**CENTRE FOR DEVELOPMENT OF ADVANCED COMPUTING (C-DAC), THIRUVANANTHAPURAM, KERALA**

## A PROJECT REPORT ON

**“A Comprehensive Security Audit on**

**zero.webappsecurity.com”**

**SUBMITTED TOWARDS THE**



PG-DCSF September 2023

**BY**

**Group Number - 06**

**Dinesh Zatke PRN: 230960940014**

**Mohd Naseem Ansari PRN: 230960940027**

**Yogesh Avarkar PRN: 230960940060**

**Vidhi Chaudhari PRN: 230960940055**

**Rohit Ujjir PRN: 230960940043**

**Under The Guidance Of**

**Mr. Jayaram P. Mr. Sreedeep A. L.**

**Course Co- Ordinator Project Guide**

# TABLE OF CONTENTS

**Contents Page No.**

[Abstract 3](#_TOC_250007)

[Introduction 4](#_TOC_250006)

[Literature Survey 5](#_TOC_250005)

[Scope and Objectives 6](#_TOC_250004)

[Methodology 7](#_TOC_250003)

1. Phases of security auditing 8
2. Tools 10
3. Vulnerabilities 22

[Conclusion 32](#_TOC_250001)

[References 33](#_TOC_250000)

# ABSTRACT

A security audit is a systematic and comprehensive examination of an organization's information systems, technology infrastructure, processes, and policies to assess the effectiveness of security measures, identify vulnerabilities, and ensure compliance with security best practices and standards. The primary objective of a cybersecurity audit is to evaluate the organization's ability to protect its digital assets, sensitive information, and technology resources from cyber threats and attacks. The goal of this project is to identify vulnerabilities in a web application, in order to provide recommendations for improving security. The project will involve DNS reconnaissance and mapping of the target application, identification of potential vulnerabilities through manual testing following the OWASP Top 10 and using more than 8 well known automated tools like the Nessus so that we don’t miss on the important vulnerabilities

The project aims to assess the security posture of the web application, identify potential attack vectors, analyze the scope of the vulnerabilities and their impact, and develop a remediation plan to address the identified security issues. The project also involves an actionable set of security recommendations to ensure the secure operation of the web applications

**Keywords:** Security Auditing, Web Application Security, Vulnerabilities, D N S Reconnaissance, OWASP Top 10.

# INTRODUCTION

In an ever-evolving landscape of technological advancements and digital transformation, ensuring the security and integrity of systems, networks, and data has become paramount. As organizations increasingly rely on digital infrastructure to operate, communicate, and store sensitive information, the potential risks and vulnerabilities also escalate. A proactive approach to identifying, mitigating, and managing these risks is essential to safeguarding an organization's assets, reputation, and stakeholder trust.

This Security Audit Project Report delves into the comprehensive assessment conducted to evaluate the security posture of **Zero Bank Website**. The primary objective of this security audit was to systematically examine the effectiveness of existing security measures, policies, and practices, and to recommend improvements that align with industry best practices and regulatory requirements. By performing a thorough analysis of the organization's information technology infrastructure, data handling procedures, and access controls, this audit aims to provide actionable insights for enhancing the organization's overall security framework.

The report is structured to provide a clear understanding of the audit scope, methodology employed, findings uncovered, and subsequent recommendations. Additionally, it underscores the importance of a security-centric mind-set within the organization's culture and emphasizes the significance of continuous monitoring and adaptation to counter the ever-changing threat landscape.

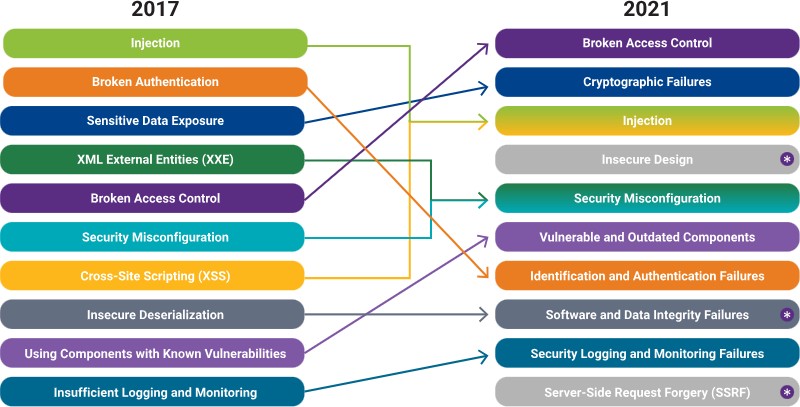
In the subsequent sections, we will explore the key aspects of the security audit, highlighting its significance in the context of modern-day cyber threats and underscoring the collaborative efforts undertaken by the audit team to ensure the confidentiality, integrity, and availability of **Zero Bank**'s critical assets.

# LITERATURE SURVEY

The OWASP Top 10 is a well-known list of the top 10 most critical security risks commonly found in web applications. Including these in your Security Audit Project Report helps to highlight key vulnerabilities that should be addressed. As of my last update in September 2021, here's the OWASP Top 10 list:

OWASP Top 10 Security Risks - 2021

1. Broken Access Control
2. Cryptographic Failures
3. Injection
4. Insecure Design
5. Security Misconfiguration
6. Vulnerable and Outdated Components
7. Identification and Authentication Failure
8. Software and Data Integrity Failures
9. Security Logging and Monitoring Failures
10. Server-Side Request Forgery



Figc1.1: OWASP Top 10 (2017 Vs 2021)

Reference Link: https://owasp.org/www-project-top-ten/

# SCOPE AND OBJECTIVES

The scope of the project involves a comprehensive evaluation of its digital infrastructure, applications, and data protection mechanisms. The primary focus will be on identifying vulnerabilities, weaknesses, and potential threats that could compromise the confidentiality, integrity, and availability of resources of the website. The security audit will cover both technical and operational aspects, including the assessment of software, network architecture, user access controls, and adherence to relevant security standards and best practices. The security audit will be done both manually and automatically using latest and legitimate tools available.

The project will also extend to evaluating user authentication mechanisms, encryption practices, and incident response procedures. The audit will primarily concentrate on online security, as the project is done completely online.

The main objective of the project is to identify the vulnerabilities by Conducting a thorough assessment of the website's infrastructure to identify potential security vulnerabilities, such as SQL injection, cross-site scripting (XSS), file upload vulnerabilities, password policy etc.

Further we aim to provide actionable recommendations and best practices to address identified vulnerabilities and enhance the overall security posture and reputation of the website, thereby enhancing user trust and safeguarding user data.

# METHODOLOGY

The current security systems need to be tested for both substantive and compliance aspects. Compliance testing is done to assess whether controls are being applied according to the documentation offered by the client. It also checks if IT controls follow the compliance levels in accordance with management procedures and policies. In substantive testing, the adequacy of the controls is substantiated by whether they are able to protect the organization from cyber threats. These tests need an in-depth understanding of the different kinds of threats such as unauthorized access to assets including data, unusual interactions with the system, data corruption, inaccuracy in information, etc. Application controls are application-specific controls and have a high impact on individual transactions. These controls ensure and verify that all transactions are authorized, safe, and recorded. To proceed with this phase of the audit, there is a need for a deep understanding of the working of the system. For this analysis, a brief description of the application is required, along with details of transactions including volume, involved data, and flow. Most organizations either use local area networks for their operations. This leads to the risk of access by unauthorized users if not monitored and protected properly. The fundamental requirement of a network is to be accessible by only authorized users. Controls should be implemented to eliminate issues like data corruption, data loss, or interception while being transmitted.

## IT Audit standards

The IT audit should comply with internationally accepted security standards. Some of these are mentioned below:

* **ISO Compliance:** The ISO publishes a slew of guidelines that ensure reliability, quality, and safety. ISO 27001 is suitable for information security requirements.
* **PCI DSS Compliance:** These standards apply to any company that is involved with customer payments. This is necessary to ensure that all transactions are secure and protected.

# PHASES OF SECURITY AUDIT

There are 4 significant phases in a security audit:

### Planning phase

**Preliminary information gathering and assessment**

Planning is an integral part of any audit. In the beginning, planning is done to create a process flow based on an initial reconnaissance of the entire system. The plan is updated according to the test results of the initial assessment.

### Audit scope and objective

From the above steps, the auditor gains relevant information and details to define the objective and scope of the audit in a clear and detailed format. The initial risk assessment forms an important part of the process and answers questions pertaining to three primary security goals, confidentiality, integrity, and reliability.

Risk assessment consists of ranking the potential threats from low to high, or other scientific or complex metrics. The ranking depends on the severity of the issue with respect to the extent of damage it can cause or the ease of exploitation. Vulnerabilities that are easy to exploit and those causing a high degree of damage must be ranked comparatively higher.

### Evaluating collected evidence

Through rigorous testing and prodding of the security infrastructure, various types of evidence are gathered that must be interpreted to compile the results of the audit. There are various techniques to test a system and obtain results. Evidence can be majorly 3 types:

* Documentary evidence
* System analysis
* Observation of processes

### Documenting audit results

Proper documentation of the results forms an integral part of security audit methodology. The final report should be in a very consumable format for stakeholders at all levels to understand and interpret. It must contain details such as the audit plan, audit scope, tests carried out, findings and detailed solutions, and next steps to remedy the security issues.

# Reconnaissance

Reconnaissance is the information-gathering stage of ethical hacking, where you collect data about the target system. This data can include anything from network infrastructure to employee contact details. The goal of reconnaissance is to identify as many potential attack vectors as possible.

How Reconnaissance Works

Reconnaissance generally follows seven steps:

1. Collect initial information
2. Determine the network range
3. Identify active machines
4. Find access points and open ports
5. Fingerprint the operating system
6. Discover services on ports
7. Map the network

# TOOLS

Following are the tools which we used to perform reconnaissance on website “**ktu.edu.in**” -

### DNSdumpster

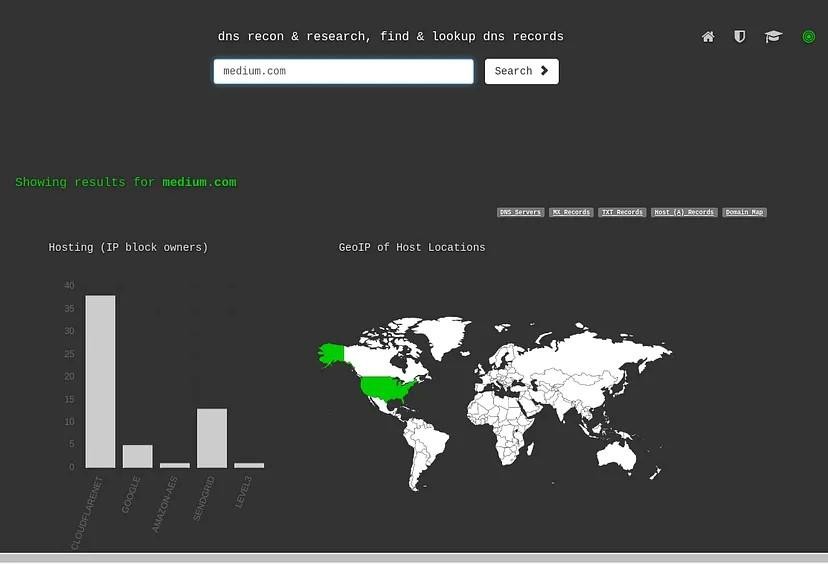
DNSdumpster is an online passive scanning tool to obtain information about domains, block addresses, emails, and all kind of information DNS related. It is a tool to perform DNS reconnaissance on target networks. The results include a variety of information that are useful for users performing network reconnaissance. Some of the information return include

* + Host subdomains
  + Different dns in format (MX, A record)
  + Geo information
  + Email

It is an open-source intelligence for the networks of your choice. With the help of this site or platform, the years can identify the attack surface or potentials. You can also analyze the security strategy related to the information of the network with the help of passive DNS reconnaissance, and eventually get rid of the threat to security.

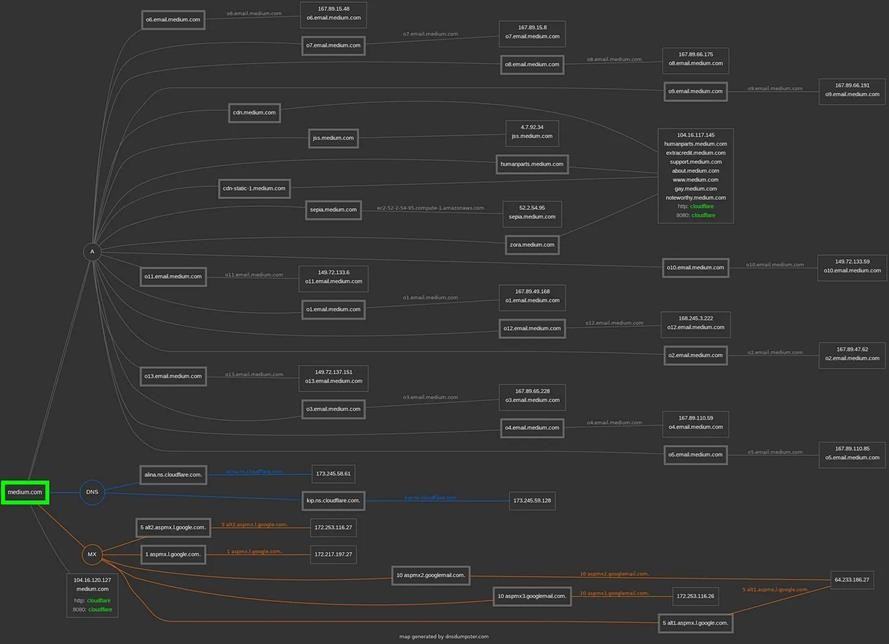
Since it’s a web-based service we only need to navigate to their url and query our target.

Example -



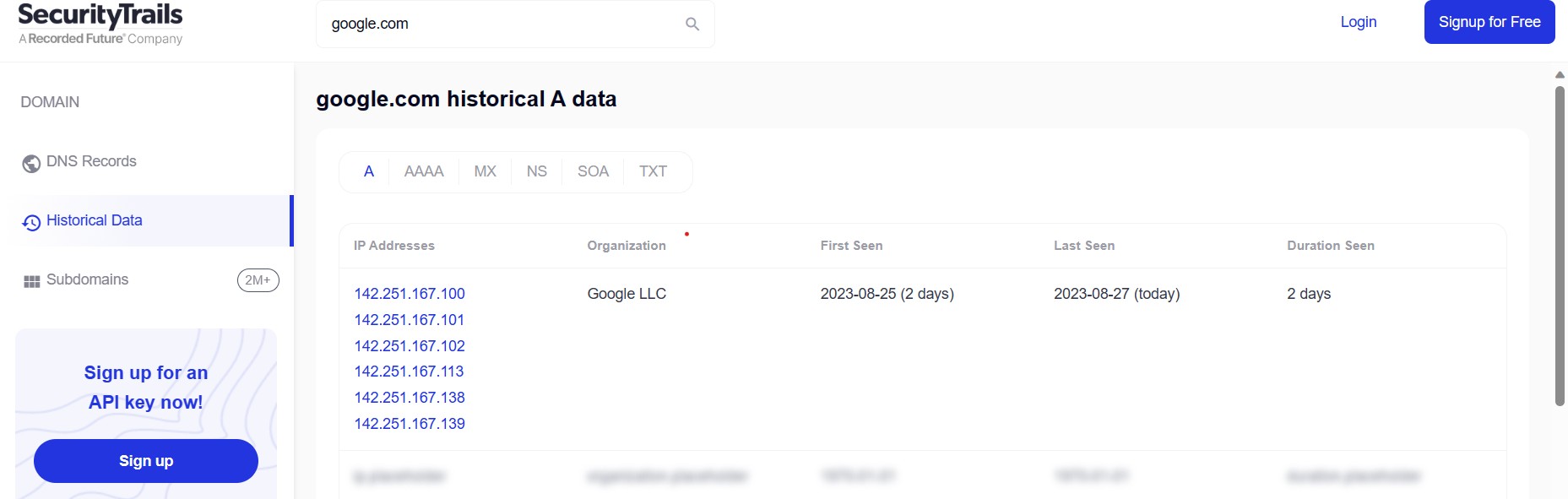
DNSdumpster

DNSdumpster gives us information about MX records, Text records and Host records. While the TXT Records is the section where you will be able to find all the information related to the hosts in Sender Policy Framework (SPF) configurations, the MX Records is the area which is related to all the emails that goes for the domain.

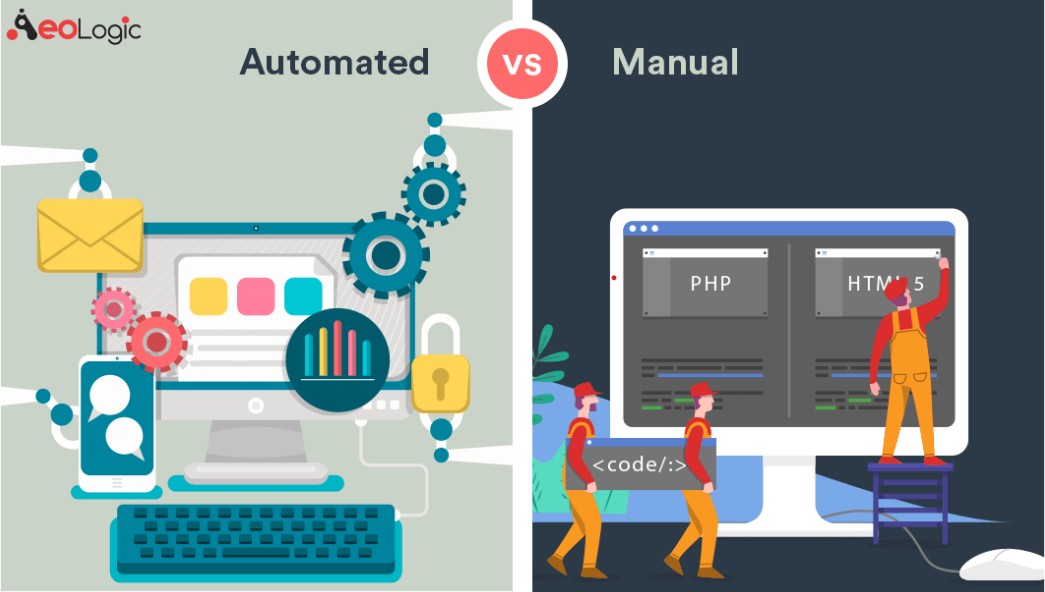
At the end it presents a nice relational picture that binds all records. Example

### Security Trails

A platform that provides data security, threat hunting, and attack surface management solutions.



**Web Application Security Testing**



**Automated Security Testing**

Automated security testing can speed up the testing process with very little effort. Automated security testing tools are good in findings common vulnerabilities (for example XSS, SQL Injection) within a very short time. However, automated scanning tools provide lots of false- positive issues that need to be verified by manual security tester.

Testing a large website manually is a very tedious task for manual security tester as they have to test one by one URL. Automation tools can help the tester to find out basic vulnerabilities quickly and they can focus their time on findings business logic and other security issues which tools cannot find.

**Manual Security Testing**

Manual security testing is performed by Penetration tester who uses his personal skills and experience to find out the vulnerabilities in the application. Some categories of vulnerabilities, such as authorization and business logic flaws, cannot be found with tools and will always require skilled Penetration tester to find them. Manual security testing is a time-consuming process and required application understanding to perform the test.

The Penetration tester r also utilizes some tools to perform testing like customized scripts, proxy tools etc. Unlike automated security testing, false positive issues are not found in manual security testing.

Automation alone is not capable to ensure that an application is thoroughly tested from a security perspective. The application that holds sensitive data required safe to host approval from Penetration tester.

We have made use of the following tools to perform automatic and manual testing on the website “**zero.webappsecurity.com**” -

## Nmap

Nmap is short for Network Mapper. It is an open-source Linux command-line tool that is used to scan IP addresses and ports in a network and to detect installed applications.

Nmap allows network admins to find which devices are running on their network, discover open ports and services, and detect vulnerabilities.

### Why use Nmap?

There are a number of reasons why security pros prefer Nmap over other scanning tools.

* First, Nmap helps you to quickly map out a network without sophisticated commands or configurations. It also supports simple commands (for example, to check if a host is up) and complex scripting through the Nmap scripting engine.
* Ability to quickly recognize all the devices including servers, routers, switches, mobile devices, etc. on single or multiple networks.
* Helps identify services running on a system including web servers, DNS servers, and other common applications. Nmap can also detect application versions with reasonable accuracy to help detect existing vulnerabilities.
* Nmap can find information about the operating system running on devices. It can provide detailed information like OS versions, making it easier to plan additional approaches during penetration testing.

Example -

### Result:

The Nmap tool has been used to conduct a network scan on the domain "zero.webappsecurity.com" while enabling OS detection (-O flag). Nmap is a versatile network scanning tool that aids in discovering open ports, services, and potentially the operating system of target hosts.

During the scan, Nmap sends various network packets to the target host and analyzes the responses to infer information about the underlying operating system. This is achieved by examining unique characteristics of the network stack implementation and behavior of the target system.

The output of the scan will include a list of open ports and services detected on the target host. Additionally, Nmap's OS detection mechanism will attempt to provide an educated guess about the operating system running on the host. This information is based on patterns in the responses received from the target system.

Please note that OS detection is not always 100% accurate, as it relies on heuristics and patterns that might be masked or altered by various factors. However, Nmap's OS detection feature can still provide valuable insights into the likely operating system running on the scanned host.

In summary, the Nmap scan with OS detection on the domain "zero.webappsecurity.com" aims to identify open ports, services, and potentially the underlying operating system of the target host. The results will aid in understanding the network infrastructure and the technology stack in use, contributing to security assessments and network management activities.

## Nessus

Nessus is an open-source network vulnerability scanner that uses the Common Vulnerabilities and Exposures architecture for easy cross-linking between compliant security tools. In fact, Nessus is one of the many vulnerability scanners used during vulnerability assessments and penetration testing engagements, including malicious attacks. Nessus is a tool that checks computers to find vulnerabilities that hackers could exploit.

Nessus works by testing each port on a computer, determining what service it is running, and then testing this service to make sure there are no vulnerabilities in it that could be used by a hacker to carry out a malicious attack.

Nessus can scan these vulnerabilities and exposures:

1. Vulnerabilities that could allow unauthorized control or access to sensitive data on a system
2. Misconfiguration (e.g. open mail relay)
3. Denials of service (Dos) vulnerabilities
4. Default passwords, a few common passwords, and blank/absent passwords on some system accounts

A screenshot of a computer

Description automatically generated

## OWASP ZAP

OWASP ZAP (short for Zed Attack Proxy) is an open-source web application security scanner. It is intended to be used by both those new to application security as well as professional penetration testers.

It has been one of the most active Open Web Application Security Project (OWASP) projects and has been given Flagship status.

When used as a proxy server it allows the user to manipulate all of the traffic that passes through it, including traffic using HTTPS.

It can also run in a daemon mode which is then controlled via a REST API.

Features

Some of the built in features include:

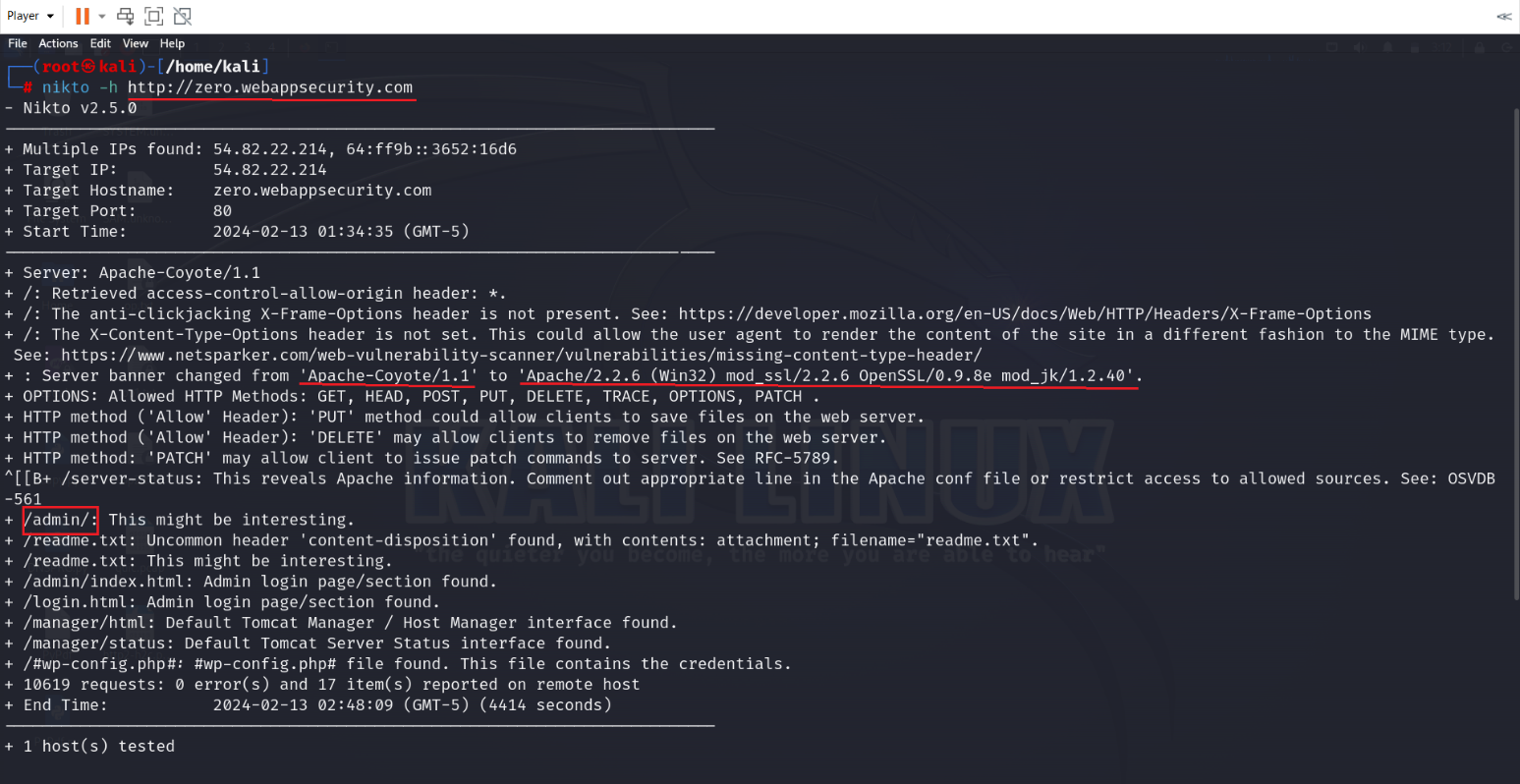
* An intercepting proxy server,
* Traditional and AJAX Web crawlers
* An automated scanner
* A passive scanner
* Forced browsing
* A fuzzer
* WebSocket support
* Scripting languages
* Plug-n-Hack support

Example -



## Nikto

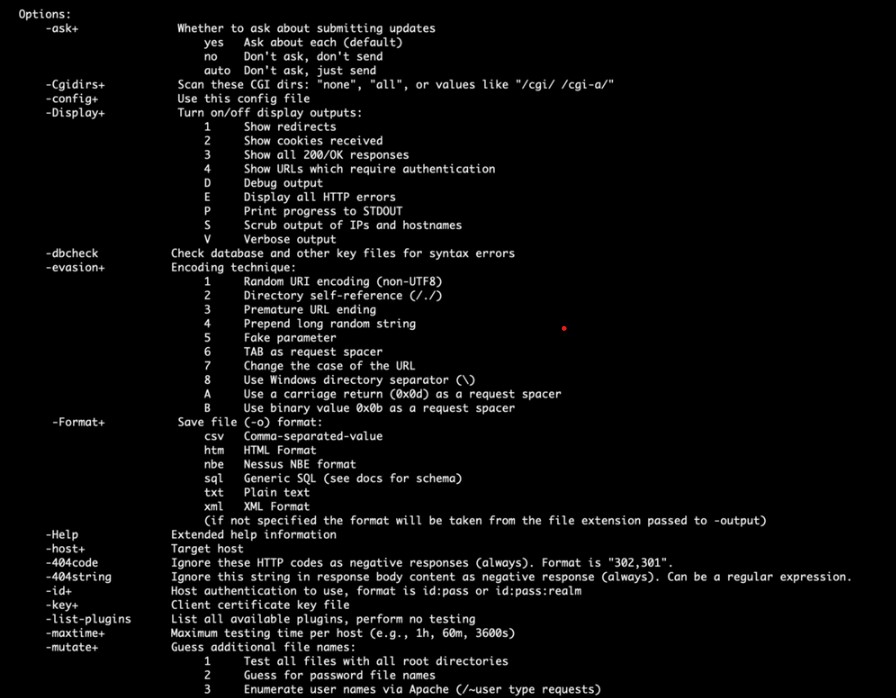
Nikto is an open-source web server and web application scanner. Nikto can perform comprehensive tests against web servers for multiple security threats, including over potentially dangerous files/programs. Nikto can also perform checks for outdated web servers’ software, and version-specific problems.



### Here are some of the things that Nikto can do:

* + Find SQL injection, XSS, and other common vulnerabilities
  + Identify installed software (via headers, favicons, and files)
  + Guess subdomains
  + Includes support for SSL (HTTPS) websites
  + Saves reports in plain text, XML, HTML or CSV
  + Guess credentials for authorization (including many default username/password combinations)

Example –



## Burpsuite

Burp or Burp Suite is a set of tools used for penetration testing of web applications. It is the most popular tool among professional web app security researchers and bug bounty hunters.

The tools offered by BurpSuite are:

### Spider:

It is a web spider/crawler that is used to map the target web application. The objective of the mapping is to get a list of endpoints so that their functionality can be observed and potential vulnerabilities can be found. Spidering is done for a simple reason that the more endpoints you gather during your recon process, the more attack surfaces you possess during your actual testing.

### Proxy:

BurpSuite contains an intercepting proxy that lets the user see and modify the contents of requests and responses while they are in transit. It also lets the user send the request/response under monitoring to another relevant tool in BurpSuite, removing the burden of copy-paste. The proxy server can be adjusted to run on a specific loop-back ip and a port. The proxy can also be configured to filter out specific types of request-response pairs.

### Intruder:

It is a fuzzer. This is used to run a set of values through an input point. The values are run and the output is observed for success/failure and content length. Usually, an anomaly results in a change in response code or content length of the response. BurpSuite allows brute-force, dictionary file and single values for its payload position. The intruder is used for:

* 1. Brute-force attacks on password forms, pin forms, and other such forms.
  2. The dictionary attack on password forms, fields that are suspected of being vulnerable to XSS or SQL injection.
  3. Testing and attacking rate limiting on the web-app.

### Repeater:

Repeater lets a user send requests repeatedly with manual modifications. It is used for:

* 1. Verifying whether the user-supplied values are being verified.
  2. If user-supplied values are being verified, how well is it being done?
  3. What values is the server expecting in an input parameter/request header?
  4. How does the server handle unexpected values?
  5. Is input sanitation being applied by the server?
  6. How well the server sanitizes the user-supplied inputs?
  7. What is the sanitation style being used by the server?
  8. Among all the cookies present, which one is the actual session cookie.

### Sequencer:

The sequencer is an entropy checker that checks for the randomness of tokens generated by the webserver. These tokens are generally used for authentication in sensitive operations: cookies and anti-CSRF tokens are examples of such tokens. Ideally, these tokens must be generated in a fully random manner so that the probability of appearance of each possible character at a position is distributed uniformly. This should be achieved both bit-wise and character-wise.

### Decoder:

Decoder lists the common encoding methods like URL, HTML, Base64, Hex, etc. This tool comes handy when looking for chunks of data in values of parameters or headers. It is also used for payload construction for various vulnerability classes. It is used to uncover primary cases of IDOR and session hijacking.

### Extender:

BurpSuite supports external components to be integrated into the tools suite to enhance its capabilities. These external components are called BApps. These work just like browser extensions. These can be viewed, modified, installed, uninstalled in the Extender window.

**Wireshark**

Wireshark is a free and open-source network protocol analyzer, often referred to as a packet sniffer. It's like having a magnifying glass for your network traffic, allowing you to see the individual packets that make up your internet connection and delve deep into their inner workings.

What it does:

Captures packets flowing across your network interface.

Decodes and analyzes those packets, presenting them in a human-readable format.

Provides detailed information about each packet, including source and destination addresses, protocols used, packet size, and data content.

Offers various filtering and search options to pinpoint specific data or troubleshoot issues.

Common uses:

Network troubleshooting: Identify the source of network slowdowns, connection problems, or errors. Analyze suspicious activity or diagnose malware infections.

Security analysis: Monitor network traffic for vulnerabilities, detect unauthorized access attempts, and investigate security incidents.

Protocol analysis: Deep dive into the inner workings of network protocols like TCP/IP, HTTP, DNS, and more. Understand how applications communicate and identify potential problems.

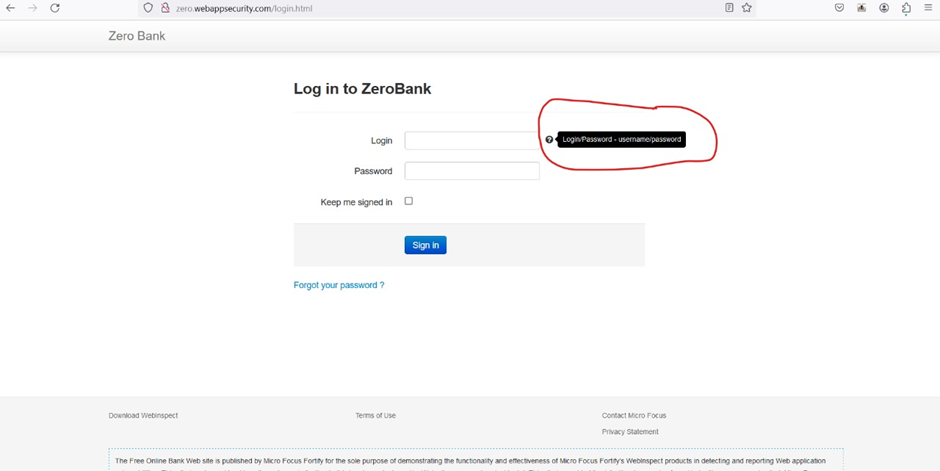
Software development: Test and debug network-based applications, analyze protocol implementations, and ensure proper data exchange.

Network education: Learn about network protocols and their behavior through hands-on exploration and analysis.

**VULNERABILITIES**

1. Weak Default Credentials:

Explanation: The system uses easily guessable default usernames and passwords, making it vulnerable to brute-force attacks and unauthorized access.



Impact: Attackers can gain access to user accounts, potentially compromising sensitive data and disrupting system functionality.

Mitigation:

* + Implement strong password policies requiring complex combinations of characters and regular password changes.
  + Disable default accounts or require immediate password changes upon first login.
  + Consider multi-factor authentication (MFA) for added security.

2. Leaked Password Recovery Mechanism:

Explanation: The password reset process sends passwords in plain text, exposing them to interception and misuse.

A screenshot of a computer

Description automatically generated

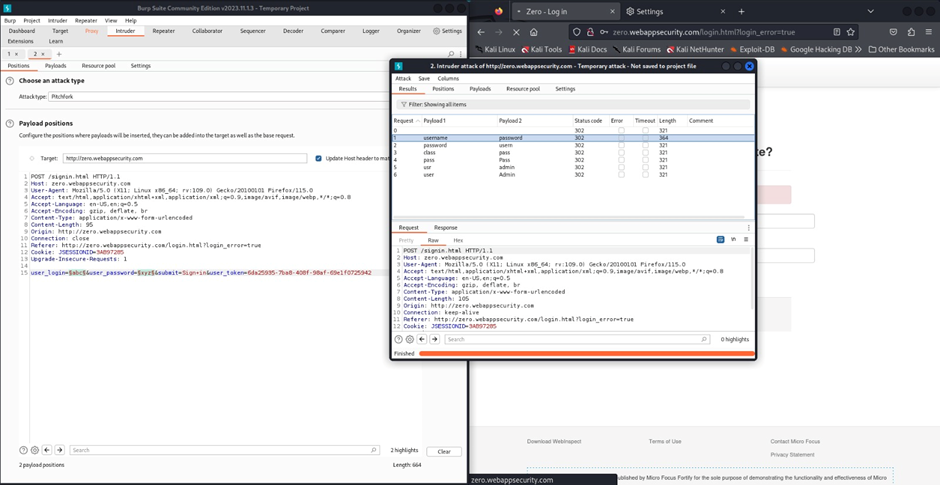
Impact: Attackers can intercept reset emails and gain unauthorized access to user accounts.

Mitigation:

* Implement password reset mechanisms that send secure tokens instead of actual passwords.
* Use email encryption to protect reset messages in transit.
* Consider implementing MFA for password resets.

3. Unencrypted Data Transmission:

Explanation: Sensitive data like login credentials or password reset tokens are transmitted without encryption, making them vulnerable to interception.



Impact: Attackers can intercept unencrypted data and gain unauthorized access to accounts or systems.

Mitigation:

* + Implement Transport Layer Security (TLS) or Secure Sockets Layer (SSL) encryption for all data transmission.
  + Use strong encryption algorithms and regularly update security certificates.

4. Missing Multi-Factor Authentication (MFA):

Explanation: The system relies solely on username and password authentication, which is susceptible to credential theft and brute-force attacks.

Impact: Attackers can gain access to accounts with compromised credentials, leading to data breaches and system disruptions.

Mitigation:

* + Implement MFA for all user accounts, requiring an additional verification factor like a code from a mobile app or security key.
  + Prioritize MFA for sensitive accounts and administrative access.

5. No Password Expiry Policy:

Explanation: Users can keep the same password indefinitely, increasing the risk of compromise over time.

Impact: Attackers can exploit weak or reused passwords to gain access to accounts, even if they haven't been recently compromised.

Mitigation:

* + Implement a password expiry policy requiring regular password changes (e.g., every 3-6 months).
  + Encourage users to create strong and unique passwords for each account.

1. Path Traversal Vulnerability

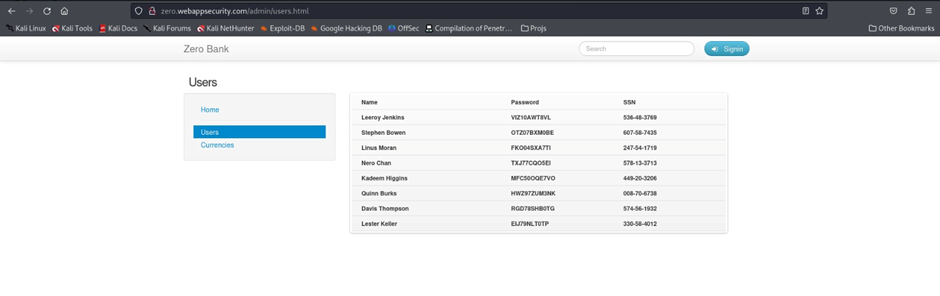
Explanation: A path traversal vulnerability occurs when an attacker manipulates user-supplied input to access files or directories outside the intended web root directory. This can be achieved by using special characters or sequences like ../ (dot-dot-slash) to navigate through the directory structure.

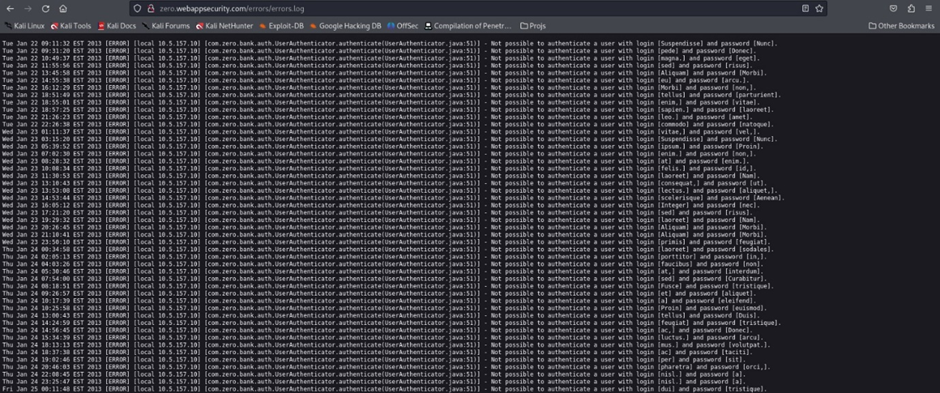
Attackers can exploit this vulnerability to:

Read sensitive files: Access configuration files, source code, or other sensitive data that shouldn't be publicly accessible.

Write or modify files: Tamper with system files, inject malicious code, or deface websites.

Gain unauthorized access: Escalate privileges and potentially take complete control of the server.





A screenshot of a computer

Description automatically generated

Impact: The impact of a path traversal vulnerability can range from mild to severe, depending on the specific context:

Data breaches: Sensitive information like user data, financial records, or intellectual property can be exposed.

Website defacement: Attackers can modify website content, causing reputational damage and loss of trust.

System compromise: In severe cases, attackers can gain complete control of the server, leading to data loss, service disruption, and potential financial losses.

Mitigation:

* Input validation and sanitization: Validate all user-supplied input using robust techniques to ensure it doesn't contain malicious path traversal characters or sequences. Sanitize the input to remove any potential threats before processing it.
* Canonicalization: Use platform-specific functions to canonicalize file paths before processing them. This ensures they are resolved to their intended location, preventing attackers from manipulating them through "../" or other sequences.
* Restrict file access: Implement strict access control mechanisms to limit access to files and directories only to authorized users or processes. This prevents unauthorized individuals from accessing sensitive data even if they exploit a path traversal vulnerability.
* Use secure frameworks and libraries: Utilize frameworks and libraries designed to handle file paths securely and prevent common path traversal vulnerabilities. These frameworks often have built-in sanitization and validation mechanisms to reduce the risk.
* Regular security updates: Keep software and libraries updated with the latest patches to address known vulnerabilities and patching techniques used by attackers. Many path traversal vulnerabilities have been addressed in past updates, so staying current is crucial.
* Security awareness training: Educate developers and users about the dangers of path traversal vulnerabilities and how to avoid them. This can help prevent accidental mistakes that could lead to exploitation.

1. No Account Lockout after Multiple Failed Logins:

Explanation: The system allows unlimited login attempts, making it vulnerable to brute-force attacks.

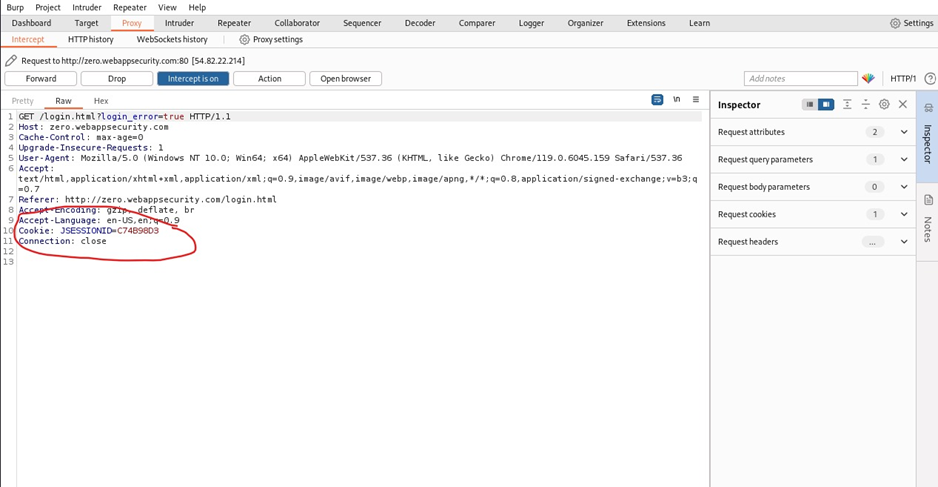
Impact: Attackers can persistently attempt to guess passwords until successful, potentially gaining unauthorized access.

Mitigation:

* Implement account lockout after a specific number of failed login attempts (e.g., 3-5 attempts).
* Increase lockout duration after subsequent attempts to deter persistent attacks.
* Consider CAPTCHA or other challenges to prevent automated password guessing.
* Section 2: Session Management and Access Control Vulnerabilities

8. Plaintext Cookies:

Explanation: Session cookies are stored in plain text, making them vulnerable to interception and manipulation by attackers.



Impact: Attackers can steal session cookies to gain unauthorized access to user accounts and impersonate legitimate users.

Mitigation:

* Set the HTTP Only flag on cookies to prevent client-side scripting access.
* Set the Secure flag on cookies to transmit them only over HTTPS connections.
* Consider using short session durations and regular session regeneration.

9. Predictable Session Identifiers:

Explanation: Session identifiers are easily guessable or generated in a predictable manner, making them vulnerable to session hijacking.

Impact: Attackers can predict or intercept session identifiers to gain unauthorized access to user accounts and sessions.

Mitigation:

* Use strong, random session identifiers generated using cryptographic algorithms.
* Regularly regenerate session identifiers to minimize the window of vulnerability.
* Implement server-side session validation to prevent unauthorized access attempts.

10. Directory Brute Forcing Vulnerability

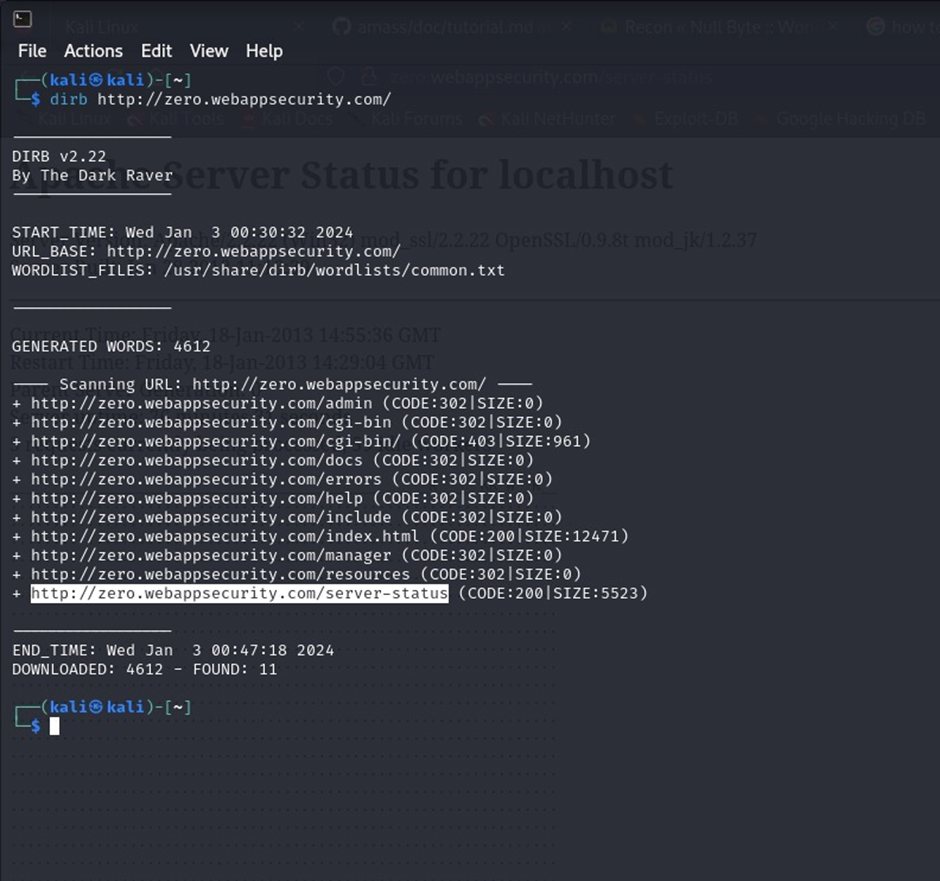
Explanation: Directory brute forcing is a technique where attackers attempt to discover hidden or unauthorized directories and files on a web server by systematically trying various directory names. They often use automated tools and dictionaries containing common directory names to automate the process.

Attackers can exploit this vulnerability to:

Unintentional directory creation: Developers or administrators might accidentally create directories that are not meant to be publicly accessible.

Misconfigured web server settings: Improper server configuration could expose hidden directories by listing them in directory indexes or error messages.

Weak password protection: If directories are protected by weak passwords, attackers can brute-force them relatively easily.



Impact:

Sensitive data exposure: Attackers might access sensitive information like configuration files, user data, or internal documents.

Website defacement: They could exploit hidden scripts or files to deface the website and damage its reputation.

Malware deployment: In some cases, attackers might upload malicious scripts or files to compromised directories, potentially leading to further infections.

Resource exhaustion: Brute-forcing attempts can consume server resources, causing performance degradation or denial-of-service attacks.

Mitigation:

* Remove unnecessary directories: Eliminate any directories that are not intended for public access.
* Disable directory listing: Ensure directory listing is disabled on the web server to prevent attackers from seeing a list of available directories.
* Secure directory permissions: Set appropriate permissions on all directories, restricting access to authorized users or processes.
* Use strong passwords: Implement strong passwords or authentication mechanisms to protect any directories with restricted access.
* Monitor access logs: Regularly monitor web server access logs for suspicious activity, like unusual requests to non-existent directories.
* Implement rate limiting: Enforce rate limits on login attempts and directory access to prevent brute-forcing attacks.
* Utilize web application firewalls (WAFs): Consider using WAFs to detect and block suspicious directory access attempts.
* Keep software updated: Maintain updated software and libraries to address known vulnerabilities that attackers might exploit.

11. Missing Cookie Security Flags:

Explanation: The system doesn't use HTTP Only and Secure flags on cookies, making them vulnerable to client-side scripting attacks and insecure connections.

Impact: Attackers can steal cookies through scripts or intercept them on unencrypted connections, leading to unauthorized access and data breaches.

Mitigation:

* Set the HTTP Only flag to prevent client-side scripting access.
* Set the Secure flag to transmit cookies only over HTTPS connections.
* Consider using additional security flags like Same Site to restrict cookie accessibility further.

1. Information Disclosure: Apache Version Exposure

Explanation: In your scenario, typing ' in the URL bar triggered the website to reveal its Apache server version. This unintentionally exposes sensitive information about the server's internal configuration.

A screenshot of a computer

Description automatically generated

Impact:

Targeted attacks: Attackers can use the specific Apache version to identify known vulnerabilities and launch targeted attacks.

Exploit development: In rare cases, the version information might be used to develop custom exploits for the exposed server.

Social engineering: Attackers could leverage the information in social engineering attempts to appear legitimate or gain user trust.

Mitigation:

Disable server version disclosure: Most web servers allow disabling version information in headers and error messages.

Secure server configuration: Implement strict access control and permissions to restrict sensitive information exposure.

Regular updates: Maintain updated software and libraries to address known vulnerabilities and patching techniques.

# CONCLUSION

We conclude that the project we undertook i.e. Security audit of website  [http://zero.webappsecurity.com/,](https://ktu.edu.in/) using OWASP Methodology for identifying vulnerabilities and potential security threats in the website has followed a thorough and systematic process for identifying vulnerabilities and carrying out required analysis.

OWASP Top 10, is a framework for list of prioritized top 10 website vulnerabilities, that helps for assessing security risks and is used to baseline required website vulnerability testing. The approach helps to improve security of the application and furthermore reduces risk by providing mitigation to reduce risk due to potential cyber-attacks and data theft.

It is an underline fact that website application testing is a repetitive methodology since it’s not possible to have full proof secure website due to cyber criminals being smart and coming up with new and complex ways to exploit and thereby impact smooth functioning of the application, which reside within the website. Therefore, yearly or half yearly website security audit should ideally ensure security of website from emerging new cyber threats and associated cyber-attacks.

# REFERENCES

1. <https://owasp.org/www-project-top-ten/>
2. <https://cheatsheetseries.owasp.org/IndexTopTen.html>
3. <https://www.kali.org/tools/nmap/>
4. <https://www.kali.org/tools/nikto/>