CSE 460 Web Application Penetration Testing

Project Report

K Bala Thripura Venkata Srivalli
AP22110011471
CSE-Y

Problem Statement:

Develop a tool to detect open directories and security misconfigurations. (57)

Modern web applications often suffer from improperly configured servers, frameworks, and environments. These misconfigurations may lead to serious security vulnerabilities like unauthorized access, data leaks, or even full system compromise. The project aims to create an automated tool that scans web applications for open/exposed directories and common security header misconfigurations.

Introduction

Security Misconfigurations:

Security misconfiguration refers to the improper or inadequate configuration of security settings within software, hardware, applications, or networks, leading to vulnerabilities that can be exploited by attackers. This can happen when default configurations are not modified, unnecessary features are left enabled, or access controls are not properly enforced. Misconfigurations are a significant source of data breaches and unauthorized access.

What it is:

- **Improper Configuration:**
  Security settings are not defined or maintained correctly, leaving systems vulnerable.
- **Default Settings:**
  Software and applications are deployed with default settings that are often insecure and should be changed.
- **Unnecessary Features:**
  Unwanted or unused features are enabled, potentially creating security holes.
- **Weak Permissions:**
  Access controls are not properly set, allowing unauthorized users to access sensitive data or systems.
- **Missing Configurations:**
  Essential security settings are entirely absent, creating gaps in the security posture.

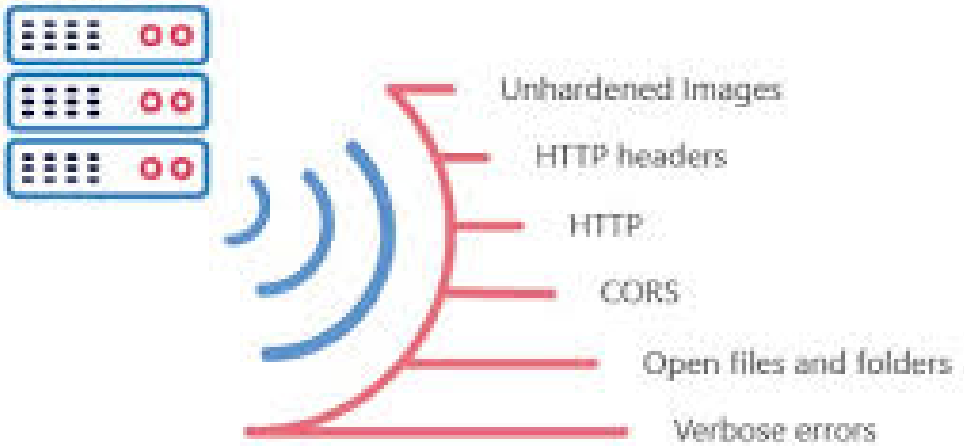
Why it's a problem:

- **Data Breaches:**
  Misconfigurations can lead to unauthorized access to sensitive data, resulting in data breaches and financial losses.
- **Unauthorized Access:**
  Attackers can exploit misconfigurations to gain access to systems and networks.

Malware Infection:
Misconfigurations can make it easier for malware to infiltrate systems and networks.
Denial-of-Service:
Misconfigured systems can be vulnerable to denial-of-service attacks.
Regulatory Violations:
Misconfigurations can lead to non-compliance with industry regulations and standards.

Examples:

- **Default Passwords:** Using default usernames and passwords for applications or services.
- **Unprotected Cloud Storage:** Leaving cloud storage buckets open to public access.
- **Open Ports:** Unnecessary ports are left open on firewalls, exposing services to attacks.
- **Outdated Software:** Failing to update software and firmware, leaving systems vulnerable to known exploits.



A6:2017 Security Misconfiguration

Open directories

An "open directory" refers to a web server where a directory is publicly accessible without requiring any authentication or password. This means anyone can browse the contents of that directory, potentially exposing sensitive information if not configured securely.

Security Risk:
They can expose sensitive files like database credentials, source code, or confidential documents, making them a significant security vulnerability.

How they occur:
Open directories often result from misconfigurations or inadequate security settings on web servers.

How i solved:

Nikto is a free, open-source, command-line web server vulnerability scanner that checks for dangerous files, outdated server software, and other security issues.

Here's a more detailed breakdown of Nikto:

Purpose:

Nikto is designed to identify vulnerabilities in web servers by scanning for various issues, including:

- Outdated or unpatched software
- Dangerous files or CGIs
- Server configuration errors
- Version-specific problems

Features:

- **Comprehensive Scanning:** Nikto performs a wide range of tests to identify potential security flaws.
- **Extensive Database:** It utilizes a database of known vulnerabilities and regularly updates it with the latest security threats.
- **Speed:** Nikto is designed to be a fast scanner.
- **Multiple Targets:** It can scan multiple ports on a single server or multiple servers.
- **Cookie Capture:** Nikto can capture and print any cookies received during the scan.

Usage:

Nikto is a command-line tool, meaning you interact with it using a terminal or command prompt.

- You can specify the target web server by its domain name or IP address.
- It can also scan specific ports.

Open Source and Free:

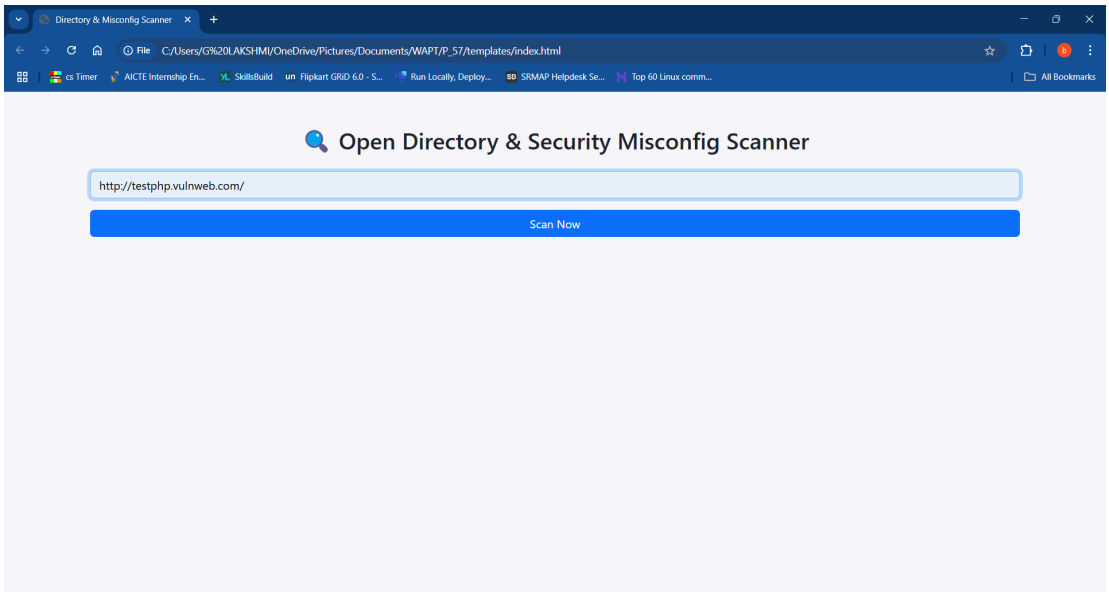
Nikto is available under the GNU General Public License, meaning it's free to use and distribute.

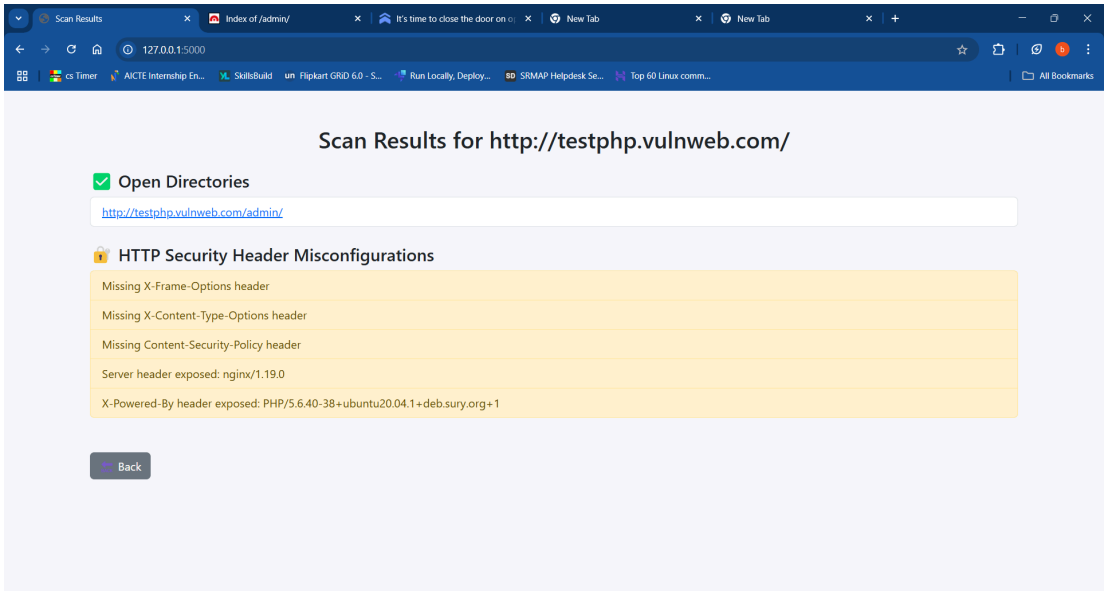
Availability:

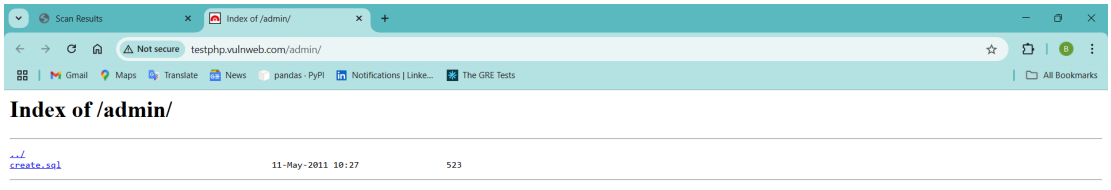
Nikto is built into Backtrack and available in the pentest/scanners/nikto directory.

Web Page Results:

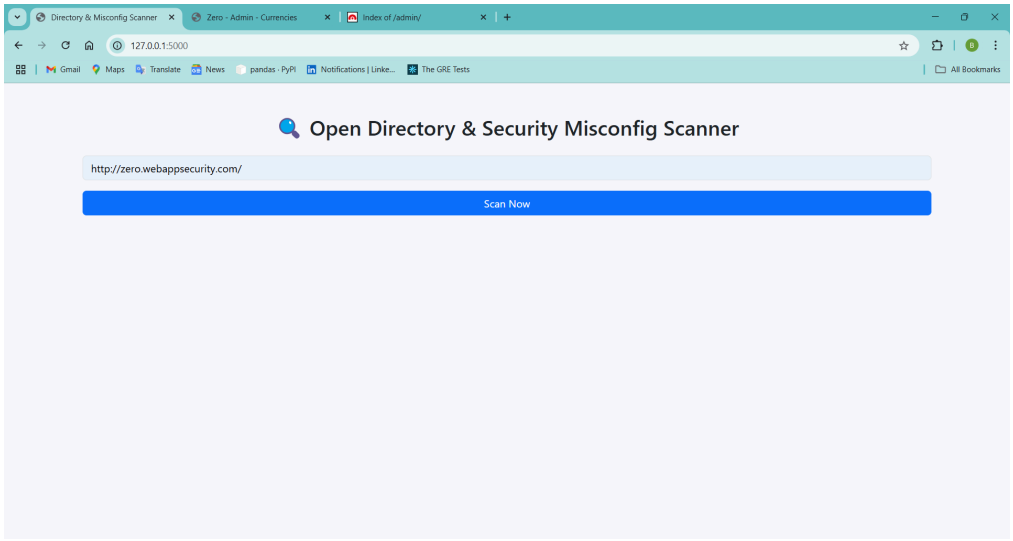
1) http://testphp.vulnweb.com/

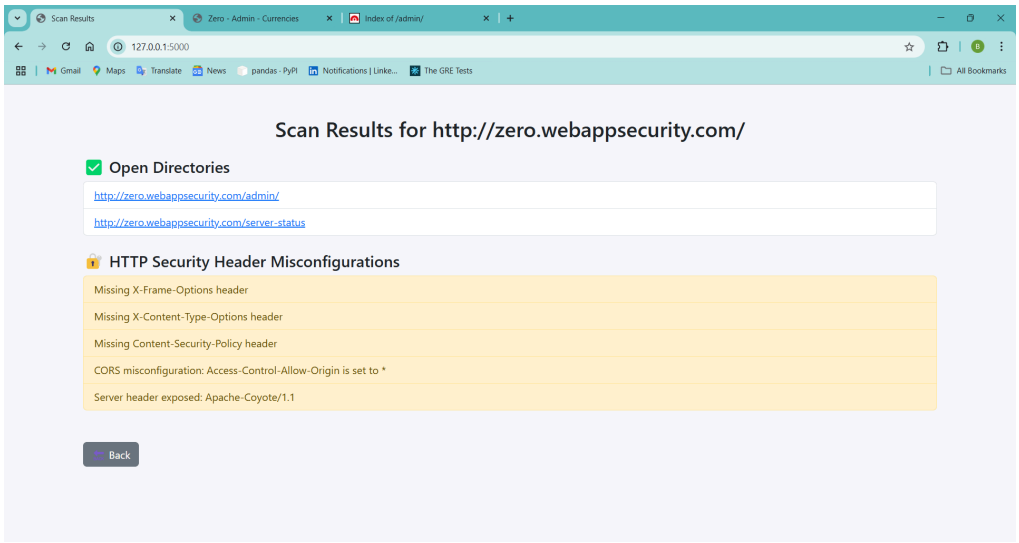


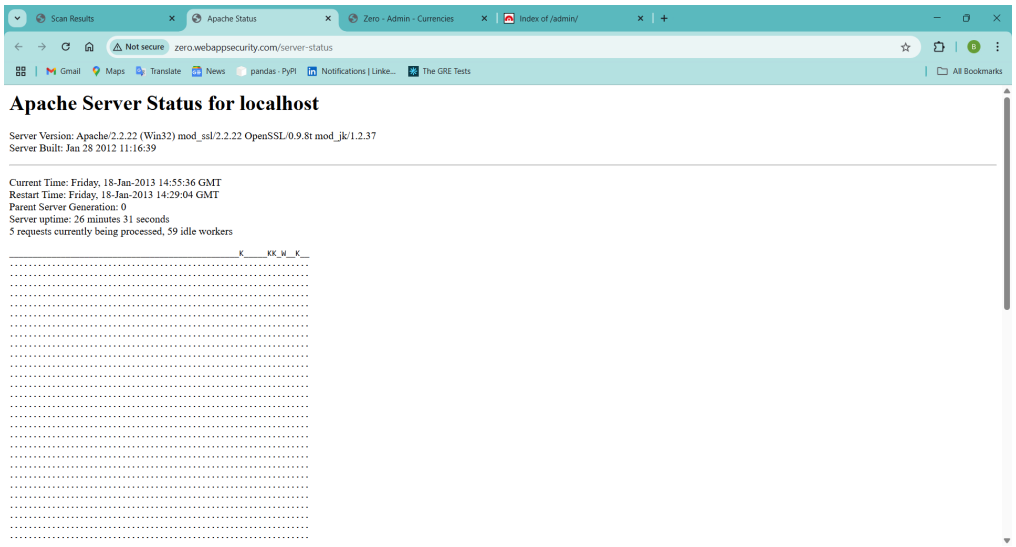


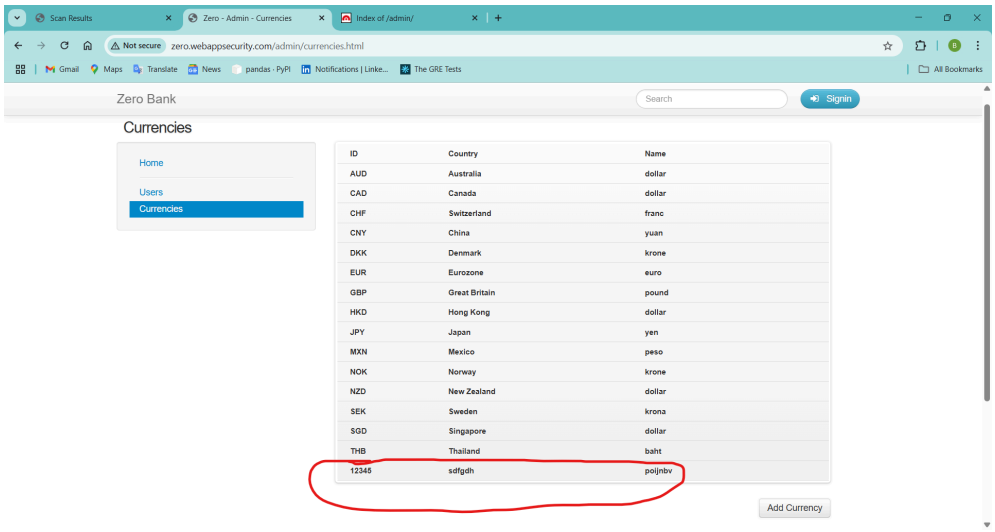


2) http://zero.webappsecurity.com/

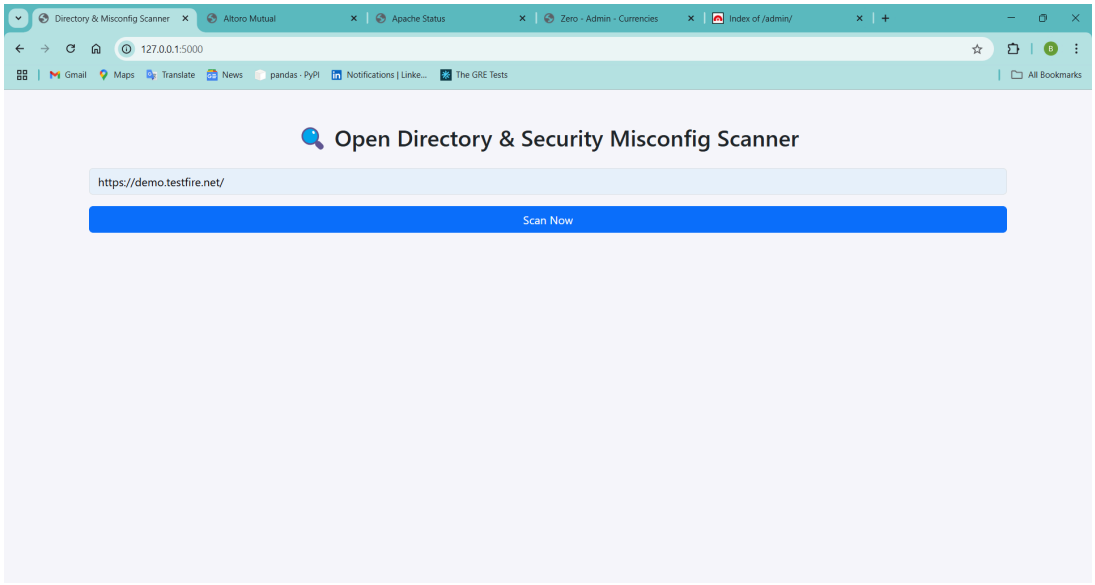


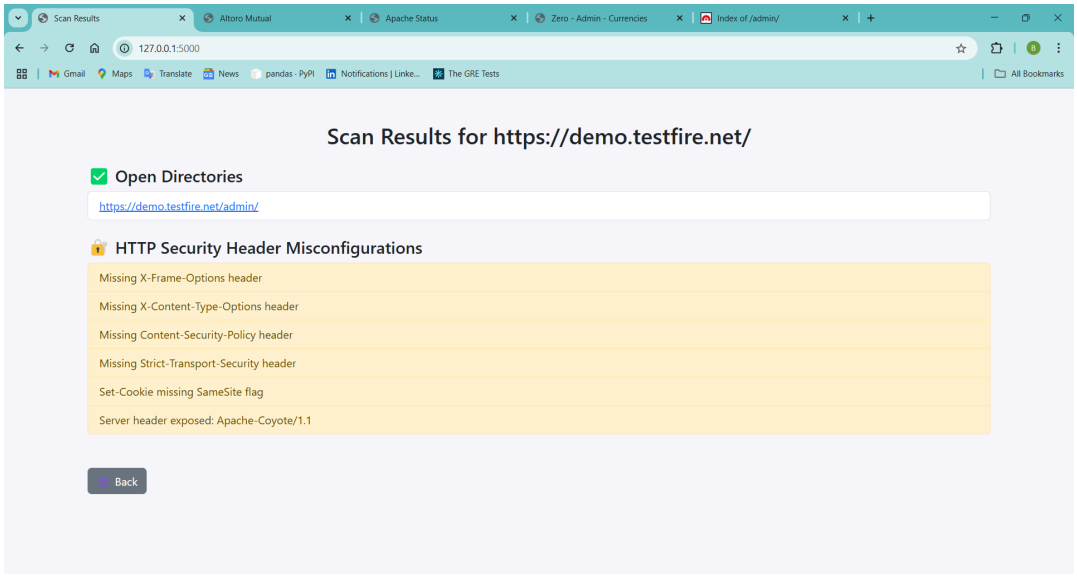


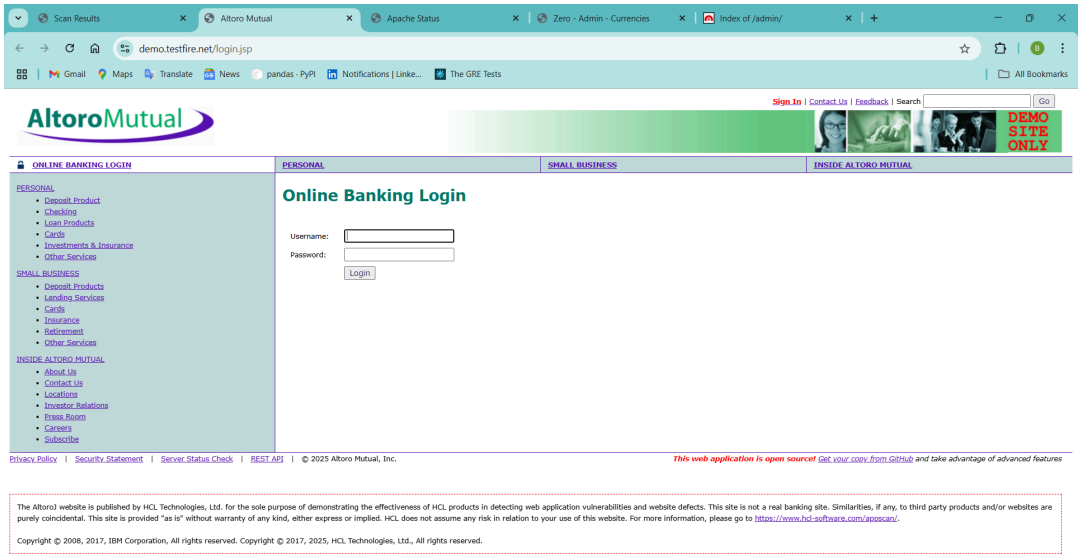




3) https://demo.testfire.net/







1. 📁 Exposed Directories and Files

Definition: Public access to internal folders or files that should not be accessible via the browser.

Examples:

- `.git/` → Can expose full codebase and commit history

- `.env` → Contains secrets and API keys
- `backup/`, `db.sql` → May contain database dumps
- `phpinfo.php` → Reveals software versions and server paths

---

2. 🔐 Missing or Misconfigured HTTP Headers

Header	Purpose	Risk if Missing
`X-Frame-Options`	Prevents clickjacking	UI redress attacks
`X-Content-Type-Options`	Stops MIME sniffing	Script injection
`Content-Security-Policy`	Controls script/resources	XSS vulnerabilities
`Strict-Transport-Security`	Forces HTTPS	Downgrade to HTTP possible
`Access-Control-Allow-Origin`	Manages CORS	Data exfiltration via JS
`X-Powered-By`, `Server`	Reveals tech stack	Easier fingerprinting by attacker

---

3. 🍪 Insecure Cookie Settings

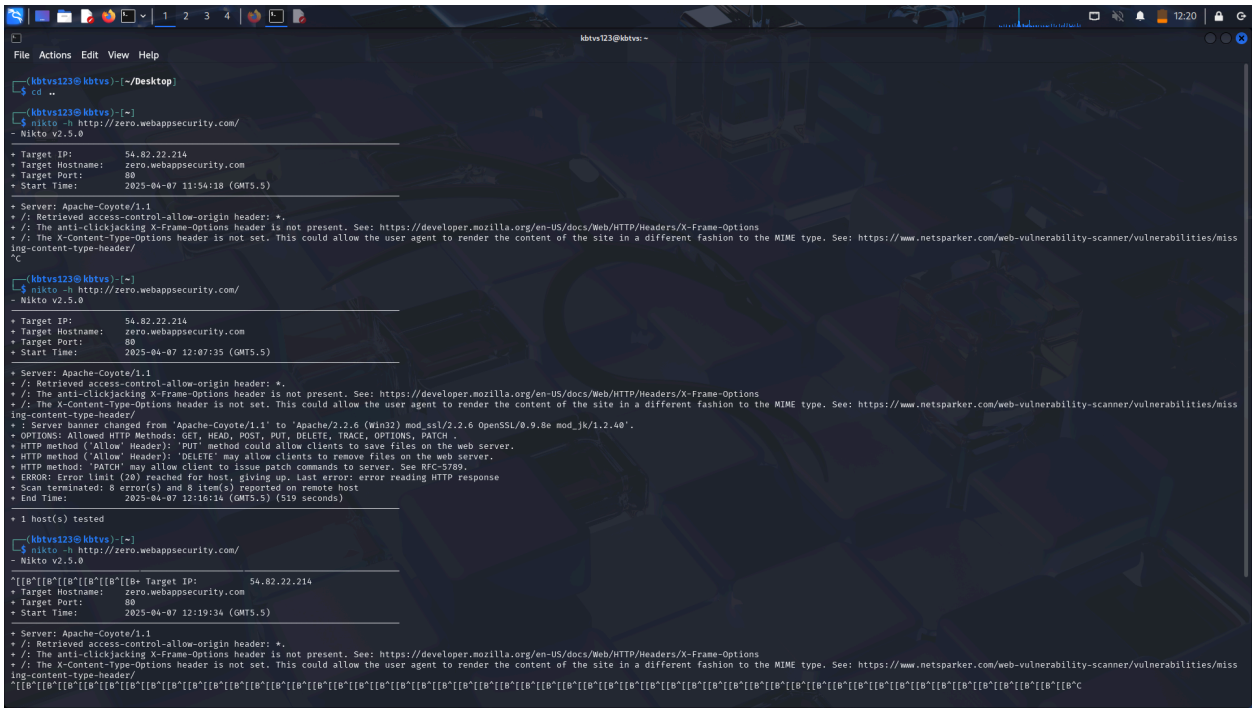
If your web app sets cookies without proper flags, they can be intercepted or stolen.

Cookie Flag	Purpose	Risk
`Secure`	Sends cookie only over HTTPS	Can be stolen via HTTP
`HttpOnly`	Blocks access via JavaScript	Helps prevent XSS
`SameSite`	Controls cross-origin access	Mitigates CSRF attacks

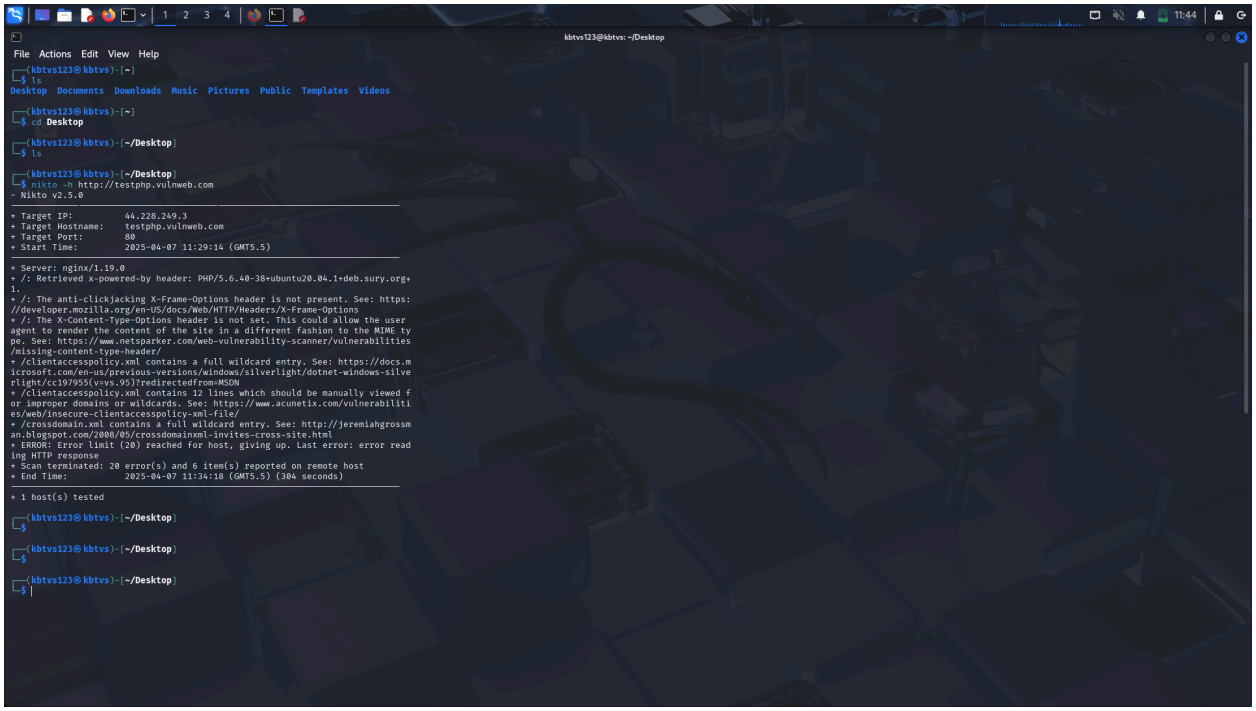
Nikto Command Results:

“nikto -h url”

nikto -h http://zero.webappsecurity.com/



nikto -h http://testphp.vulnweb.com/



nikto -h https://example.com

nikto -h http://demo.testfire.net/

Impact if the Problem is Not Solved

If open directories and security misconfigurations are left unchecked:

🚨 **Consequences Include:**

- **Data Breaches:** Exposed `.env` or database backups can leak credentials or personal data.
- **Source Code Leakage:** Public `.git` folders or backup directories can reveal proprietary code.
- **Cross-Site Scripting (XSS):** Lack of a proper Content Security Policy increases the risk.
- **Clickjacking:** Missing frame protections allow attackers to trick users into performing actions they didn't intend.
- **Session Hijacking:** Poor cookie configurations make session theft easier for attackers.
- **Reconnaissance Exposure:** Headers like `Server` and `X-Powered-By` give attackers clues about the system.

🛡️ **Long-Term Effects:**

- Regulatory penalties (GDPR, HIPAA) due to data leaks
- Loss of user trust and reputation
- Financial damages due to breaches or ransomware
- Exploitation in automated botnets and malware campaigns