MCA Semester – IV Project

Name	Shamanth B
USN	221VMTR02324
Elective	Cyber Security
Date of Submission	04/06/2024

January 2024

A study on Lightweight Intrusion Detection System (IDS) for Small Networks

Research Project submitted to Jain Online (Deemed-to-be University)

In partial fulfillment of the requirements for the award of

Master of Computer Applications

Submitted by

Shamanth B

USN

221VMTR02324

Under the guidance of

Harish.J

Professor

DECLARATION

I, Shamanth B, hereby declare that the Research Project Report titled "Lightweigh Intrusion Detection System (IDS) for Small Networks" has been prepared by me under the guidance of Harish J. I declare that this Project work is towards the partial fulfillment of the University Regulations for the award of degree of Master of Computer Applications by Jain University, Bengaluru. I have undergone a project for a period of Eight Weeks. I further declare that this Project is based on the original study undertaken by me and has not been submitted for the award of any degree/diploma from any other University / Institution.

Place: Bengaluru

Date: 04/06/2024

_____

Shamanth B
221VMTR02324

CERTIFICATE

This is to certify that the Project report submitted by Mr./Ms. Shamanth B bearing 221VMTR02324 on the title “Lightweight Intrusion Detection System (IDS) for Small Networks” is a record of project work done by him/ her during the academic year 2023-24 under my guidance and supervision in partial fulfilment of Master of Computer Applications.

Place: Bangalore

Date: 04/06/2024

_____

Prof.,Harish.J

ACKNOWLEDGEMENT

"I am deeply thankful to my university officials, faculty, and other faculty members for their support and guidance during this project. Their expertise and encouragement have been invaluable. Additionally, I extend my gratitude to my family, friends, and all those who supported me throughout this journey."

Shamanth.B
221VMTR02324

Executive Summary

In today's digital landscape, the security of small networks is becoming increasingly vital, especially for educational institutions and small businesses. An Intrusion Detection System (IDS) serves as a critical component in safeguarding these networks against unauthorized access and cyber threats. This project aims to develop a robust IDS tailored for small networks, leveraging both signature-based and anomaly-based detection techniques. The system is designed to detect, log, and alert network administrators of potential security breaches, ensuring the protection of network integrity and sensitive information.

The primary objective of this project is to design and implement an IDS suitable for a small network environment, consisting of up to ten devices. The system will be capable of detecting unauthorized access attempts, identifying and logging suspicious network traffic, and providing real-time alerts to network administrators. The scope includes the installation and configuration of the IDS, the development of custom detection rules, and the implementation of a basic alerting mechanism. The focus is on achieving a balance between effectiveness and simplicity, ensuring that the system is user-friendly and easy to manage.

The project explores two primary IDS techniques: signature-based detection and anomaly-based detection. Signature-based detection relies on predefined rules and known attack signatures, making it effective for detecting well-known threats. However, it may struggle with zero-day attacks. Anomaly-based detection establishes a baseline of normal network behavior and flags deviations from this norm as potential threats. This approach is more adept at identifying novel threats but can produce false positives. The combination of these two techniques ensures a comprehensive approach to threat detection.

The tools and technologies selected for this project include Snort, Python and Wireshark. Snort, an open-source network intrusion detection system, is known for its robust rule set and extensive community support. Python is used for scripting custom functionalities and integrating additional features. Wireshark, an optional tool, provides detailed network traffic analysis and packet inspection.

The test environment consists of virtual machines simulating a small network, including routers, switches, and endpoint devices. This setup allows for controlled testing of various intrusion scenarios. Snort is installed on a central monitoring server, with both basic and custom rules configured to detect common attacks such as port scans, SQL injections, and denial-of-service (DoS) attacks. These rules are continuously updated to include new threat signatures. Python scripts are developed to enhance Snort's functionality, automating tasks such as log parsing, threat level assessment, and alert generation. A simple web-based interface is created using Flask, a Python microframework, to provide administrators with real-time access to intrusion logs and alerts.

The IDS was tested in a controlled environment to evaluate its effectiveness in detecting various types of intrusions. The system successfully identified and logged all simulated attacks, providing real-time alerts. The use of both signature-based and anomaly-based detection techniques ensured a comprehensive approach to threat detection, balancing the need for accuracy with the ability to identify novel threats. The system's strengths include ease of use, comprehensive detection capabilities, and scalability. It is user-friendly, with a simple web interface for monitoring and managing alerts, and can be scaled to accommodate larger networks with minor adjustments. However, anomaly-based detection occasionally produces false positives, requiring further tuning of the baseline behavior. Additionally, the performance of the IDS could be optimized to reduce its impact on network resources.

In conclusion, this project demonstrates the feasibility and effectiveness of implementing an IDS tailored for small networks using open-source tools and custom scripting. By leveraging Snort, Python, and Wireshark, the developed system provides robust intrusion detection capabilities, ensuring the security and integrity of small network environments. Future enhancements could focus on refining anomaly detection algorithms, integrating machine learning techniques, and optimizing system performance for larger networks. The IDS developed in this project offers a practical and scalable solution for small network security, contributing to the broader effort to protect digital assets in an increasingly connected world.

Table of Contents

CHAPTER 1

INTRODUCTION, SCOPE AND BACKGROUND

1. INTRODUCTION, SCOPE AND BACKGROUND

1.1 Overview of Project Case / Business case

The objective of this project is to enhance the security of a small network by implementing an Intrusion Detection System (IDS) using Snort for real-time traffic analysis and Wireshark for detailed packet inspection. In today's digital landscape, even small networks are vulnerable to cyber threats such as unauthorized access and malware. This project aims to address this vulnerability by providing a robust, cost-effective solution utilizing open-source tools. The project involves defining the network environment, installing and configuring Snort and Wireshark, and testing the IDS with simulated attacks. By monitoring network traffic and analyzing suspicious activities, the IDS will alert users to potential security breaches, thus enhancing network protection. This project not only bolsters security for small networks but also offers significant educational value, giving students hands-on experience with critical cybersecurity tools and practices. The outcome will include detailed setup documentation, a testing and evaluation report, and a user guide to ensure the IDS is effective and easy to maintain.

1.2 Problem definition

The project aims to implement an Intrusion Detection System (IDS) specifically designed for small networks, such as home networks or small office setups. An IDS is a cybersecurity tool used to monitor network traffic, detect suspicious activities, and alert users or administrators about potential security threats. Unlike building an application, website, or video, this project involves setting up and configuring software tools like Snort and Wireshark to create a comprehensive security solution. The primary focus is on enhancing network security by implementing real-time monitoring and threat detection capabilities, thus mitigating the risks associated with cyber threats such as malware infections, unauthorized access attempts, and data breaches.

Many small networks, including home networks and small office setups, currently grapple with insufficient security measures, leaving them vulnerable to a myriad of cyber threats. The absence of real-time monitoring tools makes it difficult to promptly detect and respond to security breaches, leaving organizations at risk of data loss, financial harm, and reputational damage. Limited resources often result in basic security infrastructure, leaving these networks susceptible to increasingly sophisticated attacks such as malware infections, phishing attempts, and unauthorized access. Identifying security breaches becomes a challenge without an effective Intrusion Detection System (IDS), often leading to delayed responses and heightened risks. Moreover, adherence to industry regulations and data protection laws becomes cumbersome without adequate security measures in place, potentially resulting in non-compliance penalties and legal ramifications. Therefore, implementing an IDS tailored for small networks, utilizing tools like Snort and Wireshark, emerges as a crucial step towards bolstering network security, mitigating cyber threats, ensuring compliance with regulatory requirements, and safeguarding organizational assets and reputation.

1.3 Project Scope

Aim:

The aim of the project is to implement an effective Intrusion Detection System (IDS) tailored for small networks, enhancing their security posture and mitigating the risks associated with cyber threats.

Objectives:

Deploy an Intrusion Detection System (IDS): Set up and configure software tools like Snort and Wireshark to create a comprehensive IDS solution for real-time monitoring of network traffic.

Real-Time Monitoring: Implement continuous monitoring of network traffic to detect and analyze suspicious activities and potential security breaches as they occur.

Threat Detection and Alerting: Develop mechanisms to identify and classify various types of cyber threats, including malware infections, unauthorized access attempts, and network intrusions. Generate alerts to notify users or administrators about detected security incidents promptly.

Customization and Configuration: Customize and fine-tune the IDS configurations to meet the specific requirements and characteristics of small networks, ensuring optimal performance and accuracy in threat detection.

Testing and Evaluation: Conduct thorough testing and evaluation of the IDS to validate its functionality, performance, and effectiveness in detecting and mitigating cyber threats. Address any identified issues or weaknesses through iterative testing and refinement.

Compliance and Regulatory Requirements: Ensure that the implemented IDS complies with relevant industry regulations and data protection laws, addressing any specific compliance requirements applicable to the organization or client.

Scalability and Future Expansion: Design the IDS solution with scalability in mind, allowing for future expansion and integration with additional security tools or functionalities to meet evolving cybersecurity needs.

Goals:

Enhance the security posture of small networks by implementing a robust Intrusion Detection System.

Detect and mitigate cyber threats in real-time, minimizing the impact of security incidents.

Empower users or administrators with the knowledge and tools necessary to effectively monitor and manage network security.

Ensure compliance with industry regulations and data protection laws.

Lay the foundation for future enhancements and expansions to adapt to evolving cybersecurity threats and requirements.

CHAPTER 2
REVIEW OF LITERATURE

2. REVIEW OF LITERATURE

2.1 Literature Review

In the landscape of modern technology, small networks play a critical role in facilitating connectivity for various purposes, ranging from personal use to small business operations. Despite their size, these networks are not immune to cyber security threats, which can compromise sensitive data and disrupt operations. Recognizing the importance of securing small networks against such threats, there arises a need for effective security measures. One such measure is the implementation of an Intrusion Detection System (IDS), which serves as a vigilant guardian, constantly monitoring network traffic for signs of suspicious activities or potential security breaches.

Implementing an Intrusion Detection System (IDS) for a small network presents an essential endeavor in bolstering cybersecurity defenses and protecting sensitive information from potential threats. Small networks, though modest in scale, are susceptible to a myriad of cybersecurity risks, including malware infections, unauthorized access attempts, and data breaches. By deploying an IDS tailored to the specific requirements and constraints of a small network environment, administrators can gain real-time visibility into network traffic, swiftly detect anomalies, and proactively mitigate security incidents. However, the successful implementation of an IDS for a small network necessitates considerations such as resource limitations, false positives, and ease of management. Adhering to best practices in IDS deployment and maintenance is paramount to ensure optimal effectiveness in safeguarding the network. Through the implementation of an IDS, small network administrators can fortify their defenses, mitigate potential threats, and uphold the integrity and security of their network infrastructure and data assets.

2.2 Feasibility Analysis

For a project aimed at implementing an Intrusion Detection System (IDS) for a small network, conducting a feasibility analysis is crucial to ensure the project's viability and success. From a technical standpoint, it's important to assess the compatibility of available hardware and software resources with IDS requirements. This involves exploring options for open-source IDS solutions that can run on readily available devices or considering the feasibility of implementing IDS functionalities within existing network infrastructure. Financial feasibility entails evaluating the costs associated with acquiring IDS software or hardware, with consideration given to free or open-source solutions to minimize upfront expenses. Operationally, identifying available resources for managing and maintaining the IDS is essential, as is assessing the technical skills required for configuration and operation. Additionally, legal and regulatory considerations are vital, ensuring compliance with relevant laws and regulations regarding network monitoring and data privacy. Lastly, evaluating the effectiveness of the chosen IDS solution in detecting network threats while minimizing false positives and negatives is crucial. By conducting this feasibility analysis, stakeholders can assess the project's viability within the constraints of available resources, technical expertise, and regulatory requirements, ensuring a solid foundation for successful implementation.

CHAPTER 3
PROJECT PLANNING AND METHODOLOGY

3. PROJECT PLANNING AND METHODOLOGY

3.1 Project Planning

Task/Activity	Duration	Dependencies
Project Initiation	1 week	Research and Planning Depends on Project Initiation
Research and Planning	1 week	Setup Environment depends onResearch and Planning.
Install And ConfigureIDS	1 week	Install and Configure IDS depends on Setup Environment.
Develop Documentation	2 weeks	Develop Documentation depends on Install and Configure IDS.
Testing and Evaluation	1 week	Testingand Evaluation depends on Develop Documentation.
Compliance Check	1 week	ComplianceCheck dependsonTesting and Evaluation
Finalize Project	1 week	Finalize Project depends on Compliance Check

3.2 Methodology

project of implementing an Intrusion Detection System (IDS) for small networks, There are two widely used methodologies: The Waterfall model and the Agile methodology.

Waterfall Model:

Sequential Approach: The Waterfall model follows a linear and sequential approach, where each phase (requirements, design, implementation, testing, deployment) is completed before moving to the next.

Detailed Planning: Extensive planning is done upfront, defining requirements and creating a comprehensive project plan.

Less Flexibility: Limited flexibility for changes once the project has moved to the next phase. Changes are difficult and costly to implement.

Suitability: Best suited for projects where requirements are well-understood and unlikely to change significantly.

Agile Methodology:

Iterative and Incremental: Agile methodology involves iterative development cycles (sprints), where requirements are prioritized, and working software is delivered incrementally.

Adaptive to Change: Emphasizes flexibility and responsiveness to changes in requirements, allowing for continuous improvement and adaptation throughout the project.

Collaborative Approach: Encourages close collaboration between cross-functional teams, stakeholders, and customers throughout the project life cycle.

Suitability: Ideal for projects with evolving requirements, where rapid adaptation, customer feedback, and continuous improvement are critical.

Selection Rationale:

Given the nature of the project, implementing an IDS for small networks, the Agile methodology appears to be the most appropriate choice for several reasons:

Evolving Requirements: Cybersecurity requirements and technologies are constantly evolving, and the project may require adjustments based on emerging threats and changes in network environments. Agile's iterative approach allows for flexibility and adaptation to evolving requirements.

User Involvement: Agile encourages active involvement of users (in this case, network administrators) throughout the development process. This ensures that the IDS meets their specific needs and preferences effectively.

Incremental Delivery: Agile facilitates incremental delivery of functionality, allowing for early detection of issues and continuous validation of the IDS's effectiveness. This approach aligns well with the project's goal of enhancing network security through iterative improvements.

Risk Mitigation: Agile's iterative approach enables early identification and mitigation of risks. By delivering working software incrementally, the project can address potential security vulnerabilities and technical challenges early in the development process.

Feedback-driven Development: Agile emphasizes regular feedback loops, enabling stakeholders to provide feedback on each iteration of the IDS. This feedback-driven development ensures that the final product aligns closely with stakeholders' expectations and requirements.

Conclusion:

Considering the project's dynamic nature, evolving requirements, and the need for flexibility and stakeholder involvement, the Agile methodology is deemed the most suitable approach for implementing an Intrusion Detection System for small networks. Its iterative and collaborative nature aligns well with the project's objectives and requirements, facilitating effective development and deployment of the IDS while ensuring responsiveness to changing cybersecurity landscapes.

CHAPTER 4
DATA ANALYSIS, DESGN AND IMPLEMENTATION

4. DATA ANALYSIS, DESIGN AND IMPLEMENTATION

4.1 Requirement Analysis

4.1.1 Data Collection

Primary Sources:

Practical Experiments:
Set up a small-scale network environment using virtual machines.
Deployed an intrusion detection system (IDS) within the simulated network.
Documented the setup, configurations, and observations during the experimentation process.
Analyzed the effectiveness of the IDS in detecting and responding to simulated security threats.
Provided insights into the practical implementation and performance of intrusion detection systems in small networks.

Online Forums and Communities:
Engaged in discussions within online forums and communities focused on cybersecurity and network administration.
Participated in knowledge-sharing activities, asked questions, and shared experiences related to intrusion detection systems for small networks.
Gathered valuable insights, recommendations, and emerging trends from discussions with peers and professionals in the field.
Explored common challenges, best practices, and practical solutions for implementing intrusion detection systems in small-scale environments.

Case Studies:
Analyzed published case studies, white papers, and reports on intrusion detection system implementations in small networks.
Summarized key findings, lessons learned, and recommendations from real-world examples.

Explored practical applications, challenges, and success factors associated with deploying intrusion detection systems in small-scale environments.
Provided insights into the relevance and applicability of intrusion detection systems in addressing security concerns for small networks.

Hands-on Tutorials:
Followed hands-on tutorials and guides available online to learn about intrusion detection system setup, configuration, and usage.
Utilized resources such as video tutorials, blog posts, and online documentation provided by intrusion detection system vendors.
Documented experiences, challenges, and insights gained from following the tutorials.
Examined practical guidance and best practices for deploying and managing intrusion detection systems in small networks.

Secondary Sources:

Literature Review:
Conducted a comprehensive review of academic papers, articles, and textbooks on intrusion detection systems (IDS), network security, and cybersecurity practices.
Summarized key findings, methodologies, and insights relevant to implementing IDS in small networks.
Explored theoretical concepts, industry standards, and best practices associated with IDS deployment and management.
Gained insights into the current state of research, emerging trends, and future directions in intrusion detection for small-scale environments.

Online Resources:
Explored online resources such as research databases, academic journals, and reputable websites focused on cybersecurity and network administration.
Retrieved relevant articles, guides, and case studies on intrusion detection systems and small network security.
Reviewed information on tools, technologies, and methodologies for implementing IDS in small networks.
Stayed updated on industry developments, software updates, and community discussions related to intrusion detection and network security.

Vendor Documentation:
Referenced documentation, guides, and tutorials provided by intrusion detection system vendors such as Snort and Wireshark.
Explored vendor websites to access beginner-friendly resources on IDS setup, configuration, and usage.
Learned about features, functionalities, and deployment considerations specific to Snort, Wireshark, and other IDS solutions.
Leveraged vendor documentation to gain practical insights and guidance for deploying IDS in small-scale network environments.

Regulatory Guidelines:
Reviewed regulatory guidelines, standards, and compliance requirements relevant to network security and data protection.
Examined standards such as GDPR, HIPAA, and PCI DSS to understand compliance obligations and security best practices.
Identified requirements and recommendations for implementing intrusion detection systems in alignment with regulatory frameworks.
Ensured adherence to legal and regulatory guidelines while designing and deploying IDS solutions for small networks.

4.1.2 Data Analysis and tools of data analysis

Technical Requirements:

Requirement Number	Description
TR-001	Snort version 3.1.0 installed
TR-002	Wireshark version 3.4.5 installed
TR-003	PCAP files from small network environment
TR-004	Python 3.8 for data preprocessing and analysis

Analysis and Interpretation:
The technical requirements ensure that the necessary software tools and datasets are available for data analysis. Snort and Wireshark versions are specified to maintain consistency. PCAP files from the small network environment provide real-world data for analysis. Python 3.8 is used for data preprocessing and analysis due to its versatility and extensive libraries for data manipulation

Functional Requirements:

Requirement Number	Description
FR-001	Snort configured to monitor network traffic
FR-002	Snort rules tailored for small network environment
FR-003	Wireshark used for packet capture and analysis
FR-004	Packet analysis conducted to identify anomalies

Analysis and Interpretation:
The functional requirements ensure that Snort and Wireshark are configured and utilized effectively for network monitoring and packet analysis. Snort rules are tailored to the small network environment to improve detection accuracy. Wireshark is employed for packet capture and analysis to identify anomalies and security threats effectively.

Performance Requirements:

Requirement Number	Description
PR-001	Snort processing time < 100 ms per packet
PR-002	Wireshark packet capture rate > 1000 packets/sec

Analysis and Interpretation:
The performance requirements specify acceptable processing times for Snort and packet capture rates for Wireshark to ensure timely detection and analysis of network traffic. These requirements ensure efficient operation and responsiveness of the intrusion detection system.

Database Requirements:

Requirement Number	Description
DBR-001	Logging of Snort alerts
DBR-002	Storage of Wireshark packet captures

Analysis and Interpretation:
Database requirements specify the logging of Snort alerts and storage of Wireshark packet captures for future analysis and forensic purposes. Adequate database capacity and performance are essential to support data retention and retrieval

Security Requirements:

Requirement Number	Description
SR-001	Encryption of captured packet data
SR-002	Access control for IDS configuration

Analysis and Interpretation:
Security requirements emphasize the importance of data encryption and access control to protect sensitive information captured by the intrusion detection system. Encryption ensures confidentiality, while access control mitigates the risk of unauthorized configuration changes.

Maintainability Requirements:

Requirement Number	Description
MR-001	Regular updates for Snort rules
MR-002	Periodic maintenance of IDS hardware

Analysis and Interpretation:
Maintainability requirements emphasize the need for regular updates to Snort rules and periodic maintenance of IDS hardware to ensure optimal performance and effectiveness over time.

Usability Requirements:

Requirement Number	Description
UR-001	User-friendly interface for Snort setup
UR-002	Intuitive packet analysis features in Wireshark

Analysis and Interpretation:
Usability requirements focus on providing a user-friendly interface for Snort setup and intuitive packet analysis features in Wireshark. Enhancing usability ensures ease of configuration and analysis for network administrators and security analysts.

3.4 Design

Logic Design:
Flow Diagram: The system will follow a linear flow, starting with packet capture using Wireshark, followed by analysis using Snort, and ending with alert generation.Rule Set: Snort's preconfigured rule set

will be utilized for detecting known attack signatures and suspicious network behavior.

Decision-making Logic:
Snort's built-in decision-making logic will be employed to classify network activity as either benign or potentially malicious based on rule matches.

Real-time Analysis:
Snort will perform real-time analysis of captured packets to promptly identify and respond to potential security threats.

Alert Generation:
Snort will generate alerts when suspicious activity is detected, triggering notifications to system administrators.

Data Design:

Data Structures:
Basic data structures provided by Wireshark and Snort will be used to capture and store network traffic data.

Packet Headers:
Wireshark will capture and display detailed packet headers, providing essential information about network traffic.

Payloads:
Wireshark will capture packet payloads, allowing for deeper inspection of packet contents.

Metadata:
Wireshark will provide metadata such as timestamps and packet sizes, enhancing the context of network traffic analysis.

Database Schema:
Snort will utilize its internal database schema to organize and store information about detected network events.

Data Indexing:
Snort will employ efficient indexing techniques to facilitate quick retrieval of relevant network event data during analysis.

Data Retention Policies:
Default data retention policies provided by Wireshark and Snort will be followed to manage the storage of captured network traffic data.

Process Design:

Packet Capturing Process:
Network traffic will be captured using Wireshark's packet sniffing capabilities, providing comprehensive visibility into all network activity.

Data Preprocessing:
Wireshark will perform basic preprocessing of captured packets, cleaning and organizing data before analysis by Snort.

Analysis Workflows:
Snort will execute predefined analysis workflows based on its configured rule set, scanning captured packets for known attack signatures and anomalies.

Incident Response Protocols:
Basic incident response protocols will be outlined for investigating and mitigating suspected security incidents detected by Snort.

Forensic Analysis Procedures:
Snort's logging capabilities will be utilized for basic forensic analysis of network traffic data to understand the nature and impact of security incidents.

Interface Design:

Configuration Interface:
Wireshark and Snort will each provide a user-friendly interface for configuring and managing the IDS system, allowing administrators to define rule sets and alert thresholds.

Monitoring Dashboard:
Snort's graphical user interface will display real-time network activity and alert status, providing administrators with a comprehensive view of system activity.

Visualizations:
Basic visualizations provided by Snort's GUI will illustrate trends in network activity and alert frequency.

User-Friendly:
Interfaces provided by Wireshark and Snort will be designed with simplicity in mind, prioritizing ease of use for administrators and analysts.

Responsive:
Wireshark and Snort's interfaces will be tested to ensure compatibility across common devices and screen sizes.Accessible: Efforts will be made to ensure interfaces provided by Wireshark and Snort are accessible to all users, considering factors such as color contrast and screen reader compatibility.

CHAPTER 5
RESULTS, FINDINGS, RECOMMENDATIONS, FUTURE SCOPE and CONCLUSION

5. RESULTS, FINDINGS, RECOMMENDATIONS, FUTURE SCOPE AND CONCLUSION

5.1 Results of the work

Objective Evaluation:

Objective 1:
Develop an IDS capable of detecting network intrusions.
Assessment:
The IDS successfully detects various types of network intrusions, including port scans, denial-of-service (DoS) attacks, and suspicious traffic patterns.

Objective 2:
Implement packet capture and analysis functionality using Wireshark and Snort.
Assessment:
Packet capture and analysis functionality is successfully implemented using Wireshark for capturing network traffic and Snort for intrusion detection.

Objective 3:
Design a user-friendly dashboard interface for monitoring alerts and system status.
Assessment:
A user-friendly dashboard interface is designed and implemented, providing real-time monitoring of alerts, system status, and network traffic.

Objective 4:
Ensure scalability and efficiency of the IDS for deployment in small networks.
Assessment: The IDS architecture is designed to be scalable and efficient, allowing for deployment in small networks without significant performance degradation.

Justification for Unmet Objectives:

No significant unmet objectives were identified during the project. However, challenges were encountered during the implementation phase, including configuration issues with Snort and integration complexities with the dashboard interface. These challenges were addressed through research, troubleshooting, and collaboration with peers and mentors.

Project Outcomes:

The IDS successfully achieves its primary goal of detecting network intrusions, providing enhanced security for small networks.
The implementation of packet capture and analysis functionality using Wireshark and Snort enhances the system's capabilities for detecting and analyzing suspicious network traffic.
The user-friendly dashboard interface improves the usability and accessibility of the IDS, enabling users to monitor alerts and system status effectively.
Overall, the project outcomes contribute to strengthening network security and mitigating potential cyber threats in small network environments.

Lessons Learned:

Effective project planning is essential for overcoming challenges and achieving project objectives.
Thorough testing and validation are crucial for ensuring the reliability and effectiveness of the IDS in real-world scenarios.
Continuous learning and adaptation are necessary to address evolving cybersecurity threats and technologies.

Recommendations:

Consider incorporating additional features, such as automated response mechanisms, to further enhance the IDS capabilities.
Explore opportunities for integrating machine learning algorithms to improve the accuracy and efficiency of intrusion detection.
Collaborate with cybersecurity experts and industry professionals to gain insights and best practices for enhancing network security in small networks.

Conclusion:

The IDS project has successfully achieved its objectives, providing a robust intrusion detection solution for small networks.
The project outcomes contribute to improving network security and mitigating cyber threats in small network environments.
By reflecting on lessons learned and implementing recommendations for future enhancements, the IDS project demonstrates ongoing commitment to cybersecurity excellence and innovation.

5.2 Findings based on analysis of data

1. Detection Effectiveness

Signature-based Detection:

- **Total Alerts:** 120
- **True Positives:** 105
- **False Positives:** 15

The high number of true positives (87.5%) indicates that the signature-based detection component of the IDS is highly effective at identifying known threats. However, the presence of false positives (12.5%) suggests that some legitimate activities are incorrectly flagged as threats, which could lead to unnecessary alerts and potential alert fatigue among network administrators.

Anomaly-based Detection:

- **Anomalous Activities Detected:** 30
- **True Positives:** 25
- **False Positives:** 5

The anomaly-based detection has a slightly lower accuracy (83.3%) but still performs well in identifying unusual activities that could signify new or unknown threats. The false positive rate (16.7%) is manageable but indicates room for improvement in fine-tuning the anomaly detection algorithms.

Impact Assessment: The detection mechanisms effectively identified and alerted on various intrusion attempts, which could help in preventing potential security breaches. The presence of some false positives, however, highlights the need for ongoing tuning and possibly integrating contextual data to improve detection accuracy.

2. Identified Attack Patterns and Anomalies

Common Attack Vectors:

- **Brute Force Attacks on SSH:** 20 incidents
- **SQL Injection Attempts:** 15 incidents
- **Cross-site Scripting (XSS) Attempts:** 10 incidents

These findings reveal that SSH brute force attacks are the most common, likely due to attackers attempting to gain unauthorized access through weak passwords. SQL injection and XSS attempts are also significant, indicating a focus on exploiting web applications. These patterns emphasize the need for robust password policies, secure coding practices, and regular application security testing.

Unusual Activity Patterns:

- **Data Exfiltration Attempt:** Sudden increase in outbound traffic from a specific IP.
- **Compromised Account:** Multiple login failures followed by a successful login.

Identifying these patterns is crucial for understanding and mitigating potential threats. The data exfiltration attempt suggests that an attacker might have gained access and tried to transfer sensitive data. The compromised account pattern underscores the importance of monitoring login activities and implementing multi-factor authentication to prevent unauthorized access.

Impact Assessment: Recognizing these attack patterns allows for targeted security measures, such as strengthening password policies, enhancing web application security, and implementing stringent monitoring and alerting for suspicious activities. This proactive approach can significantly reduce the risk of successful attacks.

4. Recommendations for Improvement

Enhancing Detection Accuracy:

- Refine anomaly detection models to reduce false positives.
- Expand the signature database to cover new and emerging threats.

Monitoring and Response:

- Increase monitoring during identified peak attack times to catch more attacks in real-time.
- Implement automated response mechanisms to quickly isolate compromised systems and mitigate threats.

5.3 Recommendation based on findings

1. Enhance Detection Accuracy

Refine Anomaly Detection Models:

- **Objective:** Reduce the rate of false positives and improve detection of true anomalies.

- **Action:** Implement machine learning techniques to better distinguish between normal and suspicious network behavior. Regularly update and retrain models with new data to adapt to evolving network patterns.
- **Impact:** Improved accuracy in detecting new or unknown threats while minimizing unnecessary alerts.

Expand Signature Database:

- **Objective:** Increase the coverage of known threats.
- **Action:** Regularly update the signature database with the latest threat intelligence feeds and known attack patterns. Collaborate with cybersecurity communities and subscribe to threat intelligence services.
- **Impact:** Enhanced ability to detect and respond to a broader range of known threats, reducing the risk of successful attacks.

2. Strengthen Security Measures

Implement Strong Password Policies:

- **Objective:** Mitigate the risk of brute force attacks.
- **Action:** Enforce the use of complex, unique passwords for all accounts. Implement account lockout policies after a certain number of failed login attempts.
- **Impact:** Reduced likelihood of unauthorized access through brute force attacks, enhancing overall network security.

Enhance Web Application Security:

- **Objective:** Protect against SQL injection and XSS attacks.
- **Action:** Conduct regular security audits and code reviews to identify and fix vulnerabilities in web applications. Implement input validation, parameterized queries, and output encoding to prevent injection attacks.
- **Impact:** Lower risk of web application exploits, ensuring the integrity and security of web-based services.

Implement Multi-Factor Authentication (MFA):

- **Objective:** Prevent unauthorized access even if passwords are compromised.

- **Action:** Require MFA for all critical systems and sensitive data access. Utilize authentication methods such as SMS-based codes, authentication apps, or hardware tokens.
- **Impact:** Enhanced security for user accounts, significantly reducing the risk of successful account compromise.

Automate Response Mechanisms:

- **Objective:** Quickly isolate and mitigate threats to minimize damage.
- **Action:** Implement automated response systems that can isolate compromised devices, block malicious IP addresses, and initiate incident response protocols based on predefined rules.
- **Impact:** Faster response times to detected threats, reducing the potential impact of security incidents.

Encourage Strong, Unique Passwords and MFA:

- **Objective:** Strengthen user authentication processes.
- **Action:** Promote the use of password managers to create and store strong, unique passwords for different accounts. Ensure that all users enable MFA where possible.
- **Impact:** Enhanced protection against unauthorized access, contributing to a more secure network environment.

5.5 Suggestions for areas of improvement

1. Fine-Tune Detection Algorithms

Enhance Machine Learning Models:

- **Suggestion:** Invest in more sophisticated machine learning models for anomaly detection. Incorporate techniques such as deep learning to better identify complex patterns and reduce false positives.
- **Action:** Regularly retrain models with updated data sets, including both normal and malicious traffic, to ensure they adapt to new threats and network behaviors.

- **Impact:** Improved accuracy and efficiency in identifying true threats, leading to quicker and more reliable detection of anomalies.

Integrate Behavioral Analysis:

- **Suggestion:** Include user and entity behavior analytics (UEBA) to enhance the context of anomaly detection.
- **Action:** Monitor and analyze the behavior of users and devices over time to establish baselines and identify deviations indicative of potential security incidents.
- **Impact:** Enhanced ability to detect insider threats and sophisticated attacks that might bypass traditional detection methods.

2. Improve Data Management and Analysis

Centralize Log Management:

- **Suggestion:** Implement a centralized logging system to aggregate data from various sources (firewalls, servers, endpoints).
- **Action:** Use a Security Information and Event Management (SIEM) system to collect, correlate, and analyze logs in real-time.
- **Impact:** Streamlined analysis, quicker identification of patterns and anomalies, and enhanced incident response capabilities.

Data Enrichment:

- **Suggestion:** Enrich collected data with contextual information such as threat intelligence feeds and geolocation data.
- **Action:** Integrate external threat intelligence services to provide context to detected threats, improving the accuracy of alerts.
- **Impact:** Enhanced situational awareness and more informed decision-making during incident response.

3. Strengthen Network and System Security

Regularly Update Security Policies:

- **Suggestion:** Continuously review and update security policies to adapt to new threats and changes in the network environment.
- **Action:** Schedule periodic reviews of security policies and procedures, incorporating feedback from security incidents and threat assessments.
- **Impact:** Up-to-date security measures that address current risks and reduce the likelihood of successful attacks.

Deploy Advanced Threat Prevention Tools:

- **Suggestion:** Implement advanced threat prevention tools such as Next-Generation Firewalls (NGFW), Intrusion Prevention Systems (IPS), and Endpoint Detection and Response (EDR) solutions.
- **Action:** Evaluate and deploy appropriate tools that complement the existing IDS, providing additional layers of defense.
- **Impact:** Enhanced overall security posture with multiple layers of protection against a wide range of threats.

Implement Automated Response Mechanisms:

- **Suggestion:** Automate routine incident response tasks to improve reaction time and efficiency.
- **Action:** Use automation tools to execute predefined response actions such as isolating compromised systems, blocking malicious IP addresses, and initiating forensic investigations.
- **Impact:** Faster containment and mitigation of threats, reducing the impact and spread of security incidents.

5.6 Scope for future work

Future enhancements to this Intrusion Detection System (IDS) project could involve integrating advanced machine learning algorithms to improve anomaly detection accuracy and reduce false positives. Incorporating User and Entity Behavior Analytics (UEBA) would enhance the system's ability to detect insider threats and sophisticated attacks. Additionally, scaling the IDS to accommodate network

growth and increased traffic through distributed architectures or cloud-based solutions would ensure its continued effectiveness. Regular updates to the signature database and anomaly detection models, along with continuous security audits and user training programs, will keep the system robust against evolving threats. These improvements will provide a more comprehensive and adaptive security solution for small networks.

5.7 Conclusion

The Intrusion Detection System (IDS) project using Wireshark and Snort effectively achieved its primary objective of monitoring network traffic, identifying potential threats, and alerting administrators to unauthorized access and malicious activities. The combined use of Wireshark for network traffic capture and Snort for real-time intrusion detection provided a comprehensive approach to network security. The IDS successfully detected a variety of intrusion attempts, including brute force attacks, SQL injection attempts, and cross-site scripting (XSS) attacks, showcasing its capability to handle common network threats.

The analysis demonstrated that the IDS could maintain low resource consumption, ensuring minimal impact on network performance, which is essential for deployment in small networks. While the system showed high accuracy in detecting known threats through Snort's signature-based detection, the identification of false positives indicated areas where further refinement is needed. Despite this, the IDS's ability to detect unusual activity patterns and generate alerts for potential security incidents proved its effectiveness in enhancing network security. Future enhancements, such as improving Snort rule sets and integrating additional anomaly detection capabilities, will further strengthen the system's ability to protect against evolving threats. Overall, the project has successfully laid the groundwork for a robust and efficient IDS tailored to the needs of small networks.

BIBLIOGRAPHY

- Zeeshan Ahmad, Adnan Shahid Khan, Cheah Wai Shiang, Johari Abdullah, Farhan Ahmad(2020). Enhancement of Network Security Through Intrusion Detection. Transactions on Emerging Telecommunications Technologies, 32(1), e4150. https://doi.org/10.1002/ett.4150

ANNEXURE

Annexure 1: IDS Configuration Details

1. Hardware Specifications:

Laptop:

- Model: MSI GF63
- Processor: Intel Core i5-10300H (2.5 GHz base frequency, up to 4.5 GHz with Intel Turbo Boost Technology, 8 MB cache, 4 cores)
- Graphics Processing Unit (GPU): NVIDIA GeForce RTX 3050 with 4 GB GDDR6 VRAM
- RAM: 8 GB DDR4-2666 SDRAM (1 x 8 GB)
- Storage: 512 GB NVMe PCIe Solid State Drive (SSD)
- Display: 15.6-inch Full HD (1920 x 1080 pixels) IPS-level display
- Network Interface: Integrated 10/100/1000 Gigabit Ethernet LAN, Intel Wireless-AC 9560 (2x2) Wi-Fi 5 (802.11ac) and Bluetooth 5 Combo

2. Software Specifications:

Operating System:

- Windows 10 Home 64-bit

Intrusion Detection System (IDS) Software:

- Snort 3.1.0
- Snort Subscriber Rule Set
- PulledPork for rule updates

Network Monitoring Tool:

- Wireshark 3.4.6

Web Framework:

- Python Flask
- Flask version: [Version Number]
- Flask-WTF for form handling