**Systemzen**

A PROJECT REPORT

BY

Aditya Singh (E21BCAU0014)

Vedanth T. Sreenivasan (E21BCAU0009)

Archita Dubey (E21BCAU0068)

A close up of a sign

Description automatically generated

SUBMITTED TO

SCHOOL OF COMPUTER SCIENCE ENGINEERING AND TECHNOLOGY

BENNETT UNIVERSITY

GREATER NOIDA, 201310, UTTAR PRADESH, INDIA

MAY 2023

# DECLARATION

I/We hereby declare that the work which is being presented in the report entitled “System Zen”, is an authentic record of my/our own work carried out during the period from January 2023 to May 2023 at School of Computer Science, Engineering and Technology, Bennett University Greater Noida.

The matters and the results presented in this report have not been submitted by me/us for the award of any other degree elsewhere.

Signature of Candidate

Aditya Singh

(E21BCAU0014)

Vedanth T. Sreenivasan

(E21BCAU0009)

Archita Dubey

(E21BCAU0068)

# ACKNOWLEDGEMENT

I/We would like to take this opportunity to express my/our deepest gratitude to my/our mentor, **Dr. Ashima Yadav (**Provide correct name & designation) for guiding, supporting, and helping me/us in every possible way. I/we was/were extremely fortunate to have him as my/our mentor as he provided insightful solutions to problems faced by me/us thus contributing immensely towards the completion of this capstone project. I/We would also like to express my/our deepest gratitude to VC, DEAN, HOD, faculty members, family and friends who helped me/us in successful completion of this capstone project.

Signature of Candidate

Aditya Singh

(E21BCAU0014.)

Vedanth T. Sreenivasan

(E21BCAU0009)

Archita Dubey

(E21BCAU0068)

**TABLE OF CONTENTS**

[LIST OF TABLES v](#_Toc132978572)

[LIST OF FIGURES vi](#_Toc132978573)

[LIST OF ABBREVIATIONS vii](#_Toc132978574)

[ABSTRACT viii](#_Toc132978575)

[1. INTRODUCTION 1](#_Toc132978576)

[2. Background Research 1](#_Toc132978577)

[2.1. Proposed System 1](#_Toc132978578)

[2.2. Goals and Objectives 2](#_Toc132978579)

[3. Project Planning and tracking 3](#_Toc132978580)

[3.1. Project Resources 3](#_Toc132978581)

[3.2. Communication Plan 3](#_Toc132978582)

[3.3. Deliverables 4](#_Toc132978583)

[4. SYSTEM ANALYSIS AND DESIGN 5](#_Toc132978584)

[4.1. Overall Description 5](#_Toc132978585)

[4.2. Users and Roles 5](#_Toc132978586)

[4.3. Design diagrams/ UML diagrams/ Flow Charts/ E-R diagrams 6](#_Toc132978587)

[4.3.1. Use Case Diagrams 6](#_Toc132978588)

[5. User Interface 7](#_Toc132978589)

[5.1. UI Description 7](#_Toc132978590)

[5.2. UI Mockup 8](#_Toc132978591)

[6. Algorithms/Pseudo Code 9](#_Toc132978592)

[7. Project Closure 11](#_Toc132978593)

[REFERENCES 12](#_Toc132978594)

LIST OF TABLES

Table Page

[Table 1: Goal and Objectives 2](#_Toc132981057)

[Table 4: Resource Description 3](#_Toc132981058)

[Table 7: Regularly Scheduled Meetings 4](#_Toc132981059)

[Table 8: Information To Be Shared Within Our Group 4](#_Toc132981060)

[Table 11: Deliverables 5](#_Toc132981061)

[Table 12: Users & Roles 6](#_Toc132981062)

LIST OF FIGURES

Figure Page

[Figure 1: Sample use-case diagram 9](#_Toc20994053)

[Figure 2: sample 2 11](#_Toc20994054)

[Figure 3: sample 3 12](#_Toc20994055)

[Figure 4: Sample 4 13](#_Toc20994056)

[Figure 5: Sample 5 13](#_Toc20994057)

[Figure 6: Sample 6 14](#_Toc20994058)

LIST OF ABBREVIATIONS

Abbreviation Explanation of the Abbreviation

AAA Authentication Authorization and Access Control

CSP Cloud Service Provider

DNS Domain Name System

IAM Identity and Access Management

IDS Intrution Detection System

NIDS Network based Intrusion Detection System

HIDS Host only Intrusion Detection System

ABSTRACT

The Intrusion Detection System (IDS) is a crucial component of network security, designed to detect and respond to security threats in real-time. The purpose of this project was to develop a comprehensive IDS system using Python, machine learning algorithms, and a web-based user interface, aimed at enhancing the customer's cybersecurity posture.

The project had multiple objectives, including the development of a network-based IDS system, implementation of machine learning algorithms for detecting security threats, and designing a web-based user interface. The team used Python programming language, Docker for containerization, HTML, CSS, and JS for developing the user interface.

The project team aimed to create a reliable and effective IDS system that could accurately detect security threats while reducing false positives. Additionally, they aimed to develop a user-friendly interface to enable users to view security alerts and take quick action when necessary.

The team followed a communication plan that involved regular meetings and updates with stakeholders, receiving feedback on the user interface design from end-users, and providing progress updates to management and other groups.

Overall, the project successfully achieved its goals and objectives. The developed IDS system provides reliable and effective security threat detection, and the user interface is user-friendly and easy to navigate. The project team believes that this IDS system could serve as a valuable resource for organizations looking to enhance their cybersecurity posture and protect against cyber-attacks.

1. INTRODUCTION

* In recent years, with the rise of cyber-attacks and data breaches, the importance of IDS systems has only increased. Many organizations are now actively seeking out IDS solutions to protect their networks and sensitive information. Additionally, with the increasing use of cloud computing and the internet of things (IoT), there is a growing need for IDS systems that can effectively monitor and secure these complex and distributed environments.
* In terms of technology trends, machine learning and artificial intelligence (AI) are becoming increasingly important in the field of IDS. These technologies can help improve the accuracy and effectiveness of IDS systems by enabling them to learn and adapt to new threats and attack patterns in real-time. Furthermore, the use of open-source software and tools is also becoming more prevalent in the development of IDS systems, allowing for greater customization and flexibility.

1. Background Research

In recent times, the increased occurrence of cyber attacks has emphasized the need for effective intrusion detection systems (IDS) to safeguard computer networks. Conventional IDSs are rule-based and rely on signature-based methods to identify known attacks, which are ineffective against new and unknown attacks. In December 2020, a significant cyberattack was uncovered, revealing that various US government agencies and private organizations had been breached. This cyberattack was conducted on SolarWinds, a software provider, and it involved the deployment of a malware known as Sunburst on SolarWinds' Orion platform. This allowed the hackers to gain unauthorized access to sensitive information and systems And also in March 2021, it was reported that a state-sponsored group had carried out a hack on Microsoft Exchange Server, exploiting vulnerabilities in the software to gain unauthorized access to numerous email accounts. This attack was identified and reported by cybersecurity researchers who alerted the impacted organizations.

Hence, machine learning (ML) and deep learning (DL) algorithms have been introduced as effective solutions for detecting previously unknown and complex attacks. This paper aims to provide a comprehensive study of machine learning and deep learning approaches for network intrusion detection systems. The availability of datasets for training and testing machine learning models is a significant challenge in building effective IDSs.

A survey by Garcia-Teodoro et al. (2012) identified the need for a comprehensive dataset to evaluate IDSs' performance, resulting in the creation of several datasets, such as the NSL-KDD and UNSW-NB15. Supervised learning algorithms such as decision trees, neural networks, and support vector machines have been widely used for network intrusion detection systems to detect known and unknown attacks. Unsupervised learning algorithms like clustering and anomaly detection have also been explored. Random forests achieved the highest detection rate compared to other supervised learning algorithms in one study that compared their performance on the NSL-KDD dataset..

In conclusion, machine learning and deep learning algorithms have shown promising results in network intrusion detection systems. However, there is a need for further research to address the challenges of training and interpreting machine learning models and defending against adversarial attacks.

* 1. Proposed System

The aim of this project is to create an Intrusion Detection System (IDS) that utilizes Python, machine learning algorithms, and a web-based user interface to detect and respond to security threats in real-time. The project is intended to address the growing concern of cyber-attacks on computer networks, which can lead to financial losses, reputational damage, and data breaches.

The goal of this project is to provide organizations with a reliable and effective IDS system that enhances their cybersecurity posture. By using machine learning algorithms to analyze network traffic data and identify abnormal patterns, the IDS system can proactively identify and respond to potential security threats, thereby minimizing the damage caused by attacks.

This project is particularly relevant in today's environment of increasing cyber-attacks on organizations worldwide, which has been exacerbated by the rise of remote work and digital transformation. As a result, it is crucial for organizations to have strong cybersecurity measures in place to protect their assets from cyber threats. The IDS system developed in this project can provide a valuable solution to address this pressing issue.

* 1. Goals and Objectives
  + Develop a working IDS system with an accuracy rate of at least 95%.
  + Implement machine learning algorithms that can analyze and detect network traffic anomalies in real-time.
  + Create a user-friendly web-based interface for the IDS system.
  + Deploy the IDS system using Docker containers for ease of use and scalability.
  + Ensure the IDS system is compatible with a variety of network topologies and configurations.
  + Test the IDS system under various scenarios to validate its effectiveness and reliability.
  + Document the entire development process, including challenges faced and solutions implemented, for future reference and improvement.
  + Foster effective teamwork and collaboration among project members to achieve project goals and objectives within the specified timeline.

Table 1: Goal and Objectives

|  |  |  |
| --- | --- | --- |
| **#** | **Goal or Objective** | **Description** |
| 1 | Develop a comprehensive IDS system | Develop a network-based IDS system that uses machine learning algorithms to detect and respond to security threats in real-time |
| 2 | Implement a user-friendly web-based interface | Develop a user interface that allows security analysts to view and analyze security alerts in real-time |
| 3 | Ensure scalability and reliability of the system | Ensure that the system can handle large amounts of network traffic and operate reliably without downtime or errors |
| 4 | Promote collaboration and communication within the project team | Foster effective communication and collaboration among team members to ensure project success |

1. Project Planning and tracking
   1. Project Resources

Anticipated resources required for this project include a large amount of data for training and testing the machine learning models, a Capstone team to develop the project, a mentor to provide technical assistance, and software development tools. The project may also require specialized equipment for testing and implementation. It is essential to have adequate resources to ensure successful project completion.

Table 4: Resource Description

|  |  |
| --- | --- |
| **Resource** | **Resource Description** |
| Data | A lot of data is required to train and test the machine learning models. |
| Capstone Team | Our team of students will be the primary developers of the project. |
| Ashima Yadav | The mentor who will be able to provide us with technical assistance. |
| Software | Software includes development tools. |

* 1. Communication Plan
  + Communications we will provide to other groups:
  + Regular progress reports to the project stakeholders, including project sponsors, management, and team members.
  + Updates on any changes to project timelines, milestones, and objectives.
  + Notifications of any issues or risks that may impact the project schedule or quality.
  + Documentation of project deliverables and status reports.
  + Communications we need to receive from other groups:
  + Regular status updates from team members on their assigned tasks and progress.
  + Feedback from project sponsors and management on project direction and objectives.
  + Notifications of any changes or issues that may impact project timelines or deliverables

Table 7: Regularly Scheduled Meetings

|  |  |  |
| --- | --- | --- |
| Meeting Type | Frequency/Schedule | Who Attends |
| Conference Call/Skype | Weekly | Project team and mentor |
| Team Meeting | Weekly | Project team |
| Short Meeting | Weekly in class | Project team |
| Sprint Planning Meeting | Start of each sprint | Project team and mentor |
| Sprint Retrospective Meeting | End of each sprint | Project team |
| Sprint Review Meeting | End of each sprint | Project team, ***mentor, and sponsor*** |

Table 8: Information To Be Shared Within Our Group

|  |  |  |  |
| --- | --- | --- | --- |
| Who? | What Information? | When? | How? |
| Project team | Task assignments & General scrum information | Weekly | Team meetings, listing in Project Specification. |

* 1. Deliverables
  + Project Plan: A document that outlines the project's objectives, scope, schedule, budget, and resources.
  + Design Document: A detailed document that describes the system architecture, design, and implementation approach.
  + Source Code: The software application's source code that includes all the developed
  + features and functionalities, machine learning algorithms for detecting security threats, a web-based user interface, and real-time alerts and notifications.
  + Test Plan: A document that outlines the testing strategy, test cases, and expected results.
  + Final Report: A comprehensive report that summarizes the project's objectives, goals, results, and recommendations. The report will also include the documentation, user manual, test reports, training materials, and installation package.

Table 11: Deliverables

|  |  |
| --- | --- |
| **#** | **Deliverable** |
| 1 | Project Plan |
| 2 | Design Document |
| 3 | Source code |
| 4 | Test Plan |
| 5 | Final report |

1. SYSTEM ANALYSIS AND DESIGN
   1. Overall Description

Systemzen is a Host Based Intrusion Detection System (IDS), which is designed to analyze network traffic and detect any anomalies indicating a potential cyber-attack which may happen.

It is designed to operate on a Single host, allowing it to monitor all incoming and outgoing network traffic on that host. In this System machine learning algorithm is used to analyze the traffic and detect any patterns of a network threat. When an anomaly is detected, Systemzen generates an alert to notify the system of the potential attack.

The user interface is designed to be user friendly, and the system is configurable. Overall, it’s a very useful tool for those organizations which are looking to protect their data from network threats.

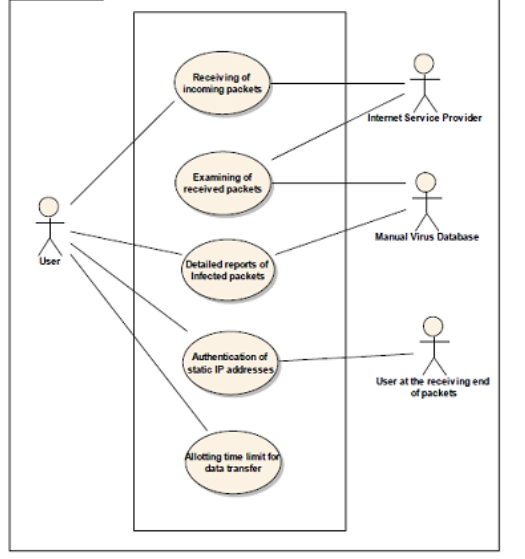
* 1. Users and Roles

Each of these user roles will have different levels of access and functionality within the system. It is important to consider the needs of each user group when designing the system to ensure that it meets the requirements of all stakeholders.

Table 12: Users & Roles

|  |  |
| --- | --- |
| **User** | **Description** |
| Developer | These users are responsible for managing the IDS system, including its configuration, monitoring, and maintenance. They have access to all the features and functionalities of the system and can configure rules, alerts, and notifications. |
| Security Analysts | These users are responsible for monitoring and analyzing the IDS alerts and notifications. They have access to the IDS system's reporting and analysis tools and can investigate security incidents and recommend appropriate responses. |
| End Users | These users may be employees or customers who interact with the systems and applications that the IDS system is designed to protect. They may have limited access to the IDS system and are primarily responsible for reporting security incidents or following security protocols. |

* 1. Design diagrams/ UML diagrams/ Flow Charts/ E-R diagrams
     1. Use Case Diagrams



1. User Interface
   1. UI Description

Our UI is and web-based UI that can be access through a web browser (port 80 localhost) The description of our UI is as follows:

Dashboards – It provides the status of our network, and it tells the number of detected

devices, number of sensors, and a graph of the network traffic.

Sensors - The sensor page provides a list of all sensors which are connected to the servers.

Events - This page provides all the network that has been detected with detailed

information.

Configurations- This page is for configuring settings

Analytics – This page is for graphical visualization for our users which provides insights

of network.

* 1. UI Mockup



Figure 6: Sample 6

1. Algorithms/Pseudo Code
2. class \_set(set):
3. pass
4. try:
5. import \_\_builtin\_\_
6. except ImportError:
7. # Python 3
8. import builtins as \_\_builtin\_\_
9. def print(\*args, \*\*kwargs):
10. ret = \_\_builtin\_\_.print(\*args, \*\*kwargs)
11. sys.stdout.flush()
12. return ret
13. try:
14. import pcapy
15. except ImportError:
16. if IS\_WIN:
17. sys.exit("[!] please install 'WinPcap' (e.g. 'http://www.winpcap.org/install/') and Pcapy (e.g. 'https://breakingcode.wordpress.com/?s=pcapy')")
18. else:
19. msg = "[!] please install 'pcapy or pcapy-ng' (e.g. 'sudo pip%s install pcapy-ng')" % ('3' if six.PY3 else '2')
20. sys.exit(msg)
21. def \_check\_domain\_member(query, domains):
22. parts = query.lower().split('.')
23. for i in xrange(0, len(parts)):
24. domain = '.'.join(parts[i:])
25. if domain in domains:
26. return True
27. return False
28. def \_check\_domain\_whitelisted(query):
29. result = \_result\_cache.get((CACHE\_TYPE.DOMAIN\_WHITELISTED, query))
30. if result is None:
31. result = \_check\_domain\_member(re.split(r"(?i)[^A-Z0-9.\_-]", query or "")[0], WHITELIST)
32. \_result\_cache[(CACHE\_TYPE.DOMAIN\_WHITELISTED, query)] = result
33. Project Closure

The System Zen project was launched with the Goal of an effective host-based intrusion detection system (HIDS) that would help many organizations to detect network threats and to act against these threats. Throughout the project we have made a significant amount of progress towards our goal.

Achievements of our project:

* 1. Developing a user-friendly interface, making it easy for our users to use the system.
  2. Using Machine Learning Algorithm - like Random Forest, Clustering and Anomaly Detection.
  3. Integration of threat detection technologies.

In Short, the project is going good so far and is giving an effective solution to protect against harmful networks in future we look forward to adding automatic delete option for deleting the harmful data when found and add more innovative features.

REFERENCES

1. ‘Network intrusion detection system: A systematic study of machine learning and deep learning approaches’ Zeeshan Ahmad, Adnan Shahid Khan, Cheah Wai Shiang, Johari Abdullah, Farhan Ahmad

2. ‘Survey of intrusion detection systems: techniques, datasets and challenges’ Ansam Khraisat, Iqbal Gondal, Peter Vamplew, Joarder Kamruzzaman

3. ‘The random forest algorithm for statistical learning’ Matthias Schonlau, Rosie Yuyan Zou