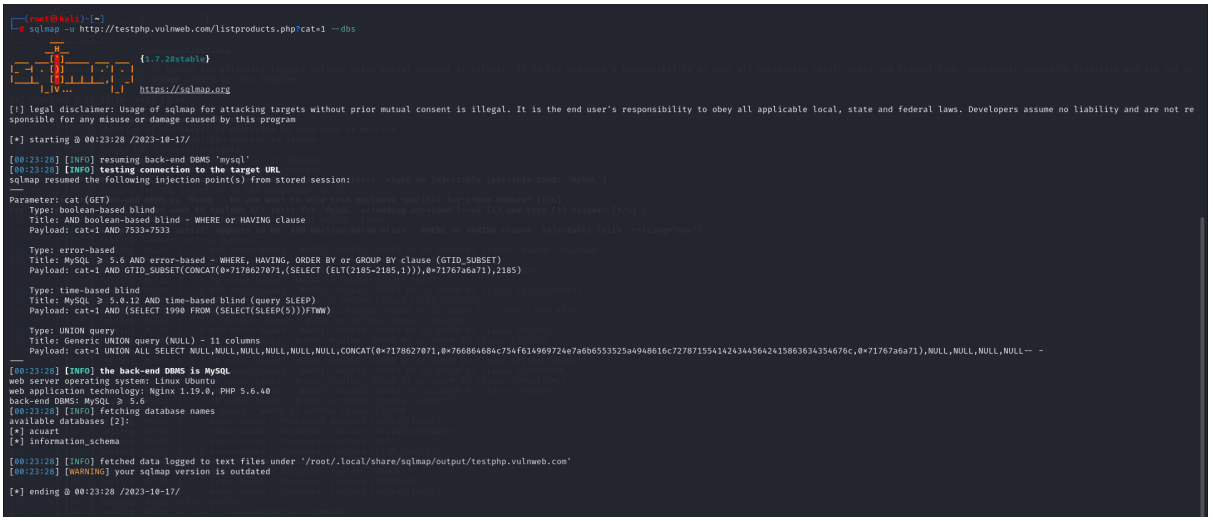
3. Open up the root terminal in Kali Linux

4. Now type the following in double quotes “sqlmap -u paste the URL here --dbs”, like the below figure.

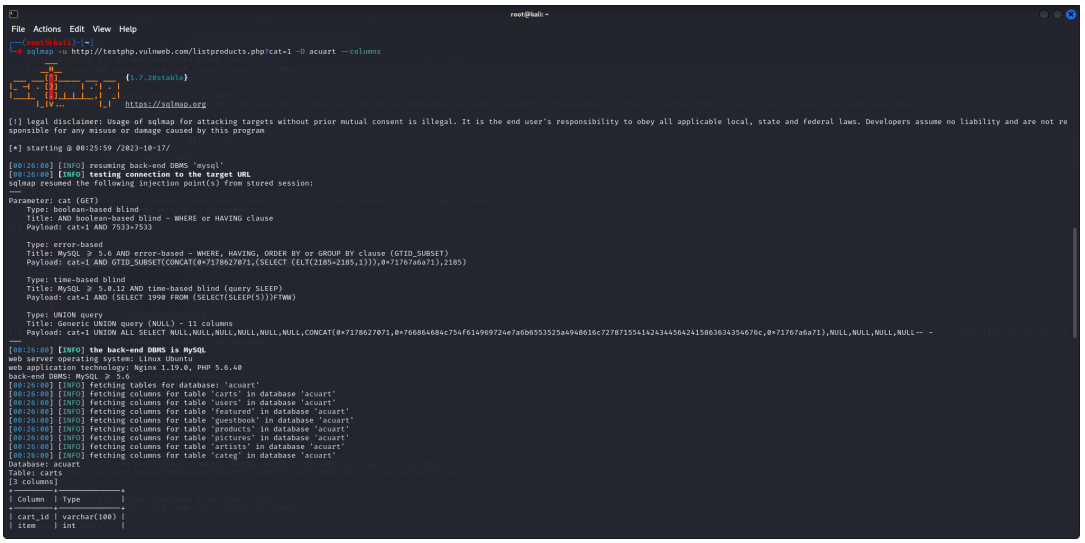


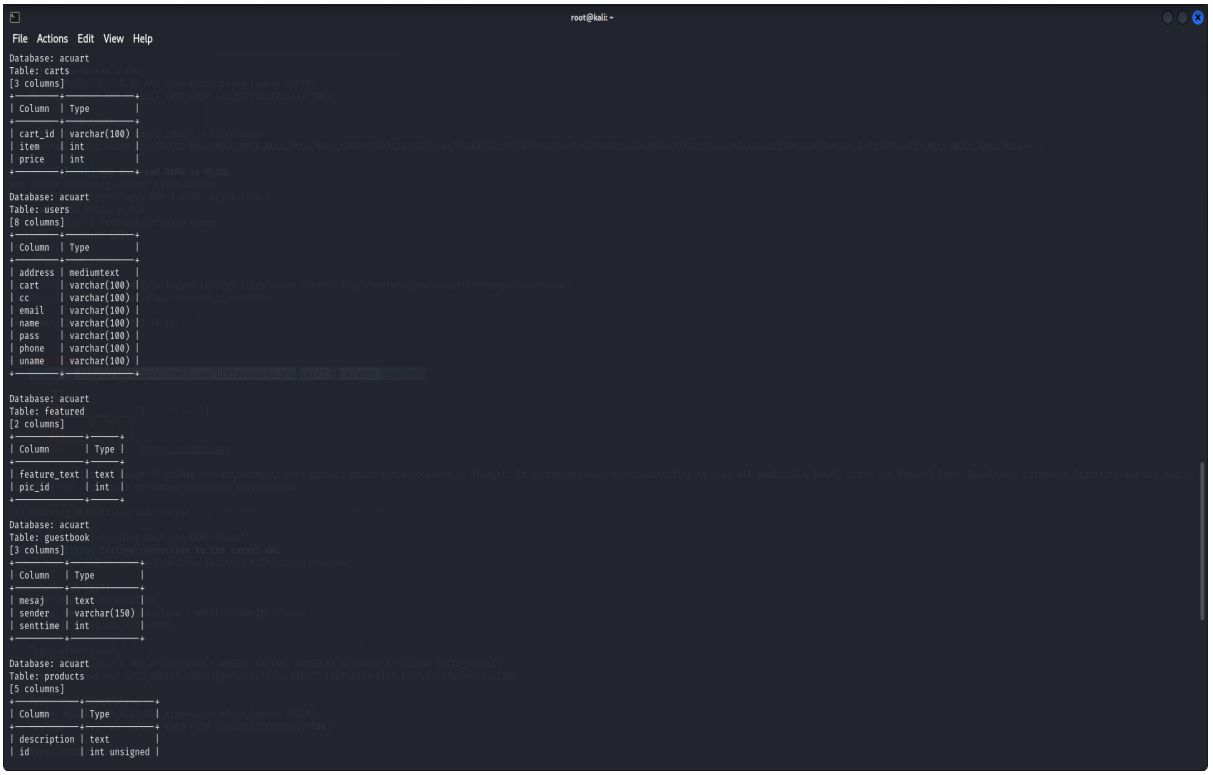
5. Now repeat the same for the 2nd parameter

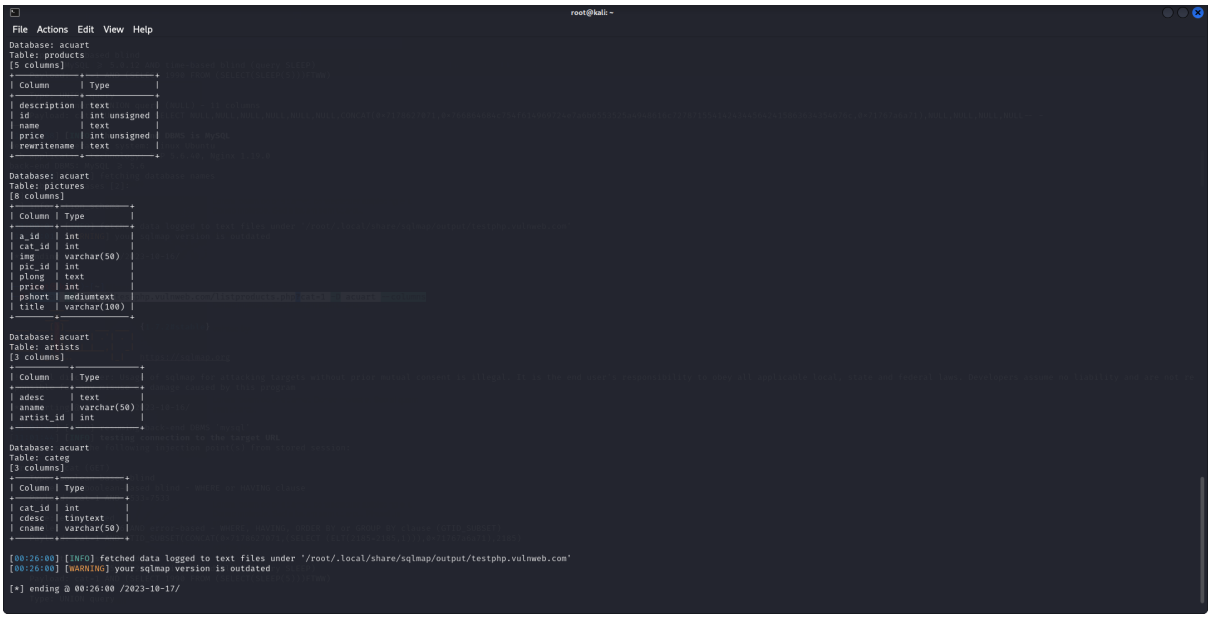




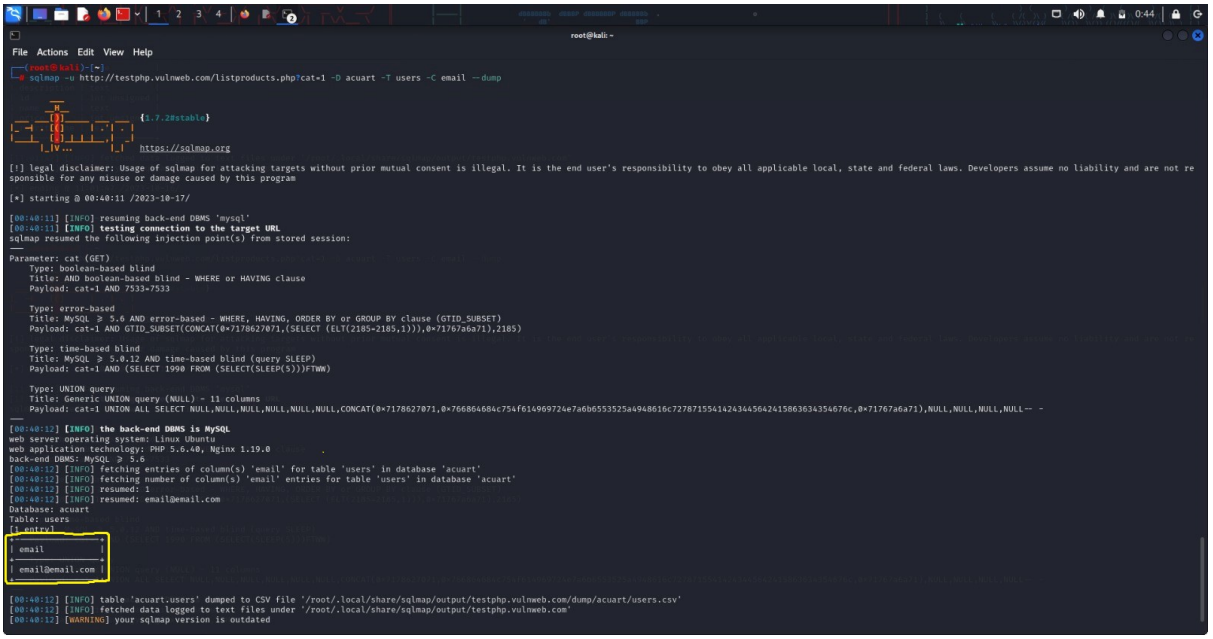
6. Now type this “sqlmap -u http://testphp.vulnweb.com/listproducts.php?cat=1 -D acuart --columns” This gives the information of everything in the database.







7. Now type this “sqlmap -u http://testphp.vulnweb.com/listproducts.php?cat=1 -D acuart -T users -C email --dump” This gives us the user email id by searching in the users table.



8. Here the the yellow highlighted box contains the email of a user
Note: I have registered to give a test email that is “email@email.com”

Mitigation: To mitigate this vulnerability, implement strict input validation routines, sanitise and filter user inputs, and use parameterized queries for database interactions. Additionally, employ web application firewalls (WAFs) to detect and block malicious input attempts. Regular security testing and code reviews are crucial to identify and address such issues.

5. Vulnerability name: Overly Permissive Cross-domain Whitelist

CWE: 942

OWASP Category: A05: 2021- Security Misconfiguration

Description: This vulnerability occurs when a web application permits overly broad cross-origin requests in its cross-domain whitelist, potentially allowing unauthorised parties to access sensitive data.

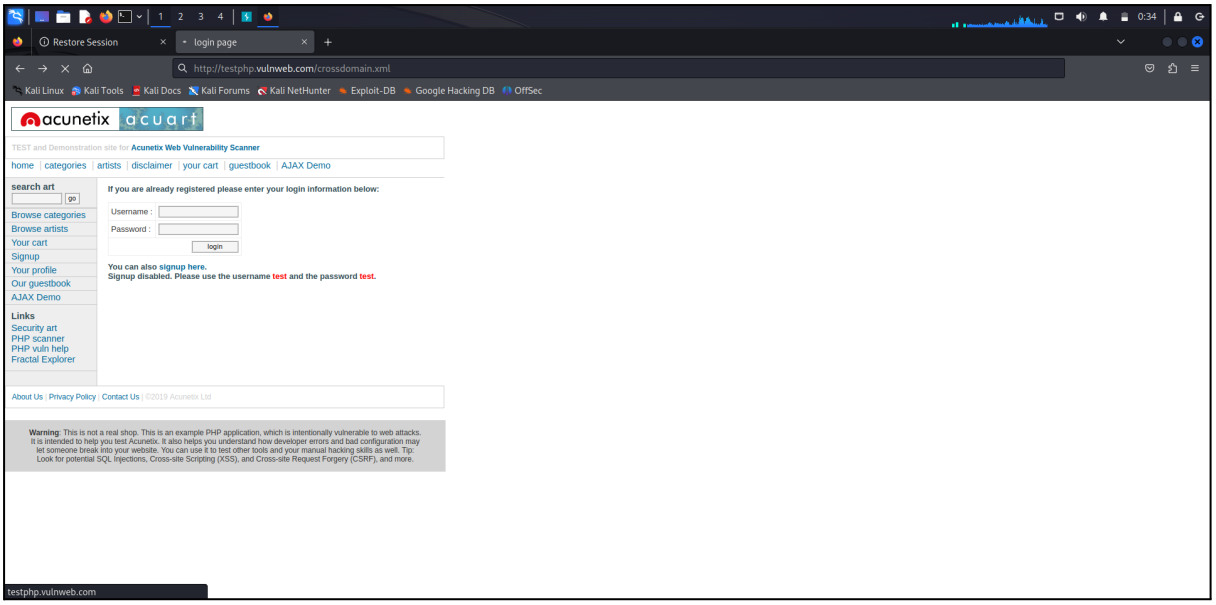
Business Impact: Allowing unrestricted cross-domain requests can enable data exposure, cross-site request forgery (CSRF), and other security breaches. This can lead to unauthorised data access, reputational damage, and legal consequences.

Vulnerability path: http://testphp.vulnweb.com/

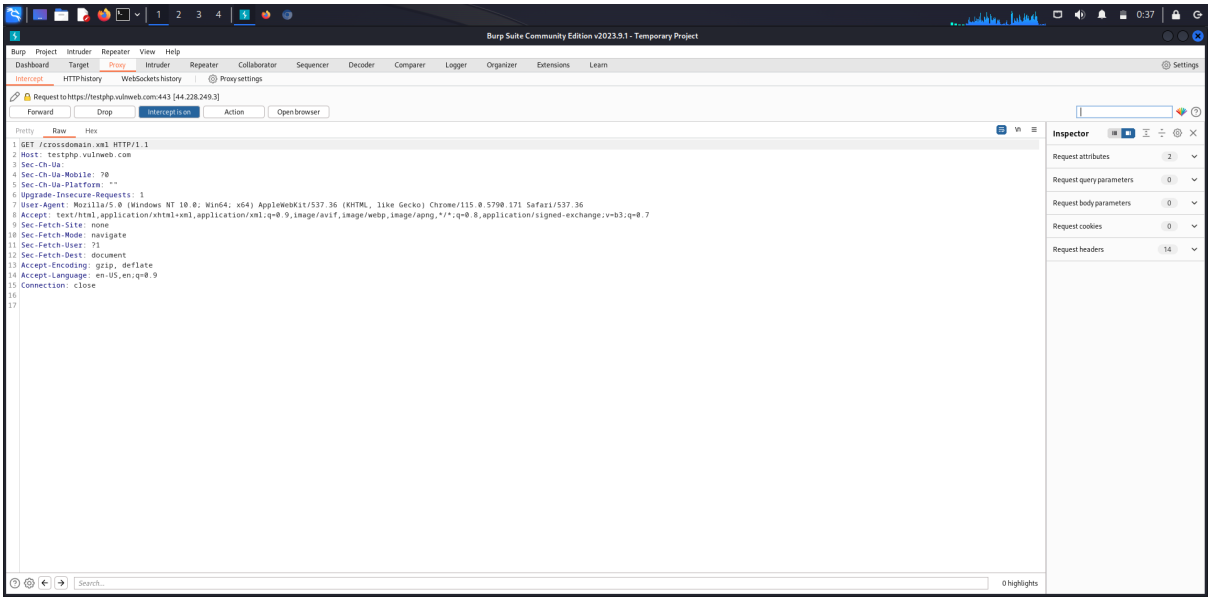
Vulnerability parameter: http://testphp.vulnweb.com/crossdomain.xml

Steps to reproduce:

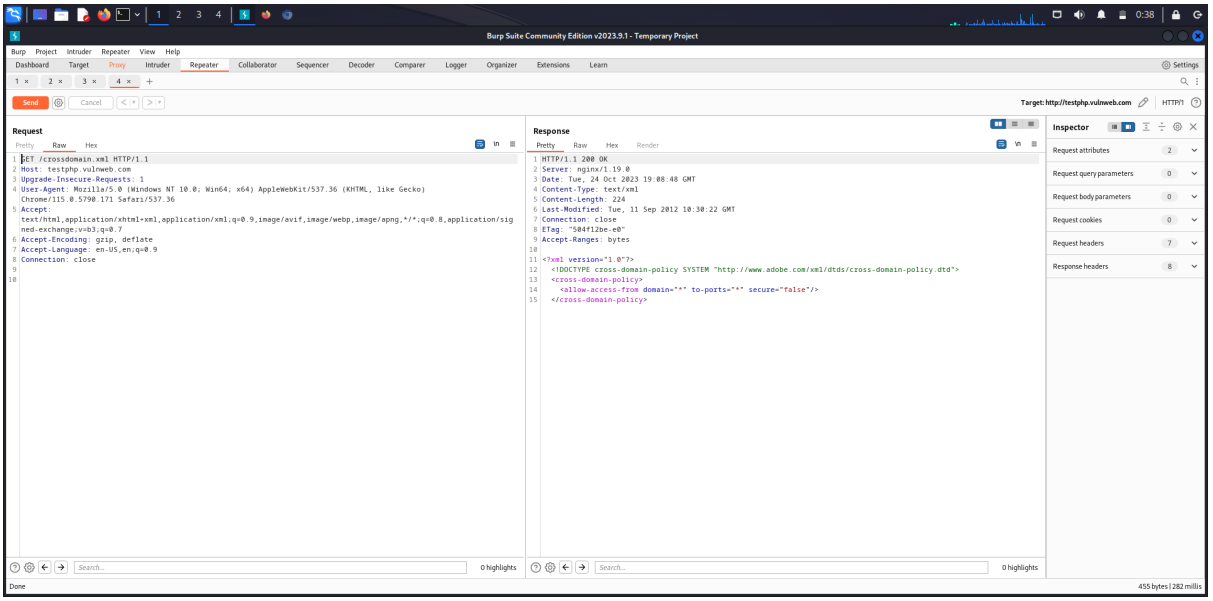
1. Access the website and use the parameter /crossdomain.xml



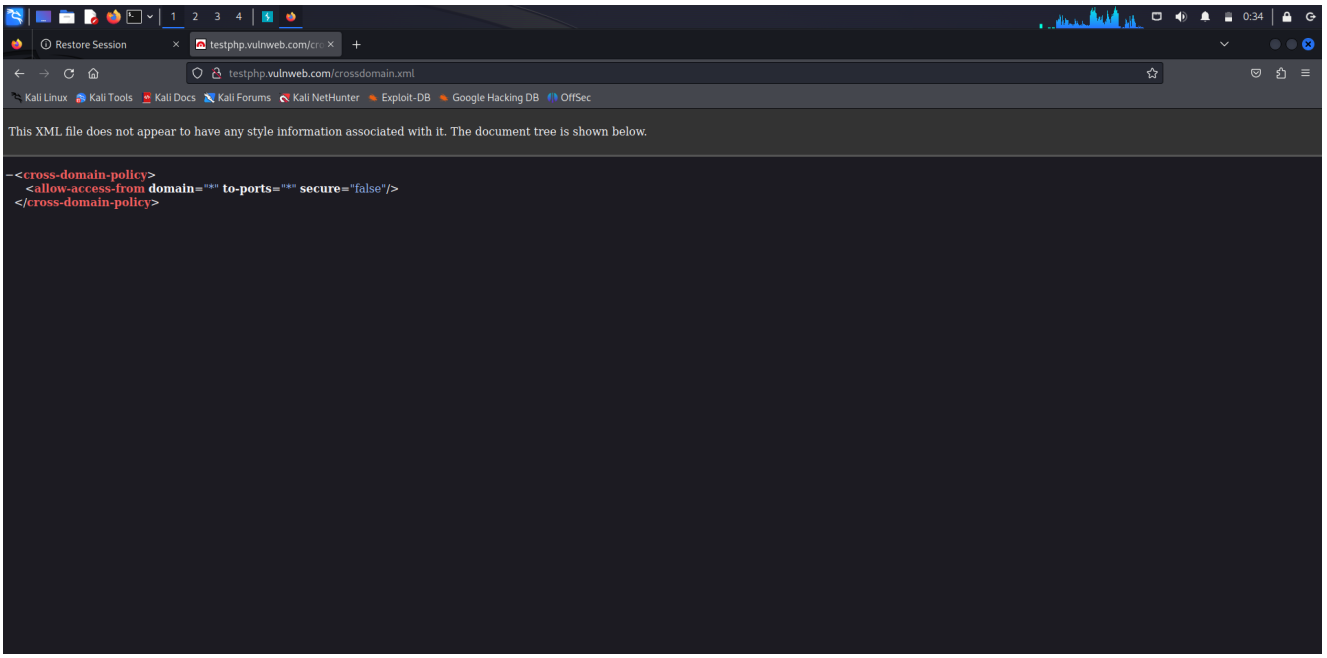
2. The information of the website entered will be shown in burpsuite.



3. Send the details under the proxy tab to the repeater tab.
4. Under the response tab, we can see the details regarding the vulnerability.



5. Website interface of the parameter.



Mitigation:

Implement a strict Content Security Policy (CSP) to limit which domains can interact with your web application.

Validate and sanitise data received from cross-origin requests to ensure it's used securely.

6. Vulnerability name: Inadequate Encryption Strength

CWE: 326

OWASP Category: A02:2021- Cryptographic Failures

Description: Inadequate encryption strength refers to the use of weak or outdated encryption algorithms and key lengths, which can be exploited by attackers to decrypt sensitive data.

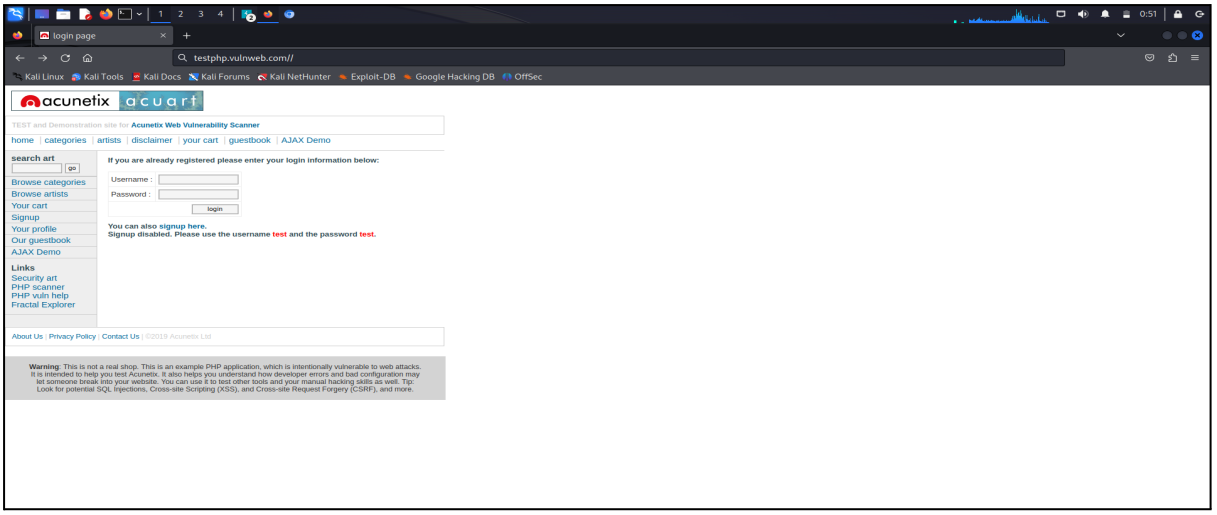
Business Impact: Weak encryption can lead to unauthorised access to confidential information, such as customer data or trade secrets. This can result in financial losses, legal liabilities, and harm to the organisation's credibility.

Vulnerability path: http://testphp.vulnweb.com/

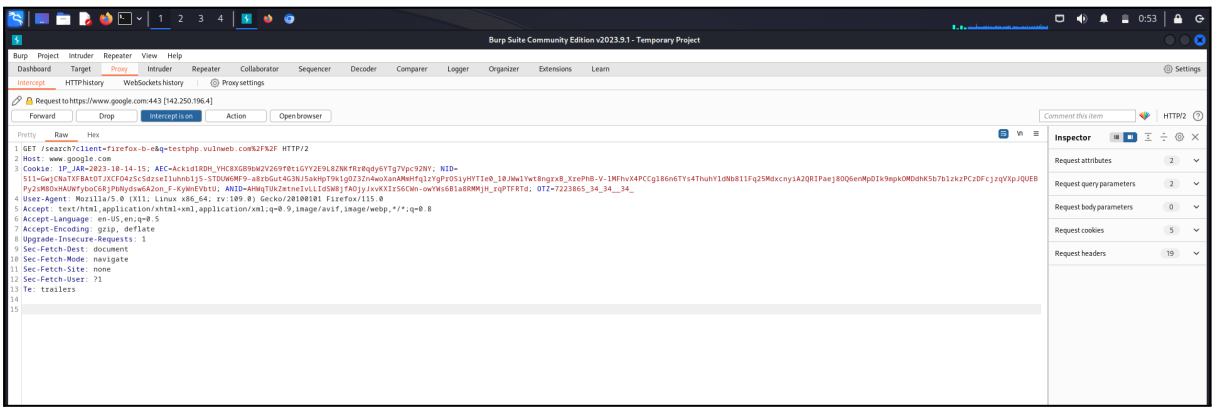
Vulnerability parameter: http://testphp.vulnweb.com//

Steps to reproduce:

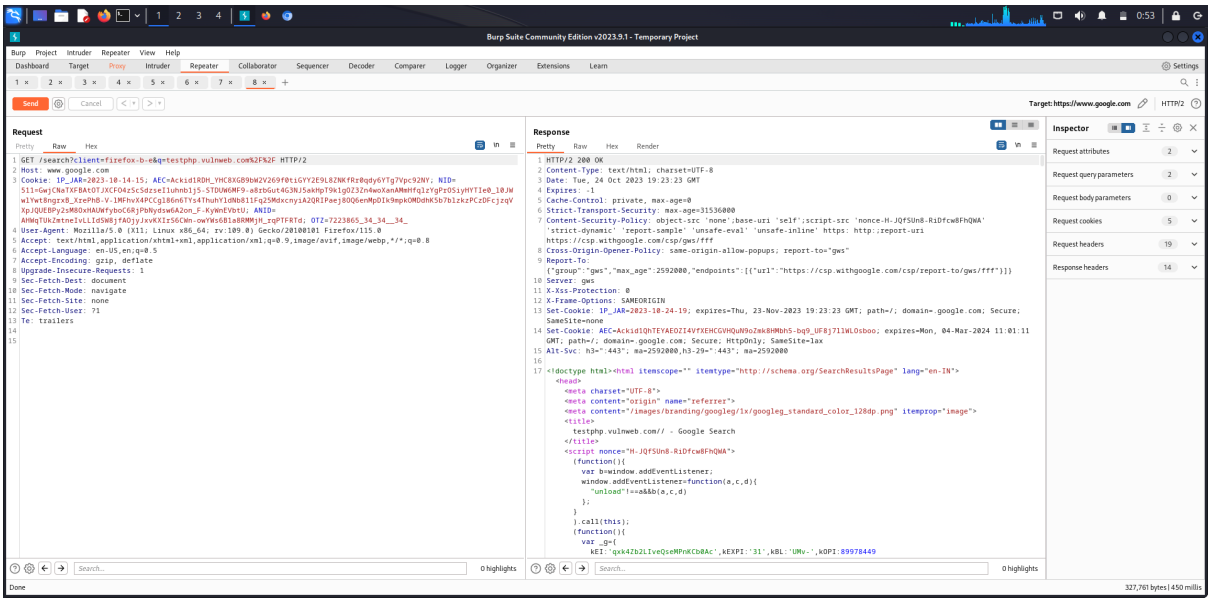
1. Access the website and use the parameter /



2. The information of the website entered will be shown in burpsuite.



3. Send the details under the proxy tab to the repeater tab.
4. Under the response tab, we can see the details regarding the vulnerability.



Mitigation:

Keep encryption algorithms and key lengths up to date, following industry best practices.

Regularly review and update cryptographic configurations to ensure they meet current security standards.

7. Vulnerability name: Failure to Sanitize Special Element

CWE: 159

OWASP Category: A03:2021- Injection

Description: This vulnerability arises when user input is not adequately sanitised, allowing malicious elements (e.g., scripts, tags) to be included in web content. This is a common vector for Cross-Site Scripting (XSS) attacks.

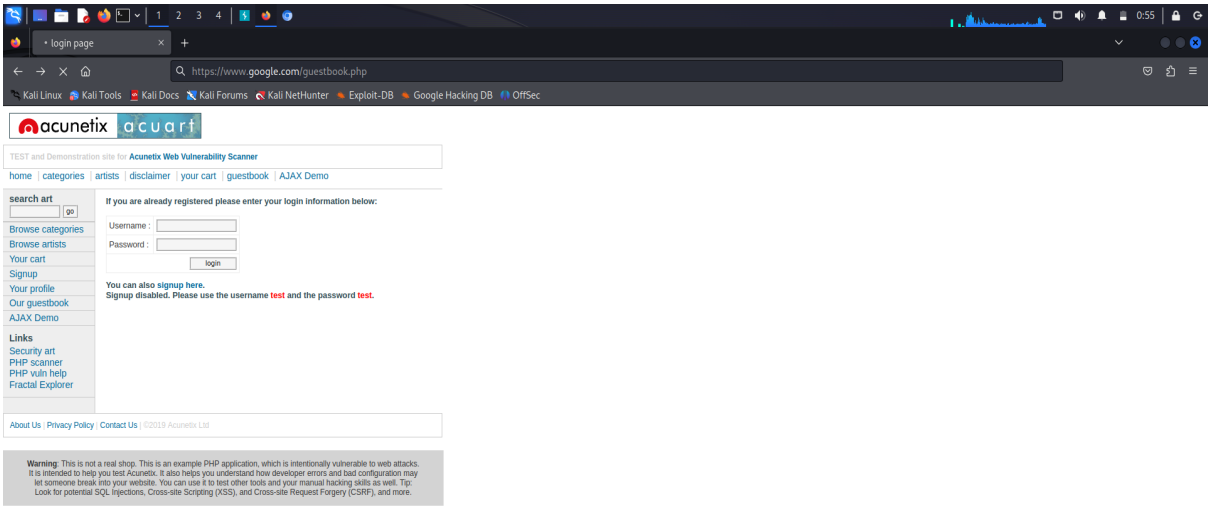
Business Impact: XSS attacks through this vulnerability can lead to compromised user accounts, data manipulation, and potential theft of sensitive information, causing reputational damage and regulatory penalties.

Vulnerability path: http://testphp.vulnweb.com/

Vulnerability parameter: http://testphp.vulnweb.com/guestbook.php

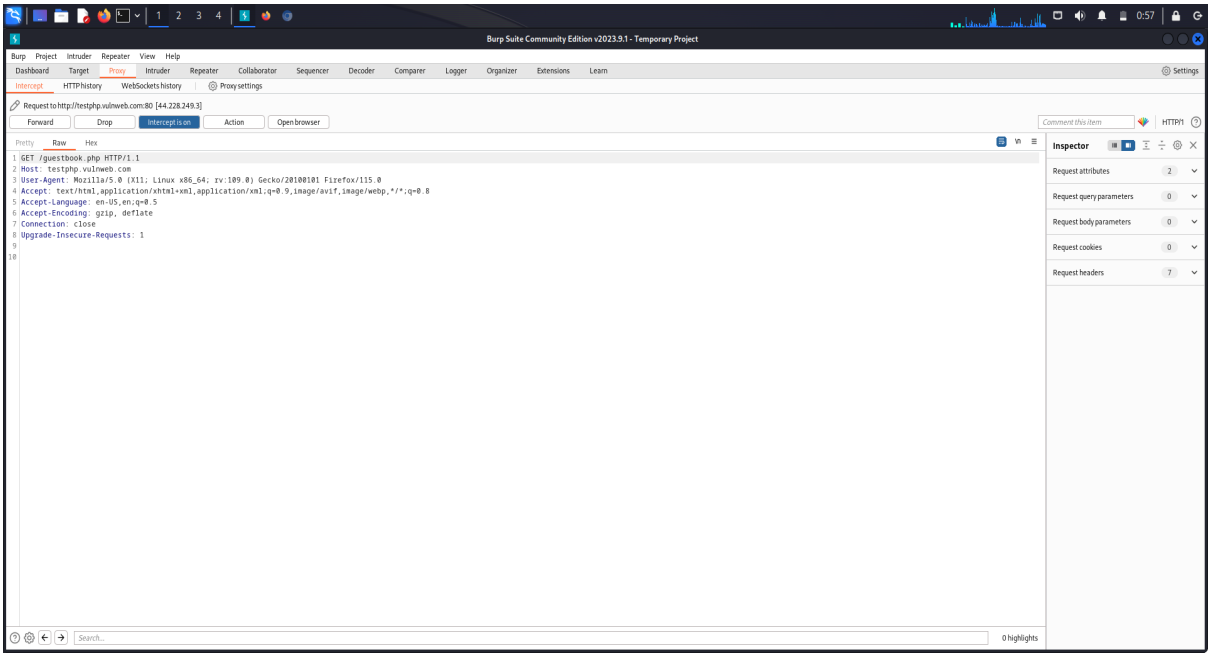
Steps to reproduce:

1. Access the website and use the parameter /guestbook.php

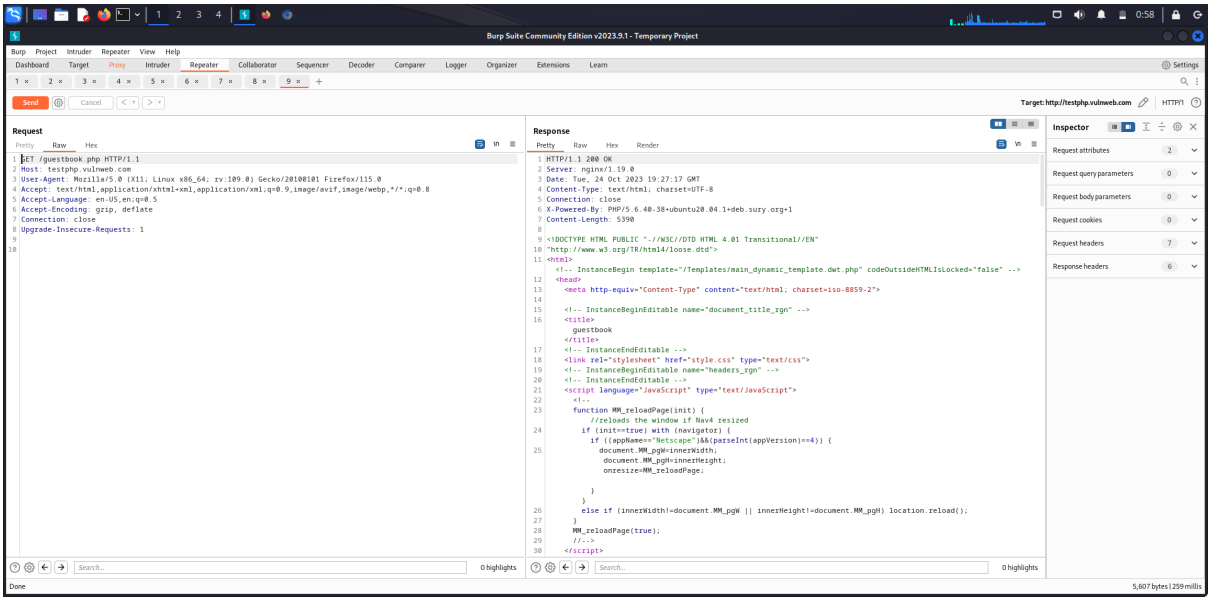


www.google.com

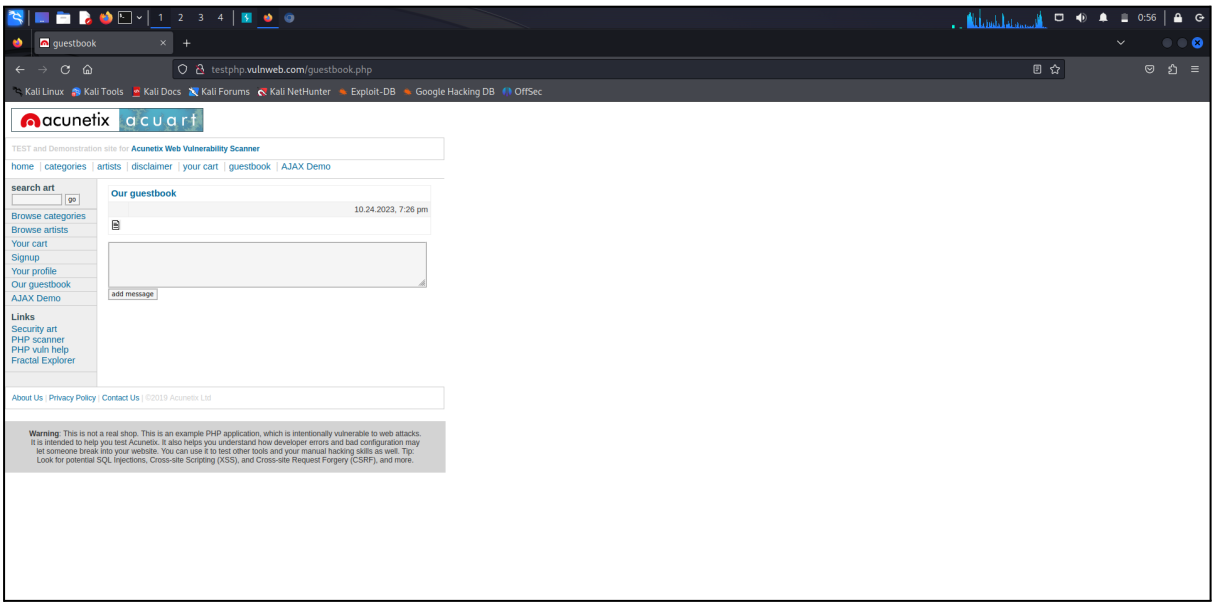
2. The information of the website entered will be shown in burpsuite.



3. Send the details under the proxy tab to the repeater tab.
4. Under the response tab, we can see the details regarding the vulnerability.



5. Website interface of the parameter.



Mitigation:

Implement input validation and output encoding to sanitise user input and prevent the inclusion of malicious elements.

Use security libraries and frameworks that offer built-in input sanitization functions.

8. Vulnerability name: Improper Encoding or Escaping of Output

CWE: 116

OWASP Category: A03:2021- Injection

Description: This vulnerability occurs when user-generated or untrusted data is not correctly encoded or escaped before being included in a web page or response, making it susceptible to Cross-Site Scripting (XSS) attacks.

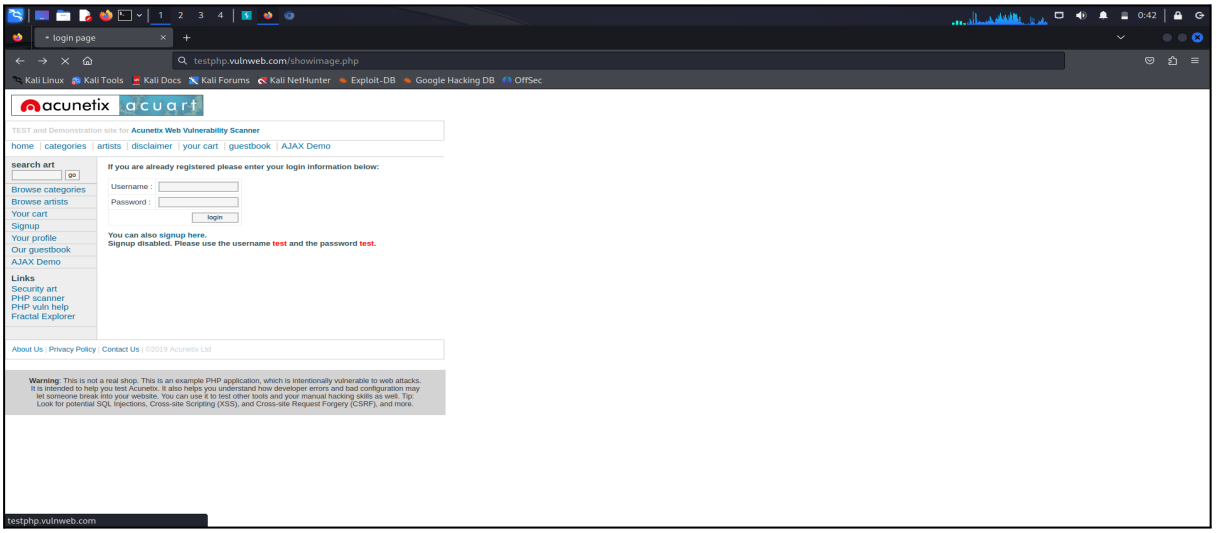
Business Impact: XSS attacks can lead to unauthorised access, data theft, and the injection of malicious scripts on a website. This can damage an organisation's reputation, result in data breaches, and compromise the security of its users.

Vulnerability path: http://testphp.vulnweb.com/

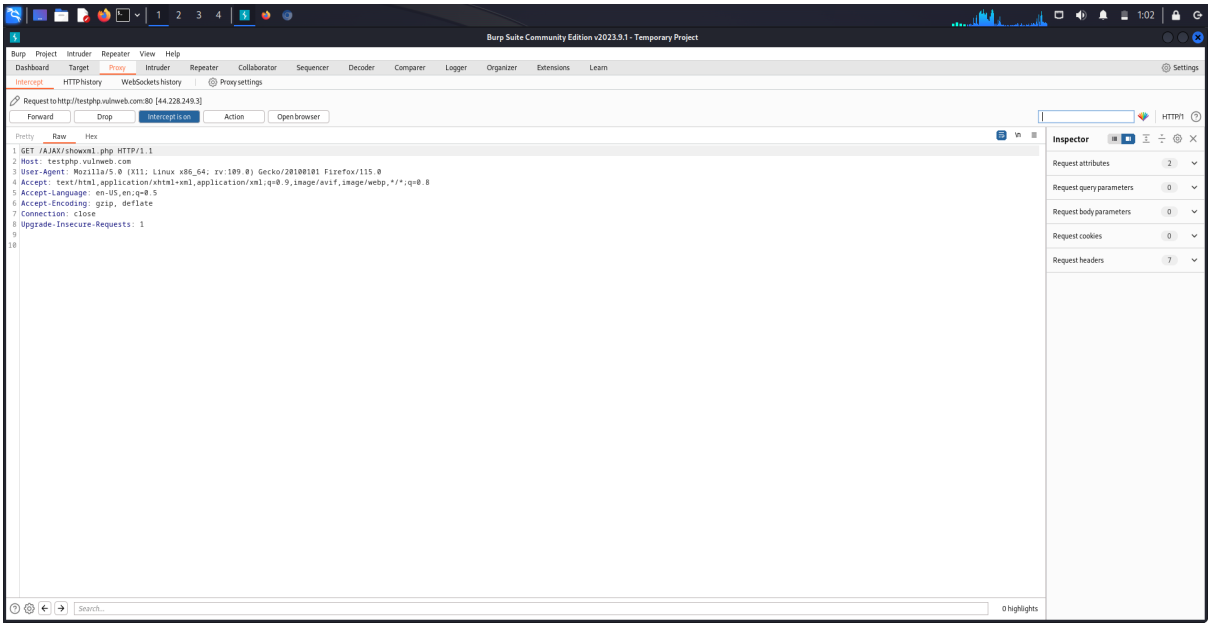
Vulnerability parameter: http://testphp.vulnweb.com/AJAX/showxml.php

Steps to reproduce:

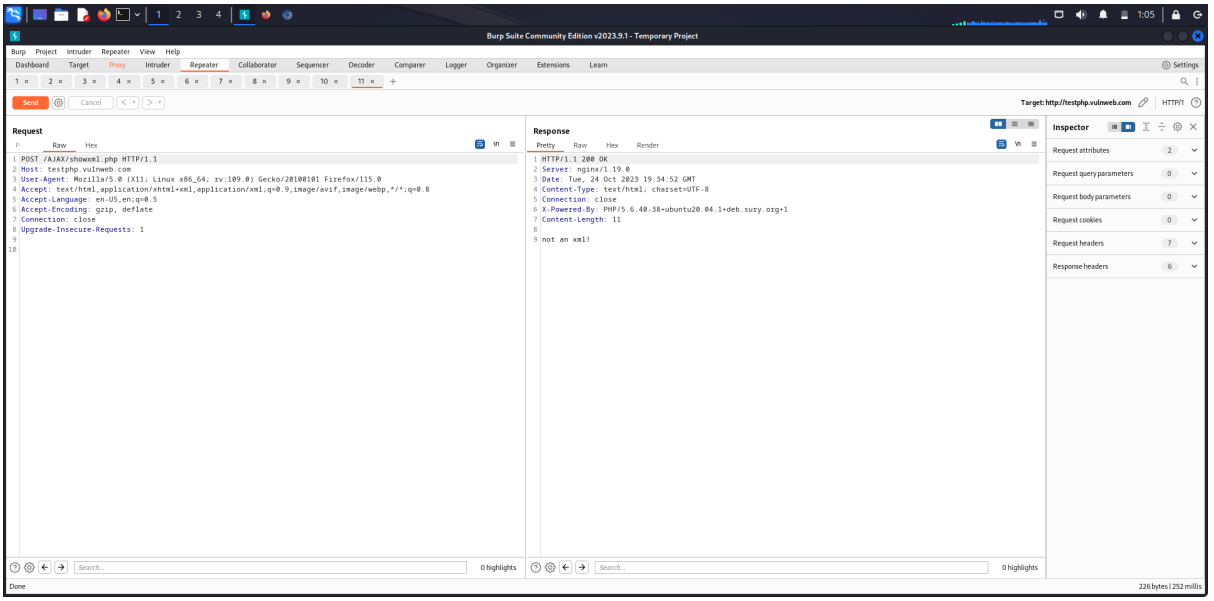
1. Access the website and use the parameter /showxml.php



2. The information of the website entered will be shown in burpsuite.

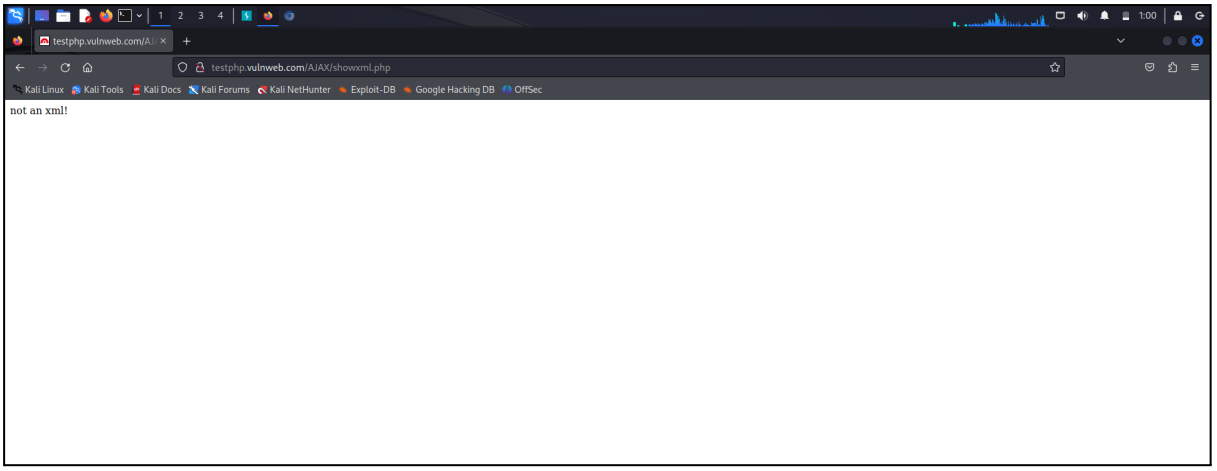


3. Send the details under the proxy tab to the repeater tab.
4. Under the response tab, we can see the details regarding the vulnerability



.

5. Website interface of the parameter.



Mitigation:

Use output encoding libraries or functions that automatically escape user-generated data when rendering it in web pages.

Implement proper context-based output encoding to ensure data is appropriately sanitised based on its usage (HTML, JavaScript, etc.).

9. Vulnerability Name: Relative Path Traversal

CWE : 23

OWASP Category: A01:2021 – Broken Access Control

Description : Relative Path Traversal is a vulnerability where an attacker manipulates file paths to gain unauthorised access to files or directories. By exploiting weaknesses in access controls, the attacker can traverse through the file system, potentially reaching files that should be restricted from their view. This can lead to unauthorised disclosure of sensitive information or even the execution of malicious code.

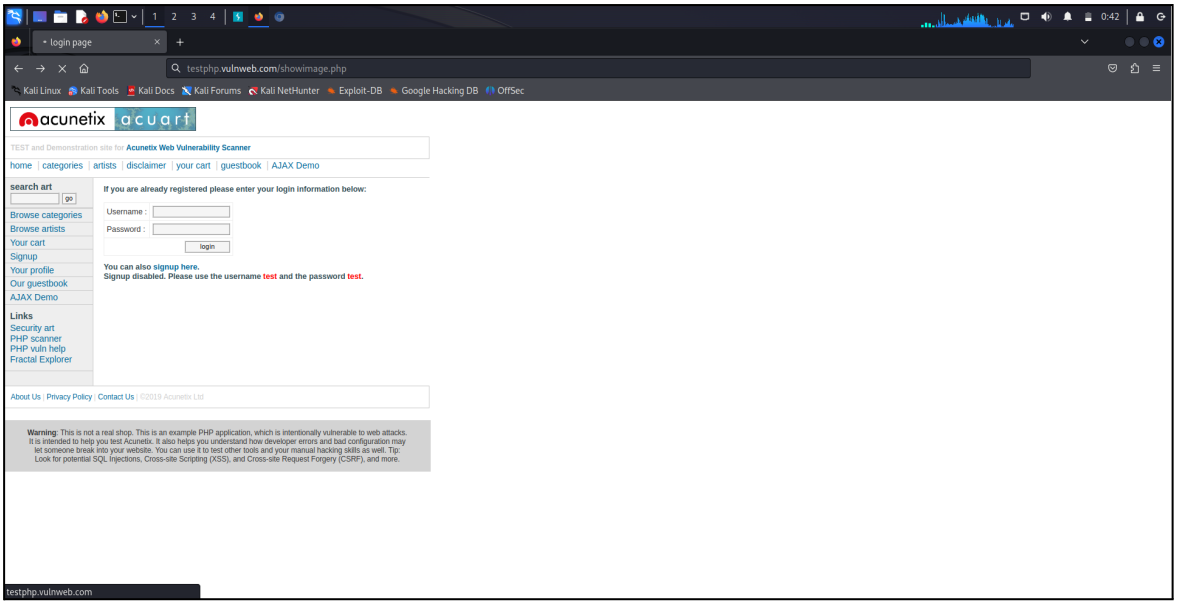
Business Impact : The impact of Relative Path Traversal can be severe. It can result in unauthorised access to critical data, potentially exposing sensitive information, proprietary code, or confidential documents. This could lead to reputational damage, legal consequences, and financial losses for affected businesses.

Vulnerability path: http://testphp.vulnweb.com/

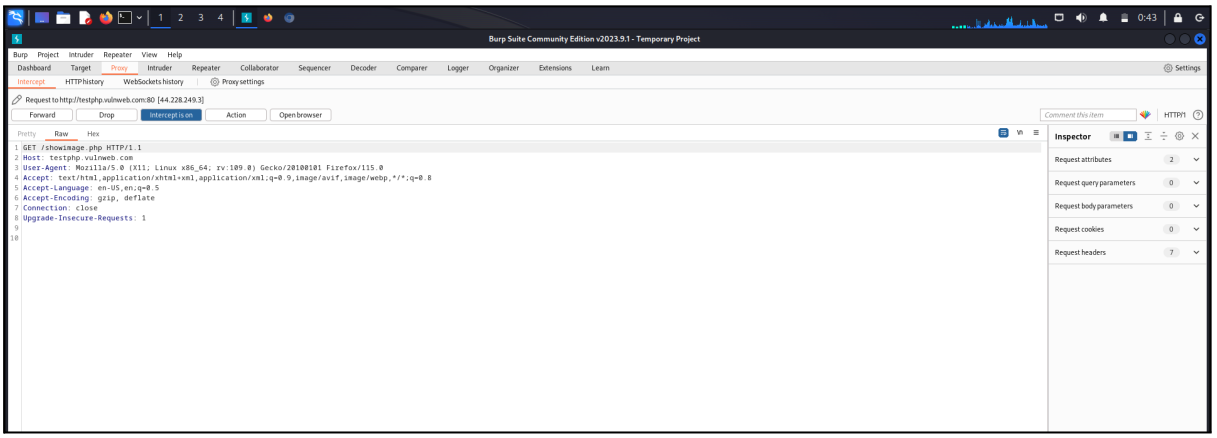
Vulnerability parameter: http://testphp.vulnweb.com/showimage.jpg

Steps to reproduce:

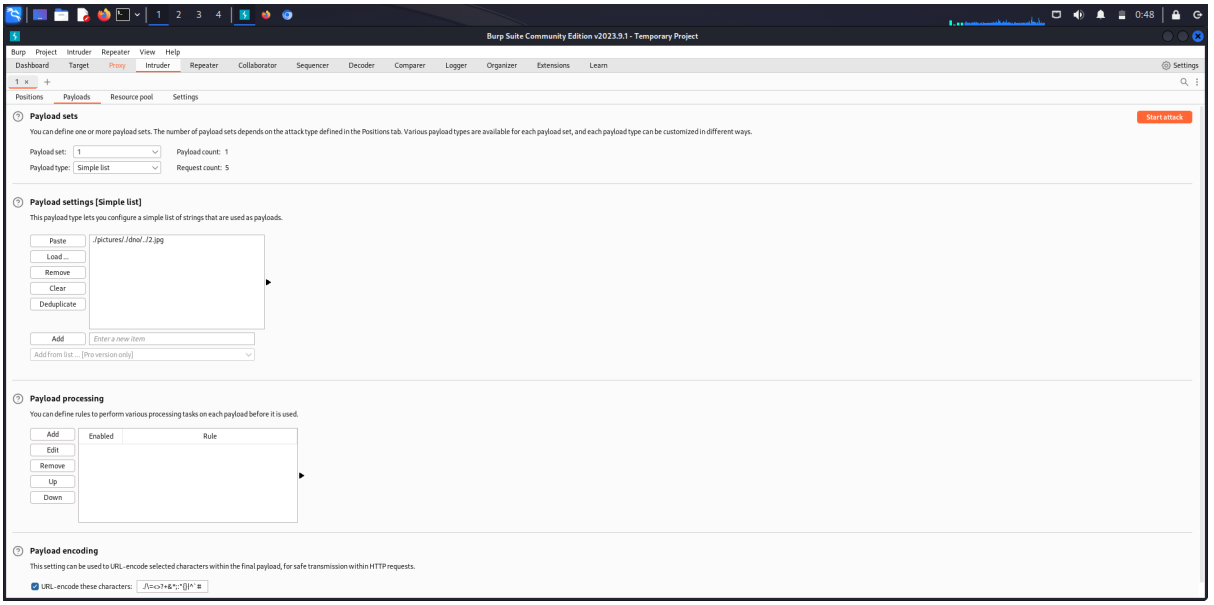
1. Access the website and use the parameter /showimage.jpg



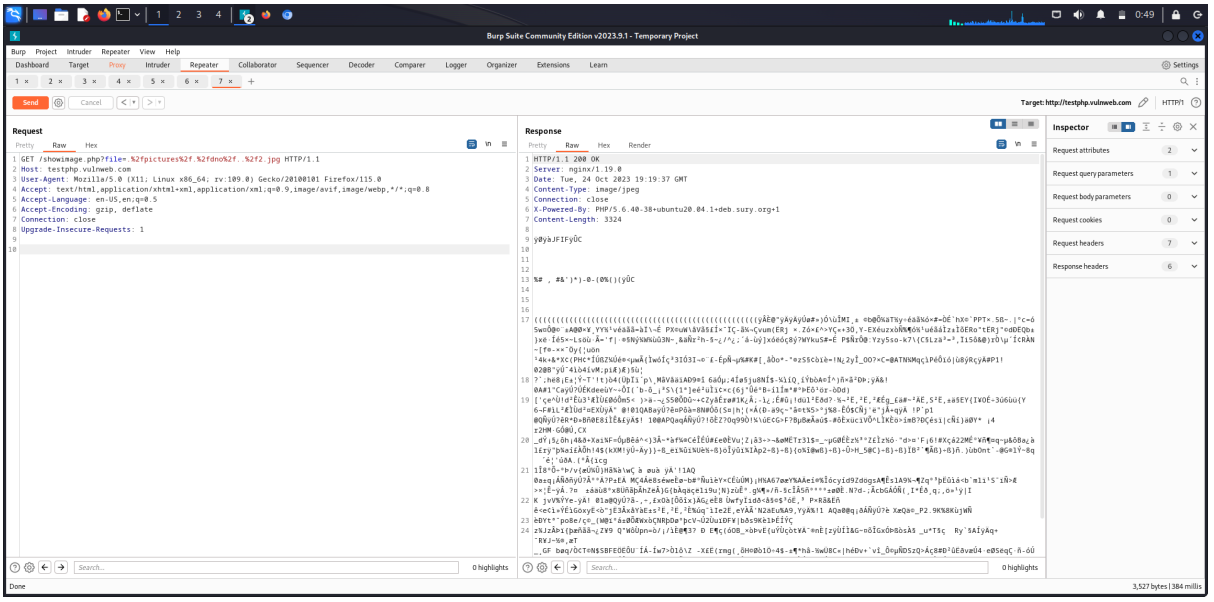
2. The information of the website entered will be shown in burpsuite.



3. Insert the payloads.



4. Send the details under the proxy tab to the repeater tab.
5. Under the response tab, we can see the details regarding the vulnerability.



Mitigation : To mitigate Relative Path Traversal, it is crucial to implement robust access controls and validate user input thoroughly. Use absolute paths or employ a secure method for resolving file paths. Additionally, consider implementing least privilege principles, ensuring that users and processes only have the minimum level of access necessary to perform their tasks. Regular security assessments, code reviews, and penetration testing can also help identify and address potential vulnerabilities in the system.

10. Vulnerability Name: Configuration

CWE : 16

OWASP Category: A06:2021 - Security Misconfiguration

Description : This vulnerability arises from inadequately managed configurations, which may include default settings, access controls, or sensitive information inadvertently exposed. Attackers exploit these misconfigurations to gain unauthorised access, escalate privileges, or retrieve sensitive data.

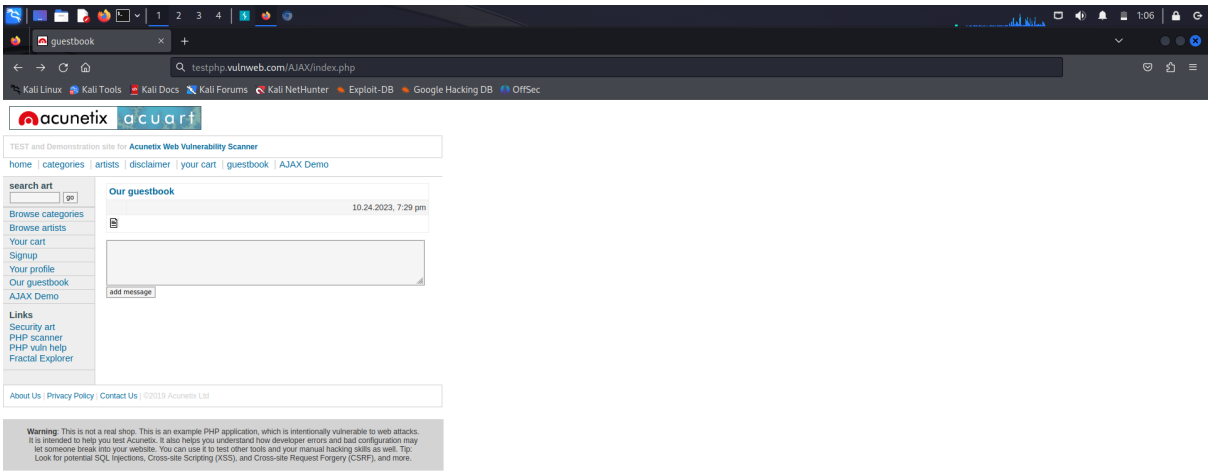
Business Impact : The business impact of a configuration vulnerability can be severe. It may lead to data breaches, financial losses, reputational damage, and legal consequences. Additionally, downtime and disruption of services can occur as a result of exploiting misconfigurations, potentially causing significant harm to an organisation's operations and reputation.

Vulnerability path: http://testphp.vulnweb.com/

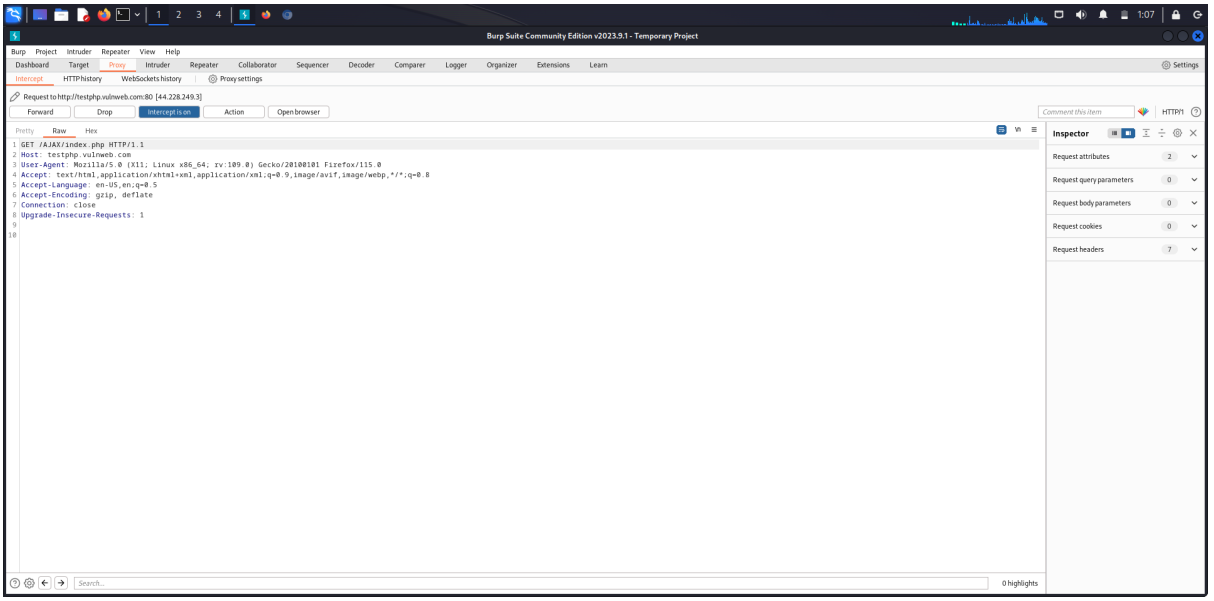
Vulnerability parameter: http://testphp.vulnweb.com/AJAX/index.php

Steps to reproduce:

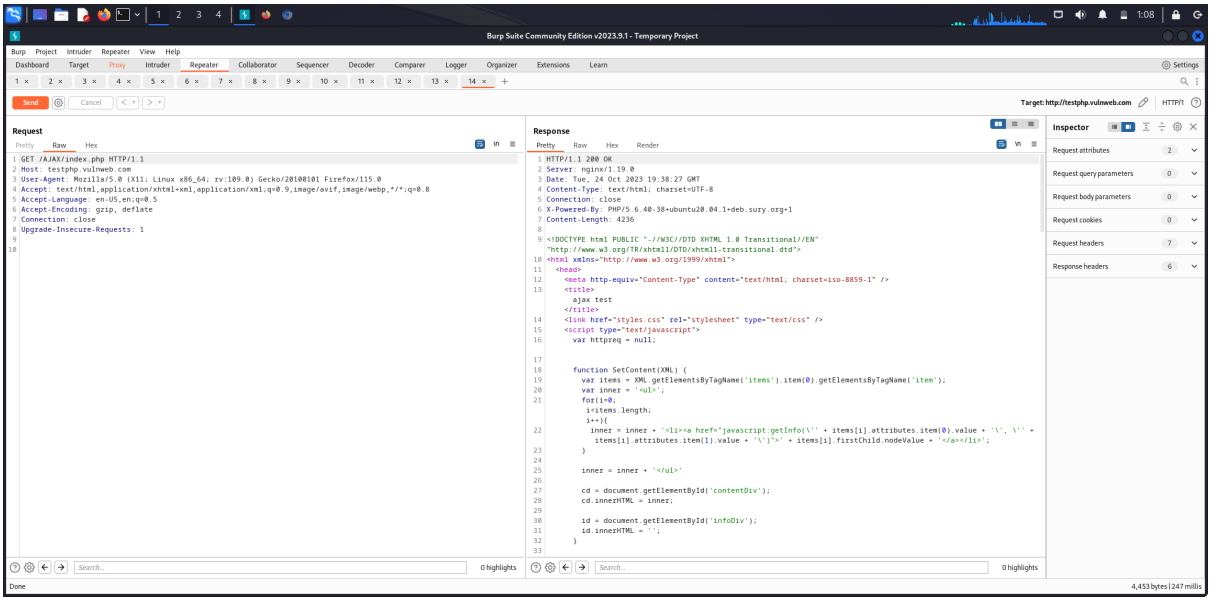
1. Access the website and use the parameter /AJAX/index.php



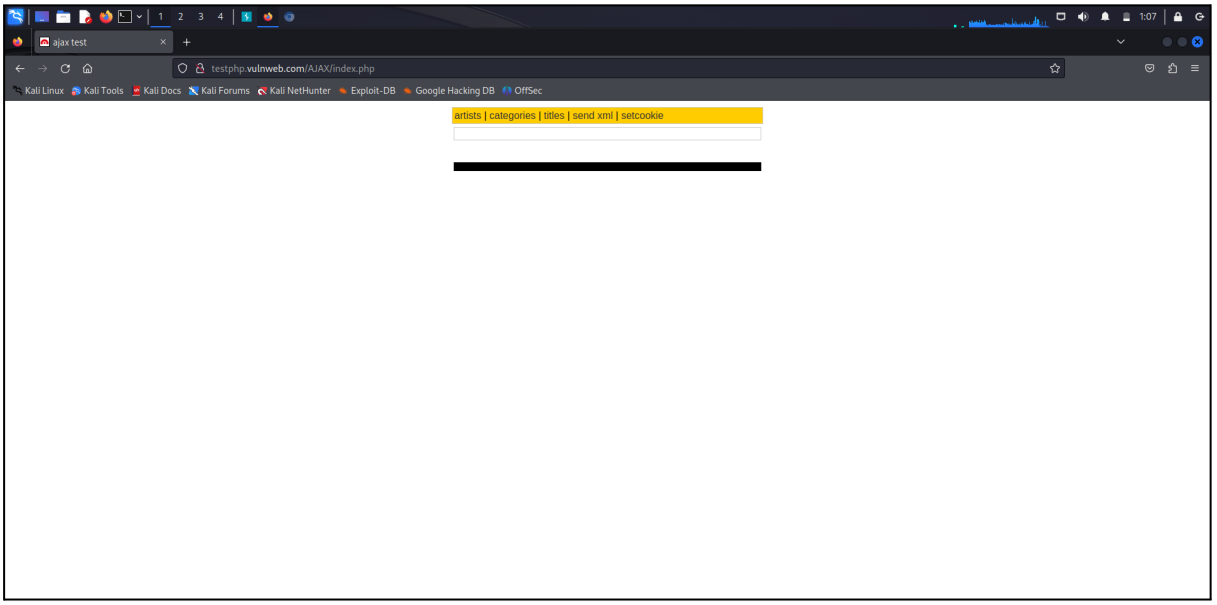
2. The information of the website entered will be shown in burpsuite.



3. Send the details under the proxy tab to the repeater tab.
4. Under the response tab, we can see the details regarding the vulnerability.



5. Website interface of the parameter.



Mitigation : To address this vulnerability, organisations should implement robust configuration management practices. This involves regular audits of system settings, strict access controls, and the removal of unnecessary services or features. Employing security best practices, such as utilising secure defaults and regularly updating configurations to align with evolving threat landscapes, is crucial in mitigating the risks associated with misconfigurations.Additionally, automated tools for configuration management and vulnerability scanning can help identify and rectify potential issues proactively. Regular training and awareness programs for staff can also contribute to a more secure configuration posture.