INSTITUTE FOR ADVANCED COMPUTING
AND
SOFTWARE DEVELOPMENT
AKURDI, PUNE

DOCUMENTATION ON

"Websecbalance using DVWA"

PG-DITISS Mar-2023

SUBMITTED BY:

GROUP NO: 18

PRATIKSHA SONWANE (233435)

SHOBHANA UNDRE (233445)

MR. KARTIK AWARI
PROJECT GUIDE

MR. ROHIT PURANIK
CENTRE COORDINATOR

TABLE OF CONTENTS

1. Introduction 1
   1.1 What are Vulnerabilities?
   1.2 Apache Web Server
   1.3 DVWA
   1.4 Burp suite
2. System Requirements 6
   2.1 Infrastructure Diagram
3. Installation 8
   3.1 Installing DVWA
   3.2 Configure DVWA
   3.3 DVWA
   3.4 DVWA Home Page
4. Installing HAProxy 11
   4.1 Installing HAProxy
   4.2 Status
5. Working 12
   5.1 DVWA Attacks
6. Logs 13
7. Report on vulnerabilities 17
   7.1 Contents for reports
   7.2 Reference Report
8. References 19

1. INTRODUCTION

In today's digital landscape, the security of web applications has become a paramount concern. With cyberattacks targeting web applications on the rise, there is an increasing demand for individuals to acquire practical skills in identifying and mitigating vulnerabilities. A load balancing mechanism is implemented to distribute security testing traffic evenly across multiple instances of DVWA. This simulates the challenges of load balancing encountered in actual scenarios. The project encourages ethical hacking by providing a safe environment to practice security testing skills.

Penetration testing involves simulating cyberattacks on a target system to identify potential security vulnerabilities and weaknesses. This proactive approach helps organizations assess the effectiveness of their security measures and discover any potential entry points for attackers. Pen testers attempt to exploit vulnerabilities in a controlled manner, mimicking real-world attacks without causing actual harm.

DVWA is a web application purposely designed to be vulnerable to various security vulnerabilities. It's often used as a training tool for learning and practicing web application security testing. DVWA provides a safe and controlled environment where individuals can explore and understand common web vulnerabilities by exploiting them in a legal and ethical manner.

1.1What are Vulnerabilities?

A vulnerability is a weakness that can be exploited by cybercriminals to g

in unauthorized access to a computer system. After exploiting a vulnerability, a cyberattack can run malicious code, install malware, and even steal sensitive data.

Vulnerabilities can be exploited by a variety of methods, including SQL injection, buffer overflows, cross-site scripting (XSS), and open-source exploit kits that look for known vulnerabilities and security weaknesses in web applications.

Many vulnerabilities impact popular software, placing the many customersusing the software at a heightened risk of a data breach, or supply chain attack. Such zero-day

exploits are registered by MITRE as a Common Vulnerability Exposure (CVE).

Types of web vulnerabilities:-(OWASP Top 10)

A01:2021-Broken Access Control moves up from the fifth position; 94%of applications were tested for some form of broken access control. The 34 Common Weakness Enumerations (CWEs) mapped to Broken Access Control had more occurrences in applications than any other category.

A02:2021-Cryptographic Failures shifts up one position to #2, previously knownas Sensitive Data Exposure, which was broad symptom rather than aroot cause.The renewed focus here is on failures related to cryptography which often leads to sensitive data exposure or system compromise.

A03:2021-Injection slides down to the third position. 94% of the applications were tested for some form of injection, and the 33 CWEs mapped into this category have the second most occurrences inapplications. Cross-site Scripting is now part of this category in this edition.

A04:2021-Insecure Design is a new category for 2021, with a focus on risks related to design flaws. If we genuinely want to “move left” as an industry, it calls for more use of threat modeling, secure design patterns and principles and reference architectures.

A05:2021-Security Misconfiguration. 90% of applications were tested for some form of misconfiguration. With more shifts into highly configurable software, it’s not surprising to see this category move up. The former category for XML External Entities (XXE) is now part of this category.

A06:2021-Vulnerable and Outdated Components was previously titled Using Components with Known Vulnerabilities and is #2 in the Top 10 community survey, but also had enough data to make the Top 10 via data analysis. Is a knownissue that we struggle to test and assess risk. It is the only category not to have any Common Vulnerability and Exposures(CVEs) mapped to the included CWEs, so a default exploit and impact weights of 5.0 are factored into their scores.

A07:2021-Identification and Authentication Failures was previously BrokenAuthentication and is sliding down from the second position, and now includes CWEs that are more related to identification failures. This category is still an integral part of the Top 10, but the increased availabilityof standardized frameworks seems to be helping.

A08:2021-Software and Data Integrity Failures is a new category for 2021,focusing on making assumptions related to software updates, critical data, and CI/CD pipelines

without verifying integrity. One of the highest weighted impacts from Common Vulnerability and Exposures/Common Vulnerability ScoringSystem (CVE/CVSS) data mapped to the 10 CWEs in this category. Insecure Deserialization from 2017 is now a part of this larger category.

A09:2021-Security Logging and Monitoring Failures was previously InsufficientLogging & Monitoring and is added from the industry survey. This category is expanded to include more types of failures, is challengingto test for, and isn't well represented in the CVE/CVSS data. However, failures in this category can directly impact visibility, incident alerting, and forensics.

A10:2021-Server-Side Request. The data shows a relatively low incidencerate with above average testing coverage, along with above-average ratingsfor Exploit and Impact potential. This category represents the scenario where the security community members are telling us this is important, even though it's notillustrated in the data at this time.

1.2Apache Web Server

Apache is considered open-source software, which means the original source code is freely available for viewing and collaboration. Being open source has made Apache very popular with developers who have built andconfigured their own modules to apply specific functionality and improve on its core features. Apache has been around since 1995 and is responsibleas a core technology that helped spur the initial growth of the internet in itsinfancy.

One of the pros of Apache is its ability to handle large amounts of traffic with minimal configuration. It scales with ease and with its modularfunctionality at its core, you can configure Apache to do what you want, how you want it. You can also remove unwanted modules to make Apachemore lightweight and efficient.

Some of the most popular modules that can be added are SSL, Server-SideProgramming Support (PHP), and Load Balancing configs to handle large amounts of traffic. Apache can also be deployed on Linux, MacOS, and Windows. If you learn how to configure Apache on Linux, you can administer Apache on Windows and Mac. The only difference would be directory paths andinstallation processes.

1.3DVWA

Damn Vulnerable Web Application (DVWA) is a PHP/MySQL web application thatis damn vulnerable. Its main goal is to be an aid for security professionals to test their skills and tools in

a legal environment, help web developers better understand the processes of securing web applications and to aid both students & teachers to learn about web applicationsecurity in a controlled class room environment.

To test WAF, we need a vulnerable web-application on which we can perform and block the attacks.

So, we have used DVWA as a web-application.

A vulnerable web application is a web application that has security weaknesses thatcan be exploited by attackers to gain unauthorized access to sensitive information, inject malicious code, or disrupt the normal operation of the application.

There are several types of vulnerabilities that can be present in a web application, including

Injection flaws: These vulnerabilities occur when untrusted user input is not properly validated or sanitized before being used by the application. This can lead to SQL injection, code injection, and other types of injection attacks.

Cross-site scripting (XSS): This occurs when an attacker is able to inject maliciousscripts into a web page viewed by other users.

Cross-site request forgery (CSRF): This occurs when an attacker is able to trick a user into performing an action on a web application without their knowledge or consent.

1.4 Burp suite

Burp Suite: is a comprehensive set of web application security testing tools developed by PortSwigger, a UK-based cybersecurity company. It's widely used by security professionals, penetration testers, and ethical hackers to identify and mitigate vulnerabilities in web applications. Burp Suite offers a range of features that assist in various stages of security testing, from identifying vulnerabilities to providing detailed reports for remediation.

Key Features of Burp Suite:

1. **Proxy :** Burp Suite's proxy allows you to intercept and modify web traffic between your browser and the target application. This is useful for manual testing and analyzing requests and responses.
2. **Scanner:** The automated scanner in Burp Suite can identify a variety of vulnerabilities,

such as SQL injection, cross-site scripting (XSS), and more. It scans the target application for potential security issues.

3. **Intruder:** Intruder enables you to automate and customize various types of attacks, such as fuzzing, to identify vulnerabilities that could be exploited.
4. **Repeater:** This tool lets you modify and replay individual HTTP requests to observe how the application responds. It's useful for manual testing and observing the impact of different inputs.
5. **Extender :** Burp Suite's extender allows you to enhance the tool's functionality by adding custom extensions. This could include additional scanning capabilities or integrations with other tools.

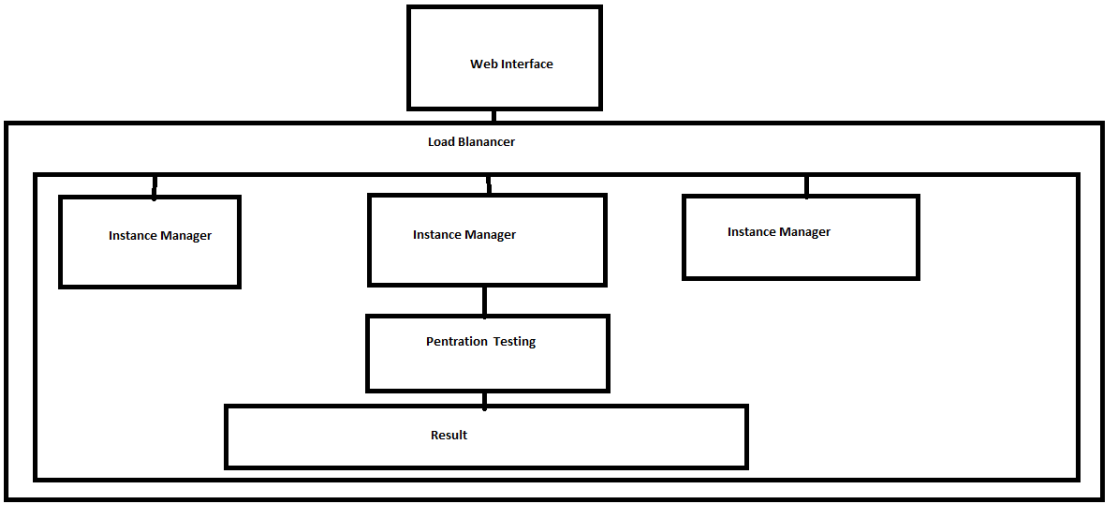
2.SYSTEM REQUIREMENTS

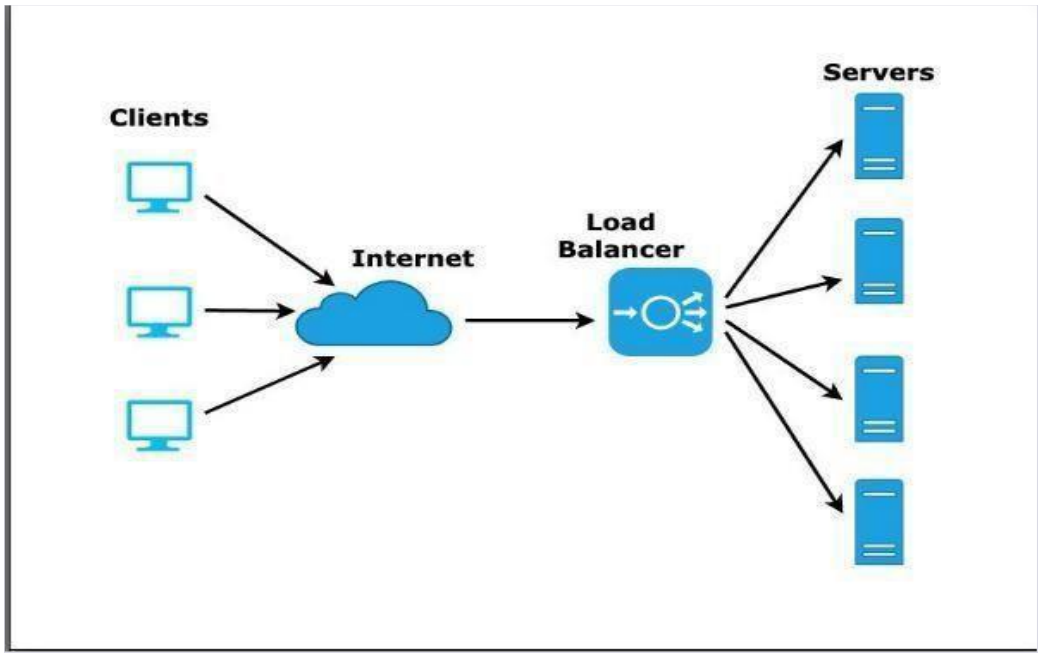
2.1 System Requirements

1. Hardware Consideration:
    a. Kali Linux
    b. RAM 16GB
    c. Hard Disk – 500GB
2. Software Architecture
    a. Apache Web Server
    b. HAProxy
    c. DVWA
3. Burpsuite

2.2Infrastructure Diagram

BASIC ARCHITECTURE:




Clients
Internet
Load
Balancer
Servers

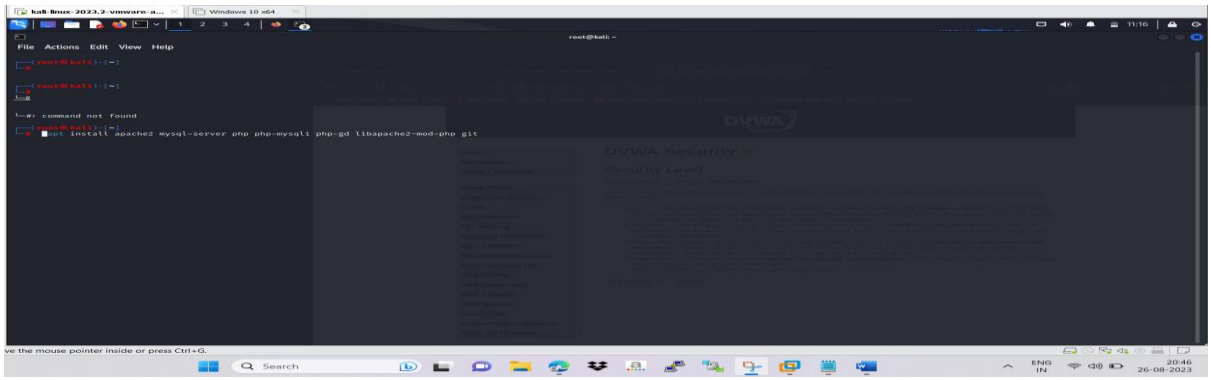
3.INSTALLATION

3.1 Installing Apache Web Server:-

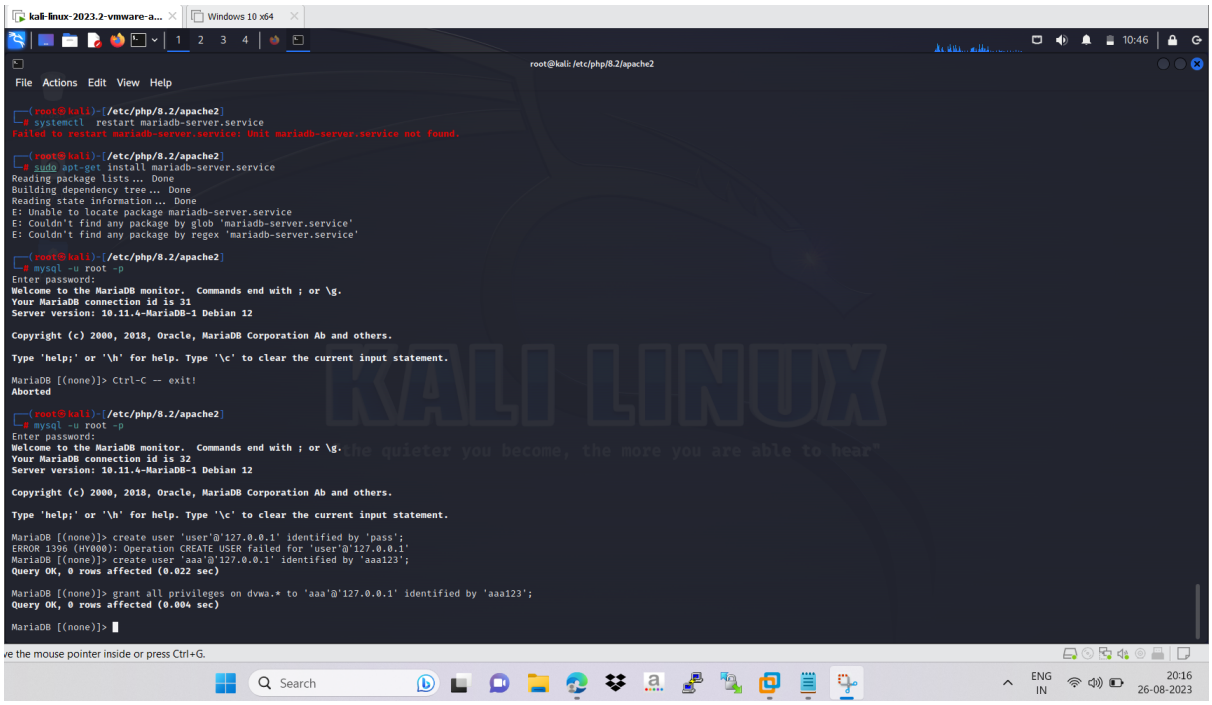
```
┌──(root㉿kali)-[~]
└─# apt-get install apache2
Reading package lists ... Done
Building dependency tree ... Done
Reading state information ... Done
apache2 is already the newest version (2.4.55-1).
0 upgraded, 0 newly installed, 0 to remove and 1789 not upgraded.

┌──(root㉿kali)-[~]
└─#
```

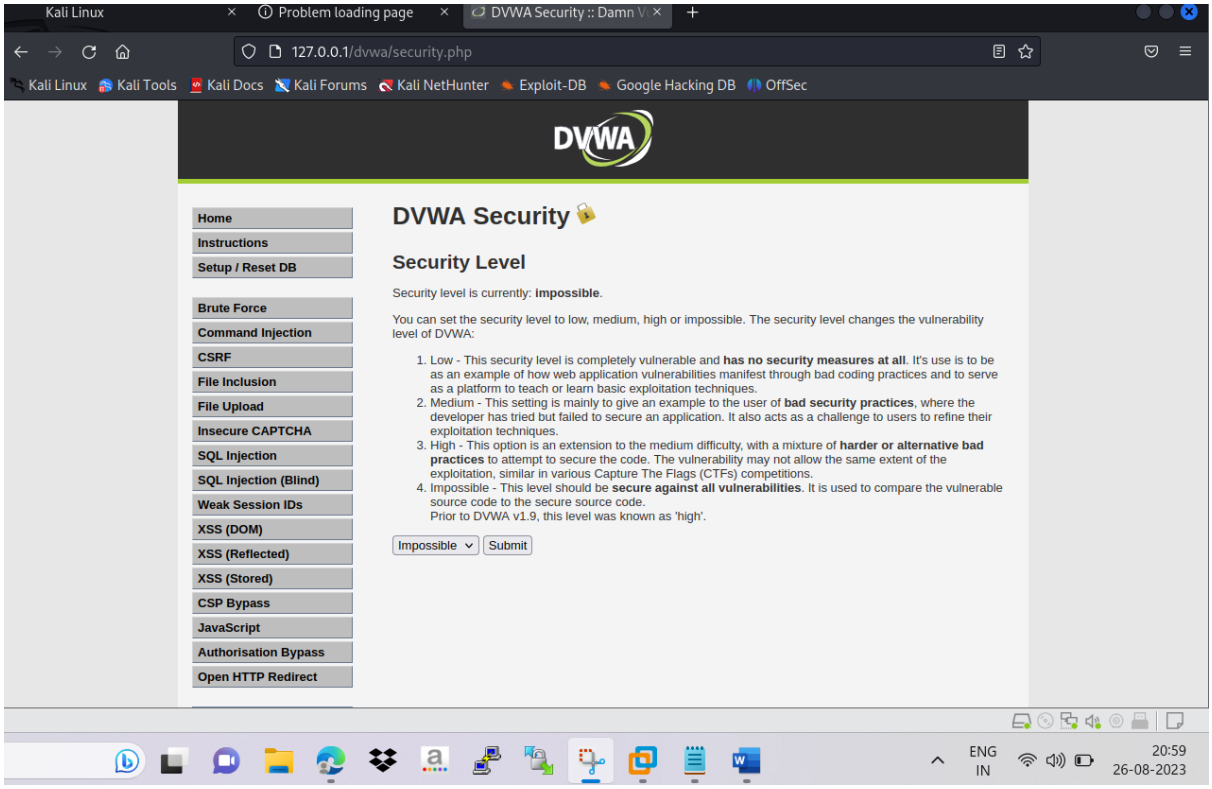
3.2 Installing DVWA :-

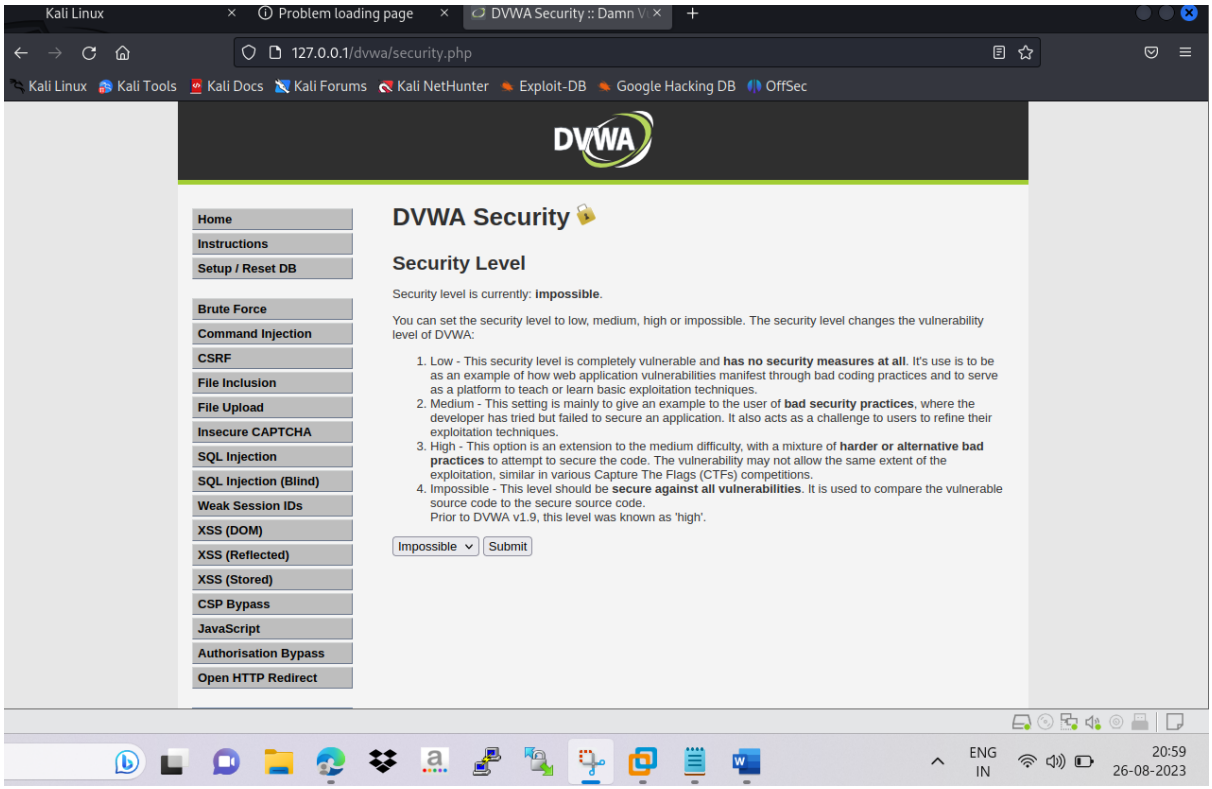


3.3 Configure DVWA



3.4 DVWA Home page

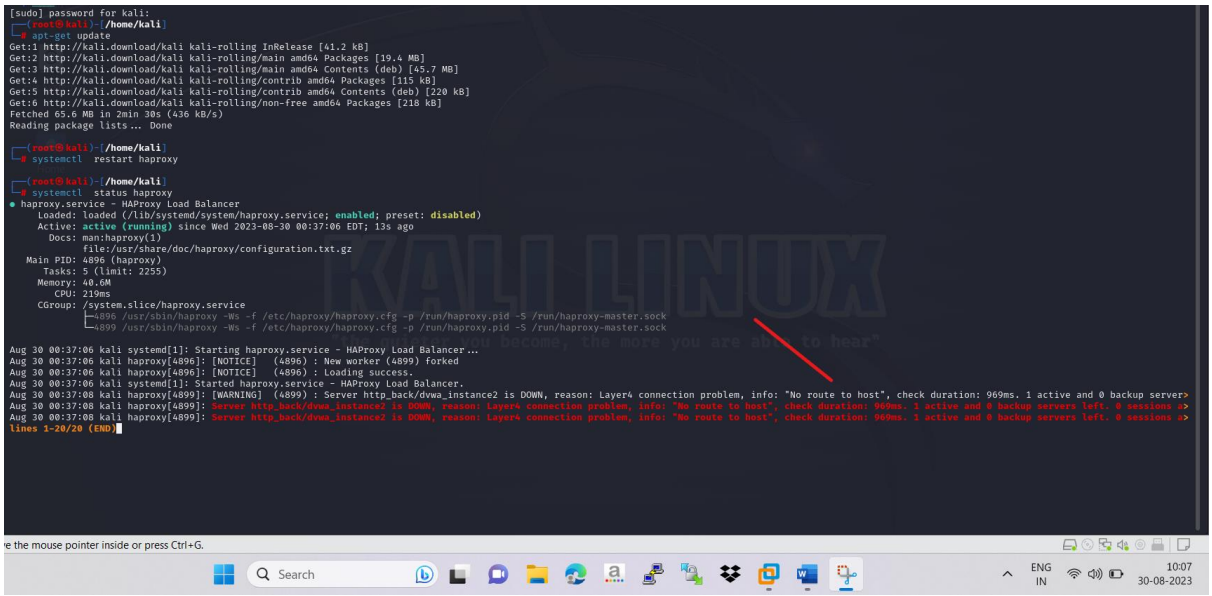



Kali Linux
Problem loading page
DVWA Security :: Damn V
127.0.0.1/dvwa/security.php
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec
DVWA
Home
Instructions
Setup / Reset DB
Brute Force
Command Injection
CSRF
File Inclusion
File Upload
Insecure CAPTCHA
SQL Injection
SQL Injection (Blind)
Weak Session IDs
XSS (DOM)
XSS (Reflected)
XSS (Stored)
CSP Bypass
JavaScript
Authorisation Bypass
Open HTTP Redirect
DVWA Security
Security Level
Security level is currently: impossible.
You can set the security level to low, medium, high or impossible. The security level changes the vulnerability level of DVWA:
1. Low - This security level is completely vulnerable and has no security measures at all. It's use is to be as an example of how web application vulnerabilities manifest through bad coding practices and to serve as a platform to teach or learn basic exploitation techniques.
2. Medium - This setting is mainly to give an example to the user of bad security practices, where the developer has tried but failed to secure an application. It also acts as a challenge to users to refine their exploitation techniques.
3. High - This option is an extension to the medium difficulty, with a mixture of harder or alternative bad practices to attempt to secure the code. The vulnerability may not allow the same extent of the exploitation, similar in various Capture The Flags (CTFs) competitions.
4. Impossible - This level should be secure against all vulnerabilities. It is used to compare the vulnerable source code to the secure source code.
Prior to DVWA v1.9, this level was known as 'high'.
Impossible
Submit
ENG
IN
20:59
26-08-2023

4 .INSTALLTION OF HAPROXY

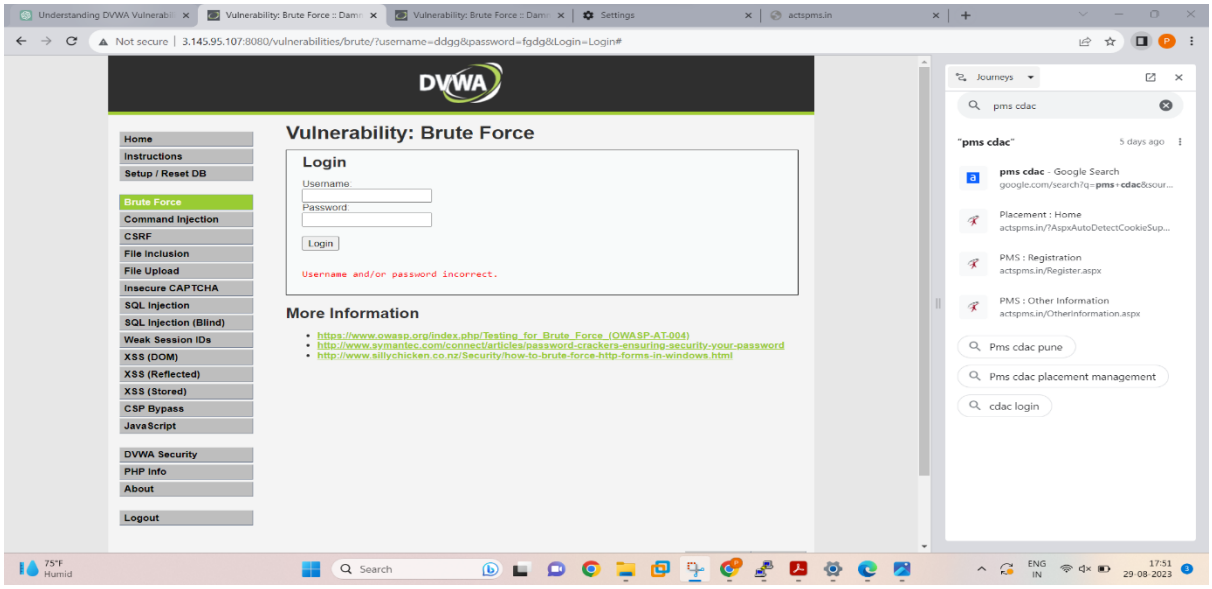
4.1 sudo apt-get update
sudo apt-get install haproxy -y

4.2 Status of HAProxy



5.WORKING

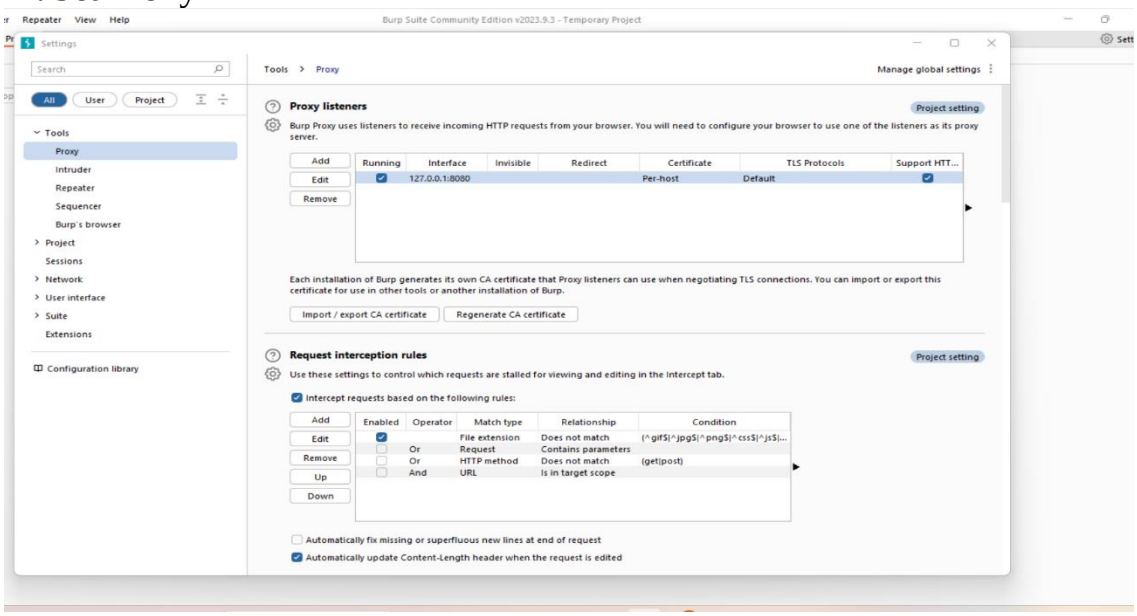
1.Bruteforce attack



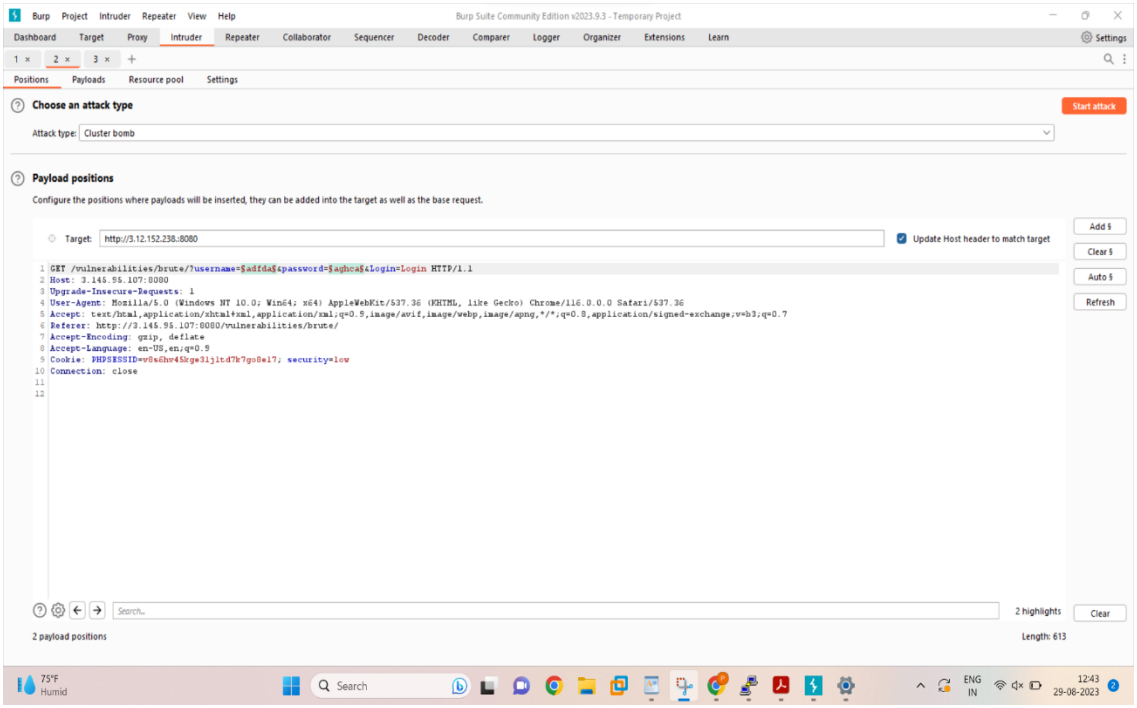
6.LOGS

Use of Burpsuit

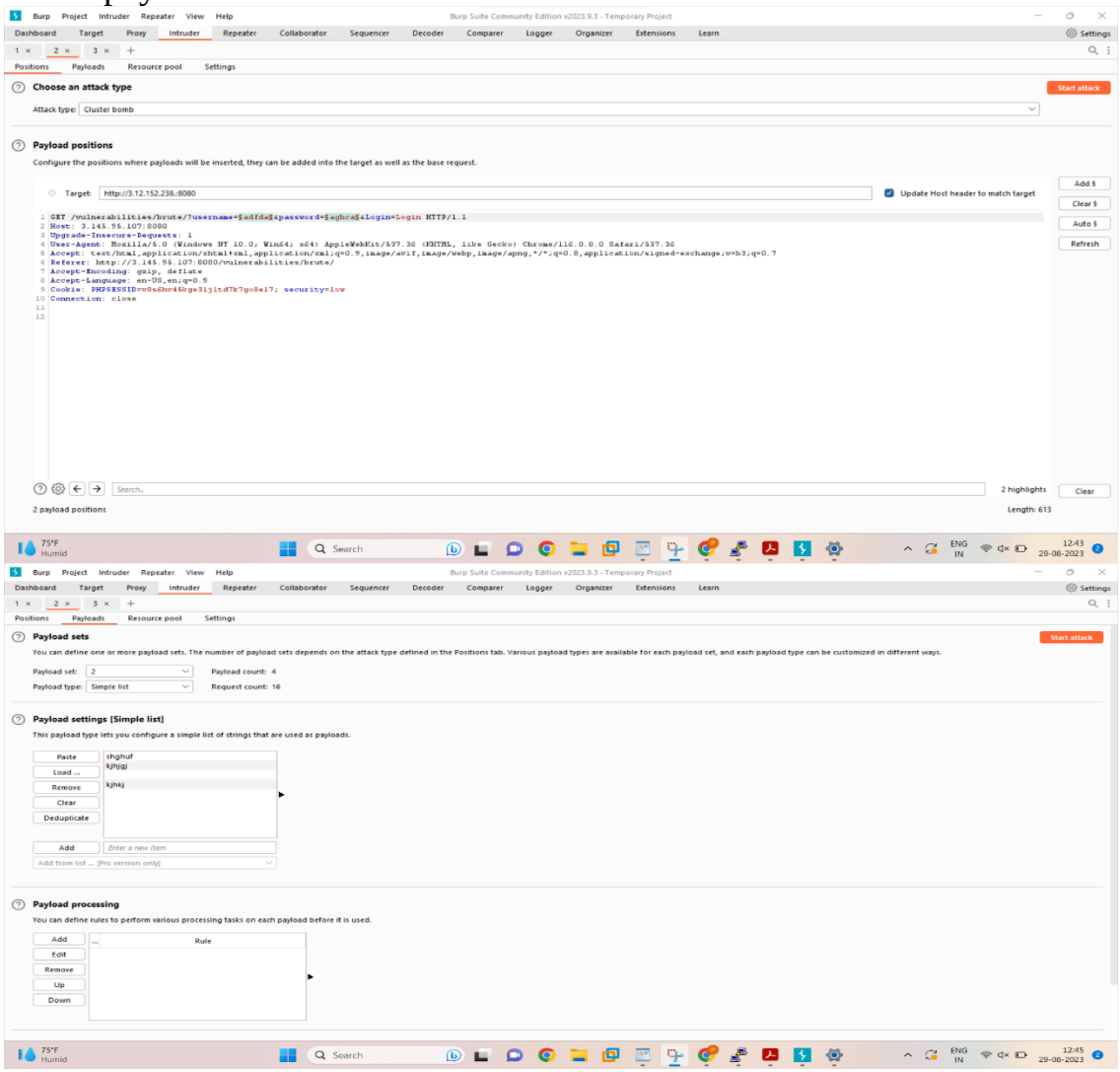
1. Set Proxy



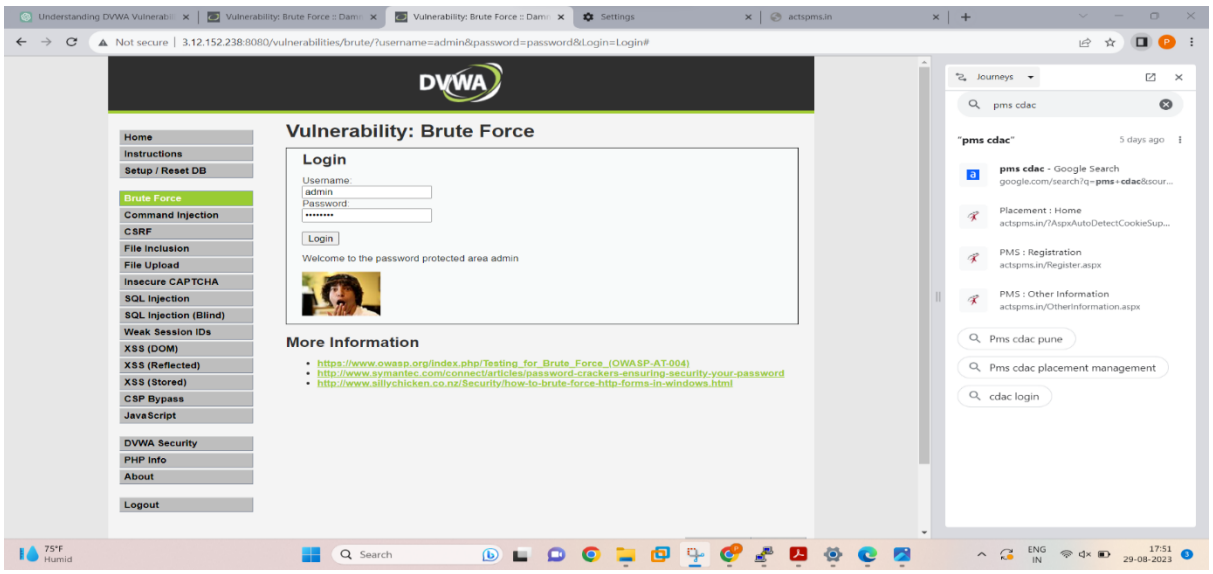
2.Set positions



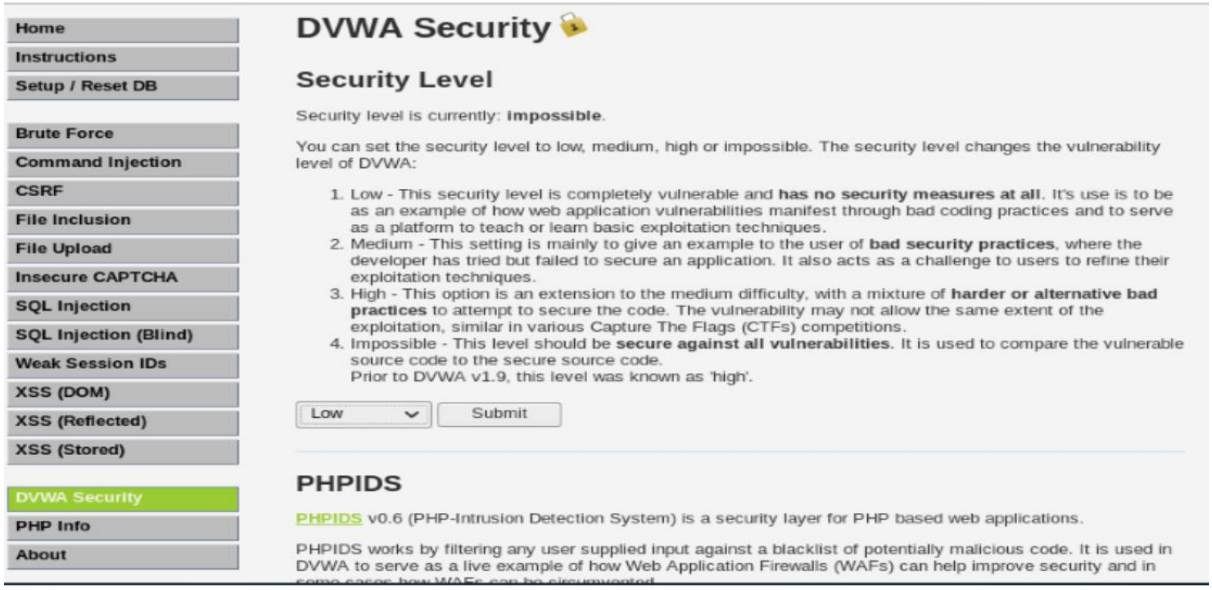
3.Add payloads



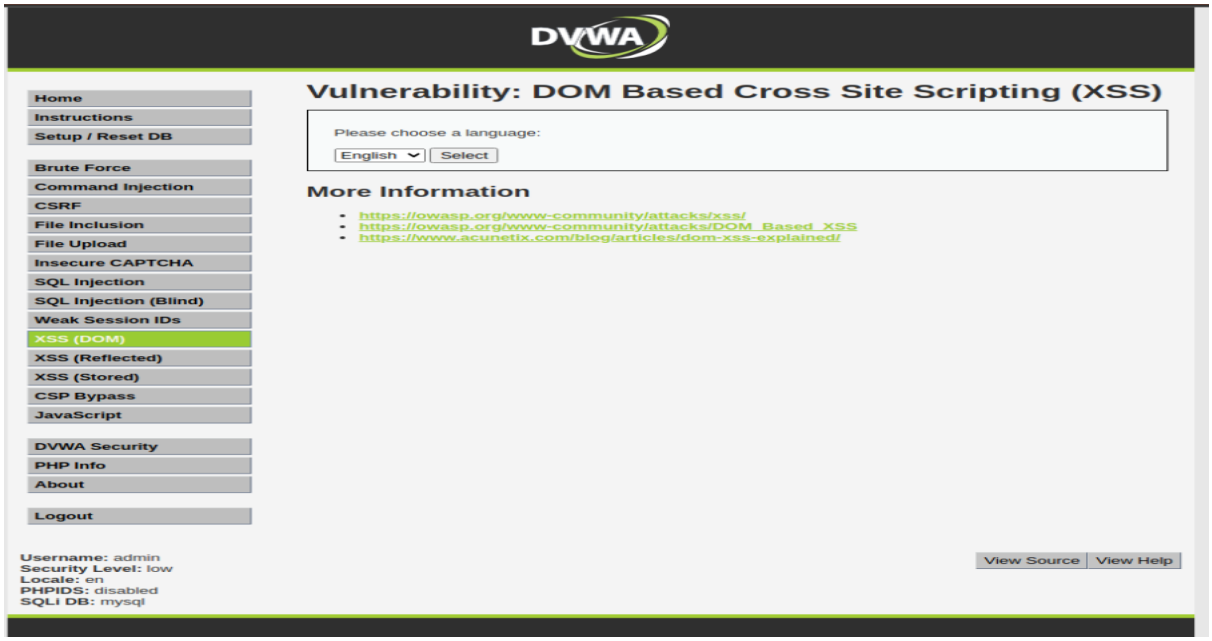
It will start and detect bruteforce attack.

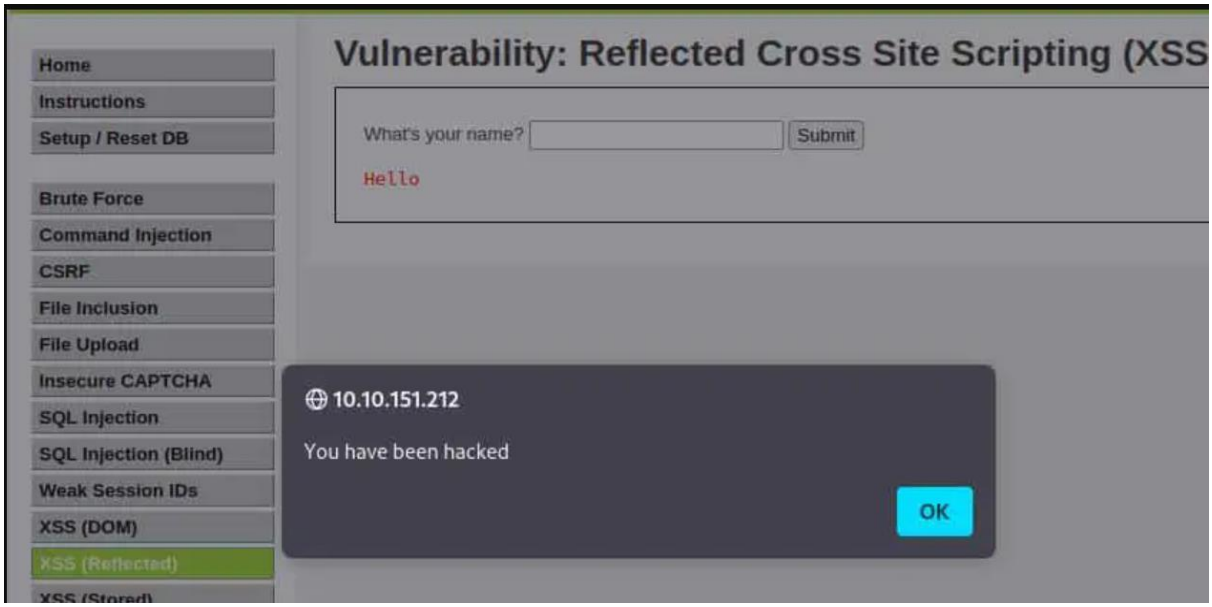


Command injections attack



DOM based cross site scripting (xss)



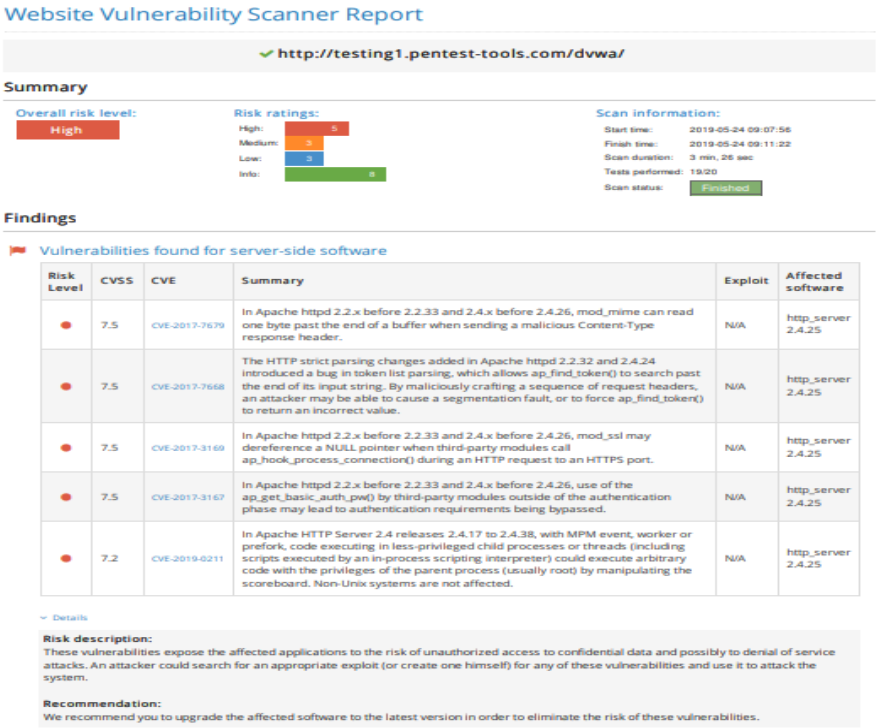


7.REPORT

7.1 Procedure for report

1. Research information about the target system Computers that can be accessed over the internet must have an official IP address. Freely accessible databases provide information about the IP address blocks assigned to an organization.
2. Scan target systems for services on offer An attempt is made to conduct a port scan of the computer(s) being tested, open ports being indicative of the applications assigned to them.
3. Identify systems and applications The names and version of operating systems and applications in the target systems can be identified by "fingerprinting".
4. Researching Vulnerabilities Information about vulnerabilities of specific operating systems and applications can be researched efficiently using the information gathered.
5. Exploiting vulnerabilities Detected vulnerabilities can be used to obtain unauthorized access to the system or to prepare further attacks.

7.2Reference report



Website Vulnerability Scanner Report

http://testing1.pentest-tools.com/dvwa/

Summary

Overall risk level: High

Risk ratings: High: 5 Medium: 3 Low: 3 Info: 8

Scan information: Start time: 2019-05-24 09:07:56 Finish time: 2019-05-24 09:11:22 Scan duration: 3 min, 26 sec Tests performed: 19/20 Scan status: Finished

Findings

Vulnerabilities found for server-side software

Risk Level	CVSS	CVE	Summary	Exploit	Affected software
	7.5	CVE-2017-7679	In Apache httpd 2.2.x before 2.2.33 and 2.4.x before 2.4.26, mod_mime can read one byte past the end of a buffer when sending a malicious Content-Type response header.	N/A	http_server 2.4.25
	7.5	CVE-2017-7668	The HTTP strict parsing changes added in Apache httpd 2.2.32 and 2.4.24 introduced a bug in token list parsing, which allows ap_find_token() to search past the end of its input string. By maliciously crafting a sequence of request headers, an attacker may be able to cause a segmentation fault, or to force ap_find_token() to return an incorrect value.	N/A	http_server 2.4.25
	7.5	CVE-2017-3169	In Apache httpd 2.2.x before 2.2.33 and 2.4.x before 2.4.26, mod_ssl may dereference a NULL pointer when third-party modules call ap_hook_process_connection() during an HTTP request to an HTTPS port.	N/A	http_server 2.4.25
	7.5	CVE-2017-3167	In Apache httpd 2.2.x before 2.2.33 and 2.4.x before 2.4.26, use of the ap_get_basic_auth_pw() by third-party modules outside of the authentication phase may lead to authentication requirements being bypassed.	N/A	http_server 2.4.25
	7.2	CVE-2019-0211	In Apache HTTP Server 2.4 releases 2.4.17 to 2.4.38, with MPM event, worker or prefork, code executing in less-privileged child processes or threads (including scripts executed by an in-process scripting interpreter) could execute arbitrary code with the privileges of the parent process (usually root) by manipulating the scoreboard. Non-Unix systems are not affected.	N/A	http_server 2.4.25

Details

Risk description:
These vulnerabilities expose the affected applications to the risk of unauthorized access to confidential data and possibly to denial of service attacks. An attacker could search for an appropriate exploit (or create one himself) for any of these vulnerabilities and use it to attack the system.

Recommendation:
We recommend you to upgrade the affected software to the latest version in order to eliminate the risk of these vulnerabilities.

Cross-Site Scripting

Vulnerable Page	Vulnerable Parameter	Method	Attack Vector
/dvwa/login.php	username	POST	http://testing1.pentest-tools.com/dvwa/login.php **POST Data:** username= <script>alert(1);</script>

1 / 8

/dvwa/vulnerabilities/brute/	username	GET	http://testing1.pentest-tools.com/dvwa/vulnerabilities/brute/?Login=Login&password=ZAP&username=%27%22%3Cscript%3Ealert%281%29%3B%3C%2Fscript%3E
/dvwa/vulnerabilities/sqli/	id	GET	http://testing1.pentest-tools.com/dvwa/vulnerabilities/sqli/?Submit=Submit&id=%27%22%3Cscript%3Ealert%281%29%3B%3C%2Fscript%3E
/dvwa/vulnerabilities/xss_r/	name	GET	http://testing1.pentest-tools.com/dvwa/vulnerabilities/xss_r/?name=%3C%2Fpre%3E%3Cscript%3Ealert%281%29%3B%3C%2Fscript%3E%3Cpre%3E
/dvwa/vulnerabilities/xss_s/	txtName	POST	http://testing1.pentest-tools.com/dvwa/vulnerabilities/xss_s/ **POST Data:** txtName= <script>alert(1);</script>
/dvwa/vulnerabilities/xss_s/	mtxMessage	POST	http://testing1.pentest-tools.com/dvwa/vulnerabilities/xss_s/ **POST Data:** mtxMessage= <script>alert(1);</script>

Details

Risk description:
The risk exists that a malicious actor injects JavaScript code and runs it in the context of a user's session in the application. This could potentially lead to various effects such as stealing session cookies, calling application features on behalf of another user, exploiting browser vulnerabilities.

Successful exploitation of Cross-Site Scripting attacks requires human interaction (ex. determine the user access a special link by social engineering).

Recommendation:
There are several ways to mitigate XSS attacks. We recommend to:
- never trust user input
- always encode and escape user input (using a Security Encoding Library)
- use the HTTPOnly cookie flag to protect from cookie theft
- implement Content Security Policy
- use the X-XSS-Protection Response Header

References:
https://www.owasp.org/index.php/Cross-site_Scripting_(XSS)
https://www.owasp.org/index.php/XSS_(Cross_Site_Scripting)_Prevention_Cheat_Sheet

SQL Injection

Vulnerable Page	Vulnerable Parameter	Method	Attack Vector
/dvwa/vulnerabilities/brute/	username	GET	http://testing1.pentest-tools.com/dvwa/vulnerabilities/brute/?Login=Login&password=ZAP&username=ZAP
/dvwa/vulnerabilities/sqli/	id	GET	http://testing1.pentest-tools.com/dvwa/vulnerabilities/sqli/?Submit=Submit&id=ZAP%27+AND+%271%27%3D%271%27+--+
/dvwa/vulnerabilities/sqli_blind/	id	GET	http://testing1.pentest-tools.com/dvwa/vulnerabilities/sqli_blind/?Submit=Submit&id=ZAP%27+AND+%271%27%3D%271%27+--+
/dvwa/vulnerabilities/xss_s/	btnSign	POST	http://testing1.pentest-tools.com/dvwa/vulnerabilities/xss_s/ **POST Data:** btnSign=Sign Guestbook" AND "1"="1" --

Details

Risk description:
The risk exists that an attacker gains unauthorized access to the information from the database of the application. He could extract information such as: application usernames, passwords, client information and other application specific data.

8.REFERENCE

- *https://www.golinuxcloud.com/dvwa-sql-injection/*
- *https://dtwh.medium.com/damn-vulnerable-web-application-dvwa-brute-force-walkthrough-722f33a3c725*
- *https://www.golinuxcloud.com/install-dvwa-kali-linux/*
- *https://linuxhint.com/install-damn-vulnerable-web-application-dvwa-kali-linux/*
- *https://cybereaon.com/wp-content/uploads/2021/05/sample_report_web.pdf*
- *https://augment1security.com/authentication/dvwa-authentication/*
- *https://www.eccouncil.org/cybersecurity-exchange/application-security/threat-mitigation-strategies-for-securing-web-applications/*