WEB SECURITY VULNERABILITY ANALYSIS IN SELECTED ETHIOPIAN GOVERNMENTAL OFFICES (USING WHITE BOX AND BLACK BOX TESTING)

A Thesis Presented

by

MERIKAT MEHARU BOKE

to

The Faculty of Informatics

of

St. Mary's University

In Partial Fulfillment of the Requirements for the Degree of Master of Science

in

Computer Science

June 2022
Addis Ababa

WEB SECURITY VULNERABILITY ANALYSIS IN SELECTED ETHIOPIAN GOVERNMENTAL OFFICES (USING WHITE BOX AND BLACK BOX TESTING)

By

MERIKAT MEHARU BOKE

Accepted by the Faculty of Informatics, St. Mary's University, in partial fulfillment of the requirements for the degree of Master of Science in Computer Science

Thesis Examination Committee:

_____

Internal Examiner

_____

External Examiner

_____

Dean, Faculty of Informatics

June 2022

DECLARATION

I, am undersigned, declare that this thesis work entitled **Web Security Vulnerability Analysis in Selected Ethiopian Government Offices (using white box and black box)** is my original work, has not been presented for a degree in this or any other universities, and all sources of materials used for the thesis work have been duly acknowledged.

Declared by

MERIKAT MEHARU

_____

Signature

Addis Ababa

Ethiopia

This thesis has been submitted for examination with my approval as advisor.

Dr. Asrat M.

_____

Signature

Addis Ababa

Ethiopia

June 2022

Acknowledgments

I owe my special gratitude to my advisor Dr. Asrat Mulatu! I would like to thank you very much for your support and understanding. I would like to extend my deepest gratitude to information network security Administration (INSA), for its cooperation in providing to permit in which white box and black box testing of governmental website.

I would also like to extend my sincere gratitude to Mr. Dessalegn W/Giorgis, cyber-Audit and Evaluation division head of INSA, for showing a direction, and I am thankful to Mr. Tilahun Ejigu, cyber security compliance and audit team leader for his valuable comments and assistance through the process of researching and writing this thesis.

Finally, I must express my gratitude to my families, coworkers specially to Yosef Dessalegn, and friends for providing me with support and encouragement thorough out my years of study and writing this thesis. With a special mention to my sister Dinknesh. M, Thank you so much for all your support and encouragement!

Table of Contents

List of Abbreviations and Acronyms

INSA	Information Network Security Administration
ZAP	Zed Attack Proxy
OWASP	Open Web Application Security Project
VA	Vulnerability Analysis
OS	Operating System
DNS	Domain Name Service
CVE	Common Vulnerability Exposures
SQLi	Sequential Query Language Injection
XSS	Cross-Site Scripting
LFI	Local File Inclusion
RFI	Remote File Inclusion
IDS	Intrusion Detection System
VAPT	Vulnerability Assessment and Penetration Testing
ISMS	Information Security Management System
BeEF	Browser Exploitation Framework
Nmap	Network Mapper
CIO	Chief Information officer
VPN	Virtual private network

List of Figures

List of Tables

Abstract

Cyber security is the action of ensuring data and data systems with suitable procedural and innovative security measures. Cyber security threats are expanding from time to time. Web security Vulnerability is an imperfection or shortcoming in a computer system, its security strategies, internal controls, or plan and execution, which may misuse to abuse the framework security policy. Web security vulnerability can influence country and can disrupt the social, financial and political realm of governments. Vulnerability analysis is a series of exercises attempted to recognize the shortcomings and gaps to exploit security vulnerabilities.

The reason of this study is to find vulnerabilities and give suggestions and rules to vulnerable systems found in web applications. We have utilized subjective approach to evaluate affect and likelihood unequivocally. The result for each appraisal has been relegate high, medium, or low vulnerability to classify the reason of this ponder is to find vulnerabilities and give recommendations and rules to vulnerable systems found in web applications. We have utilized subjective approach to survey affect and probability unequivocally. The result for each appraisal has been assign high, medium, or low vulnerability to classify them effortlessly. Test arrangement, data gathering, vulnerability analysis, and vulnerability report phases are too included. The finding of this work shows that all the possible number of vulnerabilities rate and system shortcoming or point of view attack of governmental office's web vulnerability analysis finding result by utilizing white box and black box testing. Finally, conclusions and recommendations are made based on the discoveries and analysis. The result of the research appears all the possible number of Vulnerabilities rate of governmental office web and network resource vulnerability analysis finding results of both approaches based on vulnerability impact rate or risk level by utilizing black box and white box testing.

Keywords: *Security, Web security, Vulnerability Analysis, Security Testing, Penetration Testing*

CHAPTER ONE

1 INTRODUCTION

Web security has gotten to be a developing field of concern for Ethiopian governmental offices, non-governmental offices and organizations. Web security is basic to commerce progression and to ensuring information, clients and companies from risk. Data technology Security can ensure a network by testing the network for potential threats, and persistent defense against malicious attacks [1]. Web applications are dynamic websites, which are composition of server-based programs serving client interaction and different other functionalities. Web Server security is hence a vital perspective for any organization having web server network with the web and to confirm clients utilizing their websites, for a secure online entrance [2]. In today's world, individuals store tremendous amount of information on computers and other internet-connected gadgets. The significance of cyber security comes down to the desire to keep data, information, and gadgets private and secure. They seem share sensitive information, utilize passwords to steal funds, or even alter data so that it benefits them in some way. A penetration test mimics an attacker's behavior (commonly known as hacker) but in a controlled environment to recognize and relieve possible vulnerabilities. An incredible number of organizations provide frameworks and services to evaluate security such as pen testing, risk assessment, threat modeling and even instruct ethical hacking [3] [1]. In Overview of Vulnerability Assessment and Penetration Testing Technique counting the main steps of vulnerability examination those are. Discovery: The penetrator performs data disclosure by means of a wide extend of procedures, Enumeration: the particular networks and systems that recognized through discovery, Vulnerability Identification: The vulnerability identification step is a very imperative phase in penetration testing. This permits the client to know the shortcomings of target system and where to launch the attacks. In addition, Exploitation and launching of attacks: After the vulnerabilities found on the target framework [1]. Cyber security required at company to keep their information, finances, and intellectual property secure. People require cyber security for similar reasons, in spite of the fact that intellectual property is less of a factor, and there's a higher risk of losing vital files, such as family photographs [3]. Within the case of administrative organizations or public administrations, cyber security makes a difference guarantee that the community can proceed to depend on their administrations. For example, in the event that a

cyber-attack focused on a power plant, it might cause a citywide blackout. In the event that it targeted a bank, it may take from hundreds of thousands of individuals.

1.1 BACKGROUND

Web security implies ensuring a web application or site by avoiding, detecting and reacting to cyber threats. Site vulnerability could be a misconfiguration or weakness in a web application code or site that allows an attacker to pick up some level of control of the site, and possibly the hosting server. Web attacks are a form of vindictive act performed by the hacker to extend unapproved data [4]. Most of vulnerabilities exploited through automated implies, such as botnets and vulnerability scanners. A few common sorts of site vulnerabilities, which regularly exploited by hackers, are:

SQLi: SQL injection vulnerabilities refer to areas in website code where direct user input passed to a database. Bad actors use these forms to inject malicious code, sometimes called payloads, into a website's database.

XSS: Cross-Site Scripting occurs when attackers inject scripts through un-sanitized user input or other fields on a website to execute code on the site. Cross-site scripting used to target website visitors, rather than the website or server itself.

Command Injection: Command injection vulnerabilities allow attackers to remotely pass and execute code on the website's hosting server

File Inclusion (LFI/RFI): Remote file inclusion (RFI) attacks use the include functions in server-side web application languages like PHP to execute code from a remotely stored file.

Local File Inclusion (LFI), like remote file inclusion, can occur when user input is able to modify the full or absolute path to included files. Nowadays network threats are forever changing. Hackers with malicious intent are continually attempting to infiltrate networks to steal information cyber security now in the world dynamic change. Once an attack happens, it could affect millions of people. State-run organizations can be down; Services cannot provide properly to citizens, so in this study I focused on web vulnerability test on selected organizations by using black box and white box testing.

1.2 STATEMENTS OF THE PROBLEM

Web security is vital to keeping cyber-thieves and attackers from getting to sensitive data. Without a proactive security methodology, businesses risk the spread and escalation of malware, attacks on other websites, systems, and other IT infrastructures. These days cyber space getting to be wide by means of the world and connected device. In any case, Ethiopia too weak cyber innovation and implementation compare with other nation according to Information Network Security Administration auditing report and new challenges emerge nearby growth, and increasing technological exposure. The EU Cyber Security Procedure gives an arrangement system for EU initiatives. In any case, in Ethiopia governments does not exist nether policy nor auditing service. Cyber space is wide within the world and connected with devices so that this thesis discovers vulnerabilities before attack Ethiopia governmental offices web and network infrastructure, and the problem describes about the degree of risk and attack vulnerability of governmental offices. Articulations of the issue were understood; almost Security examination and address vulnerability impact-rate or risk level. Identification of higher-risk vulnerabilities resulting from lower-risk vulnerabilities exploited in a specific way.

1.3 RESEARCH QUESTIONS

How does Governmental offices of Ethiopia perform web security vulnerability analysis?
What gaps or vulnerability exist in Ethiopian governmental offices and how is the risk level?
What is the impact of that vulnerability for Ethiopian governmental offices?

1.4 OBJECTIVES

1.4.1 General objective

The general objective of this thesis is audit and evaluate the web systems, and processes by utilizing different vulnerability scanner tools. In addition, approach to help to identify potential crevices of security on selected Ethiopian governmental organizations web security providing comprehensive view of their IT infrastructure, expediting the assessment process and

recommend ensuring basic data, identifying security loopholes, creating new security policies, and taking after the adequacy of security techniques.

1.4.2 Specific objectives

- Measure Ethiopian governmental offices web security by identifying vulnerability of web system in governmental website.
- Understanding deferent network security scanning tools.
- Understand Data gathering (reconnaissance) methodology in cyber security.
- Evaluate to the information technology use its web security.
- Determination of web vulnerability risk level: High-Risk, Medium-Risk, or Low-Risk.

1.5 SCOPE AND LIMITATION

1.5.1 Scope

- Focus on federal governmental offices network infrastructure assets use black box and white box tests.
- Identify the gaps in the existing defense and recommended cyber security mitigation.
- External Vulnerability and Internal Vulnerability in a network.

1.5.2 Limitation

If it covers the entire governmental network infrastructure in Ethiopia, the result of the research would be more comprehensive. However, due to data limitation and time constraints the student researcher has forced to focus on main of federal governmental web and network infrastructure asset.

1.6 CONTRIBUTION OF THE RESEARCH

Vulnerability analysis is a method used to discover known vulnerabilities of computing frameworks available on a network. It makes a difference to distinguish particular weak spots in application software or the operating system (OS), which might be used to crash the system or compromise it for undesired purposes. Classifies system hole in computers, systems and

communications equipment and predicts, and recommend the adequacy of countermeasures. The foremost Vulnerability analysis significances are.

- Identifying vulnerabilities and misconfigurations.
- Testing security controls and Identifying lack of security.
- Improves security policies and procedures develop cost-effective methods for implementing information security policies and procedures.
- Discover vulnerability, impact and recommend mitigation.
- Avoid vulnerability issues that the systems and provide an appropriate level of security.
- Show vulnerability levels of Ethiopian governmental web and network system.

The Expected result of this work is to identify the vulnerability of the internet and network infrastructure and verify the implementation and performance of security systems. The audit decides whether the security frameworks protect resources and maintain the confidentiality, integrity and availability of information.

1.7 ORGANIZATION OF THE REST OF THE THESIS

In this research has been found a number of vulnerabilities within the organization's network resource that could potentially lead to the investigation of sensitive data and financial losses and influence the organization's commerce reputation, and organization with a report containing the list of vulnerabilities, mentioning their risk level (low, medium or high) and defining impacts and counter measures suggestion to minimize risks. Web vulnerability analysis is usually followed by penetration testing. There's no use in conducting penetration testing before the discovered vulnerabilities are patched, as the goal of penetration testing is not just trying to get into the network but also examining the network environment 'with a new set of eyes' after the improvements are made. Vulnerabilities identified through vulnerability analysis however in this research has been applied different Testing technique include the following:

- Use Penetration testing techniques
- Information Security test and evaluation (ST&E) procedures
- Use automated vulnerability-scanning tools.
- Analysis and report the result.

CHAPTER TWO
2 LITRATURE REVIEW AND RELATED WORKS

2.1 LITRATURE REVIEW

Is the work done by others relates to what the research has been demonstrated with the current work. This survey is also where the literature related to methods that the research has been used in this current work should be introduced.

2.1.1 Vulnerability

Vulnerability is a weakness in the application, which can be an implementation bug or a design flaw that allows an attacker to cause harm to the user of the application and get extra privilege. Vulnerability is the potential risk for the system. Attacker uses this vulnerability to exploit the system and get unauthorized access and information [5]. Example: **Cross Site Scripting / XSS** is a vulnerability that can cause an attacker to send malicious code to other users. XSS interpreted as a weakness that occurs because the web server cannot validate the input data provided by the user. **SQL Injection** is a technique that misuses a security hole that occurs in the database layer of an application. This gap can occur when a programmer who creates code or script does not filter correctly from special characters used in the input data. DOS attacks (Denial-Of-Service attacks) is a type of attack on a computer or server in the Internet network by spending resources (resources) owned by the computer until the computer cannot perform its function properly so that indirectly prevent the user to gain access to services from a computer that is attacked by DOS. The main target of a denial of service is to damage the services provided so that it becomes unavailable [6].

2.1.2 Vulnerability Analysis

A vulnerability analysis is a systematic review of security weaknesses in an information system. It evaluates if the system is susceptible to any known vulnerabilities, assigns severity levels to those vulnerabilities, and recommends remediation or mitigation, if and whenever needed. The objectives of CSVA are to identify credible cyber threats to the facility, identify existing vulnerabilities, and provide risk estimates to facilitate decisions on corrective actions [7].

The purpose of vulnerability assessments is to prevent the possibility of unauthorized access to systems. Vulnerability testing preserves the confidentiality, integrity, and availability of the

system. The system refers to any computers, networks, network devices, software, web application, cloud computing, etc. There is also threats that can be prevented by vulnerability assessment include: SQL injection, XSS and other code injection attacks; Escalation of privileges due to faulty authentication mechanisms; insecure defaults – software that ships with insecure settings, such as a guessable admin passwords.

2.1.3 Types of vulnerability analysis

Host assessment: The assessment of critical servers, which may be vulnerable to attacks if not adequately tested or not generated from a tested machine image.

Network and wireless assessment: The assessment of policies and practices to prevent unauthorized access to private or public networks and network-accessible resources.

Database assessment: The database assessment or big data systems for vulnerabilities and misconfigurations, identifying rogue databases or insecure dev/test environments, and classifying sensitive data across an organization's infrastructure.

Application scans: The identifying of security vulnerabilities in web applications and their source code by automated scans on the front-end or static/dynamic analysis of source code. Vulnerability analysis is suitable to perform the test in these situations: New installed software, applied system upgrades, user policy modification, applied Security patches, and new infrastructure added [8].

2.1.4 Vulnerability Analysis Technique

The vulnerability analysis technique provides the evaluator with explicit guidance on addressing to the research problem statement vulnerabilities analysis of Ethiopian Governmental and non-governmental network asset [9]. In this work, popular VAPT techniques described Static analysis: In this technique, we do not execute any test case or exploit. We analyze the code structure and contents of the system. With this technique, we can find out about all type of vulnerabilities. In this technique, we do not exploit system. One of the big disadvantages of this technique is that it is quite slow.

Manual Testing: In this technique, we do not require any tool or any software to find out vulnerabilities. In this testing, tester uses his own knowledge and experience to find out the vulnerabilities in the system. This testing can be performing with prepared test plan (Systematic manual testing) or without any test plan (Exploratory manual testing). This technique costs cheaper compare to other techniques, because we do not need to buy any vulnerability assessment tool for

this technique. Automated Testing: In automated testing procedure, we utilize computerized vulnerability testing tools to discover out vulnerabilities within the system. These devices execute all the test cases to discover out vulnerabilities. Since of tool, repeated testing can also perform very easily. Automated testing gives superior accuracy than what other techniques give. It takes very less time and same test cases used for future operations. In any case, tools increment cost of testing. Fuzz testing: This is also known as fuzzing. In this, we input invalid or any Arbitrary Data into system and after that look for crashes and failure. This can be like robustness testing. This technique can apply with very less human interaction [5].

2.1.5 Vulnerability analysis tools

Vulnerability assessment tools designed to scan automatically new and existing threats that can target your application. It allows for the detection of vulnerabilities in applications using many ways. The Code analysis vulnerability tools analyze coding bugs. Audit vulnerability tools can find well-known rootkits, backdoor, and Trojans. Types of tools, which include Web application scanners: that test for and simulate known attack patterns. Include; ZAP, Skipfish, Grabber. Protocol scanners: that search for vulnerable protocols, ports and network services. Network scanners: that helps visualize networks and discover warning signals like stray IP addresses, spoofed packets and suspicious packet generation from a single IP address. Moreover, scanning tools like; Nikto**:** This Web server scanner tests Web servers for dangerous files, outdated server software and other problems. It performs generic and server type specific checks. It also captures and prints any cookies received. The Niko code itself is Open Source (GPL), however the data files it uses to drive the program are not.

NMAP: This tool used to find hosts and services on a computer network, in this way building a "map" of the network. To achieve its objective, Nmap sends specially crafted bundles to the target host(s) and after that analyzes the responses [10]

2.1.6 Vulnerability Analysis Metrics

Metric refers to assigning a value to an object whereas measurement is the process of estimating attributes of an object. According [11], Security metrics provide a qualitative and quantitative representation of a system or network's security level. However, using existing security metrics can lead to misleading results. This work proposed three metrics, which is the Number of Vulnerabilities (NV), Mean Vulnerabilities on Path (MVoP), and the Weakest Path (WP). The

experiment of this work used two networks to test the metrics. The results show the impact of these metrics on finding the weaknesses of the network that the attacker may use [11]. Here are some security metrics, which used to measure the performance of your Vulnerability Management (VM) program.

Mean Time to Detect: Measures how long it takes before known vulnerabilities get detected, across the organization. It refers to the time it takes from when a problem first emerges to the moment when it is detected by the right people or systems.

Mean Time to Resolve: The mean time interval taken to remediate / patch vulnerabilities after identification by the Vulnerability Assessment (VA) tool. It is the average time. This includes not only the time spent detecting the failure, diagnosing the problem, and repairing the issue, but also the time spent ensuring that the failure won't happen again. MTTR=Total maintenance time/Number of repairs

Average Window of Exposure: The time when vulnerability first publicly known to the time the impacted systems get patched.

Number of Open Critical / High Vulnerabilities: Based on Risk based Prioritization of vulnerability, considering a number of factors

Vulnerability Reopen Rate: This measures the effectiveness of the remediation process. A high rate means that the patching process is flawed.

This study used Metrics or risk calculation, which done based on **Common *Vulnerabilities and Exposures (CVE***) system. CVE stands for Common Vulnerabilities and Exposures. CVE is a glossary that classifies vulnerabilities. The glossary analyzes vulnerabilities and then uses the Common Vulnerability Scoring System (CVSS) to evaluate the threat level of a vulnerability. A CVE score often used for prioritizing the security of vulnerabilities. The following formula is used to calculate the risks.

Risk= Likelihood*impact

This means that the total amount of risk exposure is the probability of an unfortunate event occurring, multiplied by the potential impact or damage incurred by the event. If you put a value on the impact, then you can value the risk and in a simple way compare one risk factor to another.

2.2 RELATED WORKS

This study uses internal testing that performed inside an organization's network, seeking out for vulnerabilities from the inside, and external testing that refers to attacks on the organization's network resource perimeter utilizing strategies performed from outside the organization's network infrastructures and web application. Within the ponder, there has been a lot of research checked on related with Web Security Vulnerability Analysis in Ethiopia. In Network Security Vulnerability Analysis of Ethiopian Government Offices by Tilahun Ejigu [1] shows all the possible number of Vulnerabilities rate and system weakness perspective attack of governmental office network asset vulnerability analysis finding results of both approaches based on vulnerability impact rate or risk level and system technology weakness or attack perspective by using black box testing. The objective of the work is to find weak links (vulnerabilities) and give suggestions and guidelines to vulnerable entities found in its web application. A simple matrix created to assess overall exposure. The methodology of vulnerability analysis includes three phases: test preparation, conducting test and test result analysis. In any case, there's as it were white box testing method used; which will influence the quality of the result [1]. Developing black box web application penetration testing methodology using comparative criteria: By Gebrekidan Gebremedhin Mebrahtu [12]; The objective of this study is to develop a black box web application penetration testing methodology using comparative criteria to enable black web application penetration tester to conduct penetration testing on web application. The set of criteria for selecting and testing black box web application security methodologies was developed, and the methodologies was compared based on a set of criteria. The testing technique tested on a sample of four Ethiopian universities. It focuses to illuminate the challenges faced by penetration testers particularly black box web application penetration tester in selecting the correct methodology and creating a black box web application penetration testing strategy with by minimizing the restrictions of the most broadly used security testing techniques. Although the gap of this study is, White Box Web Application Security Testing Methodology and Social Engineering are not covered. In the Approach of Auditing Network Security by Anantha Sayana [9]; discovers the essential vulnerabilities related with a network can be depict in area Availability that control to guarantee availability and reliability of a network infrastructure. The center of the article is to sketch a fundamental approach to network security audit, and not to provide specific audit and

technical guidelines. In any case, it focuses on audit of network only; it does not focus on others like; application software, operating systems and databases, physical and environmental security [9]. In "Security practices and challenges at selected critical infrastructures in Ethiopia" by Tewodros Getaneh [13] examined the practices and challenges of cyber security at three selected critical infrastructures in Ethiopia. These critical infrastructures are Ethiopian Electric Power, Ethiopian Electric Utility, and Ethio-Telecom. It uses both qualitative and quantitative research approaches. Finally, it proposes a tailored cyber security framework based on INSA's Critical Mass Cyber Security Requirement Standard Version 1.0 and NIST's Framework for improving critical infrastructures cyber security version 1.1. In spite of the fact that, by using NIST framework or other cyber security systems for the challenges and level of readiness for cyber security threats and exposure is not examined [13]. Propose Vulnerability Metrics to Measure Network Secure using Attack Graph by Zaid. J. Al-Araji [11]; This work proposed three metrics, which is the Number of Vulnerabilities (NV), Cruel Vulnerabilities on Way (MVoP), and the Weakest Way (WP). The experiment of this work used two networks to test the metrics. The results show the impact of these metrics on finding the weaknesses of the organize that the attacker may use [11]. Overview of Security Metrics by Rana Khudhair Abbas Ahmed [14]; This paper gives an overview of the security metrics and its definition, needs, properties, advantages, measures, types, issues/aspects and also classifies the security metrics and clarifies its relationship with risk management. In addition, it says Effective measurement and reporting are required in order to illustrate compliance, progress viability and efficiency of controls, and ensure key alignment in an objective, reliable, and efficient way. In addition, recommends that metrics must be planned using a participatory design process including the affected security experts of the organization. [14].Vulnerability Assessment and Penetration Testing by Gaurav Bhatia [15]; the main purpose of this paper is to educate the people with respect to vulnerabilities and cyber threats. Also, describes about the technical approach for manual web-app penetration testing for maintaining the security of the net applications. Also, look for OWASP top 10 vulnerabilities in detail and its exploitation. It also contains a few courses that anybody can do for learning Penetration Testing and Vulnerability Evaluation [15]. Cyber security analysis using vulnerability analysis and penetration testing by Prashant S. Shindee [16]; this study focuses on overview and different strategies utilized in vulnerability assessment and penetration testing (VAPT). Also, it shows us VAPT process; like vulnerability analysis (data gathering, scanning, result analysis), Penetration

testing (build attack, exploitation: attack phase, result analysis). At that point, all goes to reporting. It shows briefly what to do in each step. In addition, focuses on making cyber security awareness and its significance at different level of an organization for adoption of required up to date security measures by the organization to remain ensured from different cyber- attacks. By raising advantages and disadvantages of vulnerability analysis and penetration testing, features and benefits of utilizing those techniques. Finally, it suggests existing tools have to be include with mechanisms to identify and evaluate the newly evolved vulnerabilities. This issue can be tended to by making tools so adaptable that modern attack signatures can be included for types of vulnerabilities [16]. A Study on Penetration Testing Process and Tools by Hessa Mohammed Zaher Al Shebli [17]; It talk about the significance of entrance testing, factors and components considered while conducting a penetration test with methods (black box, gray box, and white box), Penetration testing phases:(Test preparation, Test implementation, Test analysis). then show a study of tools (Nmap, BeEF, Metasploit, Nessus, and Cain and Abel), and procedures followed, role of penetration test while implementing in the IT governance in an organization and finally discussed the role of the Information Security Management Framework (ISMF), professional Ethical and technical Competency required for performing the penetration test [17]. Assessment of incident management of information security practice in Ethiopian bank by Tsedale Yohannes [18]; in this study, attempts done to look at and compare the accessible international standards and rules to utilize it in comparing with the current practice. Qualitative in-depth study was utilized to evaluate practice of data security incident management at bank x. In addition, this ponder pointed out that to what extent existing standards and guidelines are adopted in bank x's data security occurrence management process. Challenges in dealing with incidents at bank x were also revealed within the study. These challenges related to employee's awareness, need of skilled incident handlers, communication and enhancement of new threats. Finally, recommend 10 points bank x and other organization may use it for distant better; a much better; a higher; a stronger; an improved">a far better way of managing data security incidents [18]. Security testing of Ethiopian E-governmental websites using penetration-testing tools: by Habtamu Girma Abebe [19]. In this work, 11 Ethiopian websites tested by using three penetration-testing tools, which are Acunetix, Vega and NetSparker VAPT. Based on the result, almost all Ethiopian websites are vulnerable for distinctive vulnerabilities. Most websites are vulnerable for SQL Injection and XSS. The analyst compares of the security of Ethiopian websites to Turkish governmental websites. The security of

Turkish websites is much better than Ethiopian websites. In addition, the analysts compare scanning tools based on their result, Vega web vulnerable scanner was the finest one. It detects maximum number of high severity vulnerabilities from Ethiopian governmental websites than the other tools. Though the rest two scanners identify vulnerabilities, but Vega's result is best. Finally suggested all of websites should fix their websites as soon as possible. In addition, for the future, websites should have developed by developers who have sufficient knowledge about securing of websites.

Table 2.1: Summary of Review of Related Works

Authors	**Approach**	**Identified Vulnerability**	**Countermeasures**	**Research gap**	**Tool used**
Tilahun Ejigu [1]	Black box testing	Bootstrap vulnerability, CMS, Apache server vulnerability, file upload restriction vulnerability, vulnerability of Brute force attack, network open port, SQL database vulnerability.	Strong IT policy	Used only Black box, that cannot show the final result clearly	Kali Linux
Baybutt, Paul [7].	Asset-based Approach	public injuries, property damage, financial loss, loss of production or critical information, disruption of company operations, loss of reputation	Protective measures	The analysis is not detail	Authentication, firewall, intrusion detection
Anantha Sayana [9]	network architecture	Network architecture and sub domain	External network monitoring	Focuses on audit of network only	Meltigo
Gebrekidan Gebremedihn Mebrahtu [12]	Comparative criteria	HTTP TRACE/TRACK Method, Plain text	Ethiopia might build its own standard for	White Box Web Application and Social	Qualysguard, and Nessus

		authentication, and XSS	measuring the security level of web application	Engineering are not covered	
Tewodros Getaneh [13]	Design science guide lines to tailor the cyber security framework	Attack via Email, mobile computing, social media, new application development, and implementation.	Implementation framework for technical processes of cyber security at critical infrastructures.	The challenges and level of preparedness for cyber security threats is not examined	Interview and questionnaire
Habtamu Girma Abebe [19]	Exploratory type of research approach	SQL Injection and XSS	Knowledge level of developers	The work doesn't match the scope	Acunetix, Vega, and NetSparker VAPT
Nuno Seixas, MarcoVieira, Henrique Madeira, Jose´ Fonseca [20]	Pearson product-moment correlation	MFCE, WPFV, MIFS, and WVAV	Understand the typical software faults.	Does not compare more vulnerabilities of web applications written in different languages	attack simulator, automated program repair IDS
Palak Aar, Aman Kumar Sharma [21]	Scan time/Port scanning Result chart	Maximum time to perform the scan, found the least number of open ports.	Owing to broad coverage, easy to use interface, fairly fast response time and highest number of detected open ports	Characteristics of the tools like accuracy of port scanners in terms of false positive/negative rate, or usability, should be put under investigation.	By using 8 criteria
Adithyan A, Chethana R [22]	Manual web application penetration testing	SQL Injection	Use proxy mechanisms to penetration test a website		Integrated Pentest

Sheetal Bairwa1 [23]	Test run on the server side based on client server architecture.	Vulnerabilities present in the remote host	Patched the hole	More vulnerability scanning tool should be included	Nessus
Muhammad KasimLim [24]	Experimental setup, design and implementation	SQL Injection, XSS, Wordpres, and WPA2 Attacks	Using sqlmap, BeEF, wpscan, and fluxion	Evaluation of other attacks, deep analysis on the log files should be included	Kali Linux

CHAPTER THREE
3 METHODOLOGY AND RESEARCH DESIGN

This research targets to analyze the security of web at four selected Ethiopian governmental organizations. The vulnerability test on the same targets aims to provide testing security how to conduct the methodologies and to evaluate the effectiveness of the research. Since the research statement of the problem was already described detail at the chapter one problem statement section, the remaining methodologies used in this research has been be described below.

The first phase of the research was formulating the research problem the problem was initiated by personal interest to conduct research on the area of Ethical hacking vulnerability analysis. In this section has been also describe any experiments you may have run; it has been also discussing any testing methodologies and how these would be actually applied in research and discusses the strategies and the methodology of conducting vulnerability analysis. The Open Web Application Security Project (OWASP) testing guide, Information Systems Security Assessment Framework (ISAAF) used.

In this, my current work the research has been used the best methodology ISAAF (Information Systems Security Assessment Framework) standard, which aimed to help the administrator to evaluate my application, system and network controls. The methodology solved the question, how it should be vulnerability analysis and what are the step/procedures to be followed. The main approach includes three phases and nine steps.

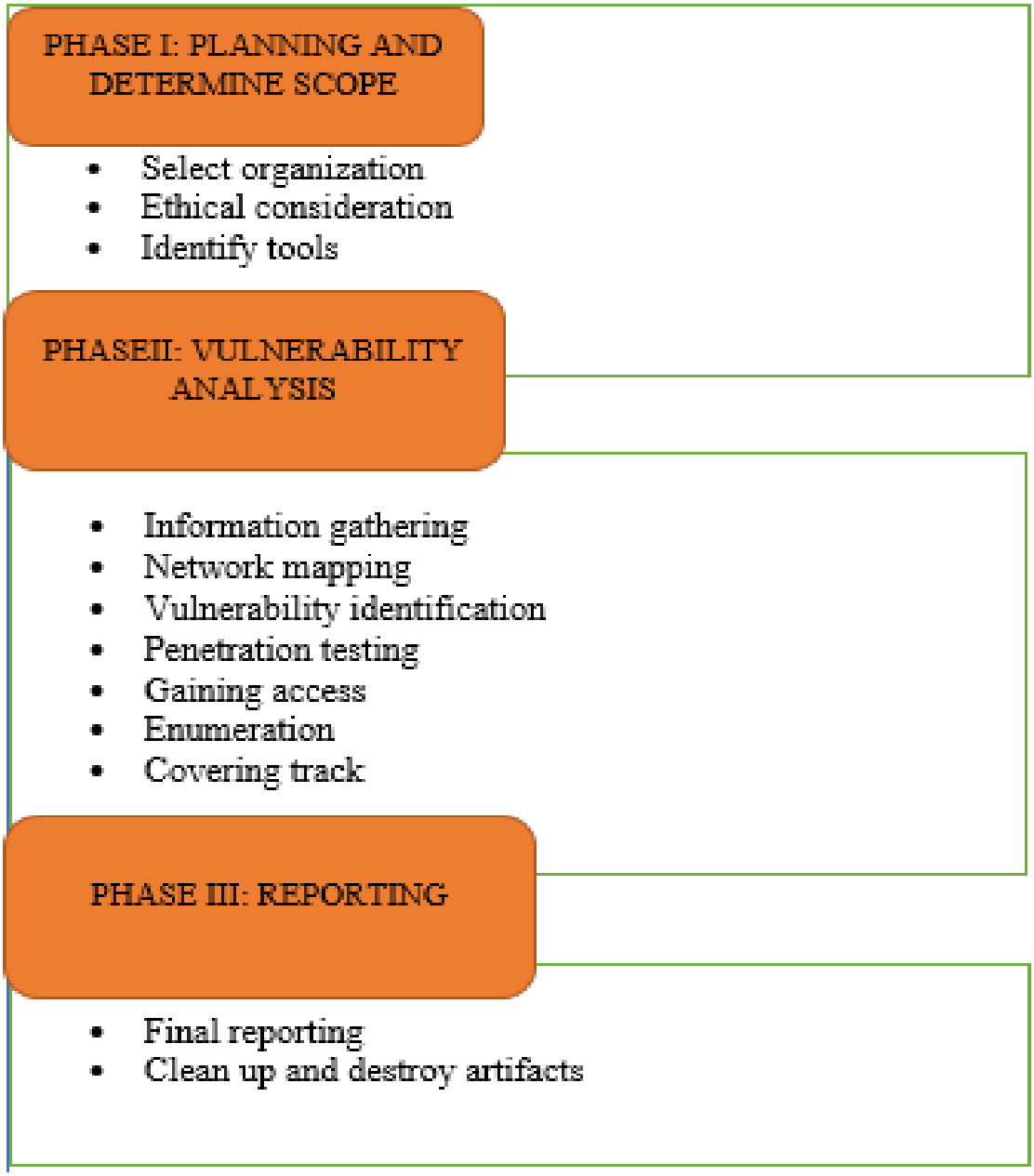


Figure 3.1: The Phases of Penetration Testing (ISSAF) Standard

3.1 General Approach

The approach of this research is qualitative type of research, because the result of the vulnerability analysis should be text based that explained briefly the observational and document analysis that applying different web security analysis tools report.

3.2 Organization Selection

Organization selection is concerned on which the organization should be one of the high assets values of the countries who have websites that can be accessed remotely and which are vulnerable. This study targets specific sector in Ethiopia government offices, so that it is possible to take some Governmental sector and make vulnerability analysis by using vulnerability analysis benchmark. In addition to that identified which sector has, most sensitive data stored. Moreover, the website will be benefit more and organizations that are more vulnerable selected.

3.3 Source of Data and Data Type

The data for this research gathered using secondary data sources. The data source involved through the use from electronic search Site: www.google.com,manually automated and vulnerability scanner tools generating result such as port scanners, ping tools, host vulnerability scanners, and network mappers

3.4 White box testing

Known as clear box testing, glass box testing, transparent box testing, and structural testing is a method of software testing that test internal structures or workings of an application, as opposed to its functionality (i.e., black box testing). In white-box testing an internal perspective of the system, as well as programming skills important to design test cases. The tester chooses inputs to exercise paths through the code and determine the expected outputs. The testing approach that allows testers to inspect and verify the inner workings of a software system, its code, infrastructure, and integrations with external systems.

3.5 Black box testing

Known as a Behavioral Testing/blind test, this is one where the pen-tester is given no background information besides the name of the target company and an external test, the ethical hacker goes up against the company's external-facing technology, such as their website and

external network servers. In some cases, the hacker not have been allowed to enter the company's building. This can mean conducting the attack from a remote location or carrying out the test from a truck. In the testing method in which the functionalities of software applications are tested without having knowledge of internal code structure, implementation details and internal paths. Black Box Testing mainly focuses on input and output of software applications, and it is entirely based on software requirements and specifications.

In this research have been used these types of testing. This testing approach focuses on the input that goes into the application software, and the output that produced. The testing involve does not cover the inside details such as code, server logic, and development method that test is performed from a user's point-of-view and not of the designers.

3.6 Tool selection

In this study, there is both automatically and manual testing method used to test web vulnerability. In the case of automated web vulnerability scanner, which helps, to detect available vulnerabilities from the tested system, tools are used. There are different both commercial and open-source web vulnerable scanners that allows to detect vulnerabilities. Here the research has been used both of them and the commercial scanners have trial version. By taking this trial as an advantage, the research has done to test the selected system automatically. The tools that the research has used for the testing are

- Kali Linux preinstalled penetration-testing programs.
- Online Web application testing tools
- Custom scripts for security testing
- **Acunetix:** Testing everything from Cross-site Scripting and SQL Injection to web server security. Effectively discover and remediate web application vulnerabilities.
- **Nessus:** Using this security scanner tool, utilizes plug-ins, which are separate files, to handle the vulnerability checks. This makes it easy to install plug-ins and to see which plug-ins installed to make sure. Nessus uses a server-client architecture.
- **Zenmap:** Used to map Network, like to find live hosts on a network, perform port scanning, ping sweeps, OS detection, and version detection.

- **OWASP ZAP:** (Zed Attack Proxy) is a free open-source platform-agnostic security-testing tool that scans through your web application to identity any security vulnerabilities as possible.
- **Nikto**: An open-source web server and web application scanner. It can perform comprehensive tests against web servers for multiple security threats, including over 6700 potentially dangerous files/programs. Nikto can also perform checks for outdated web servers' software, and version-specific problems.

CHAPTER FOUR
4 DETAILED ANALYSIS OF FINDINGS

Many security weaknesses discovered during the vulnerability analysis Ethiopian governmental offices web applications. That vulnerability also listed as in hosts vulnerabilities based on Likelihood and Impact. Under this section the security audit finding on the websites are listed in the tables below. All of the vulnerabilities discovered from testing and using the methodologies explained previous chapter. For this research has been selected as a sample 4 governmental office website.

Table 4.1: Governmental office website sample

Number	Name of Governmental Organization	URL Name
1	National bank of Ethiopia	https://www.nbe.com/
2	Fana broadcasting corporate	https://www.fanabc.com/
3	Ethiopian Electronic single window	https://esw.et/esw-trd/
4	Ministry of Health	https://www.moh.gov.et/site/

4.1 Web security

Web security refers to the protective measures and protocols that organizations adopt to protect the organization from, cyber criminals and threats that use the web channel. Web security is critical to business continuity and to protecting data, users and companies from risk.

Web security refers to protecting networks and computer systems from damage to or the theft of software, hardware, or data. It includes protecting computer systems from misdirecting or disrupting the services they designed to provide. As well, it is synonymous with cyber security and covers website security, which involves protecting websites from attacks. It includes cloud security and web application security, which defend cloud services and web-based applications, respectively. Protection of a virtual private network (VPN) also falls under the web security umbrella. Moreover, it is crucial to the smooth operation of any business that uses computers. If a website is hacked or hackers are able to manipulate your systems or

software, your website and even your entire network can be brought down, halting business operations.

Why web security is important? Web security is important to keeping hackers and cyber-thieves from accessing sensitive information. Without a proactive security strategy, businesses risk the spread and escalation of malware, attacks on other websites, networks, and other IT infrastructures.

4.2 Observation

This section shows experiment, that web reconnaissance gathering information about a website, such as the website design, coding, and internal structure of software, network structure, applications and services, and vulnerabilities. All the information about the targets (devices, application, network and services) using both technical (use kali Linux operating system) and nontechnical (observations and documentations) methods carried out. The following table summarizes show both technical and non-technical methods: From the vulnerability analysis testing of network infrastructure and basic network information, the application systems that includes the technology used obtained. The governmental website or target listed in this chapter above in the table 2. In this reconnaissance phase does not include each targets name and URL for purposed of governmental office security privacy based on rule of engagement agree with INSA if necessary, demo and screenshot evidence open for advisor and examiner in which except published, so the research has been seen when you ask any practical evidence any time.

4.3 Vulnerability Analysis finding Table Format

The security audit arranged in table format, which has the following rows:

Table 4.2: Vulnerability Analysis Finding Table Format

Target No.	
Vulnerability Name	
Tool used	
Vulnerability Description	
Risk Level	
Impact	
Evidence	

1. **Targets: -** are client computers, servers, network device, applications, policies etc.… that evaluated. Example: 192.168.1.1, www.mysite.com …
2. **Vulnerability name**: - A name given to a weakness on the application that could expose the organization to a security threat. Example: If complex passwords are not used then the organization will be exposed to guessing user passwords or brute force attack.
3. **Tool used**: - It describe what tool used to scan the web.
4. **Vulnerability Description**: - It is a clarification of how the vulnerability/weakness has occurred in the target.
5. **Risk level**: - a description that indicates the possibility/ probability of happening of losses, this is a result of damage and likelihood.
6. **Impact: -** is a damage that will happened, if a malicious party exploits the vulnerabilities identified. Example: A user can view a list of all files from this directory possibly exposing sensitive information.
7. **Evidence**: - shows the output of scan in terms of screenshot.

4.4 Vulnerability Analysis Findings

Table 4.3: SSL Medium Strength Cipher Suites Supported (SWEET32)

Target No	**1**
Vulnerability Name	**SSL Medium Strength Cipher Suites Supported (SWEET32)**
Tool used	**Nessus**
Vulnerability Description	The remote host supports the use of SSL ciphers that offer medium strength encryption. Nessus regards medium strength as any encryption that uses key lengths at least 64 bits and less than 112 bits, or else that uses the 3DES encryption suite.
Risk Level	**High**
Impact	Note that it is considerably easier to circumvent medium strength encryption if the attacker is on the same physical network.

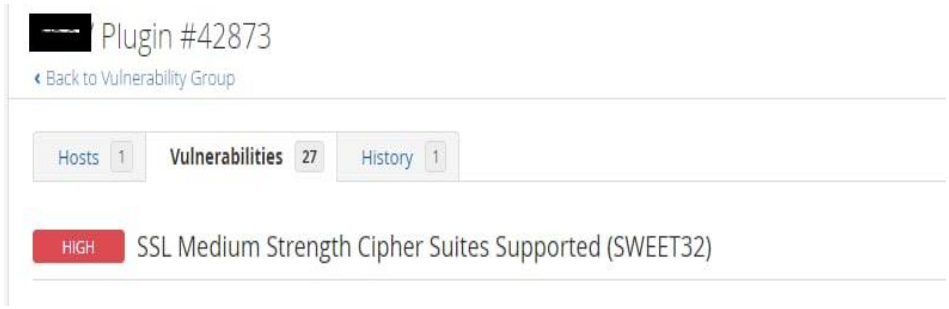
Evidence	

Table 4.4: Anti-CSRF error

Target No	**1**
Vulnerability Name	**Anti-CSRF error**
Tool used	**OWASP ZAP**
Vulnerability Description	No Anti-CSRF tokens found in a HTML submission form. A cross-site request forgery is an attack that involves forcing a victim to send an HTTP request to a target destination without their knowledge or intent in order to perform an action as the victim. The underlying cause is application functionality using predictable URL/form actions in a repeatable way. The nature of the attack is that CSRF exploits the trust that a web site has for a user. By contrast, cross-site scripting (XSS) exploits the trust that a user has for a web site. Like XSS, CSRF attacks are not necessarily cross-site, but they can be. Cross-site request forgery is also known as CSRF, XSRF, one-click attack, session riding, confused deputy, and sea surf.
Risk Level	**High**
Impact	The absence of Anti-CSRF tokens may lead to a Cross-Site Request Forgery attack that can result in executing a specific application action as another logged in user, e.g., steal their account by changing their email and password or silently adding a new admin user account when executed from the administrator account.

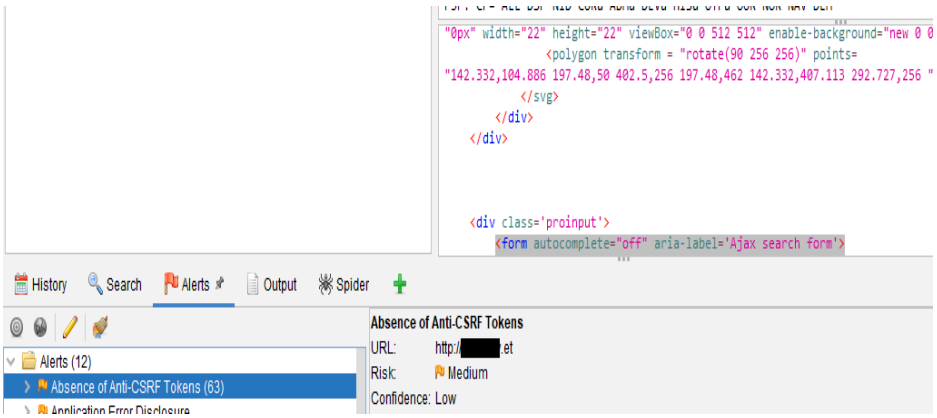
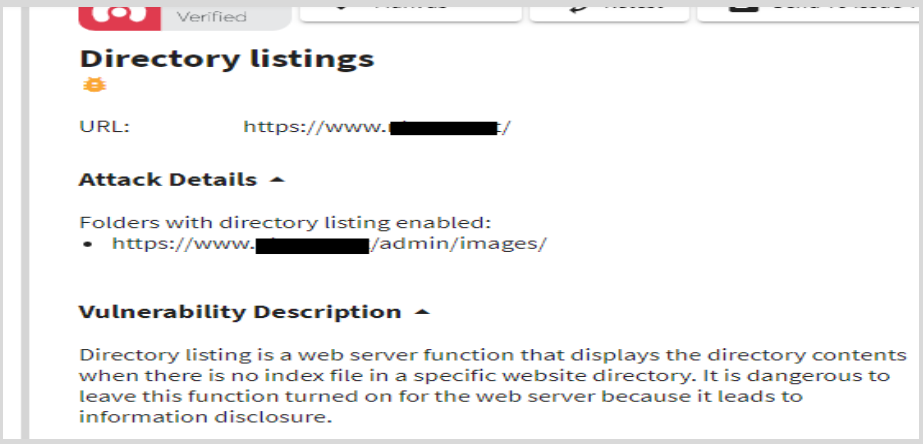
Evidence	

Table 4.5: Configuration management issues

Target No	**1**
Vulnerability Name	**Configuration management issues**
Tool used	**Manual white box (Directory traversal)**
Vulnerability Description	The web application has a php file that should not be on the web server and certainly should not be accessible remotely.
Risk Level	**High**
Impact	The consequences of these kinds of files are very dangerous and could lead to system control.
Evidence	http://www.example.com/admin/filesystem/i.php 

Table 4.6: Directory-listing Attack

Target No	**1**
Vulnerability Name	**Directory listing attack**
Tool used	**Acunetix**
Vulnerability Description	The web application lists the sensitive directories. A directory listing is inappropriately exposed, yielding potentially sensitive information to attackers. Web servers can be configured to automatically list the contents of directories that do not have an index page present. It particularly increases the exposure of sensitive files within the directory that are not intended to be accessible to users, such as temporary files and crash dumps. Any sensitive resources within the web root should in any case be properly access-controlled, and should not be accessible by an unauthorized party who happens to know or guess the URL.
Risk Level	**Medium**
Impact	Provides an attacker with the complete index of all the resources located inside of the directory
Evidence	

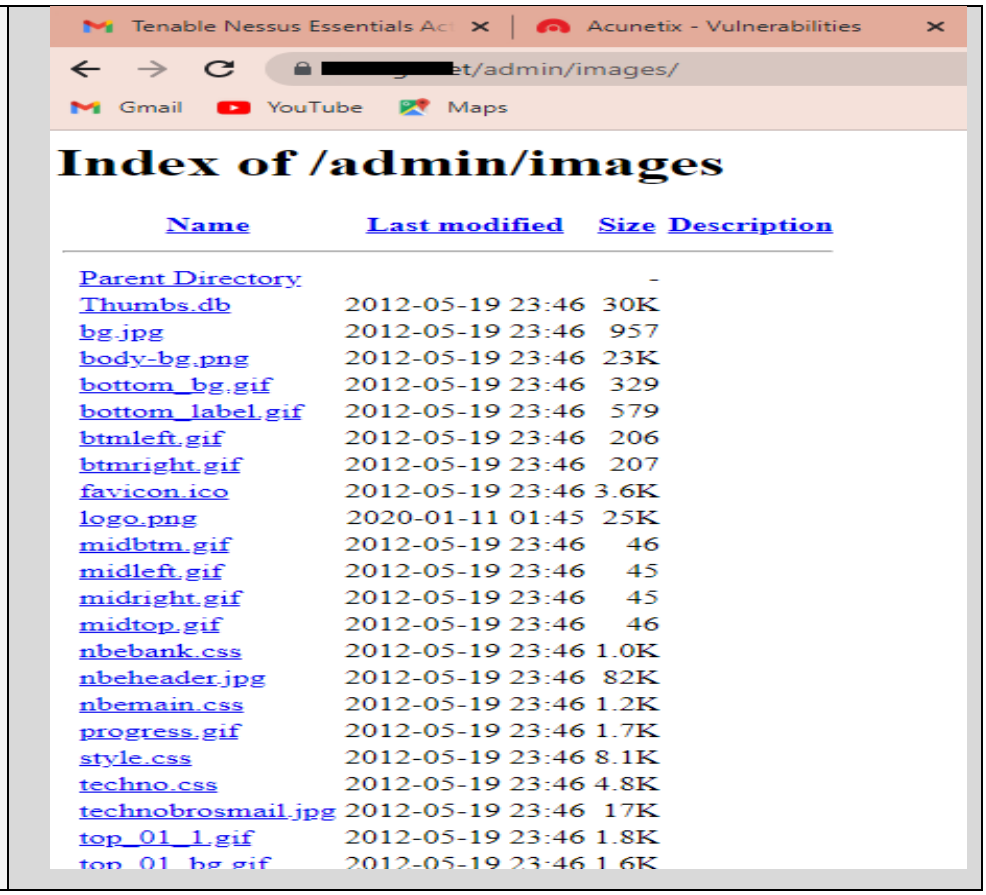


Table 4.7: Insecure http cookies are used

Target No	**1**
Vulnerability Name	**Insecure http cookies are used**
Tool used	**Owasp, Acunetix**
Vulnerability Description	Cookies without Secure flag set One or more cookies do not have the Secure flag set. When a cookie is set with the Secure flag, it instructs the browser that the cookie can only be accessed over secure SSL/TLS channels. This is an important security protection for session cookies. Cookies without HttpOnly flag set One or more cookies do not have the HttpOnly flag set. When a cookie is set with the HttpOnly flag, it instructs the browser that the cookie can only

	be accessed by the server and not by client-side scripts. This is an important security protection for session cookies.
Risk Level	**Medium**
Impact	The secure flag is not set, and then the cookie will be transmitted in clear-text if the user visits any HTTP URLs within the cookie's scope. An attacker may be able to induce this event by feeding a user suitable links, either directly or via another web site. Cookies could be sent over unencrypted channels.
Evidence	

Table 4.8: Content-Type-Options Header Missing

Target No	**1**
Vulnerability Name	**X-Content-Type-Options Header Missing**
Tool Used	**Nikto & owasp zap**
Vulnerability Description	The Anti-MIME-Sniffing header X-Content-Type-Options was not set to 'nosniff'. This allows older versions of Internet Explorer and Chrome to perform MIME sniffing on the response body, potentially causing the response body to be interpreted and displayed as a content type other than the declared content type. Current (early 2014) and legacy versions of Firefox will use the declared content type (if one is set), rather than performing MIME-sniffing
Risk Level	**Medium**

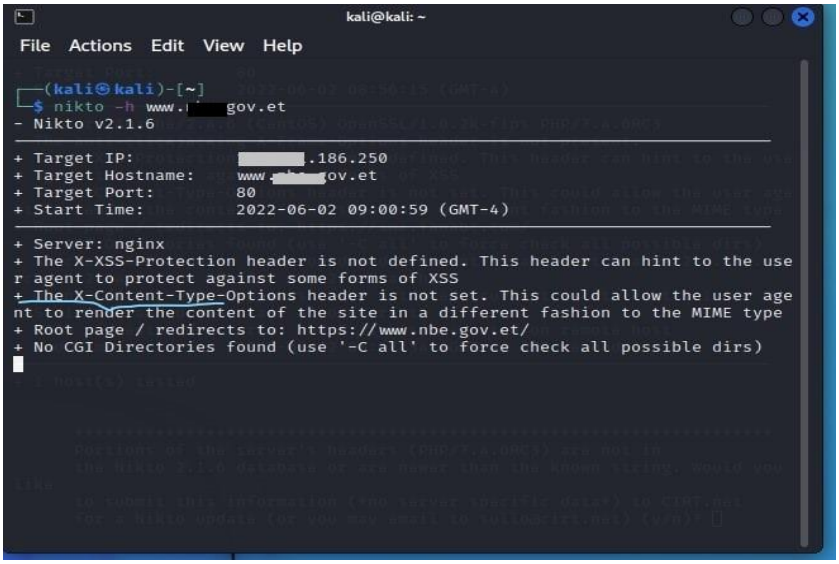
Impact	The HTTP 'X-Content-Type-Options' response header prevents the browser from MIME-sniffing a response away from the declared content-type. The server did not return a correct 'X-Content-Type-Options' header, which means that this website could be at risk of a Cross-Site Scripting (XSS) attack.
Evidence	

Table 4.9: Login page password-guessing attack (Brute-force attack)

Target No	**1**
Vulnerability Name	**Login page password-guessing attack (Brute-force attack)**
Tool used	**Acunetix**
Vulnerability Description	The website login page does not have any protection against password-guessing attacks (brute force attacks). A common threat web developer's face is a password-guessing attack known as a brute force attack. A brute-force attack is an attempt to discover a password by systematically trying every possible combination of letters, numbers, and symbols until you discover the one correct combination that works.
Risk Level	**Low**

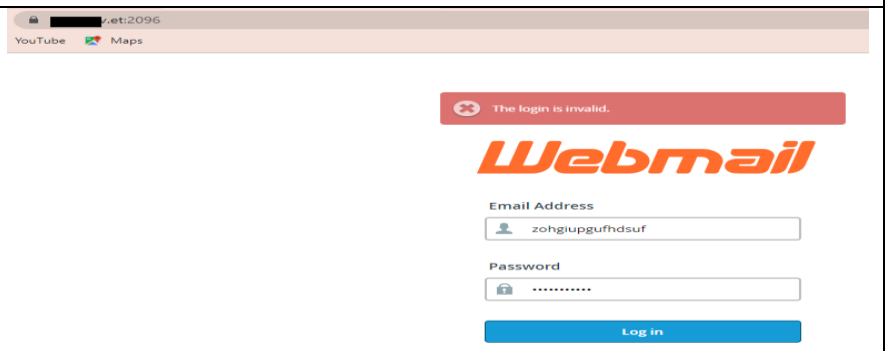
Impact	An attacker may attempt to discover a weak password by systematically trying every possible combination of letters, numbers, and symbols until it discovers the one correct combination that works.
Evidence	

Table 4.10: WordPress username enumeration

Target No	**2**
Vulnerability Name	**WordPress username enumeration**
Tool Used	**Acunetix**
Vulnerability Description	WordPress includes a REST API that can be used to list the information about the registered users on a WordPress installation. The REST API exposed user data for all users who had authored a post of a public post type. WordPress 4.7.1 limits this to only post types, which have specified that they should be shown within the REST API.
Risk Level	**High**
Impact	An unauthenticated attacker can gain access to the list of users on a WordPress installation. This can be exploited by bots, which are launching brute-force password guessing attacks on WordPress websites.

Evidence	  

Table 4.11: Open port 445

Target No	**2**

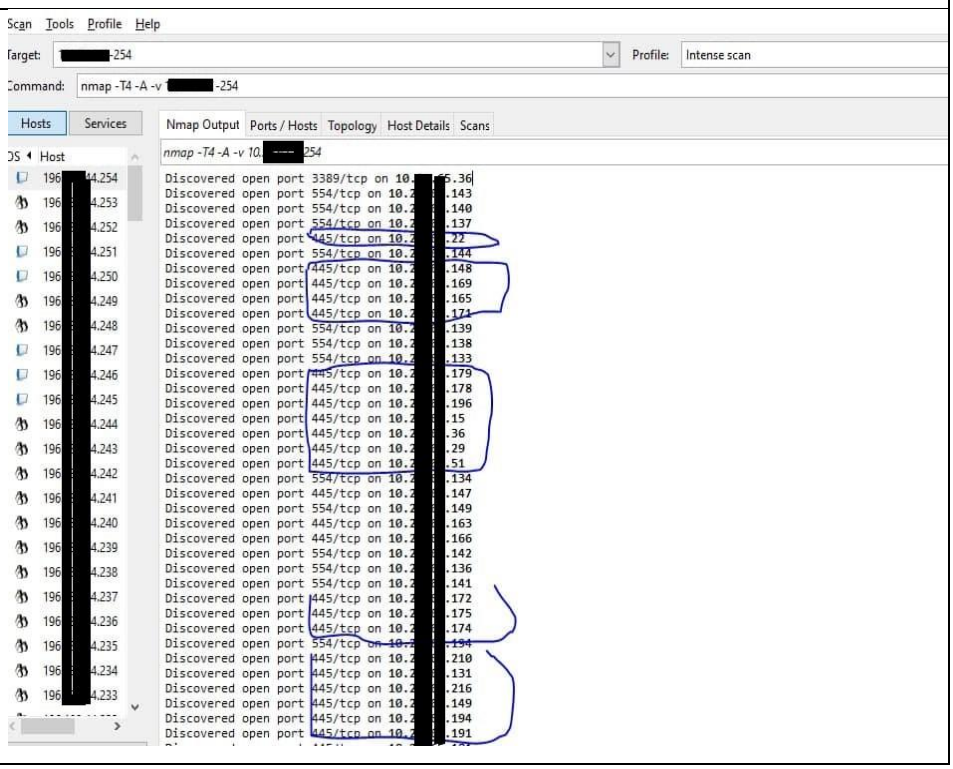
Vulnerability Name	**Open port 445**
Tool Used	**Nmap**
Vulnerability Description	**TCP port 445** is direct **TCP**/IP MS Networking access without the need for a NetBIOS layer. This service is only implemented in the more recent versions Windows starting with Windows
Risk Level	**Medium**
Impact	The attackers can use open ports as an initial attack vector. Furthermore, listening ports on a local network can be used for lateral movement. It is a good practice to close ports or at least limit them to a local network.
Evidence	

Table 4.12: Internal network share resource

Target No	**2**
Vulnerability Name	**Internal network share resource**

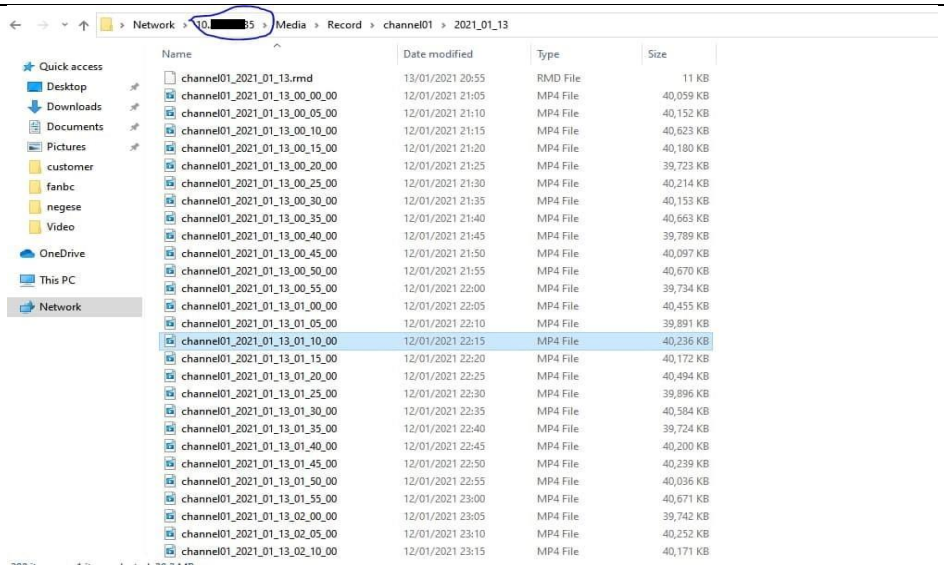
Tool Used	**Manual**
Vulnerability Description	Network resources share; refer to computer data, information, or hardware devices that can be easily accessed through a local area network (LAN) or enterprise intranet.
Risk Level	**Medium**
Impact	Sensitive file/data can be easily accessed through a local area network (LAN) or enterprise intranet for an authorize person.
Evidence	

Table 4.13: Cookies without Secure flag set fail.

Target No	**2**
Vulnerability Name	**Cookies without Secure flag set**
Tool Used	**Acunetix, Manual cookie editor**
Vulnerability Description	One or more cookies does not have the Secure flag set. When a cookie is set with the Secure flag, it instructs the browser that the cookie can only be accessed over secure SSL/TLS channels. This is an important security protection for session cookies
Risk Level	**Medium**

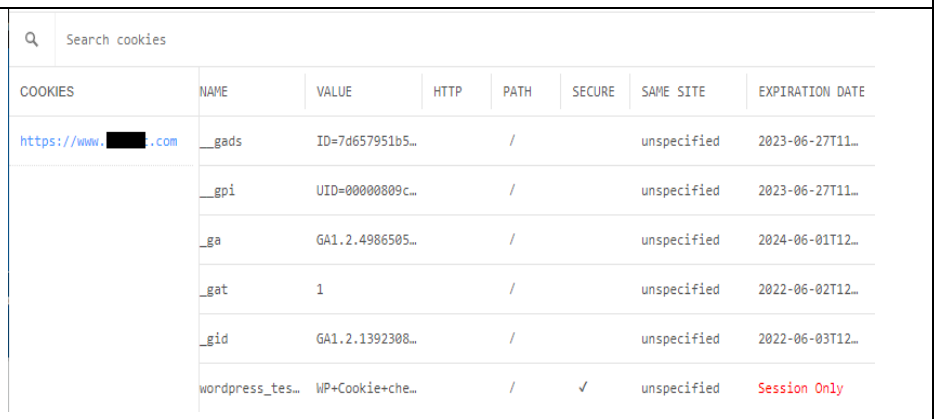
Impact	If the secure flag is not set, then the cookie will be transmitted in clear text if the user visits any HTTP URLs within the cookie's scope. An attacker may be able to induce this event by feeding a user suitable links, either directly or via another web site.
Evidence	 

Table 4.14: Clickjacking: X-Frame-Options header missing.

Target No	**2**
Vulnerability Name	**Clickjacking: X-Frame-Options header missing**
Tool used	**Acunetix, Nikto, owasp zap**
Vulnerability Description	Clickjacking (User Interface redress attack, UI redress attack, UI redressing) is a malicious technique of tricking a Web user into clicking on something different from what the user perceives they are clicking on, thus potentially

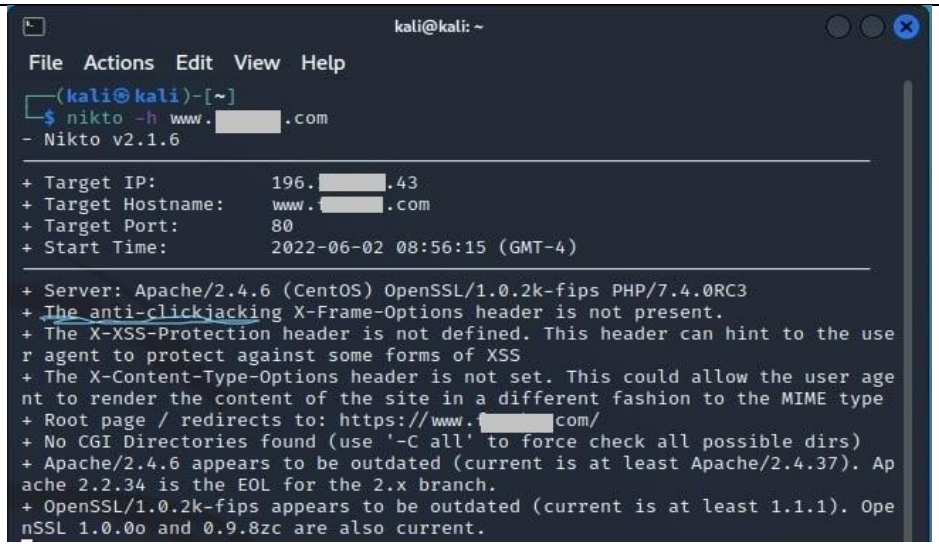
	revealing confidential information or taking control of their computer while clicking on seemingly innocuous web pages.
Risk Level	**Medium**
Impact	The X-Frame-Options HTTP response header can be used to indicate whether a browser should be allowed to render a page in a <frame> or <iframe>. Sites can use this to avoid clickjacking attacks, by ensuring that their content is not embedded into other sites. This vulnerability affects Web Server.
Evidence	

Table 4.15: X-Content-Type-Options Header Missing

Target No	**2**
Vulnerability Name	**X-Content-Type-Options Header Missing**
Tool Used	**Nikto, owasp zap**
Vulnerability Description	The Anti-MIME-Sniffing header X-Content-Type-Options was not set to 'nosniff'. This allows older versions of Internet Explorer and Chrome to perform MIME sniffing on the response body, potentially causing the response body to be interpreted and displayed as a content type other than the declared content type. Current (early 2014) and legacy versions of Firefox will use the declared content type (if one is set), rather than performing MIME-sniffing

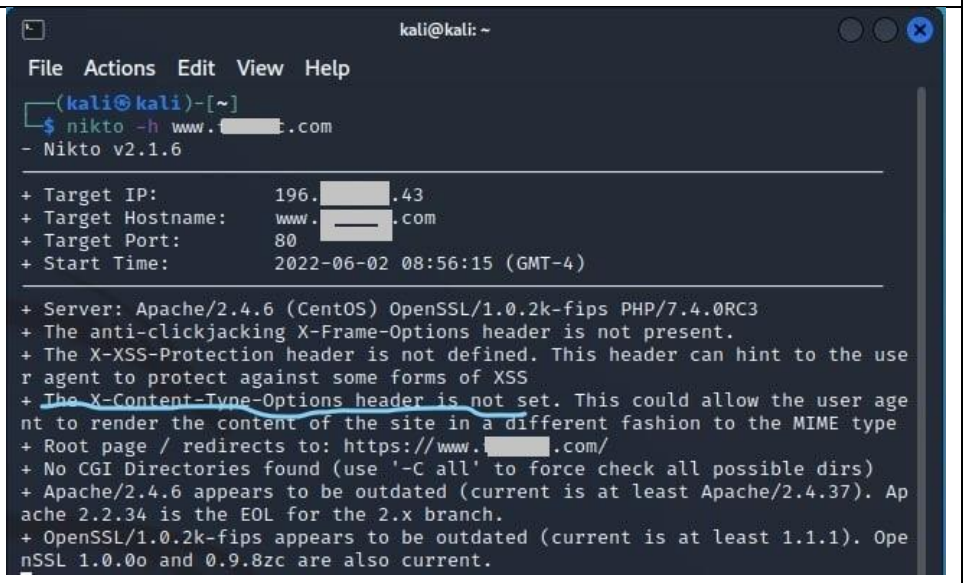
Risk Level	**Medium**
Impact	The HTTP 'X-Content-Type-Options' response header prevents the browser from MIME-sniffing a response away from the declared content-type. The server did not return a correct 'X-Content-Type-Options' header, which means that this website could be at risk of a Cross-Site Scripting (XSS) attack.
Evidence	

Table 4.16: Login page password-guessing attack (Brute-force attack)

Target No	**2**
Vulnerability Name	**Login page password-guessing attack (Brute-force attack)**
Tool used	**Acunetix**
Vulnerability Description	The website login page does not have any protection against password-guessing attacks (brute force attacks). A common threat web developer's face is a password-guessing attack known as a brute force attack. A brute-force attack is an attempt to discover a password by systematically trying every possible combination of letters, numbers, and symbols until you discover the one correct combination that works.
Risk Level	**Low**

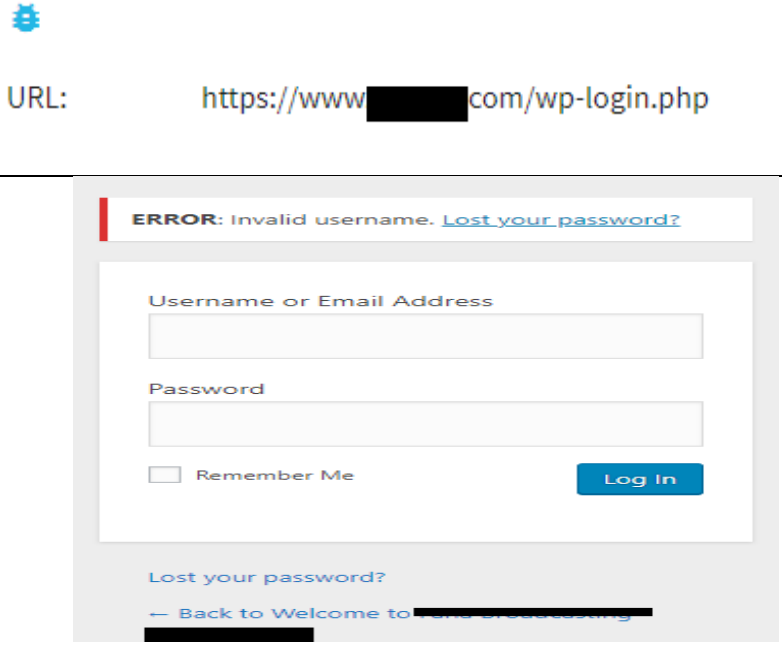
Impact	An attacker may attempt to discover a weak password by systematically trying every possible combination of letters, numbers, and symbols until it discovers the one correct combination that works.
Evidence	**Login page password-guessing attack** 

Table 4.17: SSL Medium Strength Cipher Suites Supported (SWEET32)

Target No	**3**
Vulnerability Name	**SSL Medium Strength Cipher Suites Supported (SWEET32)**
Tool used	**Nessus**
Vulnerability Description	The remote host supports the use of SSL ciphers that offer medium strength encryption. Nessus regards medium strength as any encryption that uses key lengths at least 64 bits and less than 112 bits, or else that uses the 3DES encryption suite.
Risk Level	**High**
Impact	Note that it is considerably easier to circumvent medium strength encryption if the attacker is on the same physical network.

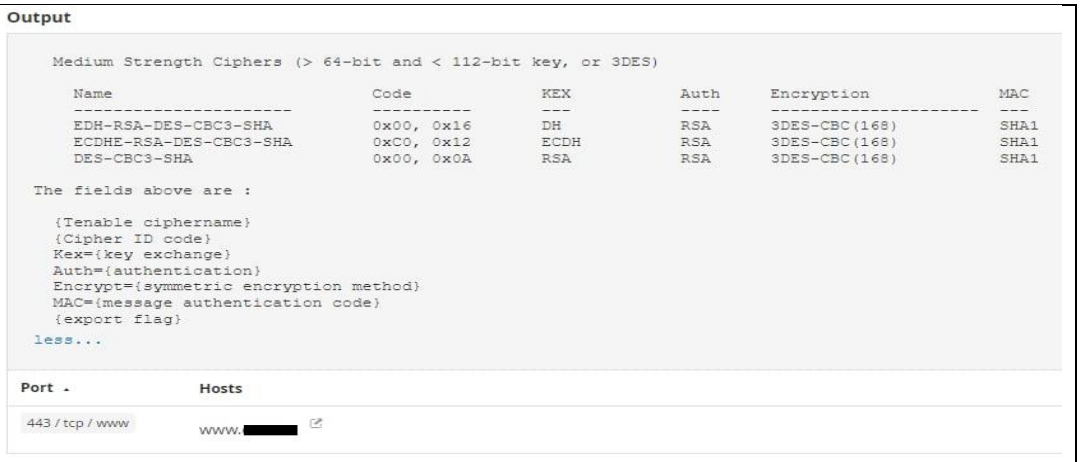
Evidence	

Table 4.18: Cross Site Scripting (XSS)

Target No	**3**
Vulnerability Name	**Cross Site Scripting (XSS)**
Tool used	**Acunetix**
Vulnerability Description	Cross-Site Scripting (XSS) attacks are a type of injection, in which malicious scripts are injected into otherwise benign and trusted websites. XSS attacks occur when an attacker uses a web application to send malicious code, generally in the form of a browser side script, to a different end user. Flaws that allow these attacks to succeed are quite widespread and occur anywhere a web application uses input from a user within the output it generates without validating or encoding it.
Risk Level	**High**
Impact	XSS can cause a variety of problems for the end user that range in severity from an annoyance to complete account compromise. The most severe XSS attacks involve disclosure of the user's session cookie, allowing an attacker to hijack the user's session and take over the account. Other damaging attacks include the disclosure of end user files, installation of Trojan horse programs, redirect the user to some other page or site, or modify presentation of content.

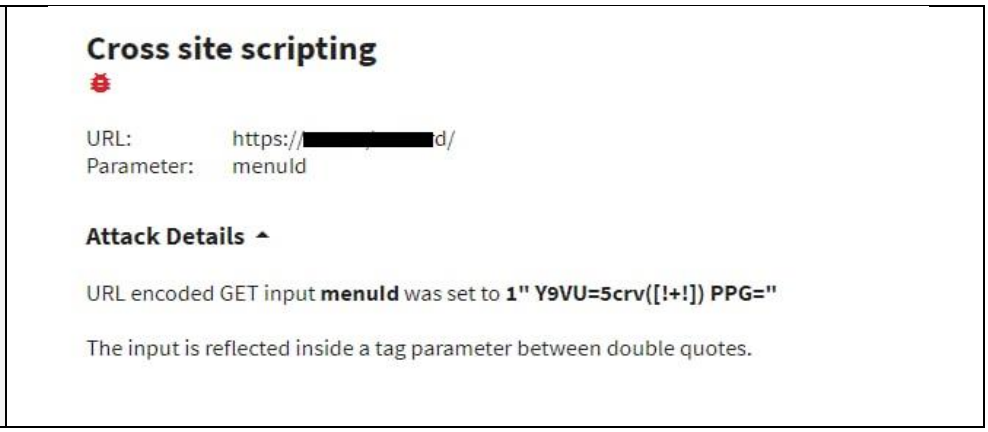
Evidence	

Table 4.19: File upload vulnerabilities

Target No	**3**
Vulnerability Name	**File upload vulnerabilities**
Tool used	**Acunetix**
Vulnerability Description	When a web server allows users to upload files to its file system without sufficiently validating things like their name, type, contents, or size. These pages allow visitors to upload files to the server. Various web applications allow users to upload files (such as pictures, images, sounds,). Uploaded files may pose a significant risk if not handled correctly. A remote attacker could send a multipart/form-data POST request with a specially crafted filename or mime type and execute arbitrary code.
Risk Level	**High**
Impact	If the uploaded files are not safely checked, an attacker may upload malicious files.

Evidence	

Table 4.20: HTTP Strict Transport Security (HSTS) not implemented

Target No	**3**
Vulnerability Name	**HTTP Strict Transport Security (HSTS) not implemented**
Tool used	**Acunetix**
Vulnerability Description	HTTP Strict Transport Security (HSTS) tells a browser that a web site is only accessible using HTTPS. It detected that your web application does not implement HTTP Strict Transport Security (HSTS) as the Strict Transport Security header is missing from the response. HSTS is also a good method to protect yourself from cookie hijacking. Does not redirect http traffic to https so the system is insecure for the user.
Risk Level	**Medium**
Impact	It detected that your web application does not implement HTTP Strict Transport Security (HSTS) as the Strict Transport Security header is missing from the response. HSTS can be used to prevent and/or mitigate some types of man-in-the-middle (MITM) attacks

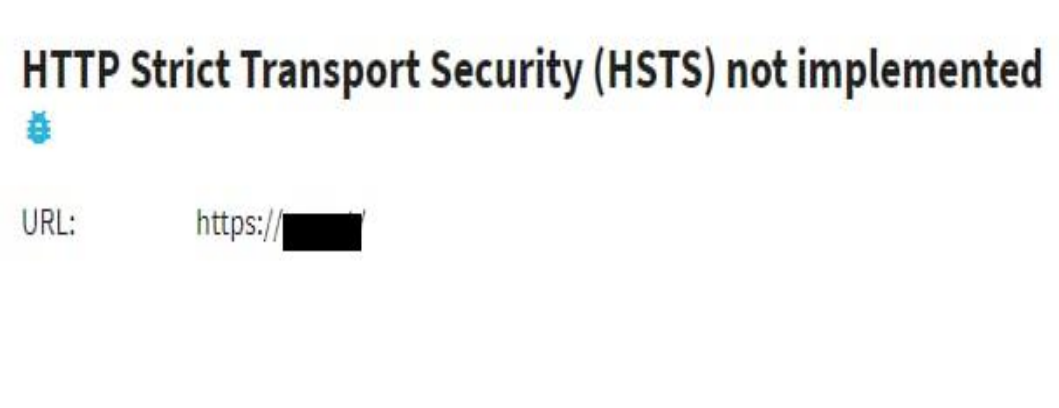
Evidence	

Table 4.21: Insecure Inline Frame (iframe)

Target No	**3**
Vulnerability Name	**Insecure Inline Frame(iframe)**
Tool used	**Acunetix**
Vulnerability Description	The Inline Frame is either configured insecurely, or not as securely as expected. This vulnerability alert is based on the origin of the embedded resource and the iframe's sandbox attribute, which can be used to apply security restrictions as well as exceptions to these restrictions
Risk Level	**Low**
Impact	IFrame sandboxing enables a set of additional restrictions for the content within a frame in order to restrict its potentially malicious code from causing harm to the web page that embeds it. The Same Origin Policy (SOP) will prevent JavaScript code from one origin from accessing properties and functions - as well as HTTP responses - of different origins. The access only allowed if the protocol, port and the domain match exactly.

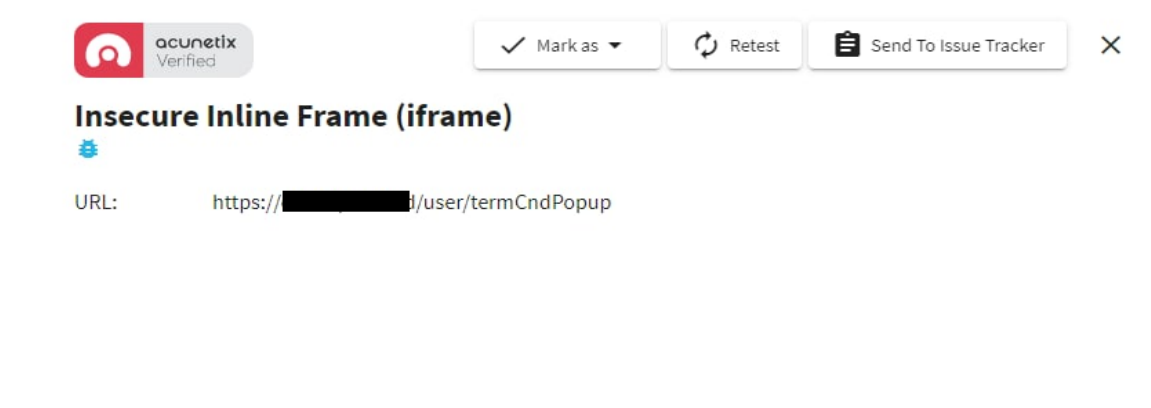
Evidence	

Table 4.22: Vulnerable JavaScript libraries

Target No	**4**
Vulnerability Name	**Vulnerable JavaScript libraries**
Tool Used	**Acunetix**
Vulnerability Description	A JavaScript library that is missing security patches can make your website extremely vulnerable to various attacks.
Risk Level	**High**
Impact	Third party JavaScript libraries can draw a variety of DOM-based vulnerabilities, including DOM-XSS, which can be exploited to hijack user accounts.

Evidence	

Table 4.23: Header file missing

Target No	**4**
Vulnerability Name	**Header File missing**
Tool used	**Acunetix, Nikto**
Vulnerability Description	Due to a missing HTTP Strict Transport Security header, an unaware user can navigate by mistake to the unencrypted version of the web application or accept invalid certificates.
Risk Level	**Medium**
Impact	This leads to sensitive data being sent unencrypted over the wire.

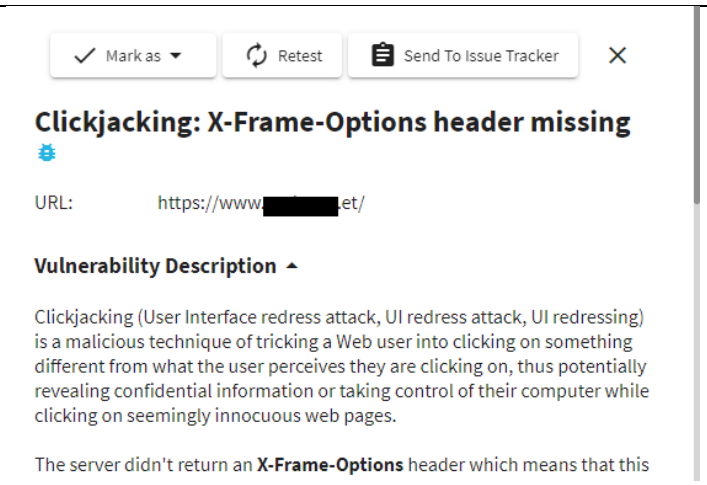
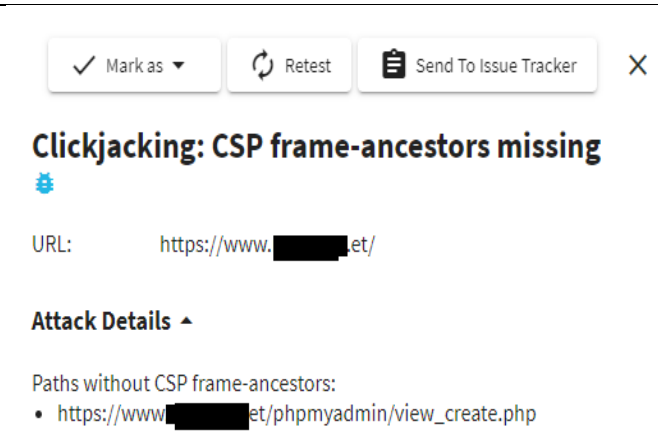
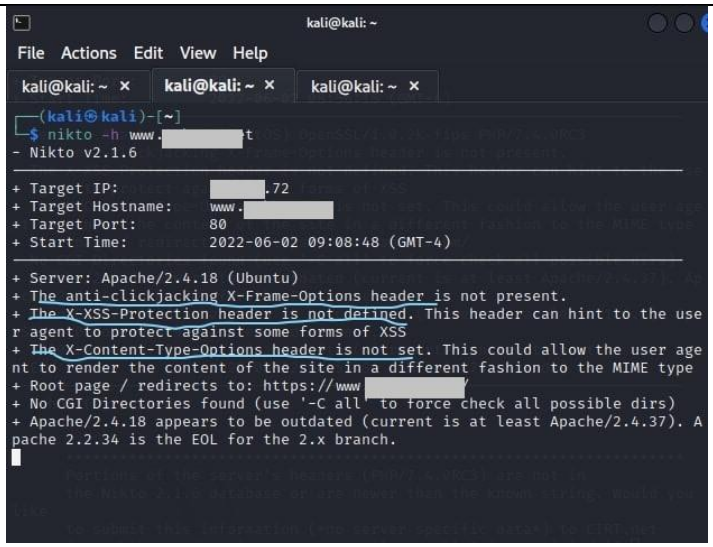
Evidence	  

Table 4.24: Apache HTTP Server 2.4.18 appears (Outdate)

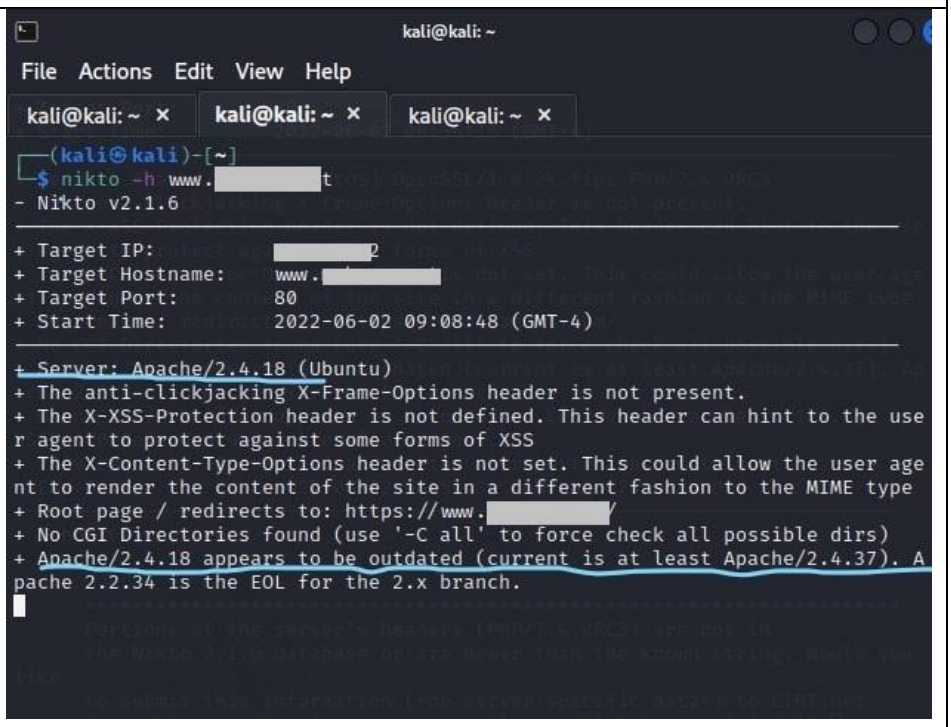
Target No	**4**
Vulnerability Name	**Apache HTTP Server Outdated**
Tool used	**Nikto**
Vulnerability Description	This affects the Apache HTTP Server through 2.2.34 and 2.4.x through 2.4.27. The attacker sends an unauthenticated OPTIONS HTTP request when attempting to read secret data. This is a use-after-free issue and thus secret data is not always sent, and the specific data depends on many factors including configuration.
Risk Level	**Medium**
Impact	Apache http allows remote attackers to read secret data from process memory if the Limit directive can be set in a user .htaccess file, or if httpd.conf has certain misconfigurations, akaOptionsbleed.
Evidence	

Table 4.25: HTTP Strict Transport Security (HSTS) not implemented

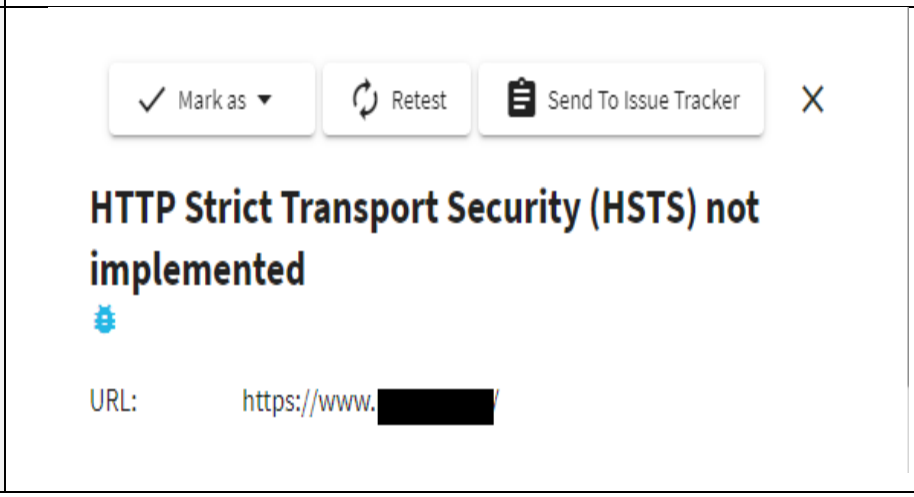
Target No	**4**
Vulnerability Name	**HTTP Strict Transport Security (HSTS) not implemented**
Tool used	**Acunetix**
Vulnerability Description	HTTP Strict Transport Security (HSTS) tells a browser that a web site is only accessible using HTTPS.
Risk Level	**Medium**
Impact	It was detected that your web application does not implement HTTP Strict Transport Security (HSTS) as the Strict Transport Security header is missing from the response.
Evidence	

Table 4.26: Directory Listing

Target No	**4**
Vulnerability Name	**Directory Listing**
Tool Used	Acunetix
Vulnerability Description	Listing directory contents when no index file is present in a common misconfiguration. The directory contents can provide useful information to an attacker, especially if there are files that are not meant to be

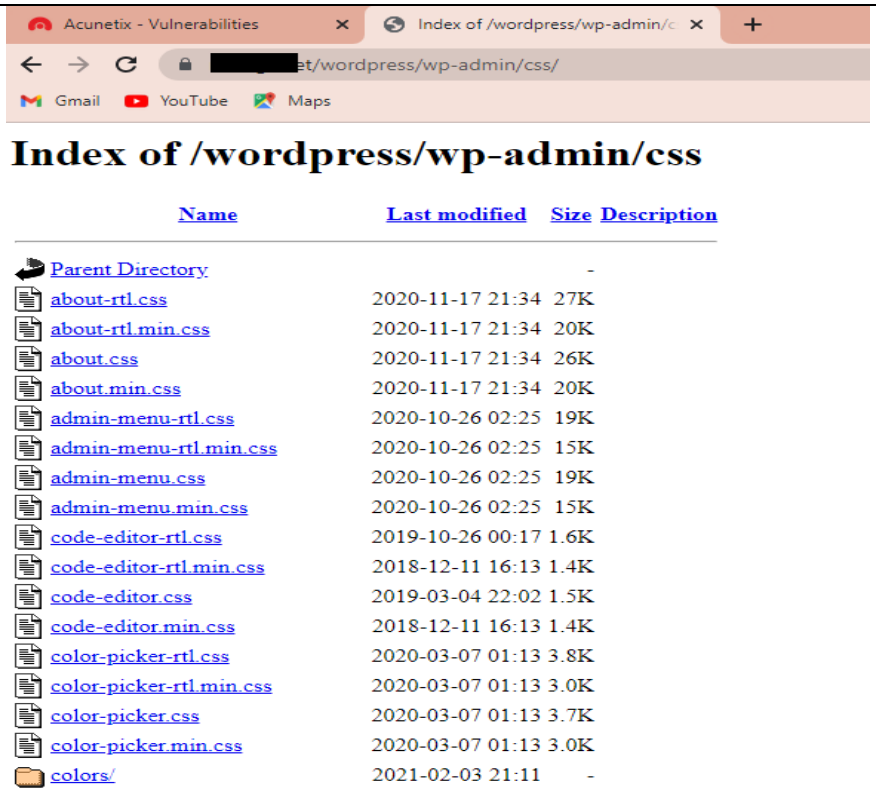
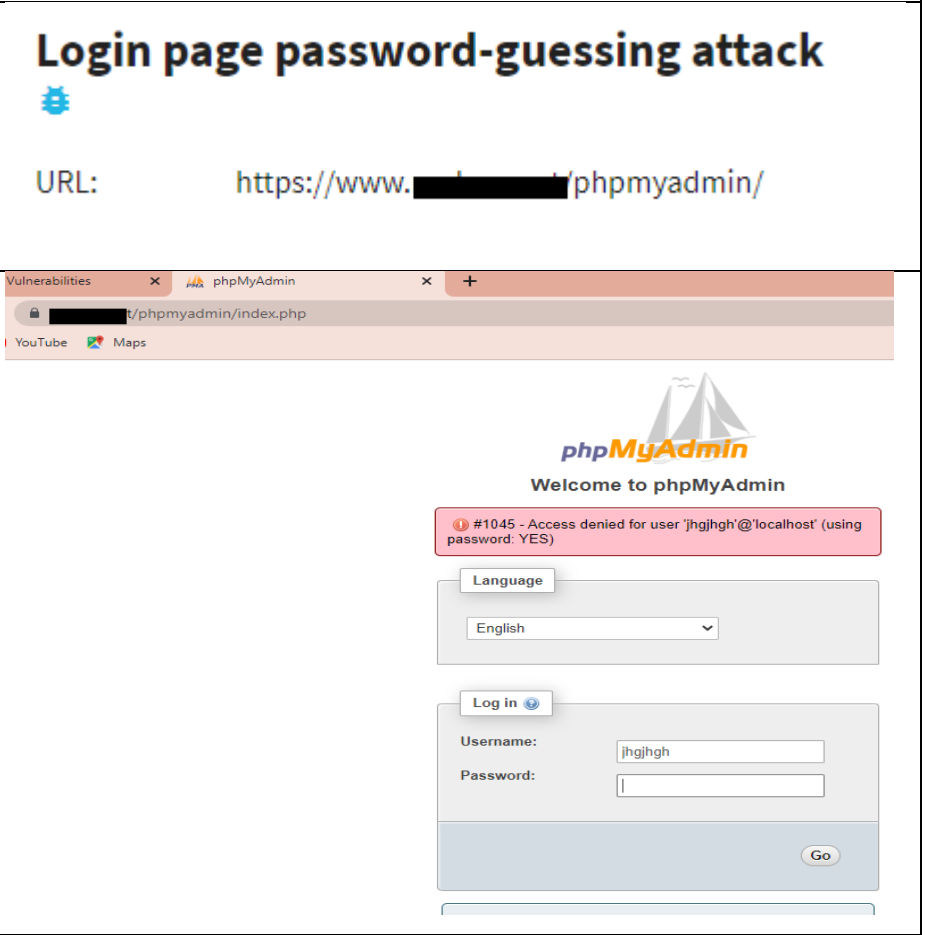
	accessible, such as source code or backups. The directory listing may also provide useful information about the habits of the server administration and/or web developers, such as file naming convention, that could be used to increase the probable success of brute-force or other attacks.
Risk Level	**Medium**
Impact	• The server is outputting the contents of directories. • This could expose files not meant for user retrieval (old htaccess files, backups, source code). • The directory listing may additionally provide useful information about the system layout and characteristics, such as naming conventions used by the developers and administrators. • This information can increase the probability of success for blind attacks and brute force guessing.
Evidence	

Table 4.27: Login page password-guessing attack (Brute-force attack)

Target No	**4**
Vulnerability Name	**Login page password-guessing attack(Brute-force attack)**
Tool used	**Acunetix, owasp zap**
Vulnerability Description	The website login page does not have any protection against password-guessing attacks (brute force attacks). A common threat web developer's face is a password-guessing attack known as a brute force attack. A brute-force attack is an attempt to discover a password by systematically trying every possible combination of letters, numbers, and symbols until you discover the one correct combination that works.
Risk Level	**Medium**
Impact	An attacker may attempt to discover a weak password by systematically trying every possible combination of letters, numbers, and symbols until it discovers the one correct combination that works.
Evidence	

4.5 Summary of Findings

The discovered of vulnerabilities analysis summarized based on risk level that covers compromise Confidentiality, Integrity and Availability (CIA) on services and applications over the website. The risk level analysis (Low, Medium, and High) discovered on the services and applications described as follows:

Table 4.28: Risk Level analysis

Risk Level	High Risk	Medium Risk	Low Risk
Number of vulnerabilities	6	6	2
Percentage (%)	37.5%	37.5%	25%

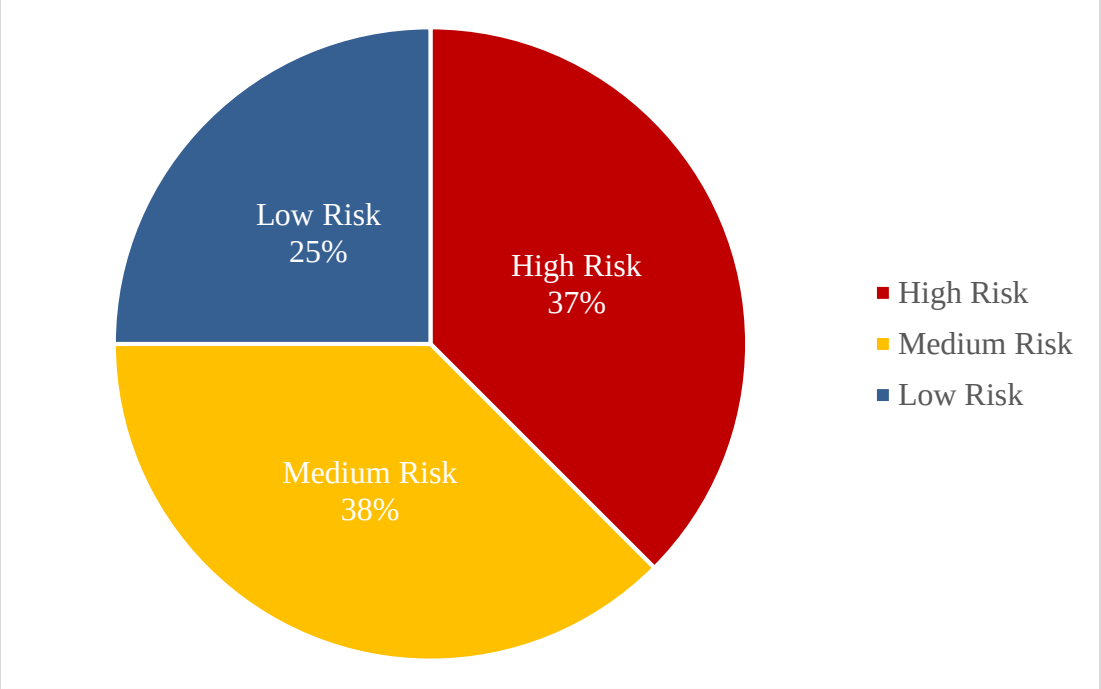


Figure 4.1: Impact Rate

The above pie chart and table has been summarized all vulnerability analysis finding results of both approaches based on vulnerability impact rate or risk level perspective.

4.6 Risk Calculation

All over the document, each risk calculated has been listed in a figure under this section as a finding and categorized as a **High-Risk**, **Medium-Risk**, or **Low-Risk**. The study has been used the following Risk calculation formula to calculate the risks. The risk calculation is done based on ***Common Vulnerabilities and Exposures (CVE***) system provides a reference-method for publicly known information-security vulnerabilities and exposures) that is a catalog of known security threats according to the CVE website and Chief information officer (CIO) reports directly to the chief executive officer analysis.

Risk= Likelihood*impact.

Which denotes that the total amount of risk exposure is the probability of an unfortunate event occurring, multiplied by the potential impact or damage incurred by the event. If you put a value on the impact, then you can value the risk and in a simple way compare one risk factor to another.

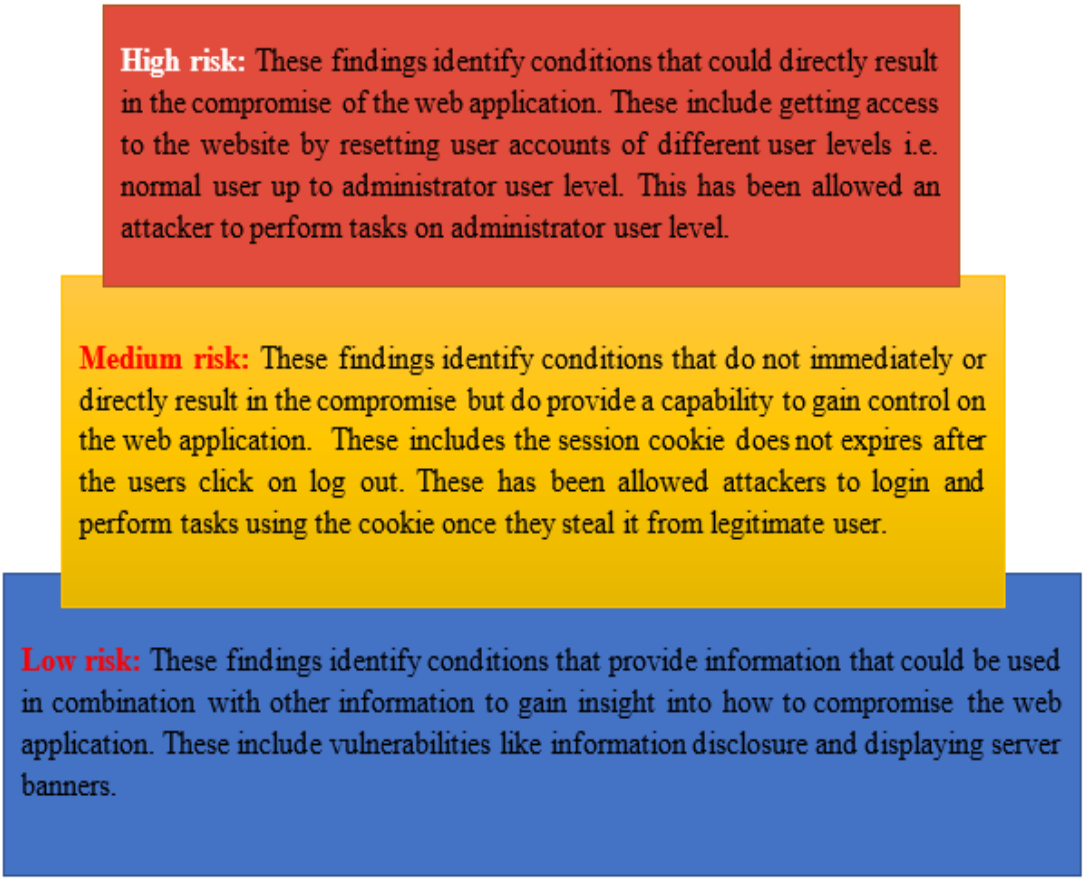


Figure 4.2: Risk level description

CHAPTER FIVE
5 CONCLUSIONS AND RECOMMENDATIONS

This chapter is categorized into two sections. The conclusion of the research and the recommendations for the future work.

5.1 Conclusions

The main goal of this study were asses the Ethiopian governmental office web and check if the system is vulnerable. Ethiopian governmental offices websites exposed to some vulnerabilities that may led to system down and its infrastructure compromise. Those vulnerabilities should have to be eliminated as soon as possible. As vulnerability analysis is a fundamental part of a company's security plan that can be conducted in-house or by a trusted IT partner. Vulnerability analysis is a comprehensive method to identify the Penetration testing in a system. It offers benefits such as prevention of financial loss; compliance to industry regulators, customers and shareholders; preserving corporate image; proactive elimination of identified risks before damage. The research has been used white box and black box penetration testing, depending on the specific objectives to be achieved. The security of a website vulnerability analysis adapting any pen-test methodology does not necessarily provide a complete picture of the vulnerability analysis process, which execute pen-test methodology. The research had carried out by identifying:

- The sampling technique to be used
- The procedures that should be applied the test methodology and
- Active devices, services and applications have been tested.
- Vulnerability analysis techniques
- The kali Linux tools to be used for the tests.
- Security metrics to be used to perform risk analysis

Vulnerabilities are identified in the work shows that the web system and all devices and applications mentioned on the scope of this research had been tested against a well-known vulnerability on common vulnerability databases (CVE). However, executing risk analysis and impact was proposed only for the discovered vulnerabilities. Vulnerability analysis and scanning to search known vulnerabilities on Ethiopian governmental office hosts and web applications

based on low, medium and high-risk factors are considered. All the available vulnerabilities identified and verified; these include in this research:

- Application and Database vulnerabilities
- Enumeration of Identified vulnerabilities
- Verification of the identified vulnerabilities
- Hosts vulnerabilities analysis.

Vulnerability analysis can be an efficient and cost-effective strategy to protect the organization's systems against attacks; however, vulnerability analysis should be following a comprehensive methodology format to present the system test results via governmental office network asset. One of the most important parts of the test analysis phase is the preparation of remediation, which includes all necessary corrective measures for the identified vulnerabilities.

5.2 Recommendations

The goal of this research is to have a vulnerability analysis on web security, and governmental ICT infrastructures is to defend information from unauthorized access, use, disclosure, disruption, modification, destruction. From this point of view, misconfiguration of services and applications exposed the website for the vulnerabilities that could have enormous impact on governmental office network infrastructure. Finally in this research has recommend to put forward our strong remind to origination: The governmental office network asset needs to have strong IT policy for

- Use strong security policy
- Updating web technology
- Upgrading OSs of Networking devices on a regular basis
- IT service management
- Web Back end and front-end security implement
- Internet Acceptable use policy

To summarize recommendation, the vulnerability need to fix as soon as possible especially the high vulnerability. In addition, the application uses third party frameworks and libraries that should have to be update and patched on the regular basis, but currently, the websites, which has multiple types of vulnerabilities. In this work has been done only focused on some governmental office website, the other office website and network infrastructure is needs to be vulnerability analysis for the future work.

References

[1] A. M. Tilahun Ejigu, "Web Security vulnerability analysis of Ethiopan government offices," *Mukt Shabd Journal,* Vols. IX,, no. VIII, p. 6, August/2020.

[2] D. G. DivyaniYadav, "Vulnerabilities and Security of Web Applications," *ResearchGate,* p. 6, December 2018.

[3] C. S. Tiago Vieira, "Web Applications Security and Vulnerability Analysis Financial Web Applications Security Audit," *International Journal of Innovative Business Strategies (IJIBS),* vol. 2, no. 2, p. 9, December 2016.

[4] Jennifer.A, R. Senthil Kumaran Arul Louise, "Development of Vulnerability Scanner," *International Research Journal of Engineering and Technology (IRJET),* vol. 05, no. 07, p. 4, July 2018.

[5] Narayan, Jai, "Vulnerability Assessment and Penetration Testing in Web," *2017 International Conference on Computing, Communication, Control and Automation (ICCUBEA),* pp. 1-6, August 2015.

[6] Rahul Johari, Ishveen K., "Penetration Testing in IoT Network," *2020 5th International Conference on Computing, Communication and Security (ICCCS),* pp. 1-7, May 18,2021.

[7] Baybutt, Paul, "Cyber Security Vulnerability Analysis: An Asset-Based Approach," *Process safety progress,* Vols. 22, No.4, pp. 220-228, December 2003.

[8] Farah Abu-Dabaseh, and Esraa Alshammari, "Automated penetration testing," *Computer Science and Information Technology,* vol. 8, no. 6, October 2018.

[9] AnanthaSayana, S., "Approach to Auditing Network Security," *Information systems control jornal,* vol. 5, p. 3, 2003..

[10] Trupti Bhosale, S.N. Mhatre, "Testing_Web_Application_using_Vulnerability scan," vol. 06, no. 05, May 2019.

[11] Zaid. J. Al-Araji, SharifahSakinah Syed Ahmad, "Propose Vulnerability Metrics to Measure Network Secure using Attack Graph," *International Journal of Advanced Computer Science and Applications,* vol. 12, no. 5, June 1 2021..

[12] Mebrehtu, Gebrekidan Gebremedihn, "Developing black box web application penetration testing methodology using comparative criteria," June 2015.

[13] Tewodros Getaneh, "Cyber security practices and challenges at selected critical infrastructures in Ethiopia: Towards Tailoring cyber security framework," p. 123, June 2018.

[14] Ahmed, RanaKhudhair Abbas, "Overview of Security Metrics," *Science publishing group,* vol. 4, no. 4, pp. 59-64, December 5, 2016..

[15] Gaurav Bhatia, O. Bhatia , "Vulnerability Assessment and Penetration Testing," *International journal of engineering research & technology (IJERT),* Vols. 10,, no. 05, May 2021.

[16] Prashant S. Shinde, Shrikant B. Ardhapurkar, "Cyber Security Analysis using Vulnerability Assessment and Penetration Testing," *2016 World Conference on Futuristic Trends in Research and Innovation for Social Welfare (Startup Conclave),* pp. 1-5, March 2016.

[17] Shebli, Hessa Mohammed Zaher Al and Beheshti, Babak D, "A Study on Penetration Testing Process and Tools," *2018 IEEE Long Island Systems, Applications and Technology Conference (LISAT,* pp. 1-7, May 2018.

[18] Yohannes, Tsedale, "Assesment of information security incident management practice in Ethiopian bank," pp. 1-91, June, 2018.

[19] Abebe, Habtamu Girma, "Security testing of Ethiopian E-governmental websites using penetration testing tools," p. 80, 2019.

[20] Jose´ Fonseca, N. Seixas, "Analysis of Field Data on web Security Vulnerabilities," *IEEE Transactions on Dependable and Secure Computing,* vol. 11, no. 2, pp. 89-100, March/April 2014.

[21] Palak Aar, Aman Kumar Sharma, Analysis of Penetration Testing Tools, September 2017.

[22] Nagendran K, Adithyan A, Chethana R, Camillus P, Bala Sri Varshini K B, "Web Application Penetration Testing," vol. 8, no. 10, August 2019..

[23] Sheetal Bairwa, B. Mewara , "Vulnerability scanners: A Proactive approach to asses web application security," *International Journal on Computational Sciences & Applications (IJCSA),* vol. 4, p. 12, February 2014.

[24] Muhammad Kasim Lim, "Penetration Testing using Kali Linux: SQL Injection, XSS, Wordpres, and WPA2 Attacks," *Indonesian Journal of Electrical Engineering and Computer Science* , vol. 12, no. 2, pp. 729-737, November 2018.