Project Overview and Plan for M.Sc. Project:

Intrusion Detection in a Foreign Domain

Name : Panagiotis Triantafyllidis

Student ID: 11431335

May 10, 2024

Contents

Word count: 2692

1 Introduction

With the rise of cloud computing, there is an increasing need to deploy robust security measures to protect sensitive data. As companies globally migrate their user-related services and resources to cloud networks, they expose data traffic to new environments that may be vulnerable to exploitation. This exposure provides exponentially more opportunities for malicious actors to breach systems [1]. Cloud computing, while offering significant operational cost-efficiency through dynamic online provisioning of IT services, also brings cybersecurity to the forefront as one of its most critical concerns [2].

Intrusion Detection Systems (IDS) play a pivotal role in safeguarding networks and systems from unauthorized access, malicious activities, and security breaches, which would otherwise lead to serious repercussions for the CIA triad of data (confidentiality, integrity, and availability) [3]. Remote security involves monitoring all traffic between the remote user and the on-premise or cloud-based environment for suspicious connections. Intrusion Detection Systems (IDS) are primarily classified into two fundamental types based on their operating environments: Host-based IDS, which monitors local system activities such as API calls and memory usage, and Network-based IDS, which focuses on analysing network traffic, including packet sniffing and log analysis [4]. Additionally, IDS can be categorized by their approach to threat identification, the intensity of countermeasures, and the computational cost of operations [5].

This project, will focus on implementing a Signature-based Intrusion Detection System (SIDS), utilising rule-sets and policy databases to detect predefined threats [6], will primarily monitor the local host environment. Concurrently, an Anomaly-based Intrusion Detection System (AIDS), which leverages machine learning algorithms to detect novel attacks, will be employed to scrutinize network activity and communications [7]. The main role of IDS, is to identify evidence of intrusions, either while they are in progress or after the act has occurred [8]. In the context of this project, computational cost is defined as the amount of processing power and memory required to operate an Intrusion Detection System, which often employs resource-intensive machine learning algorithms to analyse and detect atypical behaviours.

1.1 Aim & Objectives

To convey and present the importance of digital security in modern technocratic life, this project's aim is to research and design an efficient Intrusion Detection System to detect security violations in a Cloud (or remote) computing environment. An efficient IDS, is defined as one that can effectively detect and identify intrusions with high accuracy and minimal false positives, while operating within the constraints of available computational and energy resources [9]. The driving principle, is to balance the fast but limited response of SIDS, with the advanced but computationally demanding assistance of AIDS, in order to create a lightweight but secure IDS.

Specifically, the project will focus on the following objectives:

1. **Development of Signature IDS :** Implement a Signature IDS, and configure it with an already available open-source Anomaly IDS, modified for suitable cooperation of the two.
2. **Enhance Detection Capabilities for Zero-Day Attacks :** Develop methods to improve the detection of zero-day attacks using a combination of signature-based and anomaly-based IDS.
3. **Optimize Resource Usage in IDS Operations :** Focus on optimizing the computational and memory requirements of the IDS to ensure it operates efficiently in resource-constrained environments.
4. **Improve Accuracy and Reduce False Positives/Negatives :** Establish methods to enhance the accuracy of intrusion detection while significantly reducing the incidence of false positives and negatives.

2 Literature Review

Cloud systems are susceptible to threats like data breaches, insider threats, and insecure APIs, due to their complex and dynamic nature, which traditional security models do not fully address [10]. IoT devices, often being resource-constrained and lacking robust security, are prone to unauthorized access and malware infections [11]. These vulnerabilities highlight the necessity of IDS to improve its capacity for security and resource efficiency. IDS analyse network traffic

and user behaviour, identifying unusual activities to prevent potential cyberattacks and mitigate risks associated with these environments [12].

2.1 Background

Initially, firewalls were basic packet filters that managed network traffic based solely on protocol rules, IP addresses, and ports, forming the primary line of defence against external threats [13]. With advancements, firewalls evolved into more sophisticated stateful inspections that understand and track the state of network connections, enhancing their ability to manage traffic dynamically. Despite this, the complexity of modern cyber threats, such as advanced persistent threats (APT) and internal breaches, exposed the design limitations of firewalls, prompting the integration with IDS techniques and other security measures to provide a layered and more effective network defence [14].

Originally, organisations used system administrators to look at consoles and audit their user activity. Through printed logs, admins were trying to identify suspicious activity, until user-base became so large, an automated means of pattern recognition was required [8]. Intrusion Detection has been a critical component of cybersecurity for decades, with the first IDS emerging in the 1980s to counteract the growing threat of cyberattacks; IDES.

In 1987, Dorothy Denning and Peter Neumann researched and developed the first model of a real-time IDS. The Intrusion Detection Expert System (IDES), as seen in Figure 1, was initially a rule-based (signature) system, trained to recognise known malicious activities (the signatures of which were stored in a database) [15], as well as utilising basic statistical models for threat identification, such as: interval timers, resource measurements and event counters [16]. This hybrid result, led to the start of advancements in IDS software, notably with the enhanced and refined, Next-generation Intrusion Detection Expert System (NIDES) [8]. This evolution of the original, incorporated more advanced and data-efficient statistical anomaly detection algorithms, a method frequently used for fraud detection at the time, to establish pattern recognition of user activity, spotting discrepancies and unusual log traces, which allowed for predicting and identifying unknown attacks [17].

Today, traditional IDS are effective against known security threats but struggle to detect new

types of intrusions, like those that evade detection methods. Modern DDoS attacks often use techniques such as spoofing source IP addresses to remain undetected by traditional IDS [18]. Implementing IDS on remote domains, such as cloud services or IoT, is challenging due to their hardware diversity. This diversity in device types makes it difficult to apply traditional IDS effectively, given their resource constraints and specific network requirements [19].

The system developed in this project, will follow a Hybrid method, trying to expand on the concepts of cost-efficiency relative to security, and most importantly, upscale the outdated security capabilities of conventional IDS.

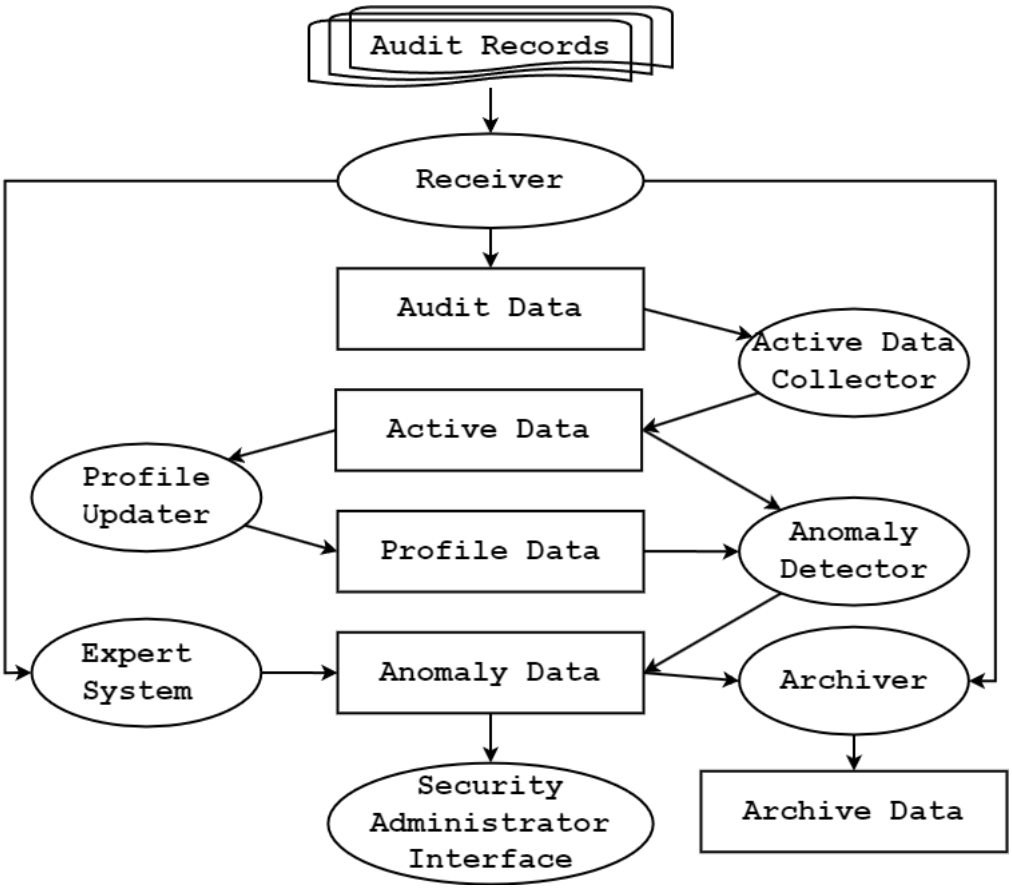

Figure 1: Flowchart of IDES Processes and Database Structure. Source : [16].

2.2 Features & Effectiveness of IDS

Modern techniques have been developed to enhance the effectiveness of IDS in detecting and mitigating cyber threats. Some key features and advancements include:

Machine Learning Integration: Utilises advanced machine learning techniques, including supervised and unsupervised learning, for dynamic adaptation to new and evolving cyber threats. IDS can also incorporate Long Short-Term Memory (LSTM) networks and attention mechanisms to capture and analyse extended data sequences, for detecting complex threats.

Optimization Algorithms: Implements optimization algorithms like Particle Inspired Optimizer (PIO) to reduce feature dimensionality, improving processing speeds and detection accuracy.

Network-based and Host-based Monitoring: The division of monitoring into Network-based IDS (NIDS) and Host-based IDS (HIDS) is intended to provide comprehensive security strategies that encompass both network and host perspectives.

2.3 Research Questions

The research questions that will be addressed are:

RQ1: How does the cost-efficiency of a Hybrid Intrusion Detection System (IDS) compare to a standalone Anomaly-based IDS (AIDS) in a cloud computing environment?
RQ2: To what extent does a Hybrid IDS exhibit false positives compared to a standalone AIDS, and what factors influence this discrepancy?
RQ3: What are the current limitations of Hybrid IDS technology in detecting sophisticated cyber threats, and what research directions could effectively address these challenges?
RQ4: How does data quality influence the effectiveness of Hybrid IDS, and what specific data pre-processing techniques could enhance its detection accuracy?

3 Research Methodology

3.1 General Statement

One challenge with modern IDS, particularly the more sophisticated Anomaly detection, is the high computational cost due to the machine learning algorithms used to identify unknown threats [9], relative to the host hardware limitations. An Anomaly IDS must process large volumes of data in a resource-constrained environment, leading to high energy and computational demands that challenge scalability [19]. Conversely, Signature IDS are valued for their speed and simplicity, as they operate primarily through pattern-matching using a database of predefined policies and rule-sets. However, SIDS are inherently limited in their ability to detect new, unseen threats, reducing its effectiveness in an area where threats continuously evolve and often outpace defensive measures [20].

The proposed solution, as seen from its design in Figure 2, will employ a dual approach. The Signature IDS as a first-line defence, and the Anomaly IDS as a second-line defence, to ensure that the system operates at low cost during routine, and high cost only during high-risk operations.

3.2 System Design

Stage 1 : Signature

The Signature IDS, will be developed using the C5.0 decision tree algorithm/classifier. Decision trees are among the most widely used methods for classification. The nodes that make up the decision tree form a rooted tree, which is a directed tree with a node known as a `root` that lacks any incoming edges [21]. The C5.0 decision trees produce results by differentiating the data based on one characteristic at a time. Sets of criteria defined at the nodes can be used to categorise new data [18]. Once a decision tree is established, it can be utilised to identify test samples, by being employed as a rule-set to ascertain whether a test sample corresponds to malware or benign software [22].

For handling unknown traffic, pattern recognition will be used to discern whether a sample

exhibits normal or abnormal behaviours. If the pattern matches an attack signature stored in the database, it triggers an alarm indicating malicious intent. Conversely, if no match is found, the request's result will be labelled `unknown` and routed to AIDS.

Stage 2 : Anomaly

The foundation of AIDS development, will be a single-class Support Vector Machine (SVM). AIDS will construct the profiles of typical operational behaviours by utilising knowledge about known attacks. To effectively detect unknown attacks, the insights gained from the SIDS stage will be utilised to train AIDS in identifying abnormal activities [23]. Training will focus on benign samples, the task will be to discern activities that deviate from the norm, indicating unusual behaviours typically associated with malware.

This approach enables the classifier to identify normal activities with greater accuracy, as ample training data for normal behaviour is readily available [24], contrary to the limited training datasets available for zero-day attacks. Therefore, in this stage anything outside identified normal behaviours, will be classified as zero-day attack. The signature of the final decision will be sent back to the SIDS, to be added to the database, and the system will be updated accordingly.

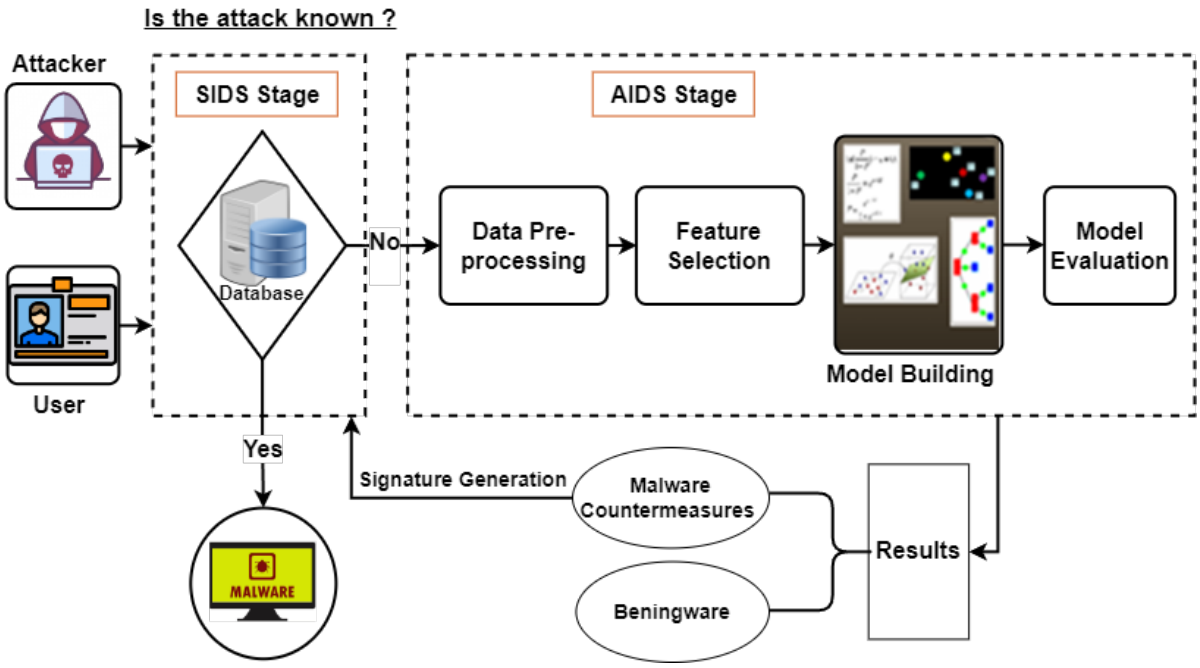

Figure 2: Flowchart of a Hybrid IDS processes. Source : [23].

3.3 Implementation

Signature Stage Implementation: The Signature IDS will be implemented using the C5.0 decision tree algorithm, with Python as the primary language choice for its rich ecosystem of machine learning libraries. Libraries such as Scikit-learn will facilitate the implementation of the decision tree classifier. MySQL will serve as the database for storing attack signatures. Additionally, Flask can be employed to develop RESTful APIs for communication between different components of the system.

Anomaly Stage Implementation: For the Anomaly IDS stage, a single-class Support Vector Machine (SVM) will be implemented using Python, leveraging libraries like Scikit-learn. Preprocessing tools for feature extraction and normalization may be implemented using Pandas and NumPy. Apache Kafka can be utilised for setting up a data pipeline architecture to efficiently handle data flow between stages. The development of RESTful APIs for communication and integration between SIDS and AIDS can be achieved using Flask or Django frameworks.

3.4 Data Collection and Analysis

The project will utilise a closed, simulated environment, leveraging host traffic data within a local network. It will also incorporate open-source datasets sourced from prominent providers such as Kaggle.com, known for its extensive collection of diverse datasets used for academic and research ML purposes.

Research will also be conducted on selecting an appropriate open-source Anomaly Intrusion Detection System (AIDS), evaluating potential candidates based on criteria such as adaptability, compatibility with existing systems, pre-implemented machine learning techniques, and simplicity to ensure the solution is not overly complex or excessive for the project's requirements. Initial candidates include systems such as Snort, Suricata, OSSEC, and NEMEA.

3.5 Testing

Testing an Intrusion Detection System (IDS) involves evaluating its effectiveness in identifying and responding to security threats. The testing process begins with establishing a controlled

environment that accurately simulates the intended operational setting of the IDS.

- **Network Configuration:** Simulating realistic traffic patterns and potential attack scenarios that the IDS is expected to handle.
- **System Setup:** The testing will be conducted on a virtual machine, to monitor traffic with the host machine and the local network.

Test types :

1. **Modular testing:** Evaluating the IDS's individual modules, such as the signature database, anomaly detection algorithms, and response mechanisms, to ensure they function correctly.

2. **Integration and Interaction Testing:** Assessing the communication and coordination between the SIDS and AIDS to verify their compatibility and effectiveness in detecting threats and updating the rule-set with new signatures.

3. **Security testing :**
   - Hybrid Tested: A combination of real and virtual systems where the IDS monitor network activities. This setup helps in understanding the IDS's effectiveness in a dynamic network.
   - Virtual Domain Simulation: Deploying the IDS within a virtual machine that mimics a foreign domain to monitor live network traffic, simulating an external attack environment.
   - Offline Simulation: Testing the IDS with pre-recorded traffic in a controlled, offline setting to evaluate its response to historical data and known threats.
   - System Log Analysis: Utilizing a virtual environment to analyse how the IDS responds to changes in system logs and file patterns, simulating internal data manipulations.
   - Live Traffic Sniffing: Employing a secondary device as a network sniffer to test the IDS against live traffic, focusing on its ability to detect unauthorized access and simulated attacks.

Operating environment :

Specification	Value
Virtual Machine	VM Ubuntu 20.04 on VirtualBox 4GB RAM, 4 cores, 50GB storage Bridged and NAT network configurations
Operating System	Microsoft Windows 10 Home
Machine Architecture	x64
RAM	8GB
CPU Model	AMD Ryzen 5 4500U @ 2.38 GHz
CPU Cores	6
Storage	500GB SSD + 500GB HDD
Network	50Gbps

Table 1: System Specifications

3.6 Evaluation

The effectiveness of the IDS is measured through various testing strategies aimed at quantifying its performance and improving its configurations.

Key performance indicators include :

- **Detection Accuracy:** Measures the IDS's ability to correctly identify both threats and benign activities.
- **False Positive Rate:** Evaluates the frequency at which legitimate actions are mistakenly flagged as threats, which is crucial for minimizing disruptions in normal operations.
- **Detection Latency:** Measure the time it takes for the IDS to detect a threat from the moment it enters the network.

The primary aim of these evaluations is to ensure the IDS can :

- Reliably identify both known and novel threats.
- Efficiently distinguish between malicious activities and legitimate operations.
- Adapt its detection mechanisms to evolving threats with minimal manual intervention.

4 Risk Considerations

The following Table 2 outlines potential risks that may arise during the project's development and implementation. Each risk is evaluated based on its probability, impact, and priority, with corresponding mitigation strategies to address and minimize the identified risks.

Risk	Consequences	Probability	Impact	Mitigation Strategy
Incompetent security	The system does not react to multiple unseen threats	Medium	High	Incorporate various types of anomalies in training datasets to ensure broad generalization capabilities.
Lack of inter-component communication	The AIDS does not react to the flags of the SIDS or AIDS does not update the SIDS databse with new signatures	Medium	High	Scale down testing to simpler threats/tests and try to identify the cause of error after ensuring the AIDS works standalone aswell.
Complexity of integration	Higher complexity of configuring and connecting the AIDS with the SIDS to work together, in a harmonised process.	Medium	Medium	Research and read documentation of similar open-source projects on Github/Gitlab, seek help from experienced developers, consider changing the selected AIDS.
Resource Allocation Limits.	The testing environment may have limited resources, which could restrict the scalability tests and affect the performance of IDS under high load conditions.	Medium	Low	Carefully plan resource allocation to optimize the available computational power and memory. Consider cloud-based or distributed computing resources to scale up simulations when needed.
Hardware Failure.	Loss of data and possible disruption to project timeline.	Low	Medium	Monitor hardware health metrics, and have contingency plans in place for quick recovery or replacement in case of failure, such as backups (cloud and local) and version control GitHub.

Table 2: Table of possible risk considerations for the project.

5 Ethics and Professional Considerations

5.1 Ethics

Developing an Intrusion Detection System (IDS) carries ethical importance that extends beyond privacy, impacting trust in cloud services and potential legal implications for providers. Ethical deployment of an IDS involves a careful data handling protocol compliant with laws like GDPR. An ethical framework is crucial as the IDS evolves, requiring responsible disclosure of vulnerabilities to prevent misuse. Additionally, ethical practices necessitate honouring licensing and proper attribution of open-source resources to maintain research integrity.

5.2 Professionalism

Maintaining professionalism in IDS development involves adhering to ethical codes, promptly addressing deviations, and taking corrective measures. Transparency and thorough quality assurance are essential, ensuring all project phases are openly conducted and subject to professional review. Regular updates with the supervisor will help deliver an ideal artifact that aligns with the standards.

6 Planning

The Gantt Chart in Figure 3 shows the project schedule, detailing essential tasks from January to September, factoring in breaks and exams. It organizes the workflow into segments, covering all project phases from initial research to final implementation and marking important milestones and deliverables.

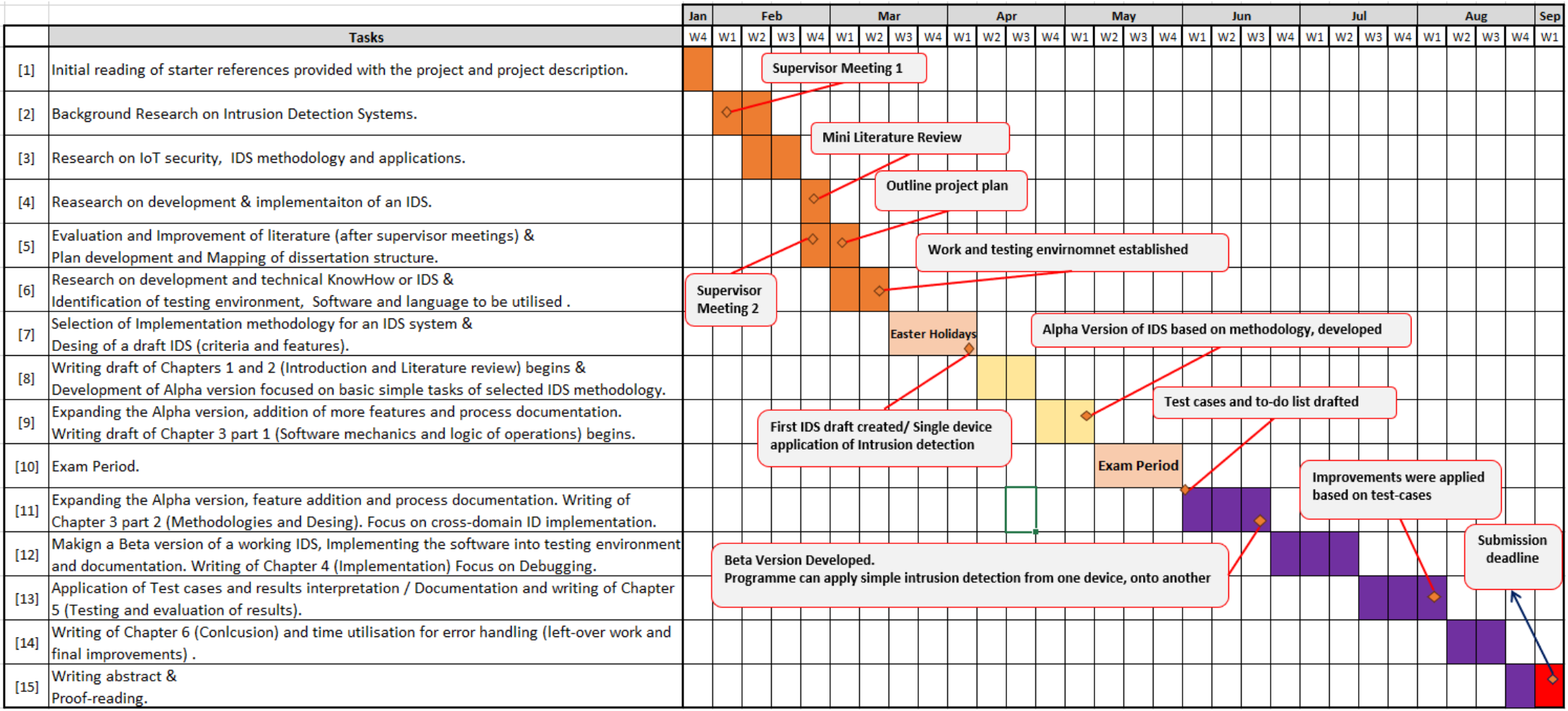

Figure 3: Gantt Chart

7 References

[1] V. Morgunov, Y. Shmelev, K. S. Services, and K. I. CERT, *Overview of iot threats in 2023*, https://securelist.com/iot-threat-report-2023/110644/, 2019 (cited on p. 1).

[2] I. Aljamal, A. Tekeoğlu, K. Bekiroglu, and S. Sengupta, "Hybrid intrusion detection system using machine learning techniques in cloud computing environments," in *2019 IEEE 17th international conference on software engineering research, management and applications (SERA)*, IEEE, 2019, pp. 84–89. DOI: 10.1109/SERA.2019.8886794 (cited on p. 1).

[3] A. Martin, A. Rashid, H. Chivers, G. Danezis, S. Schneider, and E. Lupu, "The cyber security body of knowledge," in University of Bristol, 2019, ch. Distributed Systems Security, Version 1.0. [Online]. Available: https://www.cybok.org/ (cited on p. 1).

[4] L. N. Tidjon, M. Frappier, and A. Mammar, "Intrusion detection systems: A cross-domain overview," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 4, pp. 3639–3681, 2019. DOI: 10.1109/COMST.2019.2922584 (cited on p. 1).

[5] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, "Survey of intrusion detection systems: Techniques, datasets and challenges," *Cybersecurity*, vol. 2, no. 1, p. 20, Jul. 2019, ISSN: 2523-3246. DOI: 10.1186/s42400-019-0038-7. [Online]. Available: https://doi.org/10.1186/s42400-019-0038-7 (cited on p. 1).

[6] P. Ioulianou, V. Vasilakis, I. Moscholios, and M. Logothetis, "A signature-based intrusion detection system for the internet of things," *Information and Communication Technology Form*, 2018 (cited on p. 1).

[7] S. Aljawarneh, M. Aldwairi, and M. B. Yassein, "Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model," *Journal of Computational Science*, vol. 25, pp. 152–160, 2018 (cited on p. 1).

[8] R. A. Kemmerer and G. Vigna, "Intrusion detection: A brief history and overview," *Computer*, vol. 35, no. 4, supl27–supl30, 2002 (cited on pp. 1, 3).

[9] S. U. Jan, S. Ahmed, V. Shakhov, and I. Koo, "Toward a lightweight intrusion detection system for the internet of things," *IEEE access*, vol. 7, pp. 42 450–42 471, 2019 (cited on pp. 2, 6).

[10] W. Ahmad, A. Rasool, A. R. Javed, T. Baker, and Z. Jalil, “Cyber security in iot-based cloud computing: A comprehensive survey,” *Electronics*, vol. 11, no. 1, 2022, ISSN: 2079-9292. DOI: 10.3390/electronics11010016. [Online]. Available: https://www.mdpi.com/2079-9292/11/1/16 (cited on p. 2).

[11] R. Roman, J. Zhou, and J. Lopez, “On the features and challenges of security and privacy in distributed internet of things,” *Computer networks*, vol. 57, no. 10, pp. 2266–2279, 2013 (cited on p. 2).

[12] E. Benkhelifa, T. Welsh, and W. Hamouda, “A critical review of practices and challenges in intrusion detection systems for iot: Toward universal and resilient systems,” *IEEE communications surveys & tutorials*, vol. 20, no. 4, pp. 3496–3509, 2018 (cited on p. 3).

[13] S. M. Bellovin and W. R. Cheswick, “Network firewalls,” *IEEE communications magazine*, vol. 32, no. 9, pp. 50–57, 1994 (cited on p. 3).

[14] J. Liang and Y. Kim, “Evolution of firewalls: Toward securer network using next generation firewall,” in *2022 IEEE 12th Annual Computing and Communication Workshop and Conference (CCWC)*, IEEE, 2022, pp. 0752–0759 (cited on p. 3).

[15] D. E. Denning, “An intrusion-detection model,” *IEEE Transactions on software engineering*, no. 2, pp. 222–232, 1987. DOI: 10.1109/TSE.1987.232894 (cited on p. 3).

[16] S. Listgarten, D. L. Edwards, P. G. Neumann, H. S. Javitz, and A. Valdes, “IDES: The Enhanced Prototype,” 1995. [Online]. Available: http://www.csl.sri.com/papers/4sri/ (cited on pp. 3, 4).

[17] D. Anderson, T. Frivold, and A. Valdes, “NIDES: A summary,” 1995. [Online]. Available: http://www.csl.sri.com/papers/4sri/ (cited on p. 3).

[18] A. Khraisat, I. Gondal, P. Vamplew, J. Kamruzzaman, and A. Alazab, “A novel ensemble of hybrid intrusion detection system for detecting internet of things attacks,” *Electronics*, vol. 8, no. 11, p. 1210, 2019. DOI: 10.3390/electronics8111210 (cited on pp. 4, 6).

[19] Y. Mehmood, M. A. Shibli, U. Habiba, and R. Masood, “Intrusion detection system in cloud computing: Challenges and opportunities,” in *2013 2nd national conference on information assurance (NCIA)*, IEEE, 2013, pp. 59–66 (cited on pp. 4, 6).

[20] A. Khraisat and A. Alazab, “A critical review of intrusion detection systems in the internet of things: Techniques, deployment strategy, validation strategy, attacks, public datasets and challenges,” *Cybersecurity*, vol. 4, pp. 1–27, 2021 (cited on p. 6).

[21] I. D. Mienye, Y. Sun, and Z. Wang, “Prediction performance of improved decision tree-based algorithms: A review,” *Procedia Manufacturing*, vol. 35, pp. 698–703, 2019. DOI: 10.1016/j.promfg.2019.06.011 (cited on p. 6).

[22] A. Khraisat, I. Gondal, and P. Vamplew, “An anomaly intrusion detection system using c5 decision tree classifier,” in *Trends and Applications in Knowledge Discovery and Data Mining*, Springer, 2018, pp. 149–155, ISBN: 978-3-030-04503-6 (cited on p. 6).

[23] A. Khraisat, I. Gondal, P. Vamplew, J. Kamruzzaman, and A. Alazab, “Hybrid intrusion detection system based on the stacking ensemble of c5 decision tree classifier and one class support vector machine,” *Electronics*, vol. 9, no. 1, p. 173, 2020 (cited on p. 7).

[24] P. Manoharan, R. Walia, C. Iwendi, *et al.*, “Svm-based generative adverserial networks for federated learning and edge computing attack model and outpoising,” *Expert Systems*, vol. 40, no. 5, e13072, 2023. DOI: 10.1111/exsy.13072 (cited on p. 7).