WEB APPLICATION SECURITY TESTING: NIDS-BASED PENETRATION APPROACH

A PROJECT REPORT

Submitted by,

SHREYA MALSHETTY	**-**	**20211CCS0061**
JAYALAKSHMI D	**-**	**20211CCS0064**
CHAITHRA H BANGER	**-**	**20211CCS0114**
SRUJANA MANJUNATH	**-**	**20211CCS0116**

Under the guidance of,

Dr. SHARMASTH VALI Y

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING, CYBER SECURITY

At

GAIN MORE KNOWLEDGE
REACH GREATER HEIGHTS

PRESIDENCY UNIVERSITY

BENGALURU

JANUARY 2025

Scanned with OKEN Scanner

PRESIDENCY UNIVERSITY

SCHOOL OF COMPUTER SCIENCE ENGINEERING

CERTIFICATE

This is to certify that the Project report **"WEB APPLICATION SECURITY TESTING: NIDS-BASED PENETRATION APPROACH"** being submitted by **SHREYA MALSHETTY; JAYALAKSHMI D; CHAITHRA H BANGER; SRUJANA MANJUNATH** bearing roll number(s) **20211CCS0061; 20211CCS0064; 20211CCS0114; 20211CCS0116** in partial fulfillment of the requirement for the award of the degree of Bachelor of Technology in Computer Science and Engineering is a bonafide work carried out under my supervision.

15/01/25

Dr. SHARMASTH VALI Y
Associate Professor
School of CSE&IS
Presidency University

15/1/2025

Dr. ANANDARAJ S P
Professor & HOD
School of CSE&IS
Presidency University

Dr. L. SHAKKEERA
Associate Dean
School of CSE
Presidency University

Dr. MYDHILI NAIR
Associate Dean
School of CSE
Presidency University

Dr. SAMEERUDDIN KHAN
Pro-VC School of Engineering
Dean -School of CSE&IS
Presidency University

Scanned with OKEN Scanner

PRESIDENCY UNIVERSITY

SCHOOL OF COMPUTER SCIENCE ENGINEERING

DECLARATION

We hereby declare that the work, which is being presented in the project report entitled **WEB APPLICATION SECURITY TESTING: NIDS-BASED PENETRATION APPROACH** in partial fulfillment for the award of Degree of **Bachelor of Technology** in **Computer Science and Engineering**, is a record of our own investigations carried under the guidance of **Dr. Sharmasth Vali Y, ASSOCIATE PROFESSOR, School of Computer Science Engineering & Information Science, Presidency University, Bengaluru.**

We have not submitted the matter presented in this report anywhere for the award of any other Degree.

NAME	ROLL NUMBER	SIGNATURE
SHREYA MALSHETTY	20211CCS0061	
JAYALAKSHMI D	20211CCS0064	Jayalakshmi.
CHAITHRA H BANGER	20211CCS0114	
SRUJANA MANJUNATH	20211CCS0116	

Scanned with OKEN Scanner

ABSTRACT

Imagine receiving an email from hackers informing you that your home address, date of birth, name, and bank information have all been compromised. They are asking for cash in exchange for keeping your personal information hidden. It sounds like a frightening scenario, doesn't it? We employ an NIDS-based approach to web application penetration testing in order to solve this issue. Web application penetration testing is a continuous security assessment technique that simulates actual attacks to evaluate the security of web applications. The main objective is to find any potential flaws, incorrect configurations, or vulnerabilities that malicious users can utilize to jeopardize an online application's availability, confidentiality, or integrity. An NIDS can be used in conjunction with more conventional penetration testing techniques that concentrate on software flaws to identify fraudulent behavior at the network level. By detecting fraudulent activity at the network level, an NIDS can supplement traditional penetration testing methods that focus on software defects. By merging conventional web application penetration testing with Network Intrusion Detection Systems (NIDS), this study aims to enhance the detection of security flaws at the network and application levels.

Scanned with OKEN Scanner

ACKNOWLEDGEMENT

First of all, we indebted to the **GOD ALMIGHTY** for giving me an opportunity to excel in our efforts to complete this project on time.

We express our sincere thanks to our respected dean **Dr. Md. Sameeruddin Khan**, Pro-VC, School of Engineering and Dean, School of Computer Science Engineering & Information Science, Presidency University for getting us permission to undergo the project.

We express our heartfelt gratitude to our beloved Associate Deans **Dr. Shakkeera L and Dr. Mydhili Nair,** School of Computer Science Engineering & Information Science, Presidency University, and **Dr. Andaraj S P**, Head of the Department, School of Computer Science Engineering & Information Science, Presidency University, for rendering timely help in completing this project successfully.

We are greatly indebted to our guide **Dr. Sharmasth Vali Y**, Associate Professor and Reviewer **Mr. Tanveer Ahmed,** Assistant. Professor, School of Computer Science Engineering & Information Science, Presidency University for his/her inspirational guidance, and valuable suggestions and for providing us a chance to express our technical capabilities in every respect for the completion of the project work.

We would like to convey our gratitude and heartfelt thanks to the PIP2001 Capstone Project Coordinators **Dr. Sampath A K, Dr. Abdul Khadar A and Mr. Md Zia Ur Rahman,** and Git hub coordinator Mr. **Muthuraj.**

We thank our family and friends for the strong support and inspiration they have provided us in bringing out this project.

Shreya Malshetty

Jayalakshmi D

Chaithra H Banger

Srujana Manjunath

Scanned with OKEN Scanner

LIST OF TABLES

Sl. No.	Table No.	Table Caption	Page No.
1	2.1	Literature Survey	8

Scanned with OKEN Scanner

LIST OF FIGURES

Sl. No.	Figure No.	Caption	Page No.
1	6	Network Penetration Testing	23
2	7.5	Gnatt Chart	28
3		APPENDIX -A	36
		SCREENSHOTS	
	i, ii	Nessus	36
	iii, iv	Nmap	37
	v, vi, viii	Owsap Zap	38
4		APPENDIX-B	40
		ENCLOSURES	
		Plagiarism Check Report	40
		SDG Mapping	41

Scanned with OKEN Scanner

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	**iv**
	ACKNOWLEDGMENT	**v**
	LIST OF TABLES	**vi**
	LIST OF FIGURES	**vii**
1.	**INTRODUCTION**	1
	1.1 Overview of Web Application Security	1
	1.1.1 Importance of Web Application security in Digital age	2
	1.1.2 Common types of Vulnerabilities in Web Application	2
	1.1.3 Risks associated with Insecure Web Application	3
	1.2 Integration of NIDS with Penetration Testing	3
	1.2.1 How NIDS Enhances Penetration Testing	3
	1.2.2 Best practices for Integrating NIDS with Penetration Testing	4
	1.3 Key Steps in NIDS Penetration Testing Process	4
	1.3.1 Step 1: Define Testing Objectives and Scope	4
	1.3.2 Step 2: Perform Network Mapping and Vulnerability Assessment	5
	1.3.3 Step 3: Execute Penetration Testing	5
	1.3.4 Step 4: Monitor and Analyze NIDS Alerts	6
	1.3.5 Step 5: Post-Test Analysis and Reporting	6
	1.3.6 Resting and Continuous Monitoring	7
2.	**LITERATURE SURVEY**	**8**

Scanned with OKEN Scanner

CHAPTER NO.	TITLE	PAGE NO.
3.	**RESEARCH GAPS OF EXISTING METHODS**	14
	3.1 Detection of Encrypted Traffic	14
	3.2 False Positives and False Negatives in NIDS Alerts	14
	3.3 Scalability of NIDS in Large-Scale Web Application	15
	3.4 Detection of Zero-Day Attack	15
	3.5 Automated Exploitation Detection	15
	3.6 User Behavior and Analytics and Insider Threats	16
	3.7 Lack of Contextual Awareness in NIDS	16
	3.8 Real-Time Threat Intelligence Detection	16
4.	**PROPOSED METHODOLOGY**	**17**
	4.1 Planning and Preparation	17
	4.2 NIDS Configuration and Monitoring Setup	17
	4.3 Vulnerability Scanning	17
	4.4 Automated Security Testing	18
	4.5. Manual Testing	18
	4.6 Post-Exploitation	18
	4.7 Reporting	18
5.	**OBJECTIVES**	**19**
	5.1 Identify Vulnerabilities	19
	5.2 Test Detection Capabilities	19
	5.3 Analyze Network Traffic	20
	5.4 Evaluate Incident Response	20
	5.5 Continuous Improvement	21
	5.6 Documentation and Reporting	22

Scanned with OKEN Scanner

6.	**SYSTEM DESIGN AND IMPLEMENTATION**	**23**
	6.1 Software Requirements	23
	6.1.1 Network Intrusion Detection Tools (NIDS): Nmap	23
	6.1.2 Web Application Tools: OWSAP ZAP (Zed Attack Proxy)	24
	6.1.3 Vulnerability Scanners: Nessus	24
7.	**TIMELINE FOR EXECUTION OF PROJECT**	**26**
8.	**OUTCOMES**	**29**
	8.1 Enhanced Security Posture	29
	8.1.1 Improved Vulnerability Detection	29
	8.1.2 Early Detection of Threats	29
	8.1.3 Real-Time Alerts	29
	8.2 Strengthened Incident Response	29
	8.2.1 Improved Response Times	29
	8.2.2 Actionable Incident Data	29
	8.2.3 Automated Response	30
	8.3 Optimized NIDS Configuration and Performance	30
	8.3.1 Refined Detection Rules	30
	8.3.2 Reduced False Positives and Negatives	30
	8.4 Comprehensive Vulnerability Management	30
	8.4.1 Prioritized Vulnerabilities	30
	8.4.2 Continuous Vulnerability Monitoring	30
	8.5 Documentation and Reporting	30
	8.5.1 Detailed Penetration Testing Reports	30
	8.5.2 Recommendations for Future Security Improvements	30
9.	**RESULTS AND DISCUSSIONS**	**31**
10.	**CONCLUSION**	**32**

Scanned with OKEN Scanner

REFERENCES 33
APPENDIX-A 36
APPENDIX-B 40

Scanned with OKEN Scanner

CHAPTER-1
INTRODUCTION

Web application security testing is an essential practice for identifying vulnerabilities, ensuring the security and integrity of web applications, and preventing unauthorized access or exploitation. Given the critical role that web applications play in modern digital ecosystems, securing these applications is paramount to protecting sensitive data, ensuring privacy, and maintaining business operations. With the increasing sophistication of cyber-attacks, traditional testing methods are no longer sufficient on their own to identify and mitigate emerging threats.

One of the advanced methodologies in web application security testing is the use of Network Intrusion Detection Systems (NIDS) along with penetration testing. By leveraging NIDS, penetration testers can gain deeper insights into network traffic and identify threats that may otherwise go unnoticed, especially those targeting the underlying infrastructure and communication channels.

This approach not only helps in identifying vulnerabilities but also provides actionable intelligence for mitigating risks in real-time. By adopting a NIDS-based penetration testing strategy, organizations can better anticipate potential attacks and reinforce their web applications with stronger, more proactive security measures.

1.1 Overview of Web Application Security

As the world becomes more digitally connected, web applications play a crucial role in delivering services, storing sensitive data, and enabling business operations. Web application security refers to the practices, methodologies, and tools used to protect web applications from security threats, vulnerabilities, and attacks. However, this increased reliance on web-based services also makes them an attractive target for attackers. Therefore, ensuring the security of web applications is critical to prevent data breaches, financial losses, and damage to reputation.

Scanned with OKEN Scanner

In this section, we'll dive into several key aspects of web application security, including its importance, common vulnerabilities, and risks associated with insecure applications.

1.1.1 Importance of Web Application Security in Digital Age

Web applications are the backbone of most online services, from e-commerce and social media platforms to banking and healthcare applications. They allow users to interact with systems and access information over the internet. As businesses increasingly rely on web applications for operations, security becomes a fundamental concern. The main reasons for prioritizing web application security are:

- **Protection of Sensitive Data:** Web applications often store and process sensitive user data, including personal identification information (PII), financial records, and login credentials. A breach of this data can lead to identity theft, financial fraud, and regulatory consequences.
- **Ensuring Business Continuity:** Security vulnerabilities in web applications can disrupt business operations, either by enabling cyberattacks like Distributed Denial of Service (DDoS) attacks or through exploitation of flaws that allow attackers to hijack services or disable systems.
- **Protecting Brand Reputation:** A data breach or compromised web application can lead to a loss of customer trust, damaged brand reputation, and long-term financial impacts.

1.1.2 Common types of Vulnerabilities in Web application

Web application vulnerabilities are weaknesses or flaws in the design, implementation, or configuration of an application that can be exploited by attackers. Some of the most common vulnerabilities include:

- **SQL Injection (SQLi)** - SQL Injection is one of the oldest and most dangerous types of web application vulnerabilities. It occurs when an attacker is able to inject malicious SQL code into an input field (e.g., search bars, login forms) that is then executed by the database.
- **Cross-Site Scripting (XSS)** - Cross-Site Scripting (XSS) occurs when an attacker injects malicious scripts (usually JavaScript) into a trusted website or web application.

Scanned with OKEN Scanner

- **Cross-Site Request Forgery (CSRF) -** CSRF exploits the trust a web application has in the user's browser. It involves tricking the user into submitting a request (such as changing their account settings) without their knowledge, leveraging their active session.

- **Broken Authentication** - Broken authentication vulnerabilities occur when an application's authentication mechanisms (login systems, password resets, session management) are weak or improperly implemented.

- **Security Misconfiguration -** Security misconfigurations can occur in any layer of an application stack (e.g., the server, database, application, cloud services) when they are incorrectly configured or left with default settings.

- **Sensitive Data Exposure -** Sensitive data exposure happens when an application does not adequately protect sensitive data, such as passwords, credit card numbers, or health records, during storage or transmission.

1.1.3 Risks associated with Insecure Web applications

- **Data Breaches**
- **Denial of Service (DoS) and Distributed Denial of Service (DDoS) Attacks**
- **Malware Injection**
- **Reputation Damage**

1.2 Integration of NIDS with Penetration testing

The integration of **Network Intrusion Detection Systems (NIDS)** with **penetration testing** creates a robust, multi-layered security approach that enhances the ability to detect, analyze, and respond to potential threats in real time. While penetration testing helps uncover vulnerabilities in web applications through simulated attacks, NIDS focuses on continuously monitoring network traffic to detect malicious activities. Combining these two methodologies creates a more comprehensive security posture for web applications, providing both proactive testing and real-time detection of threats.

Scanned with OKEN Scanner

1.2.1. How NIDS Enhances Penetration Testing

- **Real-Time Detection of Exploited Vulnerabilities -** During penetration testing, ethical hackers attempt to exploit vulnerabilities. NIDS can monitor this activity and provide real-time alerts on the actions of the penetration testers.

- **Automating and Streamlining the Penetration Testing Process -** NIDS can be integrated with automated vulnerability scanners, providing **continuous testing** of the web application, which can be especially useful in DevOps environments.

1.2.2 Best practices for Integrating NIDS with Penetration Testing

- **Clear Objectives and Scope:** Ensure that both penetration testing and NIDS are aligned in terms of goals and scope. This ensures that penetration tests focus on relevant vulnerabilities and that NIDS are set up to detect the associated attack patterns.

- **Regularly Update and Calibrate NIDS:** Keep NIDS updated with the latest threat intelligence and patterns identified in penetration testing. This helps improve the accuracy of attack detection.

- **Automate Alerting and Reporting:** Integrate automated tools to streamline the alerting process, ensuring that penetration testers and security teams receive timely notifications about suspicious activity or successful exploits.

- **Continuous Monitoring and Testing:** Leverage continuous monitoring with NIDS while conducting regular penetration tests to ensure the security posture evolves with the changing threat landscape.

- **Comprehensive Post-Test Reporting:** Combine findings from both NIDS and penetration testing in detailed reports that cover both the discovered vulnerabilities and active exploitation attempts, with actionable remediation advice.

Scanned with OKEN Scanner

1.3 Key steps in NIDS-Based Penetration Testing Process

1.3.1 Step 1: Define Testing Objectives and Scope - Before beginning any testing activities, it is essential to define the objectives and scope of the penetration test, as well as the integration of NIDS. Some key considerations include:

- **Testing Scope:** Identify which systems, web applications, and network segments are part of the testing process. This may involve specific applications, databases, or infrastructure components.
- **Goals:** Determine the goals of the penetration test, such as identifying vulnerabilities in the application, testing for unauthorized access, or evaluating the effectiveness of existing security measures (including NIDS).
- **NIDS Focus Areas:** Identify areas where NIDS will focus its monitoring efforts, such as suspicious login attempts, access to sensitive data, or unusual outbound traffic indicative of a data exfiltration attempt.

1.3.2 Step 2: Perform Network Mapping and Vulnerability Assessment - Before executing the penetration test, the network infrastructure and web application must be mapped to identify potential vulnerabilities. This includes:

- **Network Discovery:** Use tools to map the network, identify active devices, services, and open ports. This can help identify points of potential exploitation.

- **Vulnerability Scanning:** Conduct a vulnerability scan to find weak spots, misconfigurations, and known vulnerabilities that could be exploited during the penetration test. Tools such as Nessus or OpenVAS may be used to automate this process.

- **Configuration of NIDS:** Configure the NIDS to monitor the identified network segments and vulnerable systems. NIDS should be set to watch for common attack patterns, such as SQL injections, buffer overflows, and DDoS attempts, based on the identified vulnerabilities.

Scanned with OKEN Scanner

1.3.3 Step 3: Execute Penetration Testing - With the NIDS configured and the scope defined, the next step is to execute the penetration test. This involves simulating real-world attacks on the web application, network, or system to discover potential vulnerabilities. Penetration testers may use a variety of techniques:

- **Reconnaissance and Information Gathering:** Gather information about the target, such as its domain names, IP addresses, and public-facing services.
- **Exploitation:** Attempt to exploit vulnerabilities identified during the vulnerability assessment. Common penetration testing techniques include SQL injection, cross-site scripting (XSS), command injection, and privilege escalation.
- **Lateral Movement:** After gaining initial access, attempt to move laterally within the network to escalate privileges or access more sensitive data.
- **Post-Exploitation:** Simulate the actions of an attacker who has compromised the system, including attempting to extract sensitive data or maintain persistence.

During this phase, the NIDS actively monitors the network traffic generated by the penetration testers' activities. The NIDS is designed to detect any malicious activity and provide real-time alerts, helping security teams understand how well the system can detect and respond to attacks.

1.3.4 Step 4: Monitor and Analyse NIDS Alerts - While the penetration test is ongoing, the NIDS continuously monitors the network for suspicious activity. Key tasks during this phase include:

- **NIDS Alert Generation:** As penetration testers attempt to exploit vulnerabilities, the NIDS should detect patterns indicative of attacks and generate alerts. This could include signs of SQL injection attempts, unusual login patterns, or attempts to exploit known vulnerabilities.
- **Alert Analysis:** The security team should analyse the NIDS alerts in real-time to evaluate their accuracy. This analysis will determine whether the NIDS correctly detected the penetration testers' simulated attack and if there are any false positives or missed detections.

Scanned with OKEN Scanner

- **Incident Response Simulation:** Based on the NIDS alerts, security teams should practice incident response. This may involve isolating affected systems, blocking malicious IP addresses or taking other corrective actions.

1.3.5 Step 5: Post-Test Analysis and Reporting - Once the penetration testing and NIDS monitoring are complete, the next step is to analyse the results and generate a comprehensive report. This includes:

- **Penetration Test Findings:** Document the vulnerabilities and weaknesses discovered during the test, including any exploited vulnerabilities, attack vectors, and compromised systems.
- **NIDS Performance Evaluation:** Evaluate the effectiveness of the NIDS in detecting the simulated attacks. This includes analysing the quality of the alerts (e.g., true positives vs. false positives) and the system's ability to detect sophisticated attack techniques.
- **Remediation Recommendations:** Provide recommendations to fix vulnerabilities identified in the penetration test and improve NIDS configuration to enhance threat detection.
- **Security Improvements:** Suggest improvements to the overall security posture, such as patching vulnerabilities, tightening access controls, or enhancing network monitoring.

1.3.6 Step 6: Retesting and Continuous Monitoring - After addressing the findings from the penetration test and optimizing the NIDS configuration, it's important to conduct a **retest** to verify that vulnerabilities have been resolved and that the NIDS can accurately detect any new threats. This process includes:

- **Retesting Vulnerabilities:** Test any vulnerabilities that were remediated during the previous phase to ensure they are no longer exploitable.
- **NIDS Tuning:** Refine NIDS detection rules based on the penetration test findings to reduce false positives and improve detection accuracy.
- **Ongoing Monitoring:** Continue using NIDS to monitor network traffic and detect any suspicious activities on an ongoing basis, ensuring the network remains secure.

Scanned with OKEN Scanner

CHAPTER-2

LITERATURE SURVEY

Table 2.1: Literature Survey

Paper Title	Year	Authors	Advantages	Limitations
A Systematic Literature Review on Penetration Testing in Networks: Future Research Directions	2023	1.Alhamed M, 2.Rahman MMH	1. Comprehensive review of past penetration testing methods. 2. Identifies future research opportunities. 3. Categorizes various approaches for network penetration testing. 4. Helps to define a roadmap for future cybersecurity research.	1. Limited focus on real-world applicability. 2. Lacks practical testing and empirical validation. 3. Doesn't provide direct solutions, just trends. 4. May be outdated as the field of penetration testing evolves rapidly.
Modular Deep Learning-Based Network Intrusion Detection Architecture for Real-World Cyber-Attack Simulation	2024	1.Vladimir Ciric, 2.Marija Milosevic, 3.Danijel Sokolovic, 4.Ivan Milentijevic.	1. Utilizes deep learning for real-time detection. 2. Designed for real-world cyber-attack scenarios. 3. Modular architecture for easier customization and scalability. 4. Provides high accuracy in identifying intrusions.	1. High computational cost for real-time processing. 2. Requires large datasets for training. 3. Model interpretability can be challenging. 4. Performance degradation under unseen attack patterns.

Scanned with OKEN Scanner

Paper Title	Year	Authors	Advantages	Limitations
Research Trends in Network-Based Intrusion Detection Systems: A Review	2021	1.S. Kumar, 2.S. Gupta, 3.S. Arora.	1. Comprehensive analysis of the state of the art in intrusion detection. 2. Covers various methodologies including machine learning and rule-based systems. 3. Highlights emerging trends in intrusion detection technologies. 4. Offers insights into the future of network security.	1. May lack deep technical insights for implementation. 2. General overview without much empirical data. 3. Broad scope makes it less focused on specific methods. 4. May not fully address modern security challenges.
Enhancing Web Application Security through Automated Penetration Testing with Multiple Vulnerability Scanners	2023	1.K. Abdulghaffar, 2.N. Elmrabi, 3.M. Yousefi.	1. Automates the process of penetration testing. 2. Uses multiple scanners to improve detection accuracy. 3. Reduces human effort and time in security testing. 4. Improves coverage of web application vulnerabilities.	1. Automated testing may miss complex vulnerabilities. 2. Dependency on third-party tools for scanner efficiency. 3. May result in false positives/negatives. 4. Limited applicability to dynamic or evolving web apps.

Scanned with OKEN Scanner

Paper Title	Year	Authors	Advantages	Limitations
Network Intrusion Detection System using Deep Learning	2021	1. Lirim Ashiku, 2. Cihan Dagli	1. High accuracy in detecting network intrusions. 2. Uses deep learning to adapt to new attack patterns. 3. Can handle large volumes of network traffic. 4. Reduces false alarms with advanced neural network techniques.	1. Requires large and high-quality labeled datasets. 2. High computational overhead for model training and inference. 3. Deep learning models can lack interpretability. 4. Performance may degrade with evolving network environments.

Scanned with OKEN Scanner

[1] The paper by Alhamed and Rahman (2023) offers a thorough analysis of the situation of penetration testing in network security today, examining the approaches, resources, and difficulties related to this important procedure. By examining several methods for evaluating security holes and vulnerabilities, the writers draw attention to the developing methods and tools that improve penetration testing's efficacy. Additionally, they stress the necessity of sophisticated tactics to address the increasing intricacy of cyberthreats. The review also points out gaps in the current literature and makes recommendations for future research directions to enhance penetration testing procedures, like using machine learning and artificial intelligence to provide more adaptive security testing.

[2] In their study, Ciric et al. (2024) provide a modular deep learning-based architecture for network intrusion detection that is well suited for simulating actual cyberattacks. The authors offer a novel framework that improves the identification of complex cyberthreats in network environments by utilizing deep learning techniques. Through the use of a modular architecture, the system can be tailored to different attack situations and network setups, guaranteeing more precise and effective threat identification. In addition to providing information on how deep learning can be applied to enhance response times and detection accuracy, the study emphasizes the significance of modeling actual cyberattacks in order to evaluate the efficacy of intrusion detection systems. The authors also talk about how this design might change in response to new cyberthreats.

[3] Kumar, Gupta, and Arora (2021) provide a thorough review of recent research trends in network-based intrusion detection systems (NIDS), examining the advancements, challenges, and future directions in the field. The paper explores various techniques employed in NIDS, including signature-based, anomaly-based, and hybrid approaches, highlighting their strengths and limitations in detecting network intrusions. The authors also discuss the growing role of machine learning and artificial intelligence in enhancing the accuracy and adaptability of these systems. Furthermore, the review identifies key research areas that require further exploration, such as the integration of advanced algorithms, the handling of large-scale network data, and the development of more robust and real-time intrusion detection models. The work underscores the need for ongoing innovation to tackle evolving cybersecurity threats.

Scanned with OKEN Scanner

[4] Ashiku and Dagli (2021) propose a network intrusion detection system (NIDS) based on deep learning techniques, aimed at improving the accuracy and efficiency of detecting cyber threats in network environments. The authors leverage deep learning models to automatically identify malicious patterns and anomalies in network traffic, offering a robust solution for detecting sophisticated attacks that may evade traditional signature-based detection methods. Their approach integrates various deep learning algorithms, emphasizing the advantages of using these models for real-time intrusion detection. The paper also discusses the challenges associated with implementing deep learning in NIDS, such as the need for large, labeled datasets and computational resources. The authors highlight the potential of deep learning to enhance the detection capabilities of NIDS, ultimately contributing to more effective cybersecurity defenses.

[5] A thorough analysis of current research trends in network-based intrusion detection systems (NIDS) is provided by Kumar, Gupta, and Arora (2021), who concentrate on the development of detection methods and the difficulties these systems encounter. The study examines many NIDS techniques, including hybrid, anomaly-based, and signature-based models, outlining the benefits and drawbacks of each in terms of detecting network intrusions. To increase detection efficiency and accuracy, particularly when addressing sophisticated and adaptable cyberthreats, the authors stress the growing integration of artificial intelligence and machine learning in NIDS. The evaluation also identifies significant research gaps, such as the requirement for real-time detection capabilities, more scalable solutions, and the creation of more robust systems that can manage high network traffic levels.

[6] Abdulghaffar, Elmrabit, and Yousefi (2023) explore methods to enhance web application security by utilizing automated penetration testing combined with multiple vulnerability scanners. The authors propose an integrated approach that leverages various scanning tools to identify security weaknesses across different layers of web applications. By automating the testing process, the system provides a more efficient and comprehensive analysis of potential vulnerabilities, improving the overall security posture of web applications. The paper highlights the benefits of using multiple scanners to detect a wider range of vulnerabilities, reducing the likelihood of undetected security flaws.

Scanned with OKEN Scanner

Additionally, the authors discuss the challenges of automating penetration testing, including managing false
positives and ensuring the accuracy of the scanning results. The study suggests that this multi-tool approach could significantly aid in mitigating risks associated with web application security breaches.

Penetration testing is a vital component of web application security testing, which is necessary for detecting weaknesses in digital systems. Through seamless integration into many testing frameworks, a NIDS (Network Intrusion Detection System)-based solution improves security testing by keeping an eye on network traffic for questionable activities.

Scanned with OKEN Scanner

CHAPTER-3

RESEARCH GAPS OF EXISTING METHODS

Although current approaches to web application security testing, such as conventional penetration testing and the incorporation of Network Intrusion Detection Systems (NIDS), provide valuable information about possible weaknesses, there are a number of research gaps that require more investigation and creativity. The constraints of existing methods, the dynamic character of threats, and technological developments that potentially improve security testing are usually the subjects of these gaps. By identifying these loopholes, web application security procedures can become more effective and future research can be guided.

3.1 Detection of Encrypted Traffic:

- Current Limitation: Modern web applications increasingly rely on encrypted communication (e.g., HTTPS) to protect data in transit. NIDS often struggles to inspect encrypted traffic, limiting its ability to detect threats like malware, data exfiltration, or man-in-the-middle attacks.

- Research Gap: Developing methods for inspecting encrypted traffic without compromising privacy and security is a major challenge. Techniques like advanced SSL/TLS inspection or leveraging machine learning to identify patterns in encrypted traffic could be further explored.

3.2 False Positives and False Negatives in NIDS Alerts:

- Current Limitation: NIDS are prone to generating false positives (alerting on harmless activities) and false negatives (failing to detect actual attacks). This compromises the accuracy and reliability of the security testing process.

- Research Gap: There is a need for improved algorithms and machine learning techniques that can reduce the occurrence of false positives and false negatives. Research into more adaptive NIDS systems that can learn from new attack patterns and network behavior is crucial.

Scanned with OKEN Scanner

3.3 Scalability of NIDS in Large-Scale Web Applications:

- *Current Limitation:* NIDS can experience performance degradation when deployed in large-scale networks with high traffic volume. This limits their ability to provide real-time monitoring for complex, high-traffic web applications.

- *Research Gap:* Developing scalable NIDS solutions capable of analyzing vast amounts of network traffic in real-time without sacrificing performance is an ongoing challenge. Distributed NIDS architectures or more efficient traffic analysis algorithms could improve scalability.

3.4 Detection of Zero-Day Attacks:

- *Current Limitation:* Existing penetration testing and NIDS-based approaches often struggle to detect zero-day vulnerabilities—previously unknown exploits that attackers can use to compromise an application.

- *Research Gap:* More research is needed in the development of proactive techniques such as anomaly-based detection, behavior analysis, and machine learning to detect previously unknown or unpatched vulnerabilities in real-time.

3.5 Automated Exploitation Detection:

- *Current Limitation:* While penetration testing often identifies vulnerabilities, the automated detection of exploits using NIDS is still an underdeveloped area. Many existing methods do not efficiently correlate vulnerabilities with actual exploitation attempts.

- *Research Gap:* The ability to automate the detection of exploit attempts using NIDS (e.g., correlating vulnerability scans with live attack traffic) could significantly improve the accuracy and speed of response during penetration tests.

Scanned with OKEN Scanner

3.6 User Behavior Analytics and Insider Threats:

- *Current Limitation:* NIDS primarily focuses on network traffic and external threats, often overlooking the risk posed by insider threats or anomalous user behavior within a web application.

- *Research Gap:* Researching and integrating User and Entity Behavior Analytics (UEBA) within NIDS could provide deeper insights into potential insider threats and unusual user activities that traditional intrusion detection might miss.

3.7 Lack of Contextual Awareness in NIDS:

- *Current Limitation:* Many NIDS tools lack contextual awareness, meaning they may miss attacks that appear legitimate in isolation but are suspicious when considered in the broader context of application behavior or business processes.

- *Research Gap:* Developing NIDS that can understand the context of web application traffic, such as the user roles, session data, or business workflows, could help identify more sophisticated attacks (e.g., business logic flaws, privilege escalation).

3.8 Real-time Threat Intelligence Integration:

- *Current Limitation:* While NIDS can detect attacks based on predefined signatures or patterns, they often lack the ability to dynamically integrate up-to-date threat intelligence to detect emerging attack methods.

- *Research Gap:* Research into integrating real-time threat intelligence feeds into NIDS, enabling more adaptive and context-aware security testing, could significantly improve detection capabilities and speed of response.

Scanned with OKEN Scanner

CHAPTER-4

PROPOSED MOTHODOLOGY

Here's a proposed methodology for conducting web application penetration testing using a network intrusion detection system (NIDS)-based approach. This methodology leverages NMAP, OWASP ZAP, Nessus to ensure a thorough evaluation of the web application's security posture.

4.1 Planning and Preparation:

-Define Scope: Identify the web application's boundaries, including subdomains, APIs, and third-party services.

-Gathering information - NMAP: Use NMAP to perform reconnaissance on the target web application. This includes scanning for open ports, services, and potential vulnerabilities.

Command example: **map -sS -p- -A <target_ip_or_domain>**

-Understand the Architecture: Review application architecture and identify components such as servers, databases, and other integrated services.

4.2 NIDS Configuration and Monitoring Setup:

-Set Up NIDS: Configure a NIDS (such as Nmap or Suricata) to monitor network traffic during the penetration test.

-Traffic Logging: Ensure that all network traffic during the testing period is logged for later analysis.

4.3 Vulnerability Scanning:

-Use Nessus: Conduct a comprehensive vulnerability scan using Nessus to identify known vulnerabilities in the web application and its components.

Scanned with OKEN Scanner

4.4 Automated Security Testing:

- OWASP ZAP: Launch OWASP ZAP to perform automated security testing on the web application.
- Use the Active Scan feature to find common vulnerabilities such as SQL injection, cross-site scripting (XSS), and more.
- Set ZAP to run in conjunction with the NIDS to capture all the network traffic generated during the scan.
- Generate a report of the vulnerabilities detected by ZAP for further analysis.

4.5 Manual Testing:

- Conduct Manual Tests: After automated testing, perform manual testing to identify vulnerabilities that automated tools may have missed. Focus on:
  - Business logic flaws
  - Session management issues
  - Insecure direct object references
  - Use ZAP's Manual Explore mode for additional insights.

4.6 post-exploitation:

- Assess Impact: Determine the potential impact of successfully exploiting vulnerabilities, such as data leakage, privilege escalation, or complete system compromise.
- Data Collection: Collect any sensitive data obtained during exploitation and evaluate the security controls in place.

4.7 Reporting:

- Compile Findings: Document all findings from automated and manual tests, including vulnerabilities, exploitation attempts, and evidence.
- Use tools like Nessus and ZAP to generate detailed reports.
- Provide Recommendations:
  - Offer actionable recommendations to mitigate identified vulnerabilities.

Scanned with OKEN Scanner

CHAPTER-5
OBJECTIVES

5.1. Identify Vulnerabilities:

The first step in the project involves identifying vulnerabilities within the web application under testing. This can be achieved through a variety of techniques, such as:

- **Manual Testing**: Checking for common vulnerabilities like SQL injection, Cross-Site Scripting (XSS), Cross-Site Request Forgery (CSRF), and other web-related attacks.
- **Automated Scanning Tools**: Using vulnerability scanners (like OWASP ZAP, Burp Suite, or Nikto) to detect known security flaws.
- **Code Review**: Analyzing the source code for weak points such as improper input validation, insecure session handling, and inadequate encryption.

Role of NIDS: A Network Intrusion Detection System (NIDS) can be used to passively monitor network traffic and identify malicious activities such as attempts to exploit vulnerabilities. By correlating signatures and anomalous behaviour, NIDS can detect exploitation attempts in real-time.

5.2. Test Detection Capabilities:

Once vulnerabilities are identified, the next step is to test the detection capabilities of the NIDS deployed in the environment. This involves simulating various attack vectors against the web application and observing how well the NIDS detects them. Testing can be done in the following ways:

- **Penetration Testing**: Conducting penetration tests, either manually or with automated tools, to simulate various attacks (e.g., brute-force attacks, XSS, SQL injection) and check if the NIDS can detect these attacks in real-time.
- **Evasion Techniques**: Attempting to bypass the NIDS through techniques like encryption, fragmentation, or tunnelling to evaluate the effectiveness of the NIDS detection and the robustness of the system.

Scanned with OKEN Scanner

- **False Positive and Negative Testing**: Ensuring the NIDS can effectively minimize false positives (incorrectly flagging benign activity as malicious) and false negatives (failing to detect actual attacks).

5.3. Analyze Network Traffic:

During penetration testing, it's crucial to analyze the network traffic between the client (user) and the web server. This involves capturing and inspecting the packets using tools like Wireshark, tcpdump, or Suricata (an open-source NIDS). The analysis focuses on:

- **Normal Traffic**: Understanding the expected patterns of communication (e.g., HTTP requests and responses) to set a baseline.
- **Suspicious Traffic**: Identifying unusual patterns or anomalies in traffic that could indicate attacks (e.g., abnormal request rates, unusual payloads, SQL injection patterns, etc.).
- **Traffic Correlation**: Correlating network traffic with attack vectors to identify if an exploit attempt is successful or if an attack is being blocked by the NIDS.

The NIDS captures and logs network traffic, which is essential for identifying signs of exploitation, lateral movement, or data exfiltration. By analysing this data, you can gain valuable insights into the attack patterns and the effectiveness of your detection mechanisms.

5.4. Evaluate Incident Response:

After simulating an attack or detecting potential malicious activity via the NIDS, it's important to evaluate the incident response protocols. This step ensures that the security team can effectively respond to attacks by:

- **Monitoring Alerts**: Ensuring the security team is receiving proper alerts from the NIDS when an attack occurs.
- **Response Time**: Testing how quickly the security team can respond to alerts, take action (e.g., blocking malicious IP addresses), and mitigate the attack

Scanned with OKEN Scanner

- **Escalation Procedures**: Evaluating if the incident escalation procedures are in place and whether the response is appropriate (e.g., notifying relevant stakeholders, activating response teams).
- **Forensics**: Analyzing the attack post-mortem to understand the attack's origin, method, and impact.

NIDS logs and alerts help incident responders to determine whether the attack was successful, which vulnerabilities were exploited, and what measures should be taken to contain and remediate the attack.

5.5. Continuous Improvement:

Web application security and NIDS detection are not one-time efforts; they require continuous improvement. This phase involves:

- **Feedback Loop**: Using the lessons learned from penetration tests, attack simulations, and incident response evaluations to improve the security posture of the application and NIDS configuration.
- **Updating Signatures and Rules**: NIDS signatures need to be frequently updated to account for new attack techniques and vulnerabilities. This ensures that the NIDS remains effective in detecting new threats.
- **Tuning**: Regularly tuning the NIDS to reduce false positives and improve detection accuracy. This might involve adjusting sensitivity levels or modifying detection rules.
- **Training**: Continuously training the security team on new attack trends and response strategies to enhance their ability to recognize and react to emerging threats.

This continuous improvement process ensures that the security testing is aligned with evolving threats, application updates, and NIDS capabilities.

Scanned with OKEN Scanner

5.6. Documentation and Reporting:

Proper documentation and reporting are critical throughout the testing process to ensure transparency, provide actionable insights, and guide future improvements. Key components of documentation include:

- **Test Plan**: Detailing the testing methodology, tools, attack vectors, and expected outcomes.
- **Vulnerability Assessment**: Documenting identified vulnerabilities and their associated risk levels (e.g., high, medium, low), as well as any recommendations for mitigation.
- **Detection Evaluation**: Recording the findings from testing the NIDS's detection capabilities, including any false positives, false negatives, and successful attack detection.
- **Incident Response Evaluation**: Providing detailed logs and reports on incident response effectiveness, including response time, actions taken, and lessons learned.
- **Recommendations for Improvement**: Offering suggestions for improving the security posture of both the web application and the NIDS, such as applying patches, enhancing detection rules, or refining incident response procedures.

Finally, a **final report** summarizing the findings, including vulnerabilities found, how well the NIDS performed, response times, and improvements needed, should be delivered to stakeholders. This report can then serve as a foundation for addressing gaps in security and enhancing future testing cycles.

Scanned with OKEN Scanner

CHAPTER-6
SYSTEM DESIGN & IMPLEMENTATION

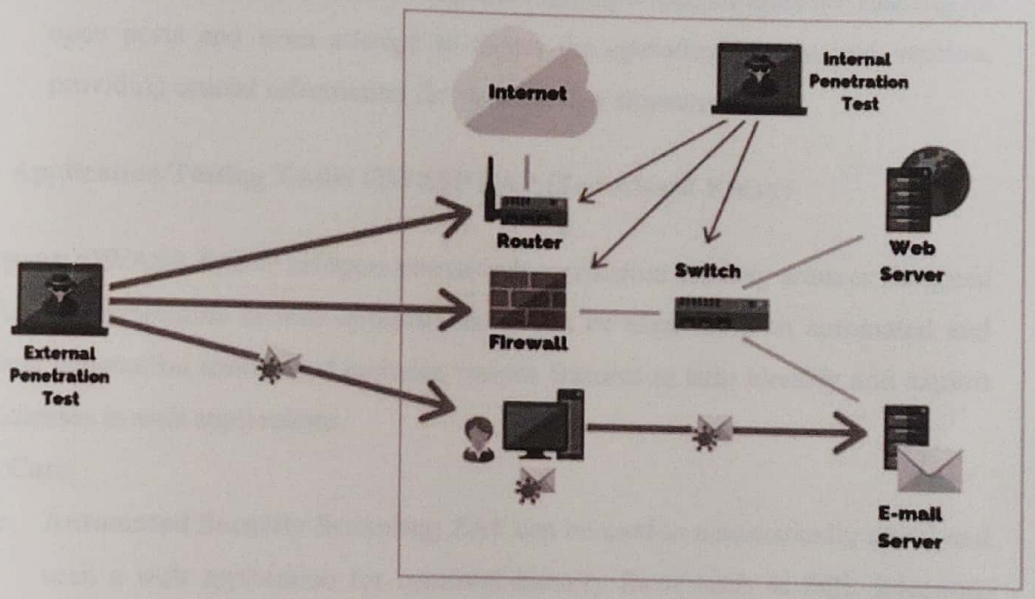


Figure 6: Network Penetration Testing

6.1 Software Requirements:

6.1.1. Network Intrusion Detection System (NIDS) Tools: Nmap

- **Purpose**: Nmap (Network Mapper) is an open-source tool used for network discovery and security auditing. It helps in identifying devices and services running on a network by sending network packets and analyzing responses. It can detect live hosts, open ports, and services that might be vulnerable to attacks.

Scanned with OKEN Scanner

- **Use Case**:
  - **Network Discovery**: Nmap can perform host discovery and map out the devices on a network.
  - **Port Scanning**: Identify open ports on remote systems, which helps in identifying potential attack vectors.
  - **Service and OS Detection**: Nmap can identify which services are running on open ports and even attempt to detect the operating system and version, providing crucial information for vulnerability assessments.

6.1.2. Web Application Testing Tools: OWASP ZAP (Zed Attack Proxy)

- **Purpose**: OWASP ZAP is an open-source web application security scanner designed to find vulnerabilities in web applications. It can be used for both automated and manual penetration testing and includes various features to help identify and exploit weaknesses in web applications.
- **Use Case**:
  - **Automated Security Scanning**: ZAP can be used to automatically crawl and scan a web application for common security flaws such as SQL injection, cross-site scripting (XSS), and security misconfigurations.
  - **Manual Testing**: Security professionals can manually interact with the application using ZAP's proxy capabilities to analyze requests and responses, potentially identifying complex vulnerabilities that automated scanners may miss.
  - **Session Management**: ZAP provides robust session management, allowing testers to track authentication and session states during testing.

6.1.3. Vulnerability Scanners: Nessus

- **Purpose**: Nessus is one of the most widely used vulnerability scanners in the cybersecurity industry. It helps security teams identify known vulnerabilities in their networks, systems, and applications. Nessus scans for vulnerabilities in both the network infrastructure and software applications.

Scanned with OKEN Scanner

- **Use Case**:
  - **Vulnerability Scanning**: Nessus scans a system or network to find potential vulnerabilities, such as missing patches, misconfigurations, and outdated software versions that could be exploited by attackers.
  - **Compliance Auditing**: Nessus can check for compliance with security standards and frameworks such as PCI DSS, HIPAA, and CIS benchmarks.

Scanned with OKEN Scanner

CHAPTER-7
TIMELINE FOR EXECUTION OF PROJECT (GANTT CHART)

7.1 Review 0

- **Title Finalization with Supervisor**: Meet with your project supervisor to discuss and finalize the project's title. This step ensures that you and your supervisor are aligned on the project's focus.
- **Literature Survey**: Start researching relevant literature in your field. This involves reading academic papers, articles, and books related to your topic. The literature survey helps you understand the existing research in the area.
- **Finalizing Objectives**: Define the specific objectives and goals of your project.

  What do you aim to achieve? What problems are you trying to solve?
- **Deciding the Methodology**: Determine the research methodology you'll use in your project, whether it's experimental, survey-based, analytical, or a combination of methods.

7.2 Review 1

- **Title**: After finalizing with your supervisor, formally decide on and document the project title.
- **Abstract:** Write the project abstract, a concise summary of your project's purpose, methodology, and expected outcomes.
- **Literature Survey**: Continue your literature survey and ensure that you've referred to at least one research paper in your work.
- **Identify**: The advantages and the disadvantages of our project.

Scanned with OKEN Scanner

- **Software and Framework Details**: List the software framework tools or platforms you plan to use.
- **References**: Site all the sources you've used in your literature survey.

7.3 Review 2

- **50% Implementation Details with Live Demo:** Implement at least half of your project and present a live demonstration.
- **50% Report Softcopy:** Submit a softcopy of the project report with 50% completion.
- **50% Journal Paper:** Showed paper to the supervisor for further guidance

7.4 Review 3

- **75% Report Softcopy**: Submit a softcopy of the project report with 75% completion.
- **Journal paper completion**: Paper completed and submitted for publication.

7.5 Final Review

- **Final Report and Submission of Project**: Complete your project, finalize the report, and submit the final project and report to your supervisor or institution.
- Complete implementation details with live demo.
- Submitting the final report to the reviewer.

Scanned with OKEN Scanner

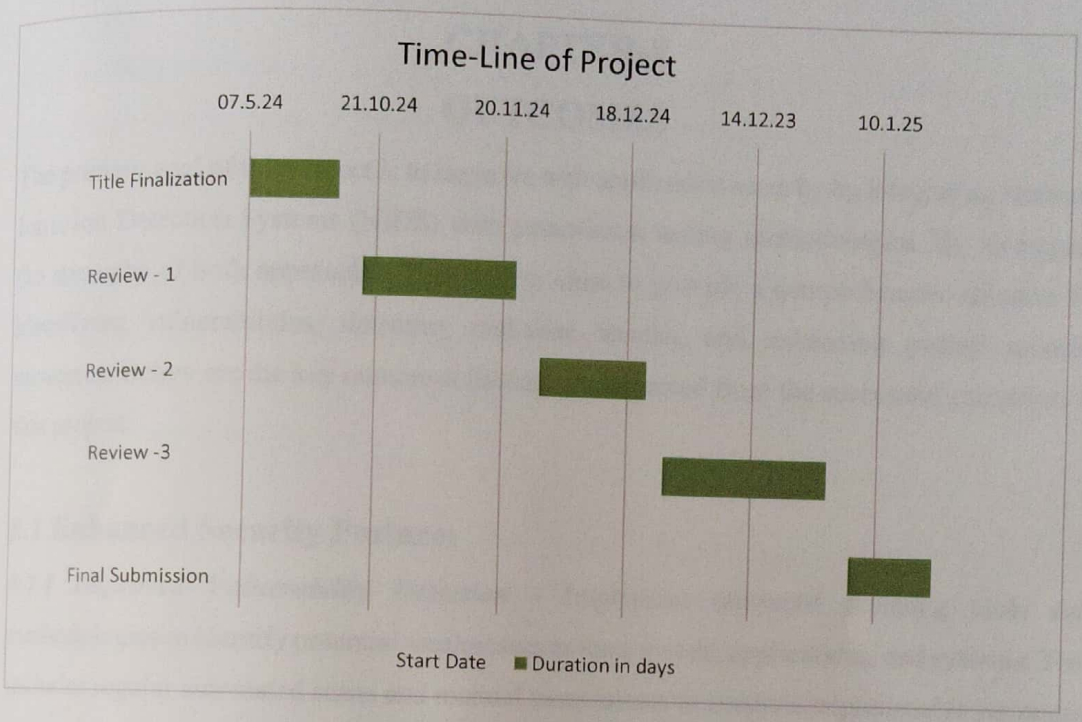


Figure 7.5 Gnatt Chart

Scanned with OKEN Scanner

CHAPTER-8
OUTCOMES

The primary goal of this project is to improve web application security by integrating Network Intrusion Detection Systems (NIDS) with penetration testing methodologies. By leveraging the strengths of both approaches, this project aims to provide a comprehensive solution for identifying vulnerabilities, detecting real-time attacks, and enhancing overall security measures. Below are the key outcomes that can be expected from the successful execution of this project:

8.1 Enhanced Security Posture:

8.1.1 Improved Vulnerability Detection - Implement advanced scanning tools and methodologies to identify potential weaknesses in the network, applications, and systems. This includes regular automated scans and manual assessments to ensure comprehensive coverage.

8.1.2 Early Detection of Threats - Utilize machine learning and behavioral analytics to detect anomalies in network traffic that may indicate a potential security breach. Early threat detection mechanisms can significantly reduce the impact of attacks.

8.1.3 Real-Time Alerts - Establish a centralized alert system that provides immediate notifications to security teams regarding suspicious activities or detected threats, ensuring rapid awareness and response.

8.2 Strengthened Incident Response:

8.2.1 Improved Response Times -Develop and implement incident response plans that streamline communication and actions among team members, reducing the time taken to react to incidents.

8.2.2 Actionable Incident Data - Ensure that incident reports provide comprehensive insights, including the nature of the threat, affected systems, and potential impact, to facilitate informed decision-making during an incident.

Scanned with OKEN Scanner

8.2.3 Automated Response - Implement automation tools that can take predefined actions in response to specific types of incidents, such as isolating compromised systems or blocking malicious traffic, to enhance response efficiency.

8.3 Optimized NIDS Configuration and Performance:

8.3.1 Refined Detection Rules - Regularly review and update detection rules to ensure they are aligned with the latest threat intelligence and are effective in identifying relevant threats without overwhelming the system.

8.3.2 Reduced False Positives and Negatives - Use advanced tuning techniques and feedback from security analysts to minimize false positives (legitimate traffic flagged as threats) and false negatives (actual threats not detected), improving the overall effectiveness of the NIDS.

8.4 Comprehensive Vulnerability Management:

8.4.1 Prioritized Vulnerabilities - Establish a risk-based approach to vulnerability management that categorizes vulnerabilities according to their potential impact and exploitability, allowing for targeted remediation efforts.

8.4.2 Continuous Vulnerability Monitoring - Implement ongoing monitoring processes to identify new vulnerabilities as they emerge and ensure timely updates to systems and applications to mitigate risks.

8.5 Documentation and Reporting:

8.5.1 Detailed Penetration Testing Reports - Conduct regular penetration tests and document findings in detail, including vulnerabilities discovered, exploited paths, and the effectiveness of existing security measures.

8.5.2 Recommendations for Future Security Improvements - Provide actionable recommendations based on the findings from assessments and incident analyses, outlining specific measures to strengthen security posture and enhance incident response capabilities.

Scanned with OKEN Scanner

CHAPTER-9
RESULTS AND DISCUSSIONS

To create a more thorough security framework, the NIDS-Based Penetration Testing Approach combines the advantages of penetration testing and Network Intrusion Detection Systems (NIDS). The success of this strategy, enhancements to security posture, and difficulties faced during deployment can all be examined.

Many challenges arose during the installation of Metasploit for this project, which might hinder its seamless integration in specific circumstances. One significant issue was the installation process failing, which was partly brought on by interference from antivirus software. Several Metasploit components were flagged by the system as potentially dangerous, resulting in incomplete installations or failed installation attempts. Furthermore, in settings with strict security procedures, the Metasploit framework's dependence on particular system configurations and security permissions may lead to further issues. Because of this, even when the program is compatible with the operating system, users may still have issues.

It is recommended to either temporarily stop antivirus software or configure it to allow uninterrupted operation of Metasploit components to minimize these issues in future installs. However, turning off antivirus software could endanger the system, thus it could be better to use other tactics, such as utilizing Metasploit in a containerized environment or virtual machine. This technique not only isolates Metasploit from the host system but also provides a more controlled and secure environment for fixing installation issues without endangering system security. Furthermore, future Metasploit endeavors can be made more efficient by leveraging cloud-based technologies or developing specific testing environments.

Scanned with OKEN Scanner

CHAPTER-10

CONCLUSION

When conducting online application penetration testing, the NIDS approach offers a more comprehensive method for finding and identifying security vulnerabilities. Popular web application dangers, the expansion of penetration testing procedures, and the function of NIDS in strengthening safety checks have all been discussed in this literature review. Penetration testing combined with NIDS deployment offers strong defense against modern threats.

We have stressed the importance of strong security testing procedures throughout this project, with a special emphasis on how combining NIDS with penetration testing provides a thorough and flexible method of detecting and addressing possible security threats instantly. Defensive mechanisms (NIDS) and offensive tactics (penetration testing) can help businesses better understand vulnerabilities that can be difficult to find with standard testing.

A multi-layered strategy combining offensive and defensive tactics is needed to ensure the security of web applications. Penetration testing based on NIDS is an effective technique that improves the capacity to find and fix vulnerabilities in network traffic and application code. Businesses may improve their entire cybersecurity posture, proactively defend against changing threats, and protect critical data and business activities by utilizing the advantages of both approaches.

Scanned with OKEN Scanner

REFERENCES

[1] Alhamed M, Rahman MMH. A Systematic Literature Review on Penetration Testing in Networks: Future Research Directions. Applied Sciences. 6986; 13, December 2023.

[2] Altulaihan EA, Alismail A, Frikha M. A Survey on Web Application Penetration Testing. Electronics. 1229; 12, May 2023.

[3] Vladimir Ciric, Marija Milosevic, Danijel Sokolovic, Ivan Milentijevic, Modular deep learning-based network intrusion detection architecture for real-world cyber-attack simulation, Simulation Modelling Practice and Theory, Volume 102916, ISSN 1569-190X; 13, March 2024.

[4] Saputra FA, Salman M, Hasim JAN, Nadhori IU, Ramli K. The Next-Generation NIDS Platform: Cloud-Based Snort NIDS Using Containers and Big Data. Big Data and Cognitive Computing, 19; 6, January 2022.

[5] S. Kumar, S. Gupta and S. Arora, "Research Trends in Network-Based Intrusion Detection Systems: A Review," in IEEE Access, vol. 9, pp. 157761-157779, 17, July 2021.

[6] Heather Lawrence, Uchenna Ezeobi, Orly Tauil, Jacob Nosal, Owen Redwood, Yanyan Zhuang, Gedare Bloom, 8890; 25, August 2024.

CUPID: A labeled dataset with Pentesting for evaluation of network intrusion detection, Journal of Systems Architecture, Volume 129, 102621,ISSN13837621; 2022.

[7] Kumar, Satish & Gupta, Sunanda & Arora, Sakshi. (2021). Research Trends in Network-Based Intrusion Detection Systems: A Review. IEEE Access. PP. 1-1; 2021.

[8] Lirim Ashiku, Cihan Dagli, Network Intrusion Detection System using Deep Learning, Procedia Computer Science, Volume 185, Pages 239247, ISSN 1877-0509; 2021.

[9] Vanin P, Newe T, Dhirani LL, O'Connell E, O'Shea D, Lee B, Rao M. A Study of Network Intrusion Detection Systems Using Artificial Intelligence/Machine Learning. Applied Sciences. 11752; 22, December 2022.

[10] Vegesna, Vinod Varma, Utilising VAPT Technologies (Vulnerability Assessment & Penetration Testing) as a Method for Actively Preventing Cyberattacks (October 25, 2023). International Journal of Management, Technology And Engineering | Volume XII, July 2022.

Scanned with OKEN Scanner

[11] Ke Chen, Man Zhang, Rufeng Liang, Junhan Chen, Jin Peng, Xun Huang, Research on the Application of Penetration Testing Frameworks in Blockchain Security, Computational and Experimental Simulations in Engineering. 7768; 20, May 2018.

[12] Fredrik Heiding, Emre Süren, Johannes Olegård, Robert Lagerström, Penetration testing of connected households,Computers & Security,Volume 126, ,103067,ISSN 0167-4048; 2023.

[13] Altulaihan, E.A.; Alismail, A.; Frikha, M. A Survey on Web Application Penetration Testing. Electronics. 12, 1229; 2023.

[14] A. Goutam and V. Tiwari, "Vulnerability Assessment and Penetration Testing to Enhance the Security of Web Application," 2019 4th International Conference on Information Systems and Computer Networks (ISCON), Mathura, India, 2019, pp. 601-605, doi: 10.1109/ISCON47742.2019.9036175; 14, September 2019.

[15] P. Vats, M. Mandot and A. Gosain, "A Comprehensive Literature Review of Penetration Testing & Its Applications," 2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), Noida, India, 2020, pp. 674-680, doi: 10.1109/ICRITO48877.9197961; 2020.

[16] Abdulghaffar, K.; Elmrabit, N.; Yousefi, M. Enhancing Web Application Security through Automated Penetration Testing with Multiple Vulnerability Scanners. Computers 235; December 2023.

[17] Soroush M. Sohi, Jean-Pierre Seifert, Fatemeh Ganji, RNNIDS: Enhancing network intrusion detection systems through deep learning,Computers & Security,Volume 102,102151,ISSN 0167-4048; 2021.

[18] Ms Khushnaseeb Roshan, Aasim Zafar, Boosting robustness of network intrusion detection systems: A novel two phase defense strategy against untargeted white-box optimization adversarial attack, Expert Systems with Applications,Volume 249, Part A,123567,ISSN 09574174; 2024.

[19] Murat Aydos, Çiğdem Aldan, Evren Coşkun, Alperen Soydan,Security testing of web applications: A systematic mapping of the literature,Journal of King Saud University - Computer and Information Sciences,Volume 34, Issue 9, Pages 6775-6792,ISSN 13191578; 23-45, July 2022.

Scanned with OKEN Scanner

[20] Tadhani, J.R., Vekariya, V., Sorathiya, V. et al. Securing web applications against XSS and SQLi attacks using a novel deep learning approach. Sci Rep 14, 1803; 2024.

[21] Aljebry, A.F., Alqahtani, Y.M., Sulaiman, N. (2022). Analyzing Security Testing Tools for Web Applications. In: Khanna, A., Gupta, D., Bhattacharyya, S., Hassanien, A.E., Anand, S., Jaiswal, A. (eds) International Conference on Innovative Computing and Communications. Advances in Intelligent Systems and Computing, vol 1387. Springer, Singapore. 3411; 16, February 2023.

[22] Rajić, B., Stanisavljević, Ž. & Vuletić, P. Early web application attack detection using network traffic analysis. Int. J. Inf. Secur. 22, 77–91; 22, November 2023.

[23] Ayyagari, M.R., Kesswani, N., Kumar, M. et al. Intrusion detection techniques in network environment: a systematic review. Wireless Netw 27, 1269–1285; 15, April 2021.

[24] Ali, M., Haque, Mu., Durad, M.H. et al. Effective network intrusion detection using stacking-based ensemble approach. Int. J. Inf. Secur. 22, 1781–1798; 10, June 2023.

[25] Chowdhary, A.; Jha, K.; Zhao, M. Generative Adversarial Network (GAN)-Based Autonomous Penetration Testing for Web Applications. Sensors 8014; 23, May 2023.

Scanned with OKEN Scanner

APPENDIX-A

SCREENSHOTS

NESSUS

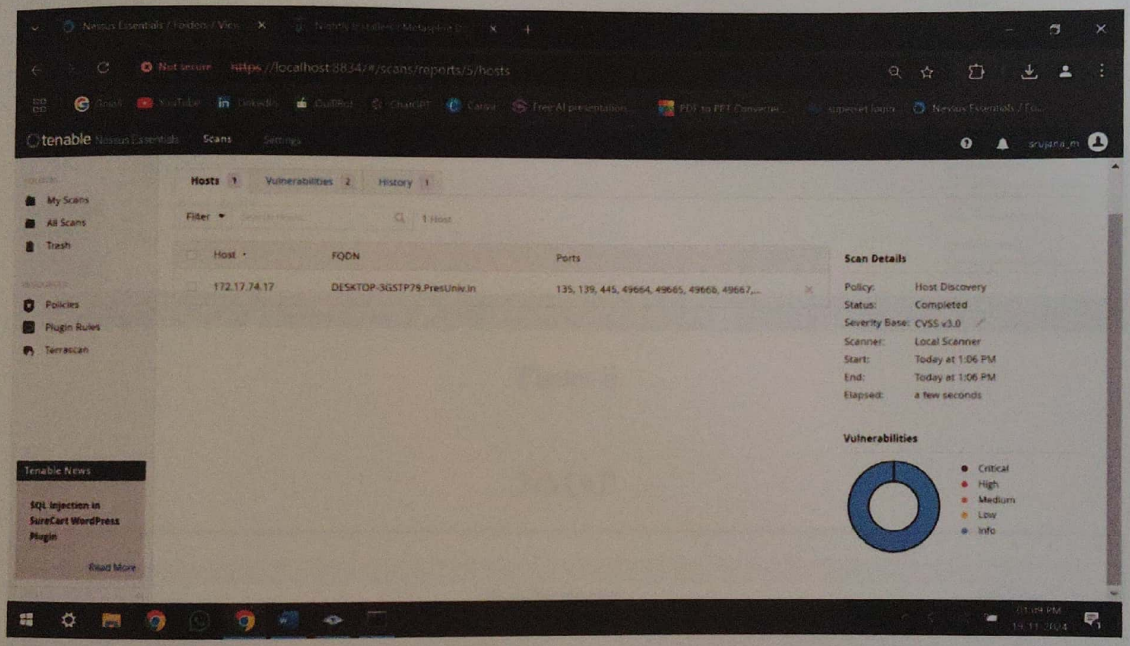


Figure i

Scanned with OKEN Scanner

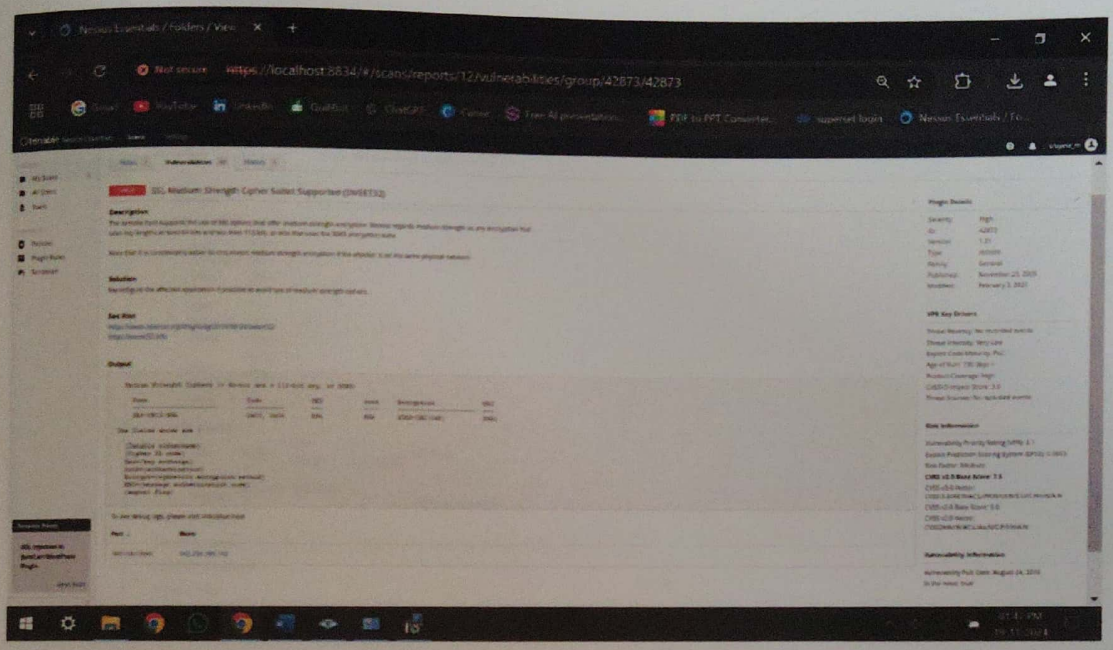


Figure ii

NMAP

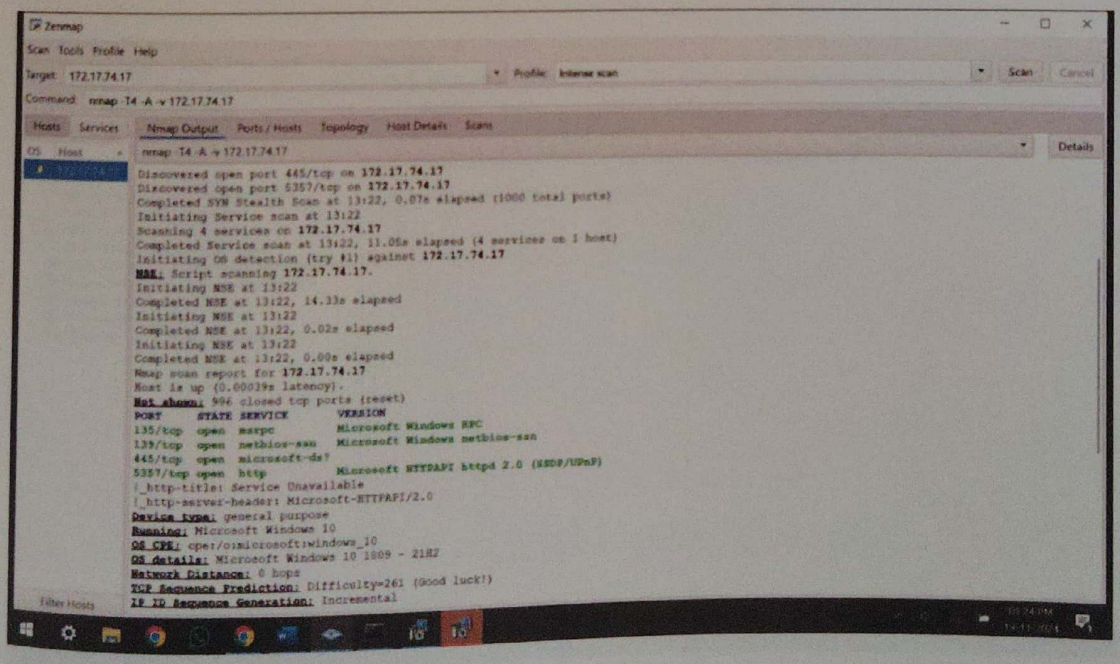


Figure iii

Scanned with OKEN Scanner

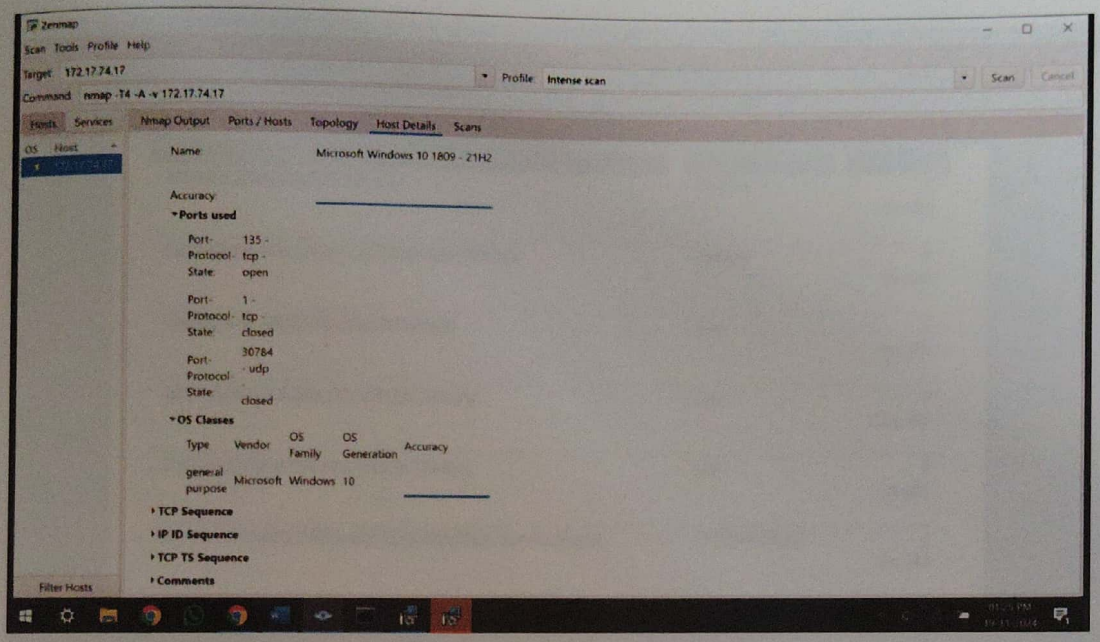


Figure iv

OWSAP ZAP

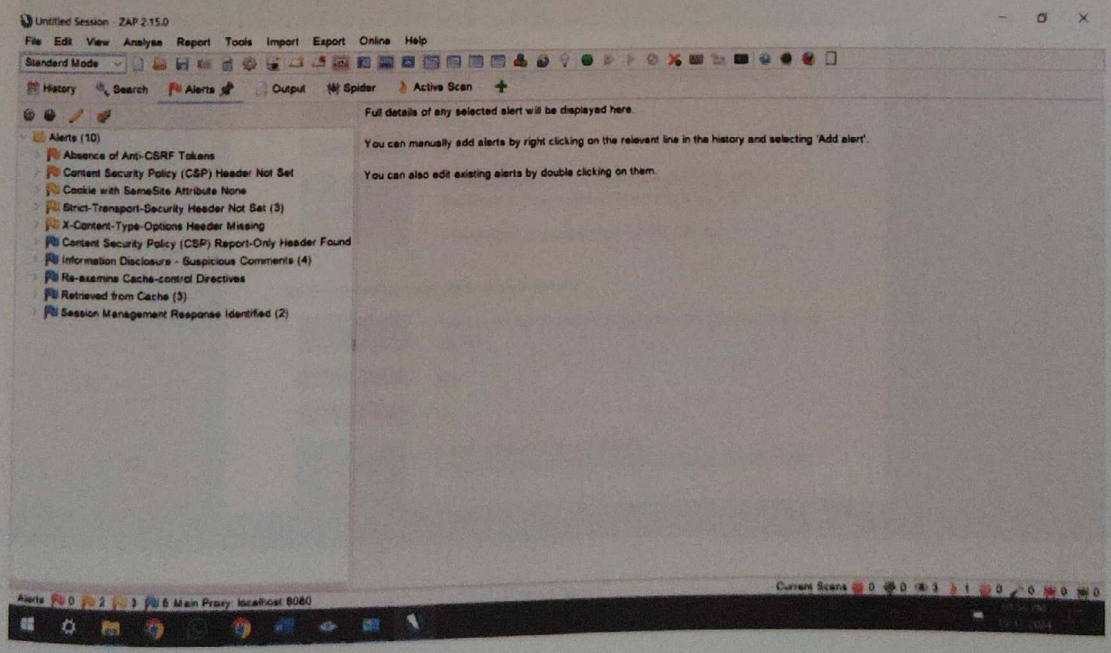
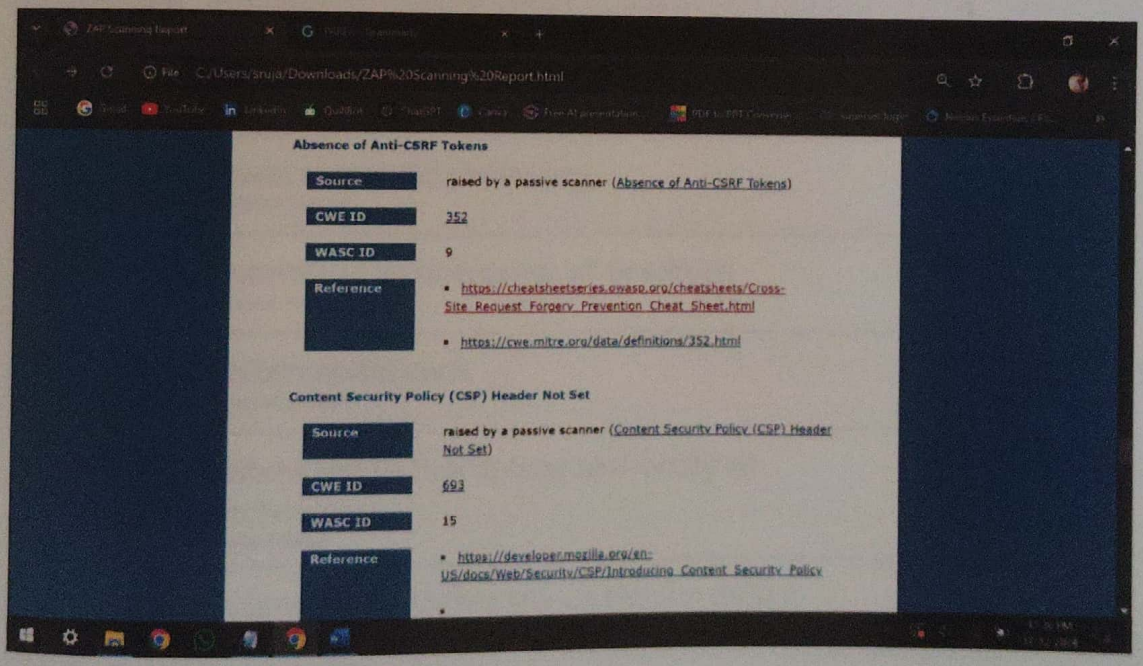


Figure v

Scanned with OKEN Scanner

Alert type	Risk	Count
Absence of Anti-CSRF Tokens	Medium	1 (9.1%)
Content Security Policy (CSP) Header Not Set	Medium	1 (9.1%)
Cookie with SameSite Attribute None	Low	1 (9.1%)
Strict-Transport-Security Header Not Set	Low	3 (27.3%)
X-Content-Type-Options Header Missing	Low	1 (9.1%)
Content Security Policy (CSP) Report-Only Header Found	Informational	1 (9.1%)
Information Disclosure - Suspicious Comments	Informational	4

Figure vi



Absence of Anti-CSRF Tokens

Source	raised by a passive scanner (Absence of Anti-CSRF Tokens)
CWE ID	352
WASC ID	9
Reference	• https://cheatsheetseries.owasp.org/cheatsheets/Cross-Site_Request_Forgery_Prevention_Cheat_Sheet.html • https://cwe.mitre.org/data/definitions/352.html

Content Security Policy (CSP) Header Not Set

Source	raised by a passive scanner (Content Security Policy (CSP) Header Not Set)
CWE ID	693
WASC ID	15
Reference	• https://developer.mozilla.org/en-US/docs/Web/Security/CSP/Introducing_Content_Security_Policy

Figure vii

Scanned with OKEN Scanner

APPENDIX-B

ENCLOSURES

REPORT PLAGARISM

SRUJANA MANJUNATH - CCS-G07(2)

ORIGINALITY REPORT

12%	8%	5%	5%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	eitca.org Internet Source	2%
2	Puthiyavan Udayakumar. "Design and Deploy a Secure Azure Environment", Springer Science and Business Media LLC, 2023 Publication	1%
3	www.semanticscholar.org Internet Source	1%
4	Submitted to University of Bradford Student Paper	1%
5	fastercapital.com Internet Source	<1%
6	Submitted to Swinburne University of Technology Student Paper	<1%
7	Submitted to University of Hertfordshire Student Paper	<1%
8	www.researchgate.net Internet Source	<1%

Scanned with OKEN Scanner

SDG – SUSTAINABLE DEVELOPMENT GOALS

The project work carried out here is named to SDG-9: Industry, Innovation, and Infrastructure

Your project focuses on improving the security of web applications using Network Intrusion Detection Systems (NIDS)-based penetration testing, which directly supports the development of secure and resilient digital infrastructures. By addressing web security vulnerabilities, you contribute to the creation of trustworthy and sustainable technological systems, which is a key aspect of SDG 9. Additionally, the research and innovation aspect of your project aligns with the goal of enhancing technological capabilities in industries and improving overall infrastructure resilience.

Scanned with OKEN Scanner

SSRG International Journal of Computer Science and Engineering
ISSN: 2348-8387/ https://doi.org/10.14445/23488387/IJCSE-VXXXX

Volume X Issue X, X-X, Month 2025
© 2025 Seventh Sense Research Group®

Original Article

A Comprehensive NIDS-Based Strategy for Web Application Penetration Testing

Srujana Manjunath[1], Shreya Malshetty[2], Jayalakshmi D[3], Chaithra Banger[4], Sharmasth Vali Y[5]

[1]*Computer Science, Cyber Security, Presidency University, Karnataka, India.*
(srujana.20211ccs0116@presidencyuniversity.in)
[2]*Computer Science, Cyber Security, Presidency University, Karnataka, India.*
[3]*Computer Science, Cyber Security, Presidency University, Karnataka, India.*
[4]*Computer Science, Cyber Security, Presidency University, Karnataka, India.*
[5]*Associate Professor, Presidency University, Karnataka, India.*
(sharmasth.vali@presidencyuniversity.in)

Received: Revised: Accepted: Published:

Abstract *- Imagine receiving an email from cybercriminals stating that all your personal information has been compromised—name, date of birth, home address, and finances. They are demanding money from you in exchange for not leaking your sensitive information. It's a terrifying situation to be in, isn't it? We adopt an NIDS-based approach for web application penetration testing in order to resolve this issue. Web application penetration testing are ongoing security evaluations that mimic actual attacks to evaluate how secure web applications are. The main objective is to find any possible flaws, configuration errors, or vulnerabilities that malicious users can use to jeopardize a web application's availability, confidentiality, or integrity. A NIDS is deployed to detect the fraudulent activities at the network level which can complement conventional penetration testing techniques that concentrate on flaws at the software. The goal of this research is to improve the identification of security vulnerabilities at the network and application levels by combining traditional web application penetration testing with Network Intrusion Detection Systems (NIDS).*

Keywords *- Intrusion Detection, Network Intrusion Detection, Penetration Testing, Vulnerability Assessment, Web Application Security*

1. Introduction

In today's internet era, networks are essential to every company that conducts virtual operations. Nevertheless, networks' interconnectedness also makes them susceptible to cyberattacks. Using network penetration testing is one method of guaranteeing network security. It includes mimicking actual network assaults to identify and address any weaknesses. Cybercriminals find web applications (such as social networking, cloud storage, e-commerce, and financial transactions) to be appealing because to their growing capability and complexity. Web application flaws can result in serious security lapses, data theft, and harm to one's reputation. However, regular network testing is very important to keep the network safe from constantly evolving cyber threats. As a result, it is crucial to make sure that these applications are secure, and penetration testing has emerged as a key tool for locating and fixing possible flaws before malicious users can take advantage of them.

Pen testing or penetration testing in networks mimic the attacks carried out by actual hackers or malicious users. This aids in identifying risks and vulnerabilities in the network's security measures. Wireless connections are intricate and compromise a mix of WAN, LAN, and Wi-Fi. Countless endpoints and devices which include PCs, servers, Internet of Things (IOT) are incorporated in the network as well. Additionally, they encompass Intrusion Prevention Systems (IPS) and firewalls. A few of these or in most cases all of these components might have a flaw that hackers could benefit to gain access unauthorizedally. The goal of network pen testing is to ensure the security of the network system and identify the loopholes before any attack arises. Therefore, a holistic approach to securing the web application ought to examine the network system alongside testing the software tool itself.

Blending Network Intrusion Detection Systems (NIDS) into penetration testing or pen testing technique is an appealing approach to tackle this challenge. NIDS are tools that keep an eye on network movements and identify questionable activities. Based on the volume of traffic flowing through all of the network devices, multiple NIDS will be required. Any

Scanned with OKEN Scanner

unusual behavior is discovered, the NIDS will notify the system administrators or the IT team to examine. By offering a more thorough, multi-layered study of web application security, this method seeks to uncover possible attack routes that could otherwise go unnoticed.

This research involves a distinctive technique that melds NIDS strategy with web application pen testing to achieve and discover the loopholes at network system levels and application levels. Integrating NIDS into this technique, the ethical hackers or security experts can effectively understand how assaults can replicate across the network. Here, in this study, we analyze how to use NIDS strategy in web application penetration testing utilizing standard tools like Nmap, OWASP ZAP, and Nessus. We assess the efficacy of this infused technique by testing a prototype web application, monitoring network traffic, evaluate NIDS alarms. The conclusions drawn reveal how to enhance web applications' security performance by disclosing loopholes of the attacks or vulnerabilities.

2. Methodologies

2.1. Techniques for Penetration Testing with OWSAP ZAP, Nmap, Nessus:

2.1.1. The Zed Attack Proxy (OWASP ZAP) –
A dynamic application security testing (DAST) tool called OWASP ZAP is intended to identify weaknesses in online applications. It may be used to perform vulnerability assessments and penetration tests both manually and automatically.

•Reconnaissance & Spidering: The target web application may be automatically crawled using the ZAP Spider tool. For additional testing, ZAP generates a thorough map of the target web application's surface area.

•Active Scanning: ZAP's Active Scanner checks the web application for replies that point to possible vulnerabilities by sending different payloads. These tests involve looking for typical web application vulnerabilities like SQL injection, Cross-Site Scripting (XSS), and Cross-Site Request Forgery (CSRF).

•Passive Scanning: During the reconnaissance stage, ZAP also uses passive scanning to find possible problems like incorrect HTTP headers, cookie settings, and poor SSL/TLS setups without having to actively engage with the application.

•Automated Reports: Following testing, provide thorough reports that include all vulnerabilities found, their risk levels, and possible fixes that can be applied to repair.

ZAP offers a thorough overview of the attack surface and weaknesses. Unusual traffic patterns or payloads that mimic attack vectors are examples of how ZAP warnings may be tracked by a Network Intrusion Detection System (NIDS) when it is integrated with one.

2.1.2. Nmap –
A strong tool for network discovery and vulnerability analysis is Nmap (Network Mapper). It is used to find open ports, services, and any security flaws in the network of interest.

•Network Discovery: Find active hosts inside the target architecture to carry out network discovery. To find devices inside the desired range, use Nmap's ICMP ping sweep (`nmap -sn`).

•Port Scanning: Perform a thorough port scan (`nmap -sS`) to find all open ports on the target web application server. This will help you identify services that are using common ports and other service ports that could be possible points of attack.

•Service Detection: Nmap's service detection feature (`nmap -sV`) may be used to determine which service versions are operating on open ports. This can assist in locating known vulnerabilities linked to software versions.

•Vulnerability Scanning: To find known vulnerabilities in services, use Nmap scripts for vulnerability scanning (`nmap --script=vuln`). These scripts may automate vulnerability tests against the target system and are a component of Nmap's NSE (Nmap Scripting Engine).

The NIDS may be set up to watch for certain attack patterns that take use of flaws discovered during the scanning phase using the data that Nmap collects.

2.1.3. Nessus –
A thorough vulnerability scanner, Nessus is especially helpful for evaluating network security, finding flaws, and offering practical remedial advice. It may check for known security holes, configuration errors, and compliance problems in both systems and online apps.

•Initial Network Scan: To find hosts and evaluate the infrastructure's overall security posture, do a first network scan. Like Nmap, but with a focus on security, Nessus will find open ports and services.

•Vulnerability Assessment: Conduct thorough vulnerability scans using Nessus on the web application as well as network infrastructure, such as firewalls, routers, and switches. Nessus scans systems and apps for more than 130,000 known vulnerabilities and configuration flaws.

Scanned with OKEN Scanner

•**Web Application Scanning: Find vulnerabilities in web** applications such as SQL injection, cross-site scripting, unsafe HTTP methods, and more by using Nessus. Critical vulnerabilities like out-of-date plugins or servers without fixes can be detected by it.

•**Report Generation: Following the scan, Nessus produces** comprehensive reports detailing the vulnerabilities found, along with information on their severity, risk assessment, and suggested corrective actions.

To create custom signatures and monitoring rules for vulnerabilities found, Nessus discoveries may be loaded into the NIDS. It is possible to set up the NIDS to indicate any efforts at exploiting a vulnerability that Nessus finds, such an Apache server that isn't patched.

2.2 Integration of OWASP ZAP, Nmap, and Nessus with Network Intrusion Detection Systems (NIDS):

2.2.1. Pre-Test Configuration:
Initially, we used Nmap to discover live hosts, open ports, and services on the target network. This information feeds into ZAP and Nessus to target specific attack vectors during the testing phase.

2.2.2. Penetration Testing:
The results obtained are fed to OWASP ZAP for active web application testing and vulnerability scanning. At the same time, Nessus provides an in-depth vulnerability assessment of the target network and application.

2.2.3. NIDS Monitoring:
Implement a comprehensive monitoring system to analyze traffic patterns for indicators of potential exploitation attempts. This involves establishing baseline traffic behaviors and continuously observing network activity for anomalies that may suggest malicious behaviors, such as unusual spikes in traffic, unexpected access to sensitive resources, or patterns typically associated with known attack vectors. Employing advanced analytics and automated alerts will enhance the ability to quickly identify and respond to potential threats, ensuring a proactive defense against intrusion.

2.2.4. Real-Time Detection:
As penetration testing progresses, the Network Intrusion Detection System (NIDS) continuously monitors network traffic for unauthorized access attempts or exploitation activities. It utilizes vulnerability signatures and attack vectors that have been identified through tools such as ZAP, Nmap, and Nessus. By analyzing this data in real-time, the NIDS can promptly detect suspicious behavior, allowing for immediate response and mitigation efforts to protect the system from potential threats. This proactive approach ensures a robust defense against various attack strategies throughout the testing phase.

2.2.5. Post-Test Analysis and Reporting:
After completing the penetration testing phase, the NIDS provides logs of any detected exploits, while Nmap and Nessus provide comprehensive assessments and remediation steps for the discovered weaknesses.

3. Implementation

Here, we outline the detailed steps for putting our all-inclusive Network Intrusion Detection System (NIDS)-based approach to web application penetration testing into practice. These tools—NMAP, NESSUS, and OWSAP ZAP—were chosen because they can perform a variety of security testing tasks on web applications, including vulnerability scanning, network traffic analysis, and real-time monitoring.

3.1 Nessus Vulnerability Report:

Numerous vulnerabilities have been found by Nessus, including a major buffer overflow vulnerability that requires urgent attention.

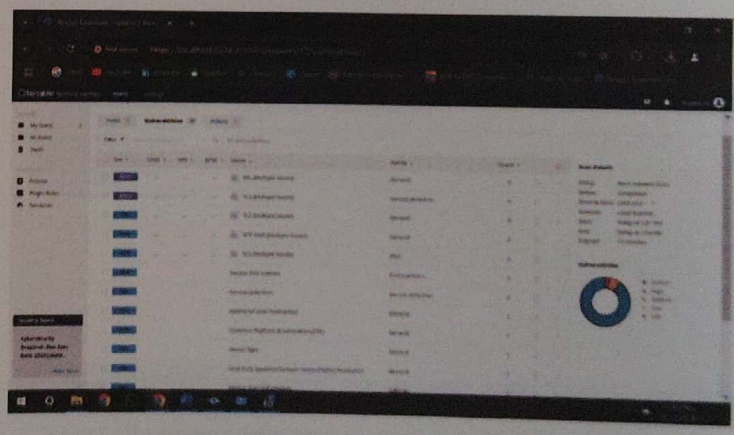


Fig. 3.1.1 Presents the Nessus vulnerability assessment report, highlighting critical findings.

Scanned with OKEN Scanner

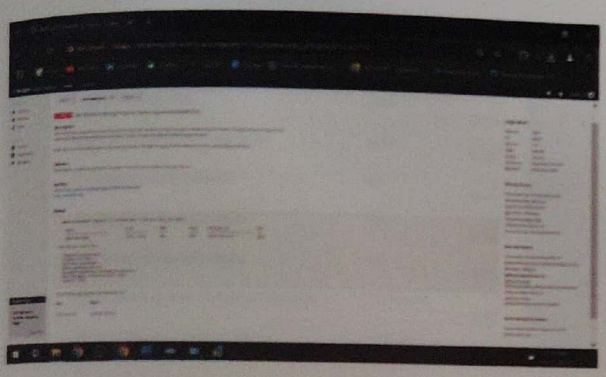


Fig. 3.1.2 Illustrates description of critical finding of a target host.

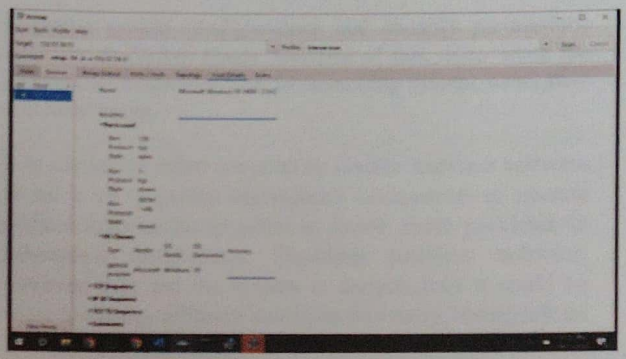


Fig. 3.2.2 Illustrates target host details after intense scanning.

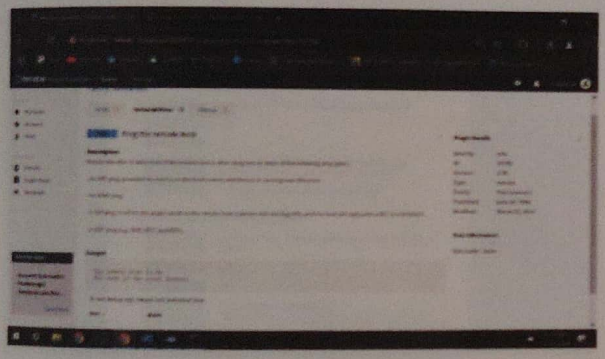


Fig. 3.1.3 Illustrates ping information of the target host.

3. 2 Nmap Scanning Result:

The target web application was first reconnaissance using Nmap. It provides vital information to further exploit vulnerabilities by assisting in the identification of open ports and services.

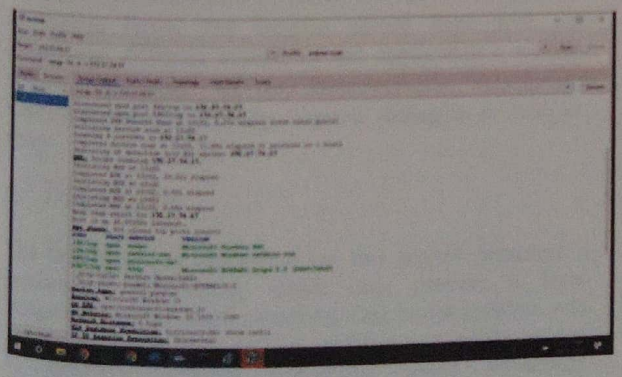


Fig. 3.2.1 Represents target host (172.17.74.17) description.

3. OWASP ZAP (Zed Attack Proxy):

Employed for active and passive security testing. The automated spider and passive scanning functionalities were used in the penetration testing to find weaknesses.

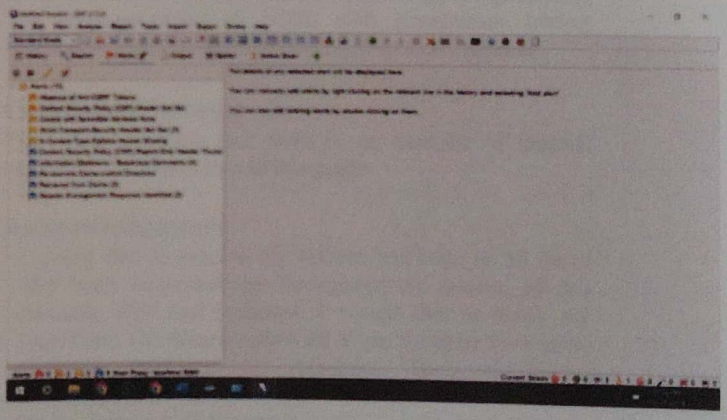


Fig. 3.3.1 Represents possible alerts of given remote target host.

Scanned with OKEN Scanner

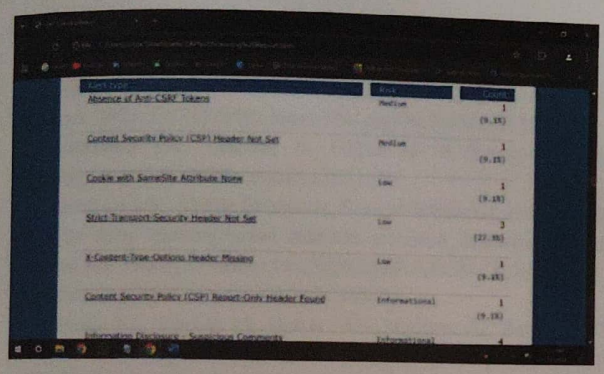


Fig. 3.3.2 Shows the number of alerts of each alert type, with the alert type's risk level

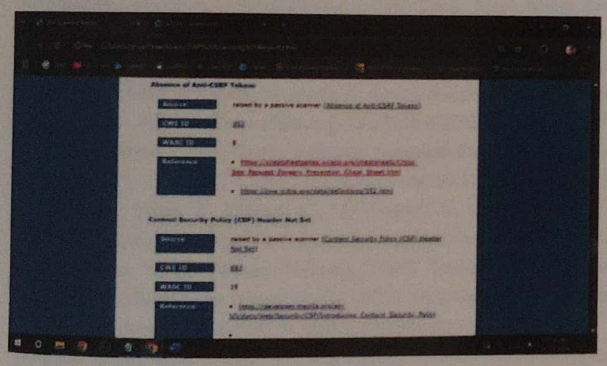


Fig. 3.3.3 Represents additional information on the types of alerts in the ZAP report.

By applying this comprehensive NIDS-based strategy, the testing effectively simulated real-world attack scenarios on the web application. Through the use of tools like Nmap, Nessus, and OWASP ZAP, we identified a diverse range of vulnerabilities, from network traffic anomalies to significant application flaws. The insights gained from each tool highlighted different aspects of the application's security posture, allowing us to constructively recommend improvements and develop targeted strategies to mitigate risks.

4. Discussions

The installation of Metasploit for this project presented a number of difficulties, which may prevent it from integrating well in some situations. One major problem was the installation process failing, which was partially caused by antivirus software interfering. Some Metasploit components were identified by the system as possible risks, which led to unsuccessful installation attempts or unfinished setups. Furthermore, in settings with strict security procedures, the Metasploit framework's dependence on particular system configurations and security permissions may lead to further issues. Because of this, even when the program is compatible with the operating system, users may still have issues.

It is advised to either temporarily disable antivirus software or set it up to permit Metasploit's components to operate without interruption in order to lessen these problems in subsequent installations. Disabling antivirus software, however, may put the system at danger, thus it could be wiser to employ different strategies like using Metasploit on a virtual machine or containerized environment. In addition to separating Metasploit from the host system, this method offers a safer and more regulated setting for resolving installation problems without jeopardizing system security. Furthermore, utilizing cloud-based technologies or creating specialized testing environments can make future Metasploit endeavours more efficient.

5. Conclusion

When performing web application penetration testing, NIDS strategy provides more holistic way to locate and identify security flaws. This literature survey has covered popular web application threats, the growth of penetration testing processes, and the role of NIDS in enhancing safety checks. The implementation of NIDS with penetration testing provides a robust protection over contemporary threats.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgments

I would like to express my sincere gratitude to all those who have supported me throughout the course of this research. First and foremost, I would like to thank my supervisor, **Dr. Sharmasth Vali Y** sir, for their invaluable guidance, patience, and expertise. Their continuous encouragement and constructive feedback were crucial to the development of this work.

I would also like to extend my thanks to my co-authors, **Shreya Malshetty, Jayalakshmi D, Chaithra Banger** for their insightful discussions and contributions. Special thanks to my university library for their assistance with plagiarism and for their help in reviewing and improving the manuscript.

Lastly, I would like to thank my family and friends for their unwavering support and understanding throughout the course of this project.

Scanned with OKEN Scanner

References

[1] Alhamed M, Rahman MMH. A Systematic Literature Review on Penetration Testing in Networks: Future Research Directions. Applied Sciences. 2023; 13(12):6986. https://doi.org/10.3390/app13126986

[2] Altulaihan EA, Alismail A, Frikha M. A Survey on Web Application Penetration Testing. Electronics. 2023; 12(5):1229. https://doi.org/10.3390/electronics12051229

[3] Vladimir Ciric, Marija Milosevic, Danijel Sokolovic, Ivan Milentijevic,
Modular deep learning-based network intrusion detection architecture for real-world cyber-attack simulation, Simulation Modelling Practice and Theory, Volume 133,2024,102916,ISSN 1569-190X,
https://doi.org/10.1016/j.simpat.2024.102916

[4] Saputra FA, Salman M, Hasim JAN, Nadhori IU, Ramli K. The Next-Generation NIDS Platform: Cloud-Based Snort NIDS Using Containers and Big Data. Big Data and Cognitive Computing. 2022; 6(1):19. https://doi.org/10.3390/bdcc6010019

[5] S. Kumar, S. Gupta and S. Arora, "Research Trends in Network-Based Intrusion Detection Systems: A Review," in IEEE Access, vol. 9, pp. 157761-157779, 2021. https://doi:10.1109/ACCESS.2021.3129775

[6] Heather Lawrence, Uchenna Ezeobi, Orly Tauil, Jacob Nosal, Owen Redwood, Yanyan Zhuang, Gedare Bloom,
CUPID: A labeled dataset with Pentesting for evaluation of network intrusion detection, Journal of Systems Architecture, Volume 129,2022,102621,ISSN13837621; https://doi.org/10.1016/j.sysarc.2022.102621

[7] Kumar, Satish & Gupta, Sunanda & Arora, Sakshi. (2021). Research Trends in Network-Based Intrusion Detection Systems: A Review. IEEE Access. PP. 1-1; https://doi.org/10.1109/ACCESS.2021.3129775

[8] Lirim Ashiku, Cihan Dagli, Network Intrusion Detection System using Deep Learning, Procedia Computer Science, Volume 185, 2021, Pages 239247, ISSN 1877-0509; https://doi.org/10.1016/j.procs.2021.05.025

[9] Vanin P, Newe T, Dhirani LL, O'Connell E, O'Shea D, Lee B, Rao M. A Study of Network Intrusion Detection Systems Using Artificial Intelligence/Machine Learning. Applied Sciences. 2022; 12(22):11752. https://doi.org/10.3390/app122211752

[10] Vegesna, Vinod Varma, Utilising VAPT Technologies (Vulnerability Assessment & Penetration Testing) as a Method for Actively Preventing Cyberattacks (October 25, 2023). International Journal of Management, Technology And Engineering | Volume XII, Issue VII, JULY 2022, Available at SSRN: ssrn.com/abstract=4612524

[11] Ke Chen, Man Zhang, Rufeng Liang, Junhan Chen, Jin Peng, Xun Huang, Research on the Application of Penetration Testing Frameworks in Blockchain Security, Computational and Experimental Simulations in Engineering, https://doi.org/10.1007/978

[12] Fredrik Heiding, Emre Süren, Johannes Olegård, Robert Lagerström,
Penetration testing of connected households,Computers & Security,Volume 126, 2023,103067,ISSN 0167-4048; https://doi.org/10.1016/j.cose.2022.103067

[13] Altulaihan, E.A.; Alismail, A.; Frikha, M. A Survey on Web Application Penetration Testing. Electronics 2023, 12, 1229. https://doi.org/10.3390/electronics12051229

[14] A. Goutam and V. Tiwari, "Vulnerability Assessment and Penetration Testing to Enhance the Security of Web Application," 2019 4th International Conference on Information Systems and Computer Networks (ISCON), Mathura, India, 2019, pp. 601-605, doi: 10.1109/ISCON47742.2019.9036175.

[15] P. Vats, M. Mandot and A. Gosain, "A Comprehensive Literature Review of Penetration Testing & Its Applications," 2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), Noida, India, 2020, pp. 674-680, doi: 10.1109/ICRITO48877.2020.9197961.

[16] Abdulghaffar, K.; Elmrabit, N.; Yousefi, M. Enhancing Web Application Security through Automated Penetration Testing with Multiple Vulnerability Scanners. Computers 2023, 12, 235. https://doi.org/10.3390/computers12110235

[17] Soroush M. Sohi, Jean-Pierre Seifert, Fatemeh Ganji, RNNIDS: Enhancing network intrusion detection systems through deep learning,Computers & Security,Volume 102,2021,102151,ISSN 0167-4048, https://doi.org/10.1016/j.cose.2020.102151

[18] Ms Khushnaseeb Roshan, Aasim Zafar, Boosting robustness of network intrusion detection systems: A novel two phase defense strategy against untargeted white-box optimization adversarial attack, Expert Systems with Applications,Volume 249, Part A,2024,123567,ISSN 09574174, https://doi.org/10.1016/j.eswa.2024.123567.

[19] Murat Aydos, Çiğdem Aldan, Evren Coşkun, Alperen Soydan,Security testing of web applications: A systematic mapping of the literature,Journal of King Saud University - Computer and Information Sciences,Volume 34, Issue 9,2022,Pages 6775-6792,ISSN 13191578, https://doi.org/10.1016/j.jksuci.2021.09.018.

[20] Tadhani, J.R., Vekariya, V., Sorathiya, V. et al. Securing web applications against XSS and SQLi attacks using a novel deep learning approach. Sci Rep 14, 1803 (2024). https://doi.org/10.1038/s41598-023-48845-4

[21] Aljebry, A.F., Alqahtani, Y.M., Sulaiman, N. (2022). Analyzing Security Testing Tools for Web Applications. In: Khanna, A., Gupta, D., Bhattacharyya, S., Hassanien, A.E., Anand, S., Jaiswal, A. (eds) International Conference on Innovative Computing and Communications. Advances in Intelligent Systems and Computing, vol 1387. Springer, Singapore. https://doi.org/10.1007/978-981-16-

Scanned with OKEN Scanner

2594-7_34

[22] Rajić, B., Stanisavljević, Ž. & Vuletić, P. Early web application attack detection using network traffic analysis. Int. J. Inf. Secur. 22, 77–91 (2023). https://doi.org/10.1007/s10207-022-00627-1

[23] Ayyagari, M.R., Kesswani, N., Kumar, M. et al. Intrusion detection techniques in network environment: a systematic review. Wireless Netw 27, 1269–1285 (2021). https://doi.org/10.1007/s11276-020-02529-3

[24] Ali, M., Haque, Mu., Durad, M.H. et al. Effective network intrusion detection using stacking-based ensemble approach. Int. J. Inf. Secur. 22, 1781–1798 (2023). https://doi.org/10.1007/s10207-023-00718-7

[25] Chowdhary, A.; Jha, K.; Zhao, M. Generative Adversarial Network (GAN)-Based Autonomous Penetration Testing for Web Applications. Sensors 2023, 23, 8014. https://doi.org/10.3390/s23188014

Scanned with OKEN Scanner

PAPER PLAGARISM CHECK

CCS-G07

ORIGINALITY REPORT

9%	6%	3%	2%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.hackthebox.com Internet Source	2%
2	Submitted to Coventry University Student Paper	1%
3	artoonsolutions.com Internet Source	1%
4	eitca.org Internet Source	1%
5	Submitted to The University of Texas at San Antonio Student Paper	1%
6	codewave.com Internet Source	1%
7	journals.plos.org Internet Source	<1%
8	Muthavarapu, Likhith. "Assessing the Robustness of Sophisticated Defender Algorithms in Penetration Testing", California State University, Sacramento, 2024 Publication	<1%

Scanned with OKEN Scanner

PAPER PUBLICATION ACKNOWLEDGEMENT

IJCSE Journal

To: SRUJANA MANJUNATH

Fri 03/01/2025 10:37

Dear sir/mam,

Final Documents and payment has been received. soon your article will get published. you will receive an acknowledgment mail after publication.

Thanking you.

FOLLOW US:

Unsubscribe

You can change your email preferences via the link above to opt-out of SSRG® emails at any time.

© 2024 Seventh Sense Research Group®

Scanned with OKEN Scanner