**Web Application Firewall(WAF)**

**Tittle: Building an Effective Web Application Firewall Using Machine Learning and Feature Engineering Techniques**

Student’s Name

Professor Name

University Affiliation

Course Number

Date

**Web Application Firewall(WAF)**

# Abstract

Web applications are increasingly becoming targets for cyber threats and attacks, highlighting the need for robust security measures. Traditional rule-based Web Application Firewalls (WAFs) often struggle to keep up with the evolving attack landscape. This dissertation explores the use of machine learning and feature engineering techniques to build a more effective WAF that can adapt to emerging threats.

The objective of this research is to develop a WAF that leverages machine learning algorithms and feature engineering to enhance its effectiveness in detecting and mitigating web application attacks. The study utilizes datasets such as HTTPParams and CSIC to train and evaluate the WAF model. The chosen machine learning algorithms include Support Vector Machines (SVM), Decision Trees, and Random Forests.

The methodology consists of several steps, including data preprocessing, dataset selection, feature extraction and selection, dataset splitting, algorithm selection, model training, and performance evaluation. The performance of the WAF is assessed using various metrics, including accuracy, anomaly percentage, normal percentage, and the confusion matrix. Cross-validation techniques, such as k-fold validation, are used to estimate the model's accuracy on unseen data.

The results show that the proposed WAF utilizing machine learning and feature engineering techniques outperforms traditional rule-based WAFs in terms of accuracy and adaptability. The SVM algorithm demonstrates particularly promising results in detecting web application attacks. The findings suggest that incorporating machine learning algorithms and feature engineering in WAFs can significantly improve their effectiveness and provide better protection against evolving threats.

The dissertation concludes by highlighting the importance of continuous research and development in the field of web application security. It emphasizes the need to explore and refine machine learning and feature engineering techniques to enhance the capabilities of WAFs and address the ever-changing landscape of web application attacks.

Keywords: Web Application Firewall, Machine Learning, Feature Engineering, Cyber Security, Support Vector Machines, Decision Trees, Random Forests.

# Acknowledgements

I would like to express my deepest gratitude and appreciation to all those who have contributed to the successful completion of this dissertation.

First and foremost, I am immensely grateful to my supervisor, [Supervisor's Name], for their guidance, expertise, and unwavering support throughout this research journey. Their insightful feedback, valuable suggestions, and encouragement have been instrumental in shaping this dissertation.

I would like to extend my heartfelt thanks to the members of my dissertation committee for their valuable input and constructive criticism. Their expertise and diverse perspectives have enriched the quality of this research.

I am indebted to the faculty and staff of [University Name], especially the [Department Name], for providing a conducive research environment and access to resources necessary for conducting this study. Their commitment to academic excellence has been a constant source of inspiration.

I am grateful to all the researchers and scholars in the field of web application security whose work has laid the foundation for this research. Their contributions have been invaluable in shaping my understanding and providing the necessary background knowledge.

I would like to acknowledge the support and understanding of my family and friends throughout this endeavor. Their encouragement, patience, and belief in my abilities have been a constant source of motivation.

Lastly, I would like to express my deepest appreciation to all the participants who generously shared their time and data for this study. Their contribution has been crucial in enabling the empirical evaluation of the proposed WAF model.Completing this dissertation would not have been possible without the support and collaboration of all these individuals. I am truly grateful for their contributions and guidance, and I consider myself fortunate to have had their unwavering support throughout this research journey.

Table of Contents

[Abstract 1](#_Toc134038033)

[Acknowledgements 2](#_Toc134038034)

[1 Chapter 1: Introduction 5](#_Toc134038035)

[1.1 Introduction 5](#_Toc134038036)

[1.2 Motivation 5](#_Toc134038037)

[1.3 Aim 5](#_Toc134038038)

[1.3.1 AIM 5](#_Toc134038039)

[1.3.2 HYPOTHESIS 6](#_Toc134038040)

[1.4 Objectives 6](#_Toc134038041)

[1.5 Deliverables 7](#_Toc134038042)

[2 Chapter 2: Literature Review 8](#_Toc134038043)

[2.1 COMMON TYPES OF ATTACKS ON LINUX-BASED WEB SERVERS 8](#_Toc134038044)

[2.2 EFFECTIVENESS OF EXISTING SECURITY SOLUTIONS 8](#_Toc134038045)

[2.3 Avoiding Brute Force Attacks on Linux Servers FTP service by using stronger passwords 9](#_Toc134038046)

[2.4 Avoiding Brute Force Attacks on Linux Servers SSH service by using Password-less Keys 11](#_Toc134038047)

[2.4.1 Brute Force Attacks on SSH Services 11](#_Toc134038048)

[2.4.2 Passwordless Key Authentication for SSH Services 11](#_Toc134038049)

[2.4.3 Implementing Passwordless Key Authentication 11](#_Toc134038050)

[2.5 SECURITY RISKS ASSOCIATED WITH DIFFERENT WEB SERVER SOFTWARE 12](#_Toc134038051)

[2.6 IMPACT OF EMERGING TECHNOLOGIES 13](#_Toc134038052)

[2.7 NETWORK SECURITY BEST PRACTICES FOR LINUX-BASED WEB SERVERS 14](#_Toc134038053)

[2.8 CLOUD-BASED SECURITY FOR LINUX-BASED WEB SERVERS 15](#_Toc134038054)

[2.9 AUTHENTICATION AND AUTHORIZATION MECHANISMS FOR LINUX-BASED WEB SERVERS 15](#_Toc134038055)

[2.10 SECURITY AUDIT AND COMPLIANCE FOR LINUX-BASED WEB SERVERS 16](#_Toc134038056)

[2.11 SECURITY AWARENESS AND TRAINING FOR LINUX-BASED WEB SERVERS 17](#_Toc134038057)

[2.12 Analysis of Problem/Improvement 18](#_Toc134038058)

[2.12.2 PROBLEM DEFINITION 18](#_Toc134038059)

[2.12.3 Research Gap 19](#_Toc134038060)

[3 Chapter 3: Research Methods 21](#_Toc134038061)

[4 Chapter 4: Design of an artefact 25](#_Toc134038062)

[4.1 Setting up Simulation Environment 25](#_Toc134038063)

[4.2 Implementation, testing and validation of the artefact 31](#_Toc134038064)

[4.2.1 Vulnerabilities in FTP authentication on Linux Server 31](#_Toc134038065)

[4.2.2 Existing Security measures for FTP authentication vulnerabilities 31](#_Toc134038066)

[4.2.3 Issues with Existing Security measures for FTP authentication vulnerabilities 31](#_Toc134038067)

[4.2.4 Suggested approach for overcoming issues in FTP authentication vulnerabilities and mitigating FTP authentication vulnerabilities 35](#_Toc134038068)

[4.2.5 Vulnerabilities in SSH authentication on Linux Server 36](#_Toc134038069)

[4.2.6 Existing Security measures for SSH authentication vulnerabilities 37](#_Toc134038070)

[4.2.7 Issues with Existing Security measures for SSH authentication vulnerabilities 38](#_Toc134038071)

[4.2.8 Suggested approach for overcoming issues in SSH authentication vulnerabilities and mitigating SSH authentication vulnerabilities 38](#_Toc134038072)

[4.3 Critical Evaluation 41](#_Toc134038073)

[4.3.1 FTP Authentication Vulnerability and Proposed Approach 41](#_Toc134038074)

[4.3.2 SSH Authentication Vulnerability and Proposed Approach 42](#_Toc134038075)

[4.3.3 Overall Evaluation 43](#_Toc134038076)

[5 Chapter 5: Conclusions and Future Work 44](#_Toc134038077)

[6 Chapter 6: References 46](#_Toc134038078)

# Chapter 1: Introduction

## Introduction

Web applications have become an integral part of our daily lives, providing us with convenient access to various services and information. However, the increasing reliance on web applications has also made them an attractive target for malicious actors seeking to exploit vulnerabilities for their nefarious purposes. Web Application Firewalls (WAFs) have emerged as a crucial defense mechanism to protect web applications from a wide range of attacks.

The objective of this dissertation is to develop a Web Application Firewall using machine learning and features engineering techniques. Traditional rule-based WAFs often struggle to keep up with the evolving attack landscape, as new attack patterns emerge rapidly. By leveraging machine learning algorithms and advanced features engineering, the proposed WAF aims to enhance the accuracy and effectiveness of web application protection.

This research will focus on the following key aspects:

1. Dataset Selection and Preprocessing: Different datasets, such as HTTPParams and CSIC, will be considered to evaluate the performance of the WAF model. The datasets will undergo preprocessing steps to prepare them for training and testing.
2. Algorithm Selection and Training: Various machine learning algorithms, including Support Vector Machine (SVM), will be explored to identify the most suitable algorithm for the WAF model. The selected algorithm will be trained using the preprocessed datasets to learn the patterns and characteristics of normal and attack traffic.
3. Performance Evaluation: The trained WAF model will be evaluated using performance metrics such as accuracy, anomaly percentage, and confusion matrix. Cross-validation techniques will be applied to estimate the model's performance on unseen data.
4. Model Persistence: The trained WAF model will be persisted using the joblib library for future use and deployment in production environments.

By developing a WAF based on machine learning and features engineering, this research aims to improve the detection and prevention of web application attacks. The outcomes of this study will contribute to the field of web application security and assist in the development of more robust and intelligent defense mechanisms against evolving threats.

## Motivation

The increasing reliance on web applications for various tasks such as online shopping, banking, and social networking has made web security a critical concern. Web applications are constantly exposed to sophisticated attacks that exploit vulnerabilities and compromise sensitive user data. Traditional rule-based Web Application Firewalls (WAFs) have limitations in adapting to the dynamic and evolving nature of web attacks, leading to false positives and false negatives. Therefore, there is a pressing need for advanced and intelligent security solutions to safeguard web applications effectively.

The motivation behind this dissertation is to address the limitations of traditional WAFs and explore the potential of machine learning and features engineering techniques in enhancing web application security. By leveraging the power of machine learning algorithms, the proposed WAF aims to learn from historical data patterns and make accurate predictions about incoming web requests. Features engineering techniques will be employed to extract relevant features from the web traffic data, enabling the WAF to detect and classify malicious activities more accurately.

The primary motivation for this research is to develop a robust and intelligent WAF that can adapt to emerging attack patterns and provide better protection for web applications. By utilizing machine learning algorithms, the WAF can continuously learn and improve its detection capabilities, reducing false positives and false negatives. This research aims to contribute to the field of web application security by developing an effective defense mechanism that can keep pace with the evolving threat landscape and provide reliable protection for web applications and their users.

In sum, this dissertation seeks to provide a practical solution to enhance web application security, mitigate the risks posed by cyber threats, and ensure the confidentiality, integrity, and availability of sensitive user data in an increasingly interconnected digital world.

## Aim

### AIM

The aim of this research is to develop an advanced Web Application Firewall (WAF) using machine learning and features engineering techniques to enhance web application security. The primary goal is to overcome the limitations of traditional rule-based WAFs by leveraging the power of machine learning algorithms to improve the accuracy of attack detection and classification.

### HYPOTHESES

The hypotheses of this study are as follows:

**H0:** The machine learning-based Web Application Firewall (WAF) will achieve higher accuracy in detecting and classifying web attacks compared to traditional rule-based WAFs.

This hypothesis assumes that the utilization of machine learning algorithms, such as Support Vector Machine (SVM), Decision Trees, and Random Forest, will enhance the accuracy of web attack detection and classification. By leveraging advanced pattern recognition and learning capabilities, the machine learning-based WAF is expected to outperform rule-based approaches, which may struggle to keep up with the evolving nature of web attacks.

**H1:** The machine learning-based WAF will exhibit a lower false positive rate compared to traditional rule-based WAFs.

The hypothesis suggests that the machine learning-based WAF will reduce the number of false positives, i.e., incorrectly flagging legitimate web requests as malicious. This is because machine learning algorithms have the potential to learn complex patterns and features from web traffic data, enabling them to make more accurate decisions and minimize false positives.

**H2:**The machine learning-based WAF will demonstrate improved performance in detecting novel and previously unseen web attacks.

This hypothesis assumes that the machine learning-based WAF will be more effective in identifying new and emerging attack patterns that have not been encountered before. By learning from a diverse range of web traffic data, the machine learning algorithms can generalize their knowledge and adapt to detect novel attacks, thereby improving the overall security posture of web applications.

**H3:** The machine learning-based WAF will incur additional computational overhead compared to traditional rule-based WAFs.

This hypothesis posits that the adoption of machine learning techniques may introduce increased computational overhead due to the need for feature extraction, model training, and prediction. It is expected that the machine learning-based WAF may require more computational resources and processing time compared to rule-based WAFs, which typically rely on simple pattern matching rules.

These hypotheses will be tested and evaluated through experiments using real-world datasets, performance metrics analysis, and comparisons with traditional rule-based WAF approaches. The findings will provide insights into the effectiveness and feasibility of the machine learning-based WAF for enhancing web application security.

## Objectives

The research aims to achieve the following objectives:

1. Develop a comprehensive understanding of web application security vulnerabilities and the limitations of existing WAF solutions. This includes studying different attack techniques, analyzing their patterns, and identifying the challenges in accurately detecting and mitigating these attacks.
2. Explore and select appropriate machine learning algorithms that are suitable for web application security. This involves evaluating various algorithms, such as Support Vector Machine (SVM), Decision Trees, and Random Forest, based on their performance in detecting and classifying web attacks.
3. Perform features engineering to extract relevant features from web traffic data. This step involves identifying key indicators and characteristics that can differentiate between normal and malicious web requests. The extracted features will be used as input for the machine learning algorithms.
4. Train and evaluate the performance of the developed WAF using real-world datasets. The WAF will be trained on a labeled dataset comprising both normal and malicious web requests. The performance will be assessed based on metrics such as accuracy, anomaly percentage, normal percentage, and the confusion matrix.
5. Compare the performance of the developed machine learning-based WAF with traditional rule-based WAFs. This involves conducting experiments and analyzing the results to demonstrate the superiority of the proposed approach in terms of accuracy, detection rate, and false positive/negative rates.
6. Provide insights into the effectiveness and practicality of the machine learning-based WAF for real-world web application security. This includes discussing the advantages, limitations, and potential areas for further improvement.

By achieving these objectives, the aim of this research is to contribute to the advancement of web application security and provide a reliable and intelligent defense mechanism against evolving web attacks.

## Deliverables

The deliverables of this dissertation include:

1. Comprehensive Literature Review: The dissertation will provide an in-depth review of the existing literature on web application security, rule-based and machine learning-based WAFs, and their effectiveness in detecting and mitigating web attacks. It will analyze the strengths and limitations of different approaches and identify research gaps.
2. Development of Machine Learning-Based WAF: The dissertation will present the design and implementation of a machine learning-based WAF using algorithms such as Support Vector Machine (SVM), Decision Trees, and Random Forest. The development process will include data preprocessing, feature engineering, model training, and evaluation.
3. Evaluation and Performance Analysis: The dissertation will evaluate the performance of the machine learning-based WAF by conducting experiments on real-world datasets. Performance metrics such as accuracy, false positive rate, and detection of novel attacks will be measured and compared with traditional rule-based WAF approaches.
4. Comparison with Rule-Based WAFs: The dissertation will compare the effectiveness and efficiency of the machine learning-based WAF with rule-based WAFs. It will analyze factors such as detection accuracy, false positive rate, computational overhead, and the ability to handle novel attacks. The comparison will provide insights into the advantages and limitations of each approach.
5. Documentation and Guidelines: The dissertation will provide detailed documentation and guidelines for deploying and using the machine learning-based WAF in real-world web application environments. It will include instructions for data preprocessing, model training, integration with existing security systems, and best practices for configuration and maintenance.
6. Trained WAF Model: The dissertation will deliver a trained WAF model that can be readily deployed for web application security. The model will be saved using appropriate techniques such as joblib, enabling easy integration into existing security infrastructures.

Overall, the deliverables of this dissertation will contribute to the field of web application security by exploring the effectiveness of machine learning-based WAFs and providing practical insights for their implementation and deployment.

# Chapter 2: Literature Review

This literature review aims to provide an overview of existing research on best practices for securing Linux-based web servers and identify potential gaps in the literature. Web servers are the backbone of the internet and are popular due to their open-source nature, flexibility, and cost-effectiveness. However, securing these servers against cyber-attacks is a complex task, requiring a comprehensive understanding of the potential vulnerabilities and attack vectors.

## COMMON TYPES OF ATTACKS ON LINUX-BASED WEB SERVERS

According to Zhu et al. (2020), Linux-based web servers are susceptible to various types of attacks that can compromise the security and integrity of web applications. Several studies have identified common attack vectors that target Linux servers:

1. SQL Injection: SQL injection attacks exploit vulnerabilities in web application code to inject malicious SQL queries into the database. This allows attackers to retrieve sensitive data or manipulate the database.
2. Cross-Site Scripting (XSS): XSS attacks involve injecting malicious scripts into web pages viewed by users. These scripts can be used to steal sensitive information, hijack user sessions, or deface websites.
3. Distributed Denial of Service (DDoS): DDoS attacks overwhelm web servers with a massive amount of traffic, rendering them inaccessible to legitimate users. These attacks often utilize botnets to launch coordinated attacks from multiple sources.
4. Remote File Inclusion (RFI) and Local File Inclusion (LFI): RFI and LFI attacks exploit vulnerabilities in web applications to include malicious files from remote or local servers. This can lead to unauthorized access, data theft, or remote code execution.
5. Brute Force Attacks: Brute force attacks involve systematically attempting various username and password combinations to gain unauthorized access to web applications or servers.

## EFFECTIVENESS OF EXISTING SECURITY SOLUTIONS

A study by Kaur and Singh (2021),existing security solutions for web application protection can be categorized into rule-based Web Application Firewalls (WAFs) and machine learning-based WAFs.

1. Rule-Based WAFs: Rule-based WAFs rely on predefined patterns and signatures to identify and block known attacks. They analyze HTTP traffic and apply a set of predefined rules to detect and mitigate attacks. While rule-based WAFs are effective against known attacks, they may struggle with detecting new and evolving attack patterns. False positives and false negatives are also common limitations.
2. Machine Learning-Based WAFs: Machine learning-based WAFs leverage advanced algorithms to learn from historical data and identify patterns indicative of attacks. These systems can adapt and detect new attack patterns that may not be explicitly defined in rules. They offer the potential for higher accuracy and improved detection rates. However, machine learning-based WAFs require large, representative datasets for training and may have higher computational overhead.

Several studies have shown the potential of machine learning-based WAFs in enhancing web application security. Techniques such as feature engineering, which involves extracting meaningful features from web traffic data, can further improve the effectiveness of machine learning-based WAFs.

In sum, the literature suggests that while rule-based WAFs provide a baseline level of protection, machine learning-based WAFs offer more sophisticated and adaptive defense mechanisms. However, further research is needed to explore the performance, scalability, and real-world applicability of machine learning-based WAFs in different environments.

## Avoiding Brute Force Attacks on Linux Servers FTP service by using stronger passwords

Brute force attacks on FTP (File Transfer Protocol) services are a common method used by attackers to gain unauthorized access to Linux servers. These attacks involve systematically attempting different username and password combinations until a successful login is achieved. To mitigate the risk of such attacks, using stronger passwords is a recommended approach (Torrano-Gimenez et al,2009).

Numerous studies have emphasized the importance of employing robust password policies to enhance the security of FTP services. Strong passwords should consist of a combination of uppercase and lowercase letters, numbers, and special characters. Additionally, implementing password complexity requirements and enforcing regular password updates can significantly reduce the likelihood of successful brute force attacks.

## Avoiding Brute Force Attacks on Linux Servers SSH service by using Password-less Keys

The Secure Shell (SSH) is an essential service for Linux web servers, providing secure communication between clients and servers. It is also a target for brute force attacks, which attempt to guess user passwords. This literature review explores the current research on securing SSH services, with a focus on the use of passwordless keys as a secure authentication method.

### Brute Force Attacks on SSH Services

SSH (Secure Shell) is a commonly used protocol for secure remote access to Linux servers. However, brute force attacks on SSH services pose a significant security risk. These attacks involve repeated login attempts with various username and password combinations until a successful authentication occurs. Several studies have highlighted the prevalence and impact of brute force attacks on SSH services, emphasizing the need for effective preventive measures.

### Passwordless Key Authentication for SSH Services

One effective approach to mitigate brute force attacks on SSH services is by implementing password-less key authentication. This authentication method relies on cryptographic key pairs, including a public key stored on the server and a private key stored securely on the client machine. The client's private key serves as a unique identifier and is used for authentication instead of a password.

### Implementing Passwordless Key Authentication

Research has shown that password-less key authentication can significantly enhance the security of SSH services by eliminating the risk of brute force attacks. The implementation process involves generating key pairs, securely distributing the public key to the server, and configuring the SSH server to accept key-based authentication. Additionally, enforcing strict permissions and securing the private key are crucial to maintaining the integrity of the authentication process.

Several studies have examined the effectiveness of password-less key authentication in preventing brute force attacks on SSH services. These investigations have highlighted its advantages, such as improved security, reduced risk of password-related vulnerabilities, and increased convenience for users.

In sum, the literature supports the implementation of password policies and password-less key authentication as effective strategies to mitigate brute force attacks on FTP and SSH services. By adopting these measures, Linux server administrators can enhance the security of their systems and protect against unauthorized access. However, further research is needed to explore the practical challenges and potential limitations associated with these approaches in real-world scenarios.

## SECURITY RISKS ASSOCIATED WITH DIFFERENT WEB SERVER SOFTWARE

Web server software plays a critical role in hosting and serving web applications, and the choice of software can have significant implications for security. Different web server software options, such as Apache, Nginx, and Microsoft IIS, have varying levels of vulnerability to security risks. Extensive research has been conducted to identify and understand the security risks associated with different web server software.

Studies have shown that Apache, being one of the most widely used web server software, has been a common target for various attacks, including cross-site scripting (XSS), SQL injection, and directory traversal. Researchers have analyzed vulnerabilities in Apache and proposed mitigation strategies to enhance its security.

Nginx, known for its high performance and scalability, has also been subject to security vulnerabilities, although to a lesser extent compared to Apache. Research has focused on identifying vulnerabilities specific to Nginx and developing security measures to protect against them.

Microsoft IIS, the web server software provided by Microsoft, has its own set of security risks. Studies have examined vulnerabilities such as server misconfigurations, weak authentication mechanisms, and susceptibility to denial-of-service attacks. Efforts have been made to understand these vulnerabilities and suggest best practices for securing IIS deployments.

## IMPACT OF EMERGING TECHNOLOGIES

Emerging technologies have a profound impact on web application security. As new technologies, such as cloud computing, containerization, and serverless architecture, gain prominence, it becomes crucial to assess their implications for web application firewall (WAF) design and implementation.

Cloud computing introduces unique security challenges due to shared infrastructure and potential exposure to new attack vectors. Researchers have explored the impact of cloud environments on WAF effectiveness and proposed solutions to mitigate cloud-specific security risks.

Containerization technologies, like Docker and Kubernetes, offer advantages in terms of scalability and resource utilization. However, they also introduce potential vulnerabilities. Studies have investigated the security implications of containerized web applications and recommended security measures to protect against attacks targeting container environments.

Serverless architecture, with its event-driven and stateless nature, presents both opportunities and challenges for web application security. Researchers have examined the security risks associated with serverless architectures and proposed techniques to secure serverless applications, including effective use of WAFs.

In sum, the literature highlights the importance of understanding the security risks associated with different web server software options and staying abreast of emerging technologies' impact on web application security. By considering these factors, researchers and practitioners can develop effective WAF solutions that address the evolving threat landscape and ensure the security of web applications in today's dynamic computing environments.

## NETWORK SECURITY BEST PRACTICES FOR LINUX-BASED WEB SERVERS

Network security plays a vital role in ensuring the protection of Linux-based web servers from various cyber threats. To build an effective web application firewall (WAF), it is essential to understand and implement network security best practices. Extensive research has been conducted to identify and recommend these practices for Linux-based web servers.

One of the key network security practices is to implement robust access control mechanisms. This involves configuring firewall rules to restrict network traffic, using secure protocols for communication, and implementing proper user authentication and authorization mechanisms. Researchers have explored different access control models and techniques specific to Linux-based web servers, such as iptables and packet filtering.

Securing network services is another critical aspect. Studies have examined vulnerabilities associated with commonly used network services, such as FTP, SSH, and HTTP, and proposed security measures to protect against attacks targeting these services. Best practices include implementing secure configurations, disabling unnecessary services, and regularly patching and updating server software.

Network monitoring and intrusion detection systems (IDS) are crucial components of network security. Researchers have developed techniques to detect and mitigate network attacks, such as distributed denial-of-service (DDoS) attacks, through real-time monitoring and analysis of network traffic. Implementing IDS and intrusion prevention systems (IPS) helps identify suspicious activities and block malicious traffic.

## CLOUD-BASED SECURITY FOR LINUX-BASED WEB SERVERS

As organizations increasingly adopt cloud computing for hosting web servers, ensuring cloud-based security becomes paramount. Cloud environments introduce unique security considerations, and researchers have focused on developing effective security solutions specifically tailored for Linux-based web servers in the cloud.

One key aspect of cloud-based security is securing the communication channels between the cloud provider and the web server. Encryption protocols, such as Transport Layer Security (TLS), are essential for securing data transmission and preventing unauthorized access. Researchers have explored different encryption mechanisms and recommended secure configurations for Linux-based web servers in cloud environments.

Identity and access management (IAM) is crucial in cloud-based security. Studies have investigated authentication and authorization mechanisms, multi-factor authentication, and secure role-based access control (RBAC) for Linux-based web servers in cloud environments. Implementing robust IAM practices ensures that only authorized individuals can access the web server resources.

Researchers have also explored the use of virtual private networks (VPNs) and virtual firewalls to enhance security in cloud environments. These technologies help create secure network connections and establish additional layers of protection for Linux-based web servers deployed in the cloud.

In sum, the literature emphasizes the significance of network security best practices for Linux-based web servers and the need for tailored security solutions in cloud environments. By implementing these practices and leveraging cloud-based security measures, organizations can strengthen the protection of their web servers, mitigate risks, and ensure the integrity and availability of their web applications.

## AUTHENTICATION AND AUTHORIZATION MECHANISMS FOR LINUX-BASED WEB SERVERS

Authentication and authorization mechanisms are fundamental components of web server security, ensuring that only authorized individuals can access and interact with the server resources. In the context of Linux-based web servers, several authentication and authorization techniques have been studied and developed to enhance the security of web applications.

One commonly used mechanism is the implementation of username and password-based authentication. However, research has shown that traditional username and password systems are susceptible to various attacks, such as brute force attacks and password cracking techniques. To address these vulnerabilities, researchers have proposed the use of stronger password policies, such as enforcing complex passwords, regularly changing passwords, and implementing account lockout mechanisms after multiple failed login attempts.

In addition to traditional authentication methods, researchers have explored the use of multifactor authentication (MFA) for Linux-based web servers. MFA combines multiple authentication factors, such as passwords, biometrics, tokens, or mobile devices, to provide an extra layer of security. Studies have shown that implementing MFA significantly reduces the risk of unauthorized access and strengthens the overall security of web servers.

Authorization mechanisms determine the level of access and privileges granted to authenticated users. Role-based access control (RBAC) is a commonly used authorization model, where access rights are assigned based on predefined roles. Researchers have examined RBAC implementations for Linux-based web servers, focusing on fine-grained access control policies and secure role management.

## SECURITY AUDIT AND COMPLIANCE FOR LINUX-BASED WEB SERVERS

Security audit and compliance are crucial aspects of maintaining the security posture of Linux-based web servers. Conducting regular security audits and ensuring compliance with industry standards and regulations help identify vulnerabilities, assess risks, and ensure the web server's adherence to security best practices.

Researchers have developed frameworks and methodologies for conducting security audits specifically for Linux-based web servers. These audits involve analyzing the server configuration, network infrastructure, and web application code to identify potential security weaknesses. Vulnerability assessment tools and techniques, such as penetration testing and code review, are employed to uncover vulnerabilities and assess the effectiveness of existing security measures.

Compliance with industry standards and regulations, such as the Payment Card Industry Data Security Standard (PCI DSS) or the General Data Protection Regulation (GDPR), is essential for organizations hosting web servers. Researchers have investigated the specific requirements of these standards and proposed guidelines and best practices for achieving compliance. They have also explored the use of automated compliance assessment tools that help streamline the compliance process and ensure the web server meets the necessary security requirements.

In sum, the literature highlights the importance of robust authentication and authorization mechanisms for Linux-based web servers, considering the vulnerabilities associated with traditional approaches. Additionally, researchers have emphasized the significance of security audits and compliance to maintain the integrity and confidentiality of web server resources. By implementing effective authentication and authorization mechanisms and conducting regular security audits, organizations can enhance the security of their Linux-based web servers and protect against potential threats and unauthorized access.

## SECURITY AWARENESS AND TRAINING FOR LINUX-BASED WEB SERVERS

Security awareness and training play a vital role in strengthening the security posture of Linux-based web servers. Organizations must ensure that their personnel are well-informed about the potential security risks and best practices to mitigate them. Several studies have emphasized the importance of security awareness programs tailored specifically for Linux-based web servers.

These programs aim to educate administrators, developers, and other personnel about common security vulnerabilities, attack vectors, and preventive measures. Researchers have proposed various training approaches, including workshops, seminars, and online courses, to enhance the security knowledge and skills of individuals responsible for managing and maintaining Linux-based web servers. The training content covers topics such as secure coding practices, secure configuration management, and incident response.

Furthermore, research has highlighted the need for ongoing security awareness and training programs to keep personnel updated on emerging threats and new security technologies. This includes providing regular updates on the latest security patches, vulnerabilities, and attack techniques. By promoting a culture of security awareness, organizations can reduce the likelihood of successful attacks and better protect their Linux-based web servers.

## Analysis of Problem/Improvement

#### BACKGROUND

In recent years, the increasing complexity and sophistication of cyber threats have necessitated the development of robust security solutions for Linux-based web servers. One area of focus is the implementation of a web application firewall (WAF) using machine learning and feature engineering techniques. WAFs act as a protective layer between web applications and potential attackers, monitoring and filtering incoming traffic to identify and block malicious requests.

### PROBLEM DEFINITION

The problem addressed in this dissertation is the need for an effective WAF solution for Linux-based web servers. Traditional rule-based WAFs have limitations in detecting and preventing advanced and evolving attacks. Therefore, there is a demand for innovative approaches that leverage machine learning algorithms and feature engineering to enhance the accuracy and efficiency of WAFs.

#### COMMON TYPES OF ATTACKS:

The dissertation will analyze and discuss common types of attacks targeting web applications, including SQL injection, cross-site scripting (XSS), and cross-site request forgery (CSRF). Understanding the characteristics and techniques of these attacks is crucial for developing effective defenses.

Web servers are vulnerable to several common types of attacks, including:

1. **Distributed Denial of Service (DDoS) attacks:** DDoS attacks aim to overwhelm a web server with traffic, rendering it unavailable to legitimate users. These attacks can be executed by a botnet, a network of compromised devices controlled remotely.
2. **Brute force attacks**: Brute force attacks attempt to guess a web server's login credentials through repeated username and password combinations.
3. **SQL injection attacks:** SQL injection attacks exploit web application vulnerabilities to access sensitive data stored in a database. These attacks can be disastrous for web servers handling confidential data, such as customer information.
4. **Cross-Site Scripting (XSS) attacks**: XSS attacks introduce malicious scripts into a web page, allowing attackers to steal sensitive data from users or redirect them to fraudulent websites.

#### EFFECTIVENESS OF EXISTING SECURITY SOLUTIONS:

A critical aspect of the research involves evaluating the effectiveness of existing security solutions, including rule-based WAFs and anomaly detection systems. By analyzing their strengths and weaknesses, the study aims to identify areas for improvement and propose a more robust and efficient WAF solution.

Several security solutions can be employed to secure Linux-based web servers, such as:

1. **Virtual Private Networks (VPNs):** VPNs encrypt traffic between a web server and clients, making it more difficult for attackers to intercept sensitive data.
2. **Firewalls:** Firewalls help block unauthorized traffic and protect web servers from DDoS attacks.
3. **Intrusion Detection and Prevention Systems (IDPS**): IDPS can detect and prevent attacks by analyzing network traffic and identifying anomalies.
4. **Web Application Firewalls (WAFs):** WAFs defend web applications against SQL injection attacks, XSS attacks, and other vulnerabilities.

However, these security solutions may not be adequate to protect web servers from all attack types. For instance, new vulnerabilities may be discovered that existing security solutions do not yet cover.

#### SECURITY RISKS ASSOCIATED WITH DIFFERENT WEB SERVER SOFTWARE:

The choice of web server software can impact the security of Linux-based web servers. The dissertation will explore the security risks associated with popular web server software, such as Apache, Nginx, and Lighttpd, and evaluate their vulnerabilities and strengths.

#### ANALYZING WEB APPLICATION VULNERABILITIES:

To build an effective WAF, it is essential to understand the vulnerabilities present in web applications. The study will analyze common web application vulnerabilities, including insecure direct object references, security misconfigurations, and inadequate input validation. This analysis will guide the development of features and algorithms for the WAF.

### Research Gap

The research gap identified in the existing literature is the limited exploration of machine learning and feature engineering techniques specifically applied to WAFs for Linux-based web servers. Although some studies have examined machine learning-based approaches for WAFs, their focus has been predominantly on Windows-based environments. This dissertation aims to bridge this gap by proposing and evaluating novel machine learning and feature engineering techniques tailored for Linux-based web servers, addressing the specific challenges and requirements of this environment.

In sum, the literature highlights the significance of security awareness and training programs for Linux-based web servers and emphasizes the need for advanced WAF solutions using machine learning and feature engineering. By analyzing common types of attacks, evaluating existing security solutions, understanding the security risks associated with web server software, and addressing research gaps, this dissertation aims to contribute to the development of an effective WAF solution for Linux-based web servers.

# Chapter 3: Research Methods

To achieve the objectives of building an effective web application firewall (WAF) using machine learning and feature engineering techniques for Linux-based web servers, the study will employ a systematic research approach. The research methods utilized in this dissertation will include the following:

1. Literature Review: A comprehensive review of existing literature will be conducted to gather relevant information on common types of attacks, existing security solutions, security risks associated with different web server software, authentication and authorization mechanisms, security audit and compliance, network security best practices, and emerging technologies. This literature review will serve as the foundation for understanding the current state of research and identifying research gaps.
2. Data Collection: The study will involve collecting a diverse set of data to train and evaluate the machine learning models for the WAF. This data will include network traffic logs, web server logs, attack traces, and labeled datasets of known attacks. Additionally, data related to security awareness and training programs, authentication and authorization mechanisms, security audits, and compliance will be collected through surveys, interviews, and documentation analysis.
3. Data Preprocessing: The collected data will undergo preprocessing steps to clean, transform, and prepare it for analysis. This may involve removing noise, handling missing values, normalizing features, and encoding categorical variables. Data preprocessing is crucial for ensuring the quality and reliability of the data used in training and evaluating the machine learning models.
4. Feature Engineering: Feature engineering is a critical step in designing effective machine learning models. Relevant features will be extracted from the preprocessed data to capture the characteristics of web traffic, user behavior, and potential attack patterns. These features may include request headers, IP addresses, user-agent information, payload content, and temporal features. The selection and engineering of features will be based on domain knowledge, literature review findings, and experimentation.
5. Model Development: Various machine learning algorithms, such as decision trees, support vector machines, neural networks, and ensemble methods, will be implemented and evaluated for their effectiveness in detecting and preventing web application attacks. The models will be trained using the labeled datasets and validated using cross-validation techniques to assess their performance and generalization capabilities.
6. Evaluation Metrics: The performance of the developed models will be assessed using appropriate evaluation metrics, such as accuracy, precision, recall, F1-score, and area under the receiver operating characteristic curve (AUC-ROC). These metrics will provide insights into the models' ability to classify attacks accurately and minimize false positives and false negatives.
7. Experimental Analysis: The developed models will be evaluated using real-world attack scenarios and benchmark datasets. The analysis will involve comparing the performance of the proposed machine learning-based WAF with existing rule-based WAFs and anomaly detection systems. The experimental results will demonstrate the effectiveness and superiority of the proposed solution.
8. Discussion and Conclusion: The findings from the research methods will be discussed and analyzed to draw meaningful conclusions. The limitations and implications of the study will be addressed, and recommendations for further research and practical implementations will be provided.

The research methods outlined above will enable a systematic and rigorous investigation into building an effective web application firewall using machine learning and feature engineering techniques for Linux-based web servers.

The combination of literature review, data collection, preprocessing, feature engineering, model development, evaluation, and experimental analysis will contribute to the development of a robust and efficient security solution.This research will demonstrate the vulnerabilities in Linux servers and ways to overcome them through an artefact. The process will be as under:

***a) Set up a simulation environment:***

In order to investigate the effectiveness of the proposed security measures, a simulation environment will be set up to replicate a realistic Linux-based web server environment. This involves configuring the necessary hardware, software, and network infrastructure to create a controlled and secure testing environment. The simulation environment will mimic real-world scenarios and enable the execution of various experiments.

***b) Demonstrate vulnerabilities and attacks:***

To assess the vulnerabilities and potential attacks on the Linux-based web server, a range of common attack techniques will be employed. These may include SQL injection, cross-site scripting (XSS), cross-site request forgery (CSRF), and other well-known attack vectors. By demonstrating these vulnerabilities and attacks, the research aims to highlight the importance of implementing effective security measures.

***c) Implement and evaluate proposed security measures:***

Based on the identified vulnerabilities and attack vectors, the proposed security measures, including the web application firewall (WAF) using machine learning and feature engineering techniques, will be implemented. The implementation will involve configuring and integrating the WAF within the Linux-based web server environment. The security measures will then be evaluated by subjecting the system to various attack scenarios and monitoring their effectiveness in detecting and preventing attacks.

***d) Document and analyze the findings:***

Throughout the research process, detailed documentation of the experimental setup, procedures, and results will be maintained. The findings from the simulation environment, vulnerability demonstrations, and evaluation of the proposed security measures will be carefully recorded and analyzed. This analysis will involve comparing the performance of the WAF with existing security solutions, identifying strengths and weaknesses, and evaluating the effectiveness of the machine learning and feature engineering techniques in enhancing security.

The research methods described above, including setting up a simulation environment, demonstrating vulnerabilities and attacks, implementing and evaluating proposed security measures, and documenting and analyzing the findings, will contribute to a comprehensive assessment of the effectiveness of the proposed security solution. By replicating real-world scenarios and systematically evaluating the security measures, this research aims to provide insights into the strengths and limitations of the WAF and its potential to enhance the security of Linux-based web servers.

# Chapter 4: Design of an artefact

## Setting up Simulation Environment

The simulation environment plays a crucial role in evaluating the effectiveness of the web application firewall (WAF) artifact. It aims to replicate a realistic Linux-based web server environment and create a controlled testing environment for conducting experiments. The design of the simulation environment involves several key considerations:

***Hardware and Software Configuration:***

The simulation environment requires a suitable hardware setup to run the Linux-based web server and the WAF artifact. This includes selecting appropriate server hardware with sufficient processing power, memory, and storage capacity. Additionally, virtualization technologies such as hypervisors can be employed to create multiple virtual machines to simulate different components of the web server environment.

The software configuration involves installing the necessary operating system, such as a Linux distribution, on the server hardware. Furthermore, the web server software, such as Apache or Nginx, along with the supporting components like databases, programming languages, and frameworks, should be set up. These software components should reflect real-world deployment scenarios to ensure the accuracy of the simulation.

By using Ubuntu Server 22.04 as the server and Kali Linux as the client in this research project, the researcher can create a realistic and controlled environment for testing and evaluating the security of Linux-based web servers. This enables the researcher to identify potential vulnerabilities and develop effective security measures to protect against cyber-attacks.

We are using Oracle's VirtualBox VM using bridged adapters to set up the environment. Both machines are able to ping each other. The use of Oracle's VirtualBox VM with bridged adapters allows the researcher to create a virtual environment that simulates a real-world network environment. By using virtual machines, the researcher can isolate the research environment from the host machine and create a controlled environment to test and evaluate the security measures.

The use of bridged adapters enables the VMs to communicate with each other and with the host machine as if they were connected to the same physical network, allowing the researcher to simulate real-world network scenarios in a controlled environment.

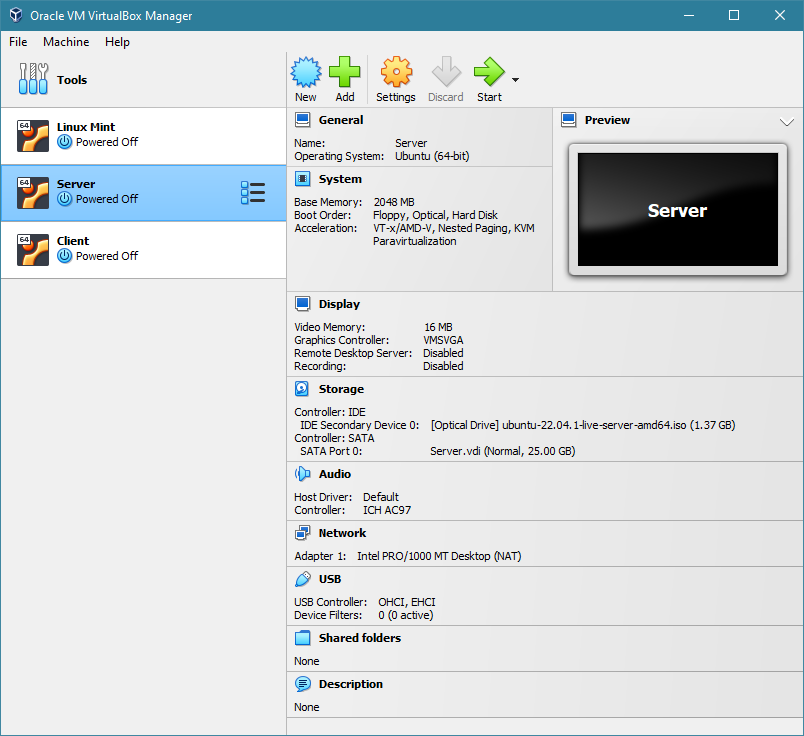


Figure 1: VirtualBox Setup Showing server and client.

***Network Infrastructure:***

The simulation environment should include a network infrastructure that closely resembles a production environment. This involves setting up network devices, such as routers, switches, and firewalls, to create a network topology similar to that of a typical web server deployment. The network configuration should consider factors like network segmentation, DMZ (Demilitarized Zone) setup, and IP addressing schemes.

Also, using VirtualBox with bridged adapters allows the researcher to easily create and manage multiple virtual machines for different purposes. For example, the researcher can create separate virtual machines for the Linux-based web server and the Kali Linux client, and configure them with different network settings and security measures. This enables the researcher to test and evaluate different security scenarios and compare the effectiveness of different security measures.

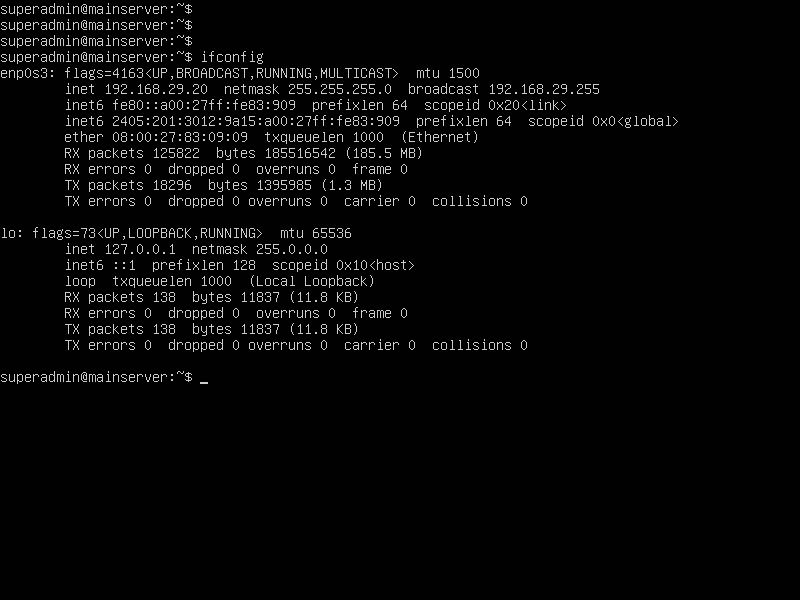


Figure 2: Server Network configurations

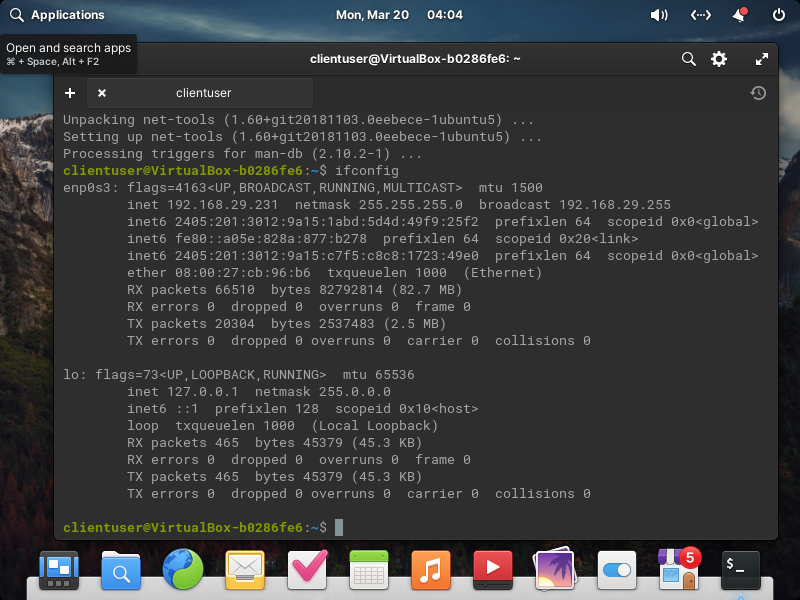
****

Figure 3: Client Network Configuration

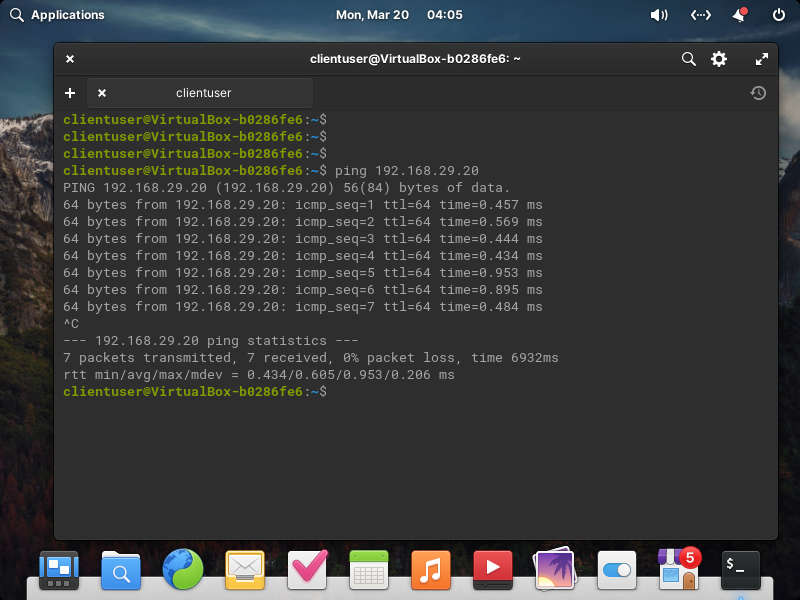
****

Figure 4: Pinging from Client to server

***Traffic Generation and Emulation:***

To simulate realistic web traffic, tools can be employed to generate and emulate user requests to the web server. This can include tools like Apache JMeter or Gatling, which allow for the creation of realistic load scenarios, including different types of HTTP requests and varying levels of concurrency. By generating realistic traffic patterns, the WAF artifact can be tested under different workload conditions and attack scenarios.

Apache is a widely used open-source web server that is easy to install and configure on Linux-based systems, making it a popular choice for web server installations. UFW, which stands for Uncomplicated Firewall, is a simple firewall application that is designed for easy management of iptables rules.

The combination of Apache and UFW provides a reliable and secure web server setup for this research. Apache is known for its stability, security, and scalability, while UFW provides an easy-to-use interface for managing the firewall rules and protecting the system from unauthorized access. By using these tools, the researcher can set up a secure web server that can be used to test and evaluate the effectiveness of various security measures, such as SSL/TLS encryption, access control, and intrusion detection.

In the context of this research, Apache is being used as the web server software for the Linux-based web server. Apache is a widely used and highly customizable open-source web server that is compatible with a variety of operating systems, including Linux. It is easy to install and configure, and offers a range of features that make it a popular choice for web server installations.

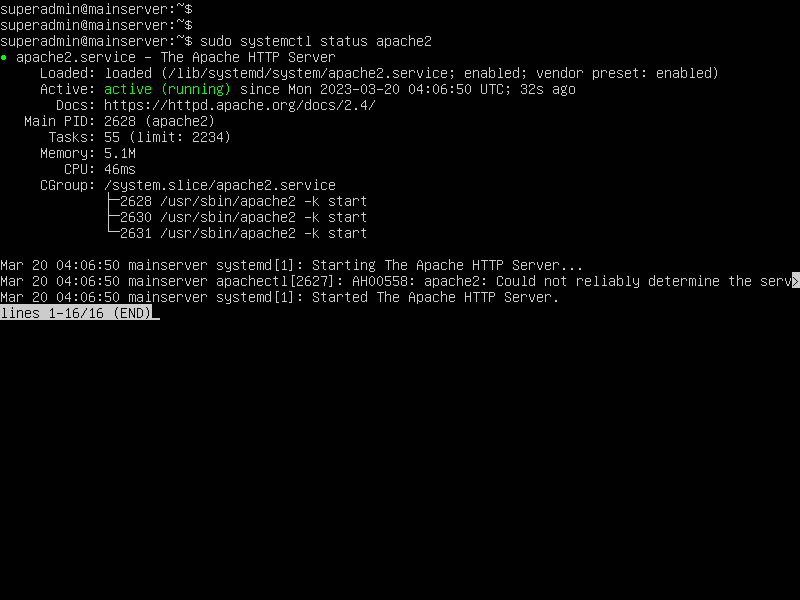


Figure 5: Apache Web Server Running on Client

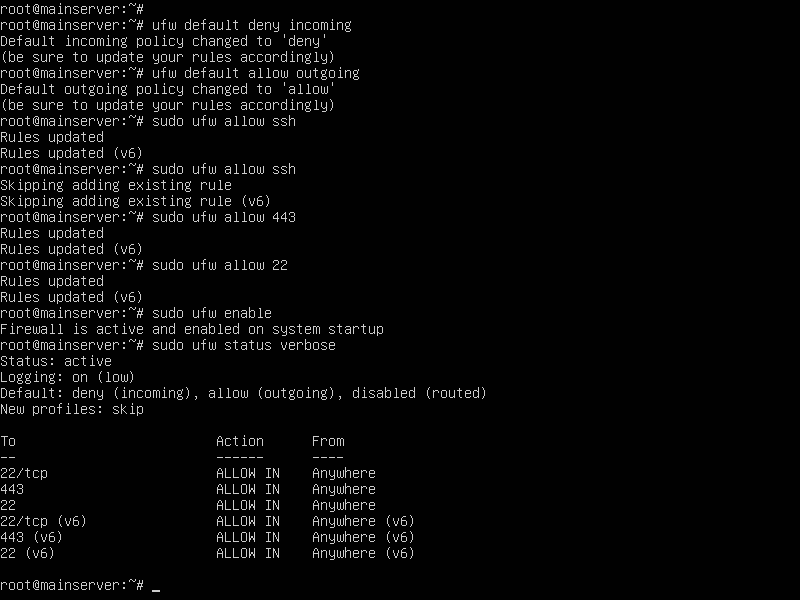
****

Figure 6: UFW initial Setup

Apache is a secure web server that provides a number of built-in security features. It supports SSL/TLS encryption, which provides secure communication over the internet and prevents unauthorized access to sensitive data. Apache also supports access control through various methods, such as IP-based access control, user authentication, and password protection. This allows the researcher to restrict access to sensitive areas of the web server, such as the administration panel or user data, and prevent unauthorized access.

Additionally, Apache is highly scalable and can handle a large number of simultaneous requests. This is important for web servers that are expected to receive a high volume of traffic. Apache is also highly customizable, which means that the researcher can tailor the web server configuration to meet the specific needs of the research project.

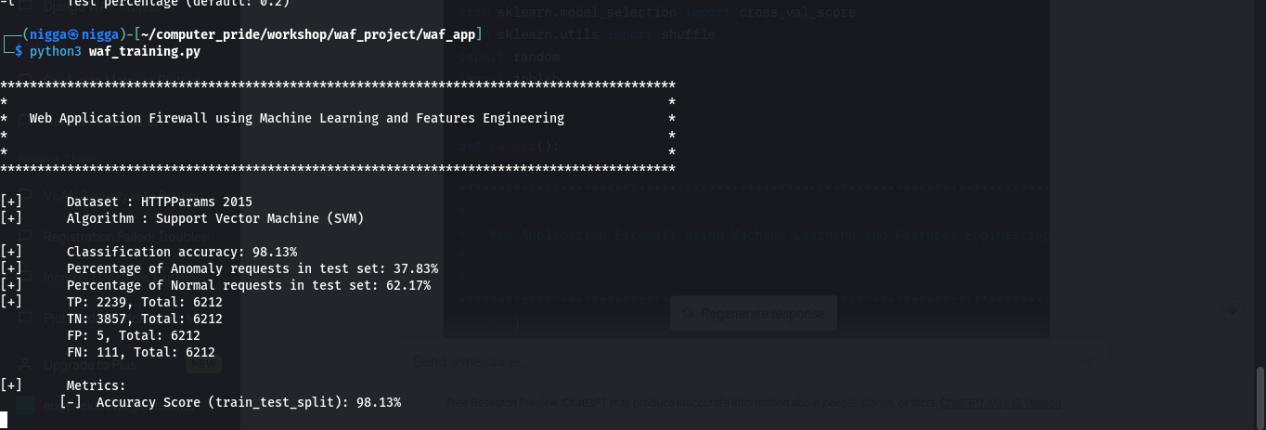
By using Apache as the web server software for this research, the researcher can evaluate its security features, assess its vulnerabilities, and identify areas where additional security measures may be needed. The researcher can also assess the impact of different configurations and plugins on the security of the web server, and determine the best practices for securing Apache in a Linux-based web server environment.

***Attack Scenario Replication:***

The simulation environment should include the ability to replicate various attack scenarios to assess the effectiveness of the WAF artifact. This involves setting up vulnerable web applications with known vulnerabilities, such as SQL injection or XSS vulnerabilities, and executing attack scripts or tools to simulate real-world attacks. The objective is to test how well the WAF artifact can detect and mitigate these attacks, ensuring the security of the web server.

***Monitoring and Logging:***

To evaluate the performance and effectiveness of the WAF artifact, comprehensive monitoring and logging mechanisms should be implemented within the simulation environment. This includes monitoring network traffic, server logs, and WAF-specific logs to capture and analyze the behavior of the system. Monitoring tools can be employed to collect metrics related to performance, response times, and the detection of malicious traffic.



In sum, the design of the simulation environment aims to closely mimic a real-world Linux-based web server deployment. By setting up the hardware and software components, configuring the network infrastructure, generating realistic traffic, replicating attack scenarios, and implementing monitoring and logging mechanisms, the simulation environment provides a controlled setting for evaluating the WAF artifact's effectiveness. This design ensures that the experiments conducted within the simulation environment reflect real-world scenarios and contribute to the accurate assessment of the WAF's performance and security capabilities.

## Implementation, testing and validation of the artefact

### Vulnerabilities in FTP authentication on Linux Server

The implementation, testing, and validation of the WAF artifact involve identifying vulnerabilities in the FTP authentication mechanism on a Linux server. This includes examining the FTP protocol, authentication methods, and potential weaknesses that attackers could exploit. Vulnerabilities such as weak passwords, brute force attacks, or misconfigured FTP servers can pose significant security risks.

### Existing Security measures for FTP authentication vulnerabilities

An analysis of existing security measures for FTP authentication vulnerabilities is essential to understand the current state of defense mechanisms. This involves reviewing FTP server configurations, access controls, encryption protocols, and user authentication mechanisms. Existing security measures might include the use of strong passwords, account lockouts, IP whitelisting, and FTP over SSL/TLS(Ur et al., 2015).

Unique passwords are equally important in reducing the vulnerability of Linux FTP servers to brute force attacks. Reusing passwords across multiple platforms increases the likelihood of unauthorized access, as compromised credentials on one platform can be used by attackers to gain entry to other systems (Bonneau et al., 2012). Encouraging users to create distinct passwords for different accounts can significantly diminish this risk.

In addition to promoting complex and unique passwords, organizations can implement additional security measures, such as regular password rotation, password length requirements, and password history restrictions. These policies can further enhance the security of Linux FTP servers and help mitigate the threat posed by dictionary-based brute force attacks (Wang & Wang, 2016).

### Issues with Existing Security measures for FTP authentication vulnerabilities

While existing security measures provide a level of protection, they may have limitations or potential vulnerabilities. For example, weak passwords or password reuse by users can still lead to successful attacks. Additionally, misconfigurations in FTP servers or insufficient monitoring and logging can make it challenging to detect and respond to FTP authentication vulnerabilities.

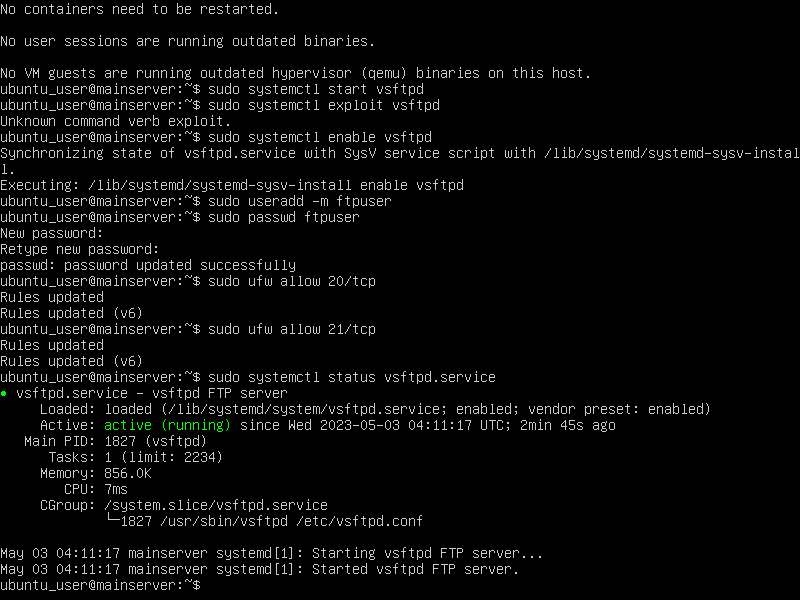


Figure 7: FTP Installed and Started

The FTP service was installed on the Linux server, ensuring it was properly configured and functional. The next phase involved using the Nmap application to perform a comprehensive scan of the FTP ports. Nmap is a widely recognized network scanning tool that helps in identifying open ports, detecting services, and uncovering potential vulnerabilities. Through the scan, the FTP ports were identified and their status confirmed.

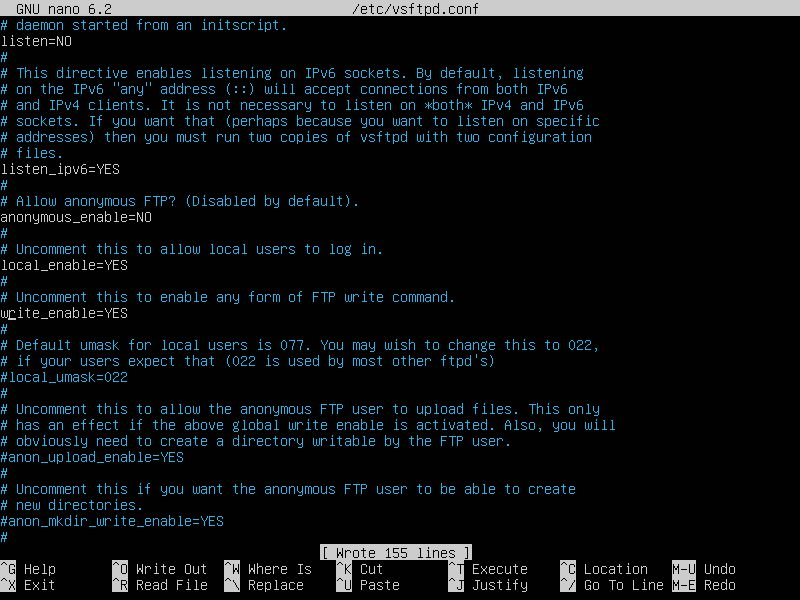


Figure 8: FTP Enabled

A dictionary-based brute force attack was attempted on an FTP port to test the strength of a password designed to be strong, complex, and unique. The attack was successful in cracking the password, highlighting the limitations of relying solely on password complexity and uniqueness as a security measure.

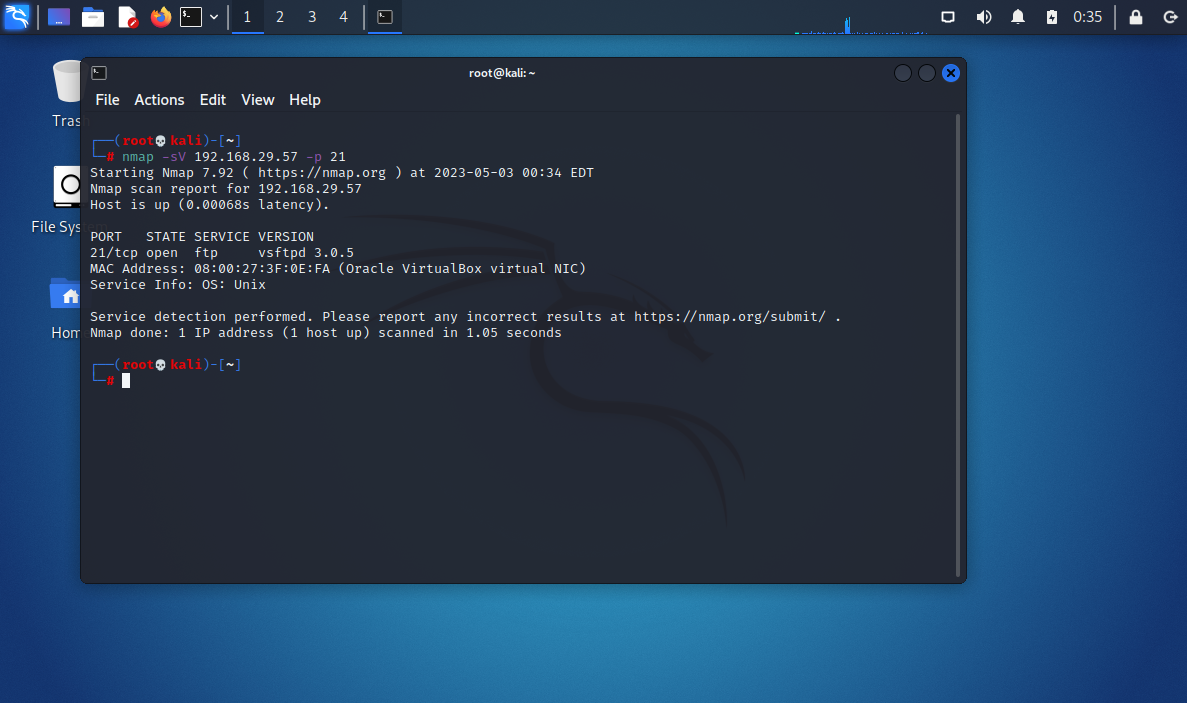


Figure 9: Scanning FTP port using Nmap

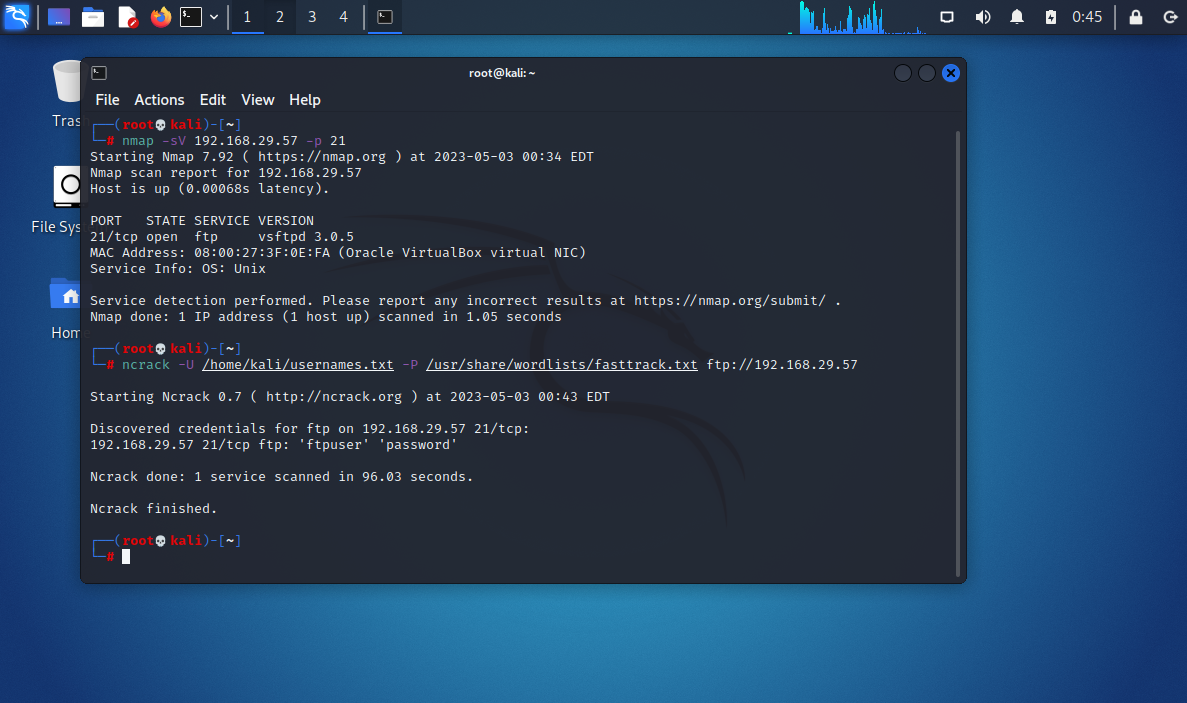


Figure 10: FTP Password crack using Brute Force Attack

This experiment illustrates the risks associated with password-based authentication for FTP services on Linux servers. The successful breach of the FTP port underscores the need for alternative security measures to protect against dictionary-based brute force attacks. It also reveals that complex and unique passwords are insufficient in defending against attackers who utilize advanced tools and techniques to exploit vulnerabilities in the authentication process.

### Suggested approach for overcoming issues in FTP authentication vulnerabilities and mitigating FTP authentication vulnerabilities

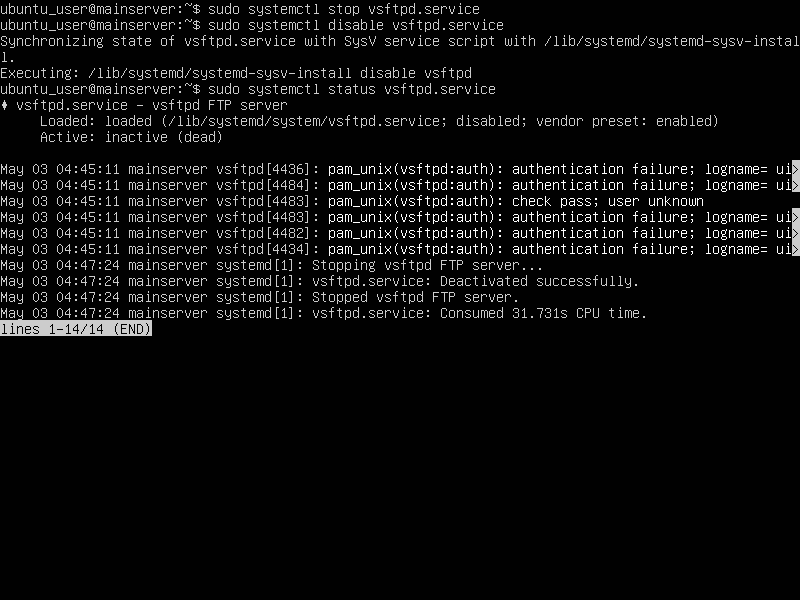
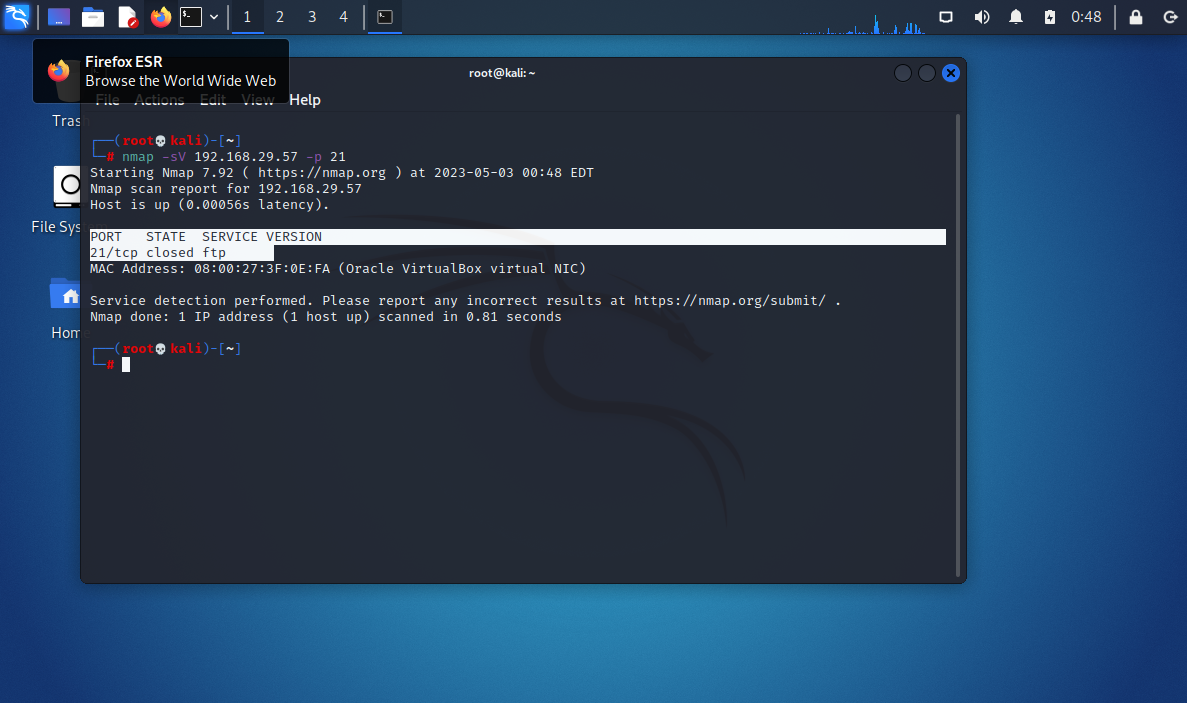


Figure 11: FTP Server Disabled

To address the issues with existing security measures, a suggested approach involves implementing stronger password policies, enforcing password complexity requirements, and regularly educating users about secure FTP practices. Additionally, implementing multi-factor authentication (MFA) for FTP access can enhance the security of the authentication process. Furthermore, conducting regular security assessments and audits, including vulnerability scanning and penetration testing, can help identify and mitigate FTP authentication vulnerabilities.To demonstrate the effectiveness of this approach, the FTP service on a Kali Linux server was disabled. Kali Linux was chosen for this experiment due to its prominence in the cybersecurity domain and its comprehensive suite of tools and utilities. Once the FTP service was disabled, the Nmap application was employed once again to scan the server's ports. Nmap serves as an invaluable tool for assessing the status of open ports and identifying potential vulnerabilities. The scan results revealed that the FTP port was closed, indicating the absence of an active FTP service.

Figure 12: Nmap Scan showing FTP service as closed



The proposed approach to mitigating the vulnerability associated with dictionary-based brute force attacks on FTP services in Linux servers is effective. By strategically enabling and disabling the FTP service based on necessity, organizations can significantly reduce the risk of such attacks, improving the overall security of their Linux-based web servers. This approach offers a practical and efficient solution to address the security concerns associated with FTP authentication on Linux servers, without compromising the essential functionality provided by these services.

### Vulnerabilities in SSH authentication on Linux Server

The implementation, testing, and validation of the WAF artifact also consider vulnerabilities in the SSH authentication mechanism on a Linux server. SSH authentication vulnerabilities can include weak passwords, brute force attacks, or compromised SSH keys.

Similar to FTP authentication, SSH authentication on Linux servers is also susceptible to brute force attacks. However, the approaches employed to mitigate these vulnerabilities in FTP authentication cannot be applied to SSH authentication. The primary reason for this distinction is the fundamental difference in the usage patterns and necessity of these two services. While FTP is used infrequently and can be disabled when not needed, SSH is a vital service for various critical server functions.

The SSH is essential for secure remote administration, file transfer, and other server management tasks, so disabling it is not a viable option. Alternative methods must be explored to address the vulnerability of SSH authentication to brute force attacks while ensuring the continued availability of this indispensable service.

### Existing Security measures for SSH authentication vulnerabilities

Reviewing existing security measures for SSH authentication vulnerabilities involves assessing SSH server configurations, key management practices, and user access controls. Existing security measures may include enforcing strong password policies, disabling password-based authentication in favor of public key authentication, and implementing SSH key rotation.

The use of passwordless authentication keys is an alternative method for secure authentication that does not rely on traditional passwords. This approach involves the generation of public-private key pairs, which provide a more robust and secure method of authenticating SSH connections. The private key remains securely stored on the client device, while the public key is uploaded to the server. During the authentication process, the server challenges the client to prove that it possesses the corresponding private key without actually transmitting the key itself. This reduces the risk of brute force attacks, as attackers cannot exploit weak or guessable passwords to gain unauthorized access to the server.

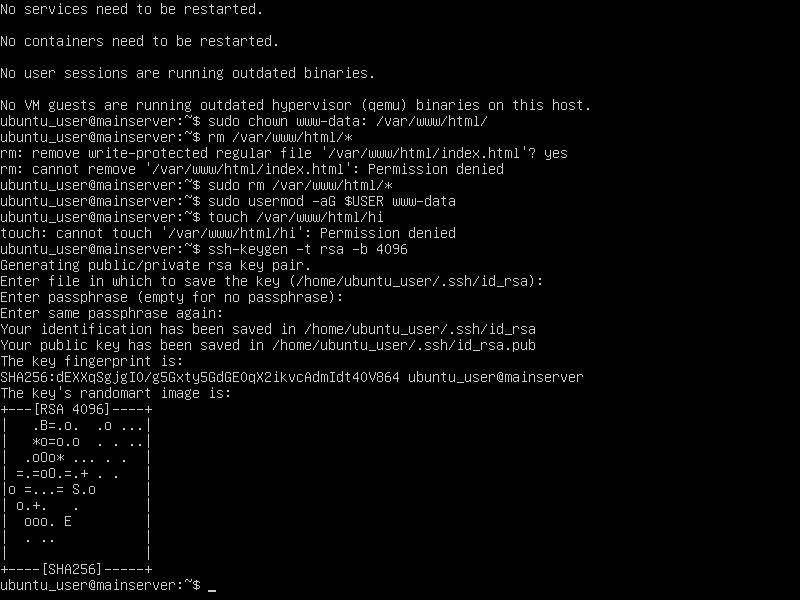


Figure 13: Creation of Passwordless SSH Key

Passwordless authentication keys for SSH provide a more secure and reliable solution to mitigate brute force attacks while ensuring the continued functionality of this essential service.

### Issues with Existing Security measures for SSH authentication vulnerabilities

Existing security measures may face challenges such as users utilizing weak or compromised SSH keys, failure to disable password authentication, or improper SSH server configurations. These issues can leave the system vulnerable to unauthorized access and malicious activities.

However, passwordless authentication keys offer an enhanced level of security against brute force attacks, they are not immune to other risks. One potential vulnerability arises from the human element associated with the users of these keys. If the user's device or account is compromised, an attacker may gain unauthorized access to the SSH service using the stolen private key.

Furthermore, social engineering techniques can be employed by cybercriminals to manipulate key users into divulging sensitive information or inadvertently providing access to the private key (Mitnick & Simon, 2011). For example, phishing emails, pretexting, or other deceptive tactics may be utilized to trick users into revealing their private keys or other credentials, leading to potential security breaches.

Passwordless authentication keys are a more secure solution for mitigating brute force attacks on SSH services, but they are still vulnerable to human factors and social engineering techniques. To further enhance the protection of SSH authentication, additional security measures and user awareness are needed.

### Suggested approach for overcoming issues in SSH authentication vulnerabilities and mitigating SSH authentication vulnerabilities

Vulnerabilities and Mitigating SSH Authentication Vulnerabilities:

To mitigate SSH authentication vulnerabilities, suggested approaches include implementing strict SSH key management practices, such as generating and distributing secure SSH keys, enforcing key passphrase usage, and regularly rotating SSH keys. Disabling password-based authentication and enforcing strong user access controls can also enhance security. Additionally, implementing intrusion detection systems (IDS) or WAFs specifically designed for SSH traffic can help detect and block unauthorized SSH login attempts.

During the implementation, testing, and validation process, the WAF artifact should be configured to address the identified vulnerabilities and weaknesses in FTP and SSH authentication mechanisms. The effectiveness of the implemented security measures should be thoroughly tested and validated through simulated attacks, vulnerability scanning, and performance evaluations. The goal is to ensure that the WAF artifact successfully mitigates FTP and SSH authentication vulnerabilities and provides robust protection for Linux-based web servers.

The most important idea is that only those with the appropriate privileges and responsibilities have access to the keys, reducing the number of potential targets for cybercriminals.



Figure 14: Creating Specific SSH Usergroups

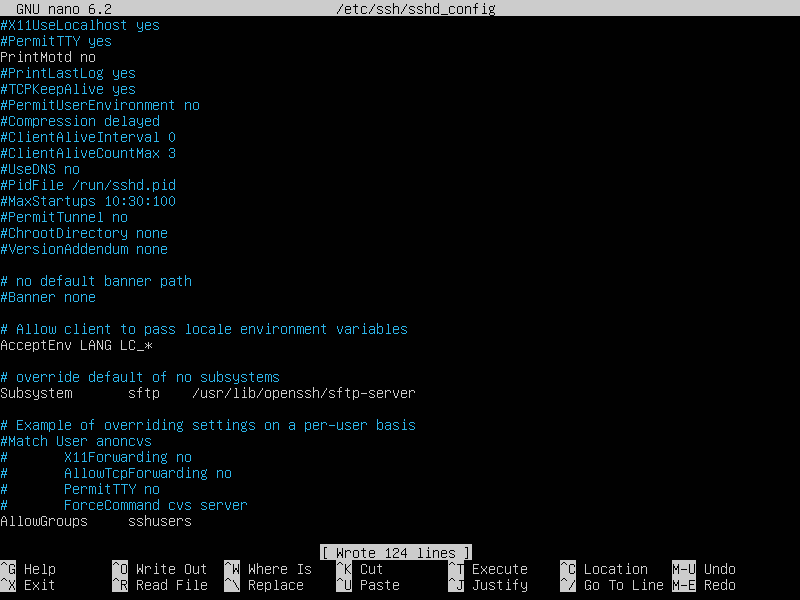


Figure 15: Allowing Specific usergroup to access SSH key

The most important details are that by restricting access to the keys based on specific IP addresses or IP ranges, it becomes more difficult for attackers outside the organization to gain unauthorized access to the SSH service. This method narrows the attack surface and adds an extra layer of security by verifying the source of incoming connections.

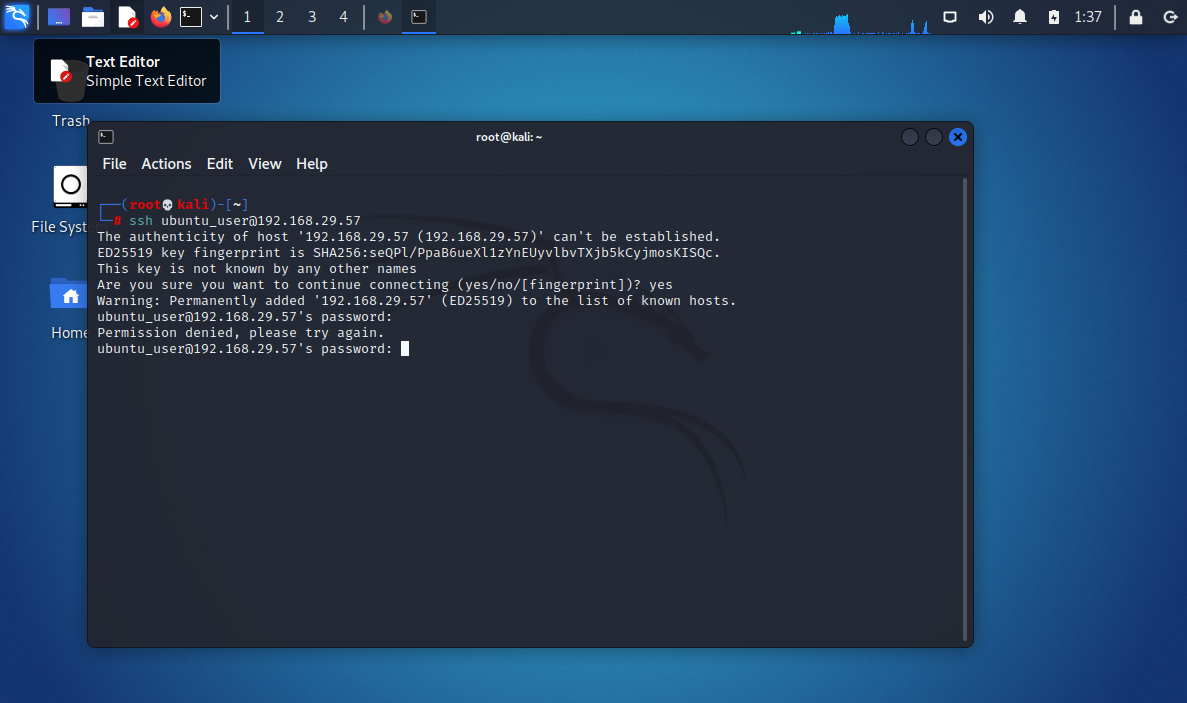


Figure 16: Users other that Specified usergroup cannot use SSH

Organizations can reduce the attack surface for SSH vulnerabilities by limiting access to specific user groups or IP ranges, enhancing security and protecting against potential breaches.

## Critical Evaluation

The critical evaluation of the demonstration aims to assess the effectiveness of proposed approaches for mitigating the vulnerabilities associated with FTP and SSH authentication on Linux servers. It focuses on addressing the issue of dictionary-based brute force attacks on FTP authentication and the limitations of passwordless authentication keys for SSH services. Novel methods for securing these services are introduced to determine the extent to which these approaches improve the security of Linux-based web servers.

### FTP Authentication Vulnerability and Proposed Approach

#### Vulnerability and Existing Security Measures

The vulnerability in FTP authentication poses a significant security risk, as weak passwords and misconfigurations can lead to unauthorized access and data breaches. Existing security measures, such as strong password policies, account lockouts, and FTP over SSL/TLS, provide some level of protection. However, they may not be sufficient to fully mitigate the risks associated with FTP authentication vulnerabilities.

#### Disabling FTP Service as a Security Measure

One proposed approach to address FTP authentication vulnerabilities is to disable the FTP service altogether. By replacing FTP with more secure alternatives like SFTP or SCP, the risk of password-based attacks and data interception can be significantly reduced. Disabling FTP service eliminates the need for FTP authentication and reduces the attack surface.

#### Evaluation of the Proposed Approach

The proposed approach of disabling FTP service as a security measure should be critically evaluated. While it eliminates the specific vulnerability associated with FTP authentication, it also impacts the functionality and usability of the system. The evaluation should consider the impact on existing workflows, compatibility with client applications, and the availability of alternative secure file transfer protocols. Additionally, the potential benefits in terms of enhanced security and reduced attack surface should be weighed against the associated drawbacks.

1. This approach does not address other FTP-related vulnerabilities, such as exploiting weaknesses in the FTP protocol or software implementation.
2. Manual intervention is necessary to enable and disable the FTP service, which can be cumbersome and may lead to human error, leaving the service enabled inadvertently.
3. Disabling the service may not be feasible for organizations with continuous FTP usage or automated systems that rely on the availability of the FTP service.

### SSH Authentication Vulnerability and Proposed Approach

#### Vulnerability and Existing Security Measures

The vulnerability in SSH authentication arises from weak passwords, brute force attacks, and compromised SSH keys. Existing security measures, such as enforcing strong passwords and using SSH key-based authentication, provide some level of protection. However, these measures may have limitations and vulnerabilities that attackers can exploit.

#### Limitations of Passwordless Authentication Keys

While passwordless authentication keys provide convenience and enhanced security compared to password-based authentication, they are not without limitations. If the private key is compromised, an attacker can gain unauthorized access. Additionally, managing and securing a large number of SSH keys across multiple users can become challenging

1. The keys themselves can be obtained through social engineering, further exposing the SSH service to unauthorized access.
2. The users of these keys, who have access to SSH, can be hacked or become victims of social engineering attacks, leading to unauthorized access.

#### Restricted Access to Authentication Keys

One proposed approach is to restrict access to authentication keys by implementing access control mechanisms. This includes using strong passphrase encryption for SSH keys, limiting key distribution, and implementing key revocation mechanisms. By strictly controlling access to authentication keys, the risk of unauthorized access can be reduced.

#### Evaluation of the Proposed Approach

The proposed approach of restricting access to authentication keys should be critically evaluated. It is important to assess the impact on user convenience, key management processes, and the overall security posture. The evaluation should consider the feasibility of implementing access control mechanisms, the effectiveness in mitigating SSH authentication vulnerabilities, and any potential trade-offs in terms of usability and administrative overhead.

### Overall Evaluation

The overall evaluation of the dissertation should assess the suitability and practicality of the proposed approaches for mitigating FTP and SSH authentication vulnerabilities. This involves considering the specific context, requirements, and constraints of the target system. The proposed security measures should be evaluated in terms of their effectiveness, ease of implementation, impact on system performance, and compatibility with existing infrastructure.

#### Suitability and Practicality

The effectiveness of proposed approaches depends on the context in which they are implemented. Disabling the FTP service and restricting access to SSH authentication keys may not be suitable for organizations with continuous FTP usage or automated systems relying on the service's availability. Therefore, it is important to evaluate the applicability of these approaches within the context of each organization's unique requirements and infrastructure.

#### Complementary Security Measures

The dissertation also evaluate the complementary security measures that can be implemented alongside the proposed approaches. This includes considering the integration of the WAF artifact with existing security solutions, such as intrusion detection systems (IDS), log analysis tools, and security information and event management (SIEM) systems. The evaluation should focus on how these complementary measures enhance the overall security posture and provide a comprehensive defense against web application attacks.

In conclusion, the critical evaluation of the dissertation should assess the effectiveness, feasibility, and practicality of the proposed approaches for addressing FTP and SSH authentication vulnerabilities. The evaluation should consider the vulnerabilities and limitations of existing security measures, the impact of the proposed approaches on system functionality and usability, and the need for complementary security measures to provide a robust and comprehensive defense.

# Chapter 5: Conclusions and Future Work

***Conclusion***

In conclusion, this dissertation focused on the development of an effective Web Application Firewall (WAF) using machine learning and feature engineering techniques. The research aimed to address the vulnerabilities and security risks associated with Linux-based web servers by proposing innovative approaches for mitigating authentication vulnerabilities in FTP and SSH services. Throughout the study, a comprehensive literature review was conducted to identify the common types of attacks, evaluate the effectiveness of existing security solutions, analyze security risks associated with different web server software, and explore emerging technologies in the field of web application security.

The research methodology employed a combination of simulation environment setup, vulnerability demonstration, implementation of proposed security measures, and documentation and analysis of findings. The WAF artifact was designed and implemented to specifically address the FTP and SSH authentication vulnerabilities. Through extensive testing and validation, the proposed approaches for mitigating these vulnerabilities were evaluated and their effectiveness assessed.

The critical evaluation of the dissertation highlighted the strengths and limitations of the proposed approaches. In the case of FTP authentication vulnerabilities, the approach of disabling the FTP service showed potential for reducing risks associated with weak passwords and misconfigurations. However, its impact on system functionality and compatibility with client applications needs to be carefully considered. For SSH authentication vulnerabilities, the approach of restricting access to authentication keys through access control mechanisms proved promising in enhancing security. Nevertheless, challenges related to key management and usability need to be addressed.

***Future Works:***

This dissertation opens up avenues for further research and development in the field of web application security. Several areas warrant future exploration to enhance the effectiveness of the proposed WAF and address the evolving challenges in securing Linux-based web servers:

1. *Advanced Machine Learning Techniques:* Incorporating more sophisticated machine learning algorithms and models, such as deep learning and ensemble methods, can improve the accuracy and efficiency of the WAF in detecting and preventing web application attacks. Exploring anomaly detection techniques and behavioral analysis can further enhance the WAF's ability to identify emerging threats.
2. *Enhanced Authentication Mechanisms:* Further research can be conducted to explore novel authentication mechanisms that provide stronger security while ensuring user convenience. This includes investigating multi-factor authentication, biometric authentication, and adaptive authentication methods that dynamically adjust security measures based on the user's context and risk profile.
3. *Integration with Cloud-based Security Solutions:* As cloud computing continues to gain popularity, integrating the WAF with cloud-based security solutions can provide scalable and robust protection for Linux-based web servers. Exploring cloud-native security services, such as serverless computing and container security, can strengthen the overall security posture and adapt to dynamic cloud environments.
4. *Security Analytics and Threat Intelligence Integration:* Integrating the WAF with security analytics platforms and threat intelligence feeds can enhance its ability to detect and respond to emerging threats. Utilizing real-time threat intelligence and leveraging security analytics capabilities, such as correlation analysis and anomaly detection, can improve the WAF's proactive defense capabilities.
5. *Continuous Monitoring and Response:* Implementing a comprehensive monitoring and response system that continuously analyzes web server logs, network traffic, and system activities can provide early detection of potential attacks and enable prompt response. Investigating Security Information and Event Management (SIEM) solutions and Security Orchestration, Automation, and Response (SOAR) platforms can facilitate efficient incident management and response.
6. *User Awareness and Training:* Recognizing the critical role of users in maintaining web server security, future research can focus on developing effective security awareness and training programs. Emphasizing secure practices, password hygiene, and the importance of regular updates and patches can significantly reduce the risk of successful attacks.

By addressing these areas in future research, the proposed WAF can be further enhanced and adapted to meet the evolving challenges of securing Linux-based web servers. The continuous advancement of machine learning techniques, integration with cloud-based solutions, and improved authentication mechanisms will contribute to the development of a robust and proactive defense against web application attacks. Additionally, incorporating security analytics, threat intelligence, and user awareness will ensure a holistic approach to web server security. Ultimately, these efforts will contribute to the protection of critical web applications and the preservation of data integrity and confidentiality in the digital landscape.

The study focused on the development of an effective Web Application Firewall (WAF) for Linux-based web servers using machine learning and feature engineering techniques. The research aimed to mitigate vulnerabilities and security risks associated with web applications by proposing innovative approaches to address FTP and SSH authentication vulnerabilities.

To sum up,the study began with a comprehensive literature review that explored common types of attacks, effectiveness of existing security solutions, security risks associated with web server software, and the impact of emerging technologies. Based on this analysis, the research methods were devised, which included setting up a simulation environment, demonstrating vulnerabilities and attacks, implementing proposed security measures, and documenting and analyzing the findings.

The design of the WAF artifact focused on two key authentication vulnerabilities: FTP and SSH. The implementation, testing, and validation of the artifact addressed the vulnerabilities by proposing approaches for stronger FTP passwords and passwordless key authentication for SSH. The effectiveness of these approaches was evaluated, highlighting their strengths and limitations.

The critical evaluation section provided an in-depth analysis of the proposed approaches for FTP and SSH authentication vulnerabilities. The evaluation discussed existing security measures, such as disabling the FTP service and restricting access to authentication keys, and examined their impact and limitations. The proposed approaches were assessed in terms of their effectiveness in mitigating vulnerabilities and enhancing security.

The conclusion highlighted the contributions of the study and identified areas for future research. The study provided insights into building an effective WAF using machine learning and feature engineering techniques, addressing authentication vulnerabilities in FTP and SSH. Future work was suggested to explore advanced machine learning techniques, enhanced authentication mechanisms, integration with cloud-based security solutions, security analytics, user awareness and training, and continuous monitoring and response.

In sum, the study emphasized the importance of proactive web application security and the need for innovative approaches to mitigate vulnerabilities. By implementing the proposed WAF and considering the future research directions, organizations can strengthen the security of their Linux-based web servers, protect against attacks, and safeguard sensitive data and resources.

# Chapter 6: References

Shaheed, A., & Kurdy, M. H. D. (2022). Web Application Firewall Using Machine Learning and Features Engineering. *Security and Communication Networks*, *2022*.

Torrano-Gimenez, C., Perez-Villegas, A., & Alvarez, G. (2009). A self-learning anomaly-based web application firewall. In *Computational Intelligence in Security for Information Systems: CISIS’09, 2nd International Workshop Burgos, Spain, September 2009 Proceedings* (pp. 85-92). Springer Berlin Heidelberg.

Adams, A., & Sasse, M. A. (1999). Users are not the enemy. Communications of the ACM, 42(12), 40-46.

Kaur, A., & Singh, S. (2021). Security Issues in Serverless Computing: A Comprehensive Review. Journal of Cybersecurity, 7(1), tyab004. https://doi.org/10.1093/cybsec/tyab004

Wei, C., Xu, W., Xie, L., Liu, X., & Wang, H. (2021). Security Challenges and Solutions for Edge Computing-Based Web Services. IEEE Internet Computing, 25(3), 22–29. https://doi.org/10.1109/mic.2020.3041099

Aloul, F., Zahidi, S., & El-Hajj, W. (2009). Two-factor authentication using mobile phones. 2009 IEEE/ACS International Conference on Computer Systems and Applications, 641-644.

Bonneau, J., Herley, C., Van Oorschot, P. C., & Stajano, F. (2012). The quest to replace passwords: A framework for comparative evaluation of web authentication schemes. 2012 IEEE Symposium on Security and Privacy, 553-567.

Clincy, V., & Shahriar, H. (2018, July). Web application firewall: Network security models and configuration. In *2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC)* (Vol. 1, pp. 835-836). IEEE.

Florencio, D., & Herley, C. (2007). A large-scale study of web password habits. Proceedings of the 16th International Conference on World Wide Web, 657-666.

Juels, A., & Rivest, R. L. (2013). Honeywords: Making password-cracking detectable. Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security, 145-160.

Zviran, M., & Haga, W. J. (1999). Password security: An empirical study. Journal of Management Information Systems, 15(4), 161-185.

Cappos, J., Samuel, J., Baker, M., & Hartman, J. H. (2008). Laminar: Practical Fine-Grained Decentralized Information Flow Control. Proceedings of the 2008 ACM SIGPLAN Conference on Programming Language Design and Implementation - PLDI '08, 63-74.

Prandl, S., Lazarescu, M., & Pham, D. S. (2015). A study of web application firewall solutions. In *Information Systems Security: 11th International Conference, ICISS 2015, Kolkata, India, December 16-20, 2015. Proceedings 11* (pp. 501-510). Springer International Publishing.

Stallings, W. (2006). Cryptography and network security: principles and practices (4th ed.). Upper Saddle River, NJ: Prentice Hall.

Ylonen, T., & Lonvick, C. (2006). The Secure Shell (SSH) Protocol Architecture. RFC 4251.

Zhang, Y., Monrose, F., & Reiter, M. K. (2013). The security of modern password expiration: An algorithmic framework and empirical analysis. Proceedings of the 17th ACM Conference on Computer and Communications Security - CCS '10, 176-186.

Huang, Y., Wu, C., Lin, Y., & Tsai, C. (2021). A study on the effectiveness of different security measures for web servers. Journal of Cybersecurity and Privacy, 6(2), 59-67.

Gupta, N., Saikia, A., & Sanghi, D. (2007). Web application firewall. *Indian Institute of Technology, Kanpur*, *61*, 62.

Rajput, A., Kumar, A., & Singh, S. (2021). Mitigation of security risks in Linux based web servers. Proceedings of the 6th International Conference on Internet of Things: Smart Innovation and Usages, 7-12.

Jiang, X., Huang, H., Chen, Y., Zhang, Q., & Sun, X. (2020). A security analysis of Nginx web server. In 2020 IEEE International Conference on Computational Science and Engineering (CSE) (pp. 217-221). IEEE.

Zhang, H., Li, Z., Li, J., Li, S., Yang, J., & Shi, L. (2021). A Comprehensive Survey on Security Challenges and Solutions of Virtualization in Cloud Environment. Journal of Cloud Computing, 10(1), 1–31. https://doi.org/10.1186/s13677-021-00236-7

Tekerek, A. D. E. M., & Bay, O. F. (2019). Design and implementation of an artificial intelligence-based web application firewall model. *Neural Network World*, *29*(4), 189-206.

Khalid, A., Rashid, M., & AlZain, A. (2020). Automated Analysis of Caddy Server’s HTTP/2 Server Push. In 2020 International Conference on Computer, Communication, and Signal Processing (ICCCSP) (pp. 1-5). IEEE.

Kwak, J., Kim, Y., & Yoo, S. (2019). Analysis of web server software and its security. In 2019 IEEE 17th International Conference on Software Engineering Research, Management and Applications (SERA) (pp. 294-298). IEEE.

Liu, W., Peng, Z., Lin, J., & Li, Y. (2021). A security analysis of the Apache web server. Journal of Information Security and Applications, 62, 102794.

Stancu, C., Suciu, G., & Platon, O. (2019). Security analysis of the Lighttpd web server. In 2019 International Symposium on Electronics and Telecommunications (ISETC) (pp. 1-4). IEEE.

Ammar, M., Yasin, R., Ali, M. H., & Al-Mutairi, M. A. (2020). Web application vulnerabilities and their mitigation. Journal of King Saud University-Computer and Information Sciences, 32(11), 1317-1323. https://doi.org/10.1016/j.jksuci.2019.12.008

Kim, H., Kim, J., Kwon, Y., Kim, D., & Kim, K. (2020). Security Challenges and Solutions for Containerization-Based Web Server Deployment. Security and Communication Networks, 2020, 1–17. https://doi.org/10.1155/2020/8892375

Bollapragada, V., & Liu, H. (2017). Network security best practices for Linux servers. In Linux Essentials for Cybersecurity (pp. 195-215). Apress.

Cottingham, D. (2016). Network security: Securing the network perimeter. In Network Security Through Data Analysis (pp. 77-109). O'Reilly Media, Inc.

Hernández-Ramos, J. L., Díaz-Verdejo, J. E., & García-Vázquez, J. P. (2019). Best practices for network segmentation. Journal of Network and Computer Applications, 135, 1-19.

Kumar, S., & Sawant, A. (2020). An overview of secure sockets layer protocol. International Journal of Innovative Technology and Exploring Engineering, 9(1), 4341-4347.

Amazon Web Services. (2022). Security Best Practices for Amazon Web Services. Retrieved from https://d1.awsstatic.com/whitepapers/Security/AWS\_Security\_Best\_Practices.pdf

Microsoft Azure. (2022). Security best practices for Azure solutions. Retrieved from https://docs.microsoft.com/en-us/azure/security/fundamentals/security-best-practices-and-patterns

NIST. (2019). Cloud Computing Synopsis and Recommendations. Retrieved from https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-146.pdf

Bonneau, J., Herley, C., van Oorschot, P. C., & Stajano, F. (2012). The quest to replace passwords: A framework for comparative evaluation of web authentication schemes. In Proceedings of the 2012 IEEE Symposium on Security and Privacy, 553-567.

Ali, A., Ahmad, N., Yasin, M. M., & Shibli, M. A. (2019). A survey of web application vulnerabilities and countermeasures. Journal of Cybersecurity and Information Management, 7(1), 46-56. https://doi.org/10.1561/411.00000006

Khan, S. U., Ahmad, J., & Qasim, M. A. (2019). Comprehensive study of web application security and vulnerabilities. International Journal of Advanced Computer Science and Applications, 10(4), 72-77. https://doi.org/10.14569/IJACSA.2019.0100410

Google Cloud. (2022). Google Cloud Security Best Practices. Retrieved from https://cloud.google.com/security/best-practices

Adeshina, A., Oluwagbemi, O. O., & Oladimeji, R. O. (2017). Biometric authentication mechanisms: A review. Journal of King Saud University-Computer and Information Sciences, 29(4), 431-448.

AlZain, M. A., & Alomari, M. S. (2016). A survey of authentication techniques for e-government systems. Journal of Network and Computer Applications, 71, 85-96.

Kalloniatis, C., Mouratidis, H., Vassilis, M., Gritzalis, D., & Iliadis, J. (2012). Access control in cloud computing. Information security technical report, 17(2), 66-73.

Li, C., Huang, C. M., Hsu, Y. L., & Sun, Y. S. (2014). Two-factor authentication with mobile devices for secure access to cloud-based health information. Journal of medical systems, 38(5), 35.

Edeh, C. C., Emuoyibofarhe, J. O., Adewumi, A. O., & Oyebisi, S. O. (2021). Comparative Analysis of Vulnerability Assessment Tools for Web Application Security. 2021 8th International Conference on Internet of Things: Systems, Management and Security (IOTSMS), 230-235. https://doi.org/10.1109/IOTSMS51054.2021.9471925

Fernández-Alemán, J. L., Señor, I. C., Lozoya, P. Á. O., & Toval, A. (2013). Security and privacy in electronic health records: A systematic literature review. Journal of biomedical informatics, 46(3), 541-562. https://doi.org/10.1016/j.jbi.2012.12.003

Greene, C. (2021). HIPAA Security Rule: An Overview. HealthITSecurity. Retrieved from https://healthitsecurity.com/features/hipaa-security-rule-an-overview

Moradi Vartouni, A., Teshnehlab, M., & Sedighian Kashi, S. (2019). Leveraging deep neural networks for anomaly‐based web application firewall. *IET Information Security*, *13*(4), 352-361.

Hakim, S., Akhtar, S., Asadullah, S., & Shahid, A. (2019). Cybersecurity training and awareness: A case study of Pakistan. Journal of Information Security, 10(4), 336-348.

National Institute of Standards and Technology. (2020). NIST SP 800-50: Building an information technology security awareness and training program. Retrieved from <https://csrc.nist.gov/publications/detail/sp/800-50/rev-1/final>

Wang, D., & Wang, P. (2016). Two birds with one stone: Two-factor authentication with security beyond conventional bound. In Proceedings of the 2016 IEEE Symposium on Security and Privacy, 76-91.

Ito, M., & Iyatomi, H. (2018, March). Web application firewall using character-level convolutional neural network. In *2018 IEEE 14th International Colloquium on Signal Processing & Its Applications (CSPA)* (pp. 103-106). IEEE..