**Network Threat Learning System**

Project Documentation Submitted

To the Faculty of School of

Computer Science and Information Technology

Asia Pacific College

In Partial Fulfillment of the Requirements for the subject

Applied Projects 2 or Software Development

By

Balarbar, Adrian

Bernardo, Jan Michael

Mercado, Gavriel

Valencia, Fatima Audrey

Verano, Ruy Josel

Table of Contents

[Project Members 5](#_Toc479983598)

[Project Team Member 5](#_Toc479983599)

[Project Adviser 5](#_Toc479983600)

[Project Professor 5](#_Toc479983601)

[Executive Summary 6](#_Toc479983602)

[I. Introduction 7](#_Toc479983603)

[A. Project Context 7](#_Toc479983604)

[B. Purpose and Description 7](#_Toc479983605)

[C. Objectives 7](#_Toc479983606)

[General Objectives 7](#_Toc479983607)

[Specific Objectives 8](#_Toc479983608)

[II. Review of Related Literature/Systems 8](#_Toc479983609)

[Cyberthreat Intelligence 8](#_Toc479983610)

[Machine Learning 9](#_Toc479983611)

[Intrusion Detection System 9](#_Toc479983612)

[Detection of Attacks 10](#_Toc479983613)

[Snort based Intrusion Detection System 11](#_Toc479983614)

[Automatic Snort IDS rule generator based on Honeypot log 11](#_Toc479983615)

[0-Day Vulnerabilities 11](#_Toc479983616)

[III. Technical Background 12](#_Toc479983617)

[IV. Methodology, Results and Discussion 13](#_Toc479983618)

[A. Requirements Analysis 13](#_Toc479983619)

[B. Requirements Documentation 13](#_Toc479983620)

[C. Design of Software, Systems, Product, and/or Processes 15](#_Toc479983621)

[D. Development and Testing 19](#_Toc479983622)

[E. Description of Prototype 19](#_Toc479983623)

[F. Implementation Plan 20](#_Toc479983624)

[G. Implementation Results 20](#_Toc479983625)

[V. Conclusions and Recommendation 20](#_Toc479983626)

[VI. Appendices 21](#_Toc479983627)

[A. Diagrams 21](#_Toc479983628)

[a) Use case 21](#_Toc479983629)

[b) Use Case Narrative 22](#_Toc479983630)

[c) Functional Decomposition Diagram 26](#_Toc479983631)

[d) Object Diagram 26](#_Toc479983632)

[e) Class Diagram 27](#_Toc479983633)

[f) Sequence Diagram 27](#_Toc479983634)

[g) Activity Diagrams 28](#_Toc479983635)

[h) Context Flow Diagram 31](#_Toc479983636)

[i) Data Flow Diagram 31](#_Toc479983637)

[j) Entity Relationship Diagram 33](#_Toc479983638)

[k) Component Diagram 34](#_Toc479983639)

[l) Packaging Diagram 34](#_Toc479983640)

[m) Deployment Diagram 35](#_Toc479983641)

[B. Project Vision and Scope 35](#_Toc479983642)

[Business Requirements 35](#_Toc479983643)

[Business Opportunity 36](#_Toc479983644)

[Business Objectives and Success Criteria 36](#_Toc479983645)

[Business Risks 36](#_Toc479983646)

[Vision of the Solution 36](#_Toc479983647)

[Vision Statement 37](#_Toc479983648)

[Major Features 37](#_Toc479983649)

[Scope of Initial Release 37](#_Toc479983650)

[Business Context 38](#_Toc479983651)

[Stakeholder Profiles 38](#_Toc479983652)

[Project Priorities 39](#_Toc479983653)

[Operating Environment 39](#_Toc479983654)

[C. Work Breakdown Structure 40](#_Toc479983655)

[D. Gantt Chart 41](#_Toc479983656)

[E. Software Requirements Specification 42](#_Toc479983657)

[I. Introduction 42](#_Toc479983658)

[II. Overall Description 44](#_Toc479983659)

[III. External Interface Requirements 45](#_Toc479983660)

[IV. System Features 46](#_Toc479983661)

[V. Other Nonfunctional Requirements 48](#_Toc479983662)

[VI. Other Requirements 49](#_Toc479983663)

[F. Estimates 49](#_Toc479983664)

[G. Bibliography 49](#_Toc479983665)

# Project Members

## Project Team Member

|  |  |  |
| --- | --- | --- |
| **Name** | **Position** | **E-Mail** |
| Bernardo, Jan Michael | Project Manager | jbernardo@student.apc.edu.ph |
| Valencia, Fatima Audrey | Systems Analyst | favalencia@student.apc.edu.ph |
| Balarbar, Adrian John | Systems Designer | avbalarbar@student.apc.edu.ph |
| Verano, Ruy Josel | Network Designer | rgverano@student.apc.edu.ph |
| Mercado, Gavriel | System Designer | gamercado@student.apc.edu.ph |

## Project Adviser

|  |  |  |
| --- | --- | --- |
| **Name** | **Position** | **E-Mail** |
| Justin David G. Pineda | Adviser | justindavidpineda@gmail.com |

## Project Professor

|  |  |  |
| --- | --- | --- |
| **Name** | **Position** | **E-Mail** |
| Mr. Manuel Sebastian "Sean" Sanchez | Professor | manuels@apc.edu.ph |

# **Executive Summary**

Zero-day vulnerabilities and exploits present a serious risk to company data. The team is proposing to develop a detection system that would be able to identify zero-day attacks. To achieve this, the team will utilize the following tools: Honeypot and SNORT. The team will set-up the Honeypot, a dummy server, then connect it to SNORT, an Intrusion Detection System (IDS), as well as a database developed by the team. If the Honeypot is attacked, the attacks will be recorded by the Honeypot. From there, the team will analyze the reports that the Honeypot produces. Through this process, the team will be able to make **general rules** that will be put in SNORT that would alert the admin of an attack. From the rules, the system will be able to generate its own specific rules based on the general rules made by the Admin. After making the rules, the system will now place them in the database, to track how frequent and severe the attacks are. This allows the team to provide detailed information of the attack that can be utilized by potential clients. This project aims to make a system that will be adaptable to different organizations, that also has various needs in terms of securing their network.

# **Introduction**

## Project Context

Every day, attackers are finding more ways to exploit and destroy systems or networks using malware, botnets, etc. The way attackers evolve is extremely alarming, finding a solution for these problems must be focused on to prevent future attacks. There are various tools that can help mitigate, prevent and detect malicious logs. The team likes to test different ways of using these tools to further enhance the performance of these tools and provide better safety precautionary measures.

* 1. Purpose and Description

Nowadays, different types of attacks emerge every now and then. Some of these attacks are created for leisure and to access highly secured confidential data. With this information, newly created attacks are harmful for any company and for the welfare of its employees; thus, detection of newly created attacks is necessary and critical. They need to determine if a simple log contains malicious intent to avoid unnecessary corruption and loss of data.

Asia Pacific College is a center for IT excellence so to further enhance the order of operations in the ITRO in terms of network security. The team will use Asia Pacific College (APC) as a piloting company to be able to test out the system. The potential market of this innovation is those companies that want added protection on their network.

## Objectives

### General Objectives

* + This project aims to provide a system that will efficiently identify zero-day attacks, learn about zero-day attacks and analyze zero-day attacks.

### Specific Objectives

* To be able to provide rules to prevent and save zero-day attacks.
  + - To effectively use a Honeypot server connected to SNORT to lure unsaved attacks.
* To be able to use correlation to identify new and unsaved types of attacks.
* To successfully adapt when new attacks are present
  1. Scope and Limitations

This project aims to help network administrators to efficiently secure their network. The project primarily focuses on detection of network attacks and correlation of attacks. The system can monitor any type of network attack, as long as it is in the parameter of the rules stored in the database.

1. **Review of Related Literature/Systems**

## Cyberthreat Intelligence

In the last several years, a disturbing trend—attackers are innovating much faster than defenders are. Large botnets are available for rent, allowing attackers to send spam or launch DDoS attacks at will. Many attackers reuse malware and command and control protocols and methods, adapting their “products” over time to keep ahead of the antimalware industry and security professionals. The idea behind cyberthreat intelligence is to provide the ability to recognize and act upon indicators of attack and compromise scenarios in a timely manner. While bits of information about attacks abound, cyberthreat intelligence (CTI) recognizes indicators of attacks as they progress, in essence putting these pieces together with shared knowledge about attack methods and processes. Cyberthreat intelligence, when used correctly, can help defenders detect attacks during—and ideally before—these stages by providing indicators of actions taken during every stage of the attack. (Shackleford, 2015)

## Machine Learning

Traditionally, network intrusion detection systems (NIDS) are broadly classified based on the style of detection they are using: systems relying on misuse-detection monitor activity with precise descriptions of known malicious behavior, while anomaly-detection systems have a notion of normal activity and flag deviations from that profile.1 Both approaches have been extensively studied by the research community for many years. However, in terms of actual deployments, the team observed a striking imbalance: in operational settings, of these two main classes most people find almost exclusively only misuse detectors in use—most commonly in the form of signature systems that scan network traffic for characteristic byte sequences. It can be surprising at first to realize that despite extensive academic research efforts on anomaly detection, the success of such systems in operational environments has been very limited. In other domains, the very same machine learning tools that form the basis of anomaly detection systems have proven to work with great success, and are regularly used in commercial settings where large quantities of data render manual inspection infeasible. (Robin Sommer, 2010)

## Intrusion Detection System

Intrusion Detection System (IDS) is meant to be a software application which monitors the network or system activities and finds if any malicious operations occur. Tremendous growth and usage of internet raises concerns about how to protect and communicate the digital information in a safe manner. Nowadays, hackers use different types of attacks for getting the valuable information. Many intrusion detection techniques, methods and algorithms help to detect these attacks. These application areas made the network an attractive target for the abuse and a big vulnerability for the community. Malicious users or hackers use the organization’s internal systems to collect information and cause vulnerabilities like Software bugs, Lapse in administration, leaving systems to default configuration. As the internet emerge into the society, new stuff like viruses and worms are imported. (S.Vijayarani, 2015)

## Detection of Attacks

Intrusion detection systems (IDSs) are usually deployed along with other preventive security mechanisms, such as access control and authentication, as a second line of defense that protects information systems. There are several reasons that make intrusion detection a necessary part of the entire defense system.

Traditional systems and applications were developed without security in mind. In other cases, systems and applications were developed to work in a different environment and may become vulnerable when deployed intrusion detection complements these protective mechanisms to improve the system security. Moreover, even if the preventive security mechanisms can protect information systems successfully, it is still desirable to know what intrusions have happened or are happening, so that we can understand the security threats and risks and thus be better prepared for future attacks. The attack can be launched in term of fast attack or slow attack. Fast attack can be defined as an attack that uses a large amount of packet or connection within a few second. Meanwhile, slow attack can be defined as an attack that takes a few minutes or a few hours to complete. Currently IDS is used as one of the defensive tools in strengthens the network security especially in detecting the first two phases of an attack either in form slow or fast attack An intrusion detection system can be divided into two approaches which are behavior based anomaly and knowledge based. Hence, unknown attacks in network intrusion pattern and characteristic might not be capture using this technique. Anything that does not correspond to the system profile is flagged as intrusive. False alarms generated by both systems are major concern and it is identified as a key issue and the cause of delay to further implementation of reactive intrusion detection system. (Amrita Anand, 2012)

## Snort based Intrusion Detection System

Snort is the signature based anomaly detection method. It captures the incoming packets that are transmitted over the network (Roesch, 1999). It incorporates rules within it and thereby performs preprocessing by itself. It mainly reduces the burden of system administrator. New rules can be included within the rule set per the occurrence of new attacks. Snort is used with statistical methods to improve the detection strategy in real time. (G.V. Nadiamma, 2013)

## Automatic Snort IDS rule generator based on Honeypot log

Intrusion detection is an attempt to monitor and detect illegal data stream. This is one way that can be used to reduce the illegal actions enters the network. The data flow to be detected is divided into two categories, namely Signature Based intrusion and Anomaly Based intrusions. Snort rule generation based on honeypot log has been developed at various university and national labs research cyber security concerns. (Sagala, 2015)

## 0-Day Vulnerabilities

One such attack that this thesis will address is the zero-day, or commonly abbreviated 0-day, vulnerability which can cause damage as it is unknown what the target of the attack might be when it is first utilized or how to guard against it. 0-day vulnerabilities are those that have just been released and may or may not have a patch against them, some of which may not be known to the vendors for a patch to be created. Several applications exist to help security analysts find these vulnerabilities. One category of these tools is Intrusion Detection Systems, or IDS, which monitor network traffic for predefined patterns. An intrusion detection system, IDS, provides a stack of hardware and software components that listen to all traffic on a hardware network interface card or from a packet capture file and scans each packet looking for predefined patterns, such as a flood of packets, or invalid packet headers which may indicate a problem. It then responds to these patterns in a predetermined matter, from recording the attempt, to warning administrators through email or visual alerts. As hackers become more creative in their attacks and begin using exploits that may or may not be known to the environment, these systems will help in analyzing potential threats. (Truhan, 2011)

1. **Technical Background**

Hardware

Any hardware will do if it is running in a Windows based OS with 64-bit OS. Best Recommended is at least Windows 7 OS, Intel(R) Core(TM) i5-2430M CPU @ 2.40GHz 2.40 GHz, 4.00GB RAM. Higher specs would be much better. Should have 2 laptop or computer, one for the system and the other one is for testing attacks.

Software

Honeybot

Honeybot 0.1.8 is a Windows based honeypot that is easy to use for logging capability and captures raw packet level data even including the keystrokes and mistakes made by hackers. Captures only malicious traffic with little to no false positives. Used for real time monitoring of the malicious attacks that is covered by the parameters of the honeypot that attack the network. and also, used to gather data that have a suspicious behavior.

Honeybot is automatically configured but the adapters that you will be monitoring will be chosen by the user.

SNORT

Snort 2.9.2.2-ODBC-MySQL-WIN32 is an Intrusion Detection System, it requires a winpcap software for snort to recognize the interfaces. Snort 2.9.2.2 is an old version of Snort that is used to be able connect it to MySQL database and record the data in it since it is used just to identify attacks and alerts. A tool used in the Windows based OS, Accessible and the command prompt in the bin folder.

XAMPP

A simple Apache distribution that makes use by the developers to create a local web server for testing and deployment purposes. Wherein, it is a user-friendly tool that uses is used to easily access the database server and store data in it. The MySQL database and Apache should be running to access the database and for the system to work properly.

NMap

It is a open source network discovery and security auditing tool. Nmap runs on all major computer operating systems, and official binary packages are available for Linux, Windows, and Mac OS X. In addition to the classic command-line Nmap executable, the Nmap suite includes an advanced GUI and results viewer (Zenmap), a flexible data transfer, redirection, and debugging tool (Ncat), a utility for comparing scan results (Ndiff), and a packet generation and response analysis tool (Nping).

# **Methodology, Results and Discussion**

* 1. Requirements Analysis

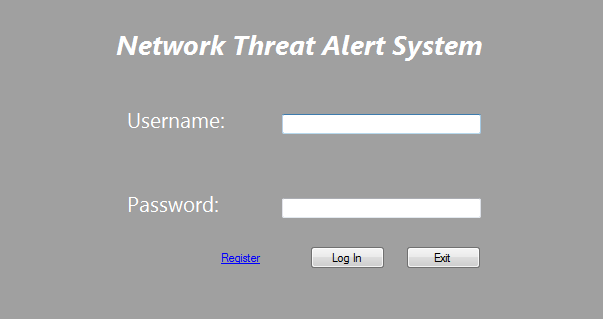
Organizations have a hard time securing their networks since most attack that happens are not identified by the security measures placed by them. The team is providing a way to help organizations to efficiently analyze gathered network data information that will help them in the long run to enhance their network security.

* 1. Requirements Documentation

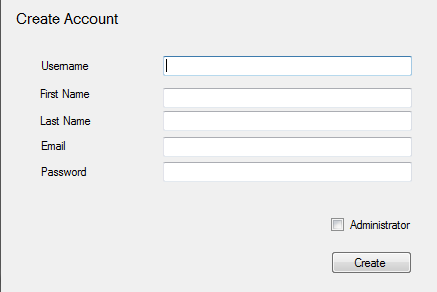
|  |
| --- |
|  |
| **Output**  ✓The Honeypot must produce a report or logs every time there is an attack to it, consisting the Number of the attack, IP Address, and Pattern of attack consisting (Time, Date, Flag, Port, Quantity)  ✓The System must automatically generate more specific rules based on the General Rule to the Correlated Rules.  ✓The System must alert the Admin every time there is a new generated rule. |
| **Input**  ✓Users must login with their user account to identify if he’s an Admin, Security Analyst or an employee.  ✓The Admin must create General Rules base on the attacks from the Honeypot.  ✓Each rule must include Rule Number, date, time, and attack type.  ✓All Logs must be stored to the database.  ✓The Employee must only be able to report attacks using a text box. |
| **Process**  ✓The System sends the Honeypot Raw Logs to Snort.  ✓The System checks the rules if there are similar attributes, and if there are, a new rule will be created.  ✓The System will store the correlated entries to the database. |
| **Performance**  ✓The Honeypot must record every second, minute and hour of an attack.  ✓The System must alert the admin (there must be a delay) when there are new attacks.  ✓The Honeypot database will delete files every seven (7) days |
| **Control**  ✓The system must provide logon security at the operating system level and at the application level.  ✓To avoid the System’s database overload the same attack must be correlated with another rule.  ✓The system must maintain separate levels of security for users and the system administrator. (like limiting the privileges of the user) |

## Design of Software, Systems, Product, and/or Processes

Login

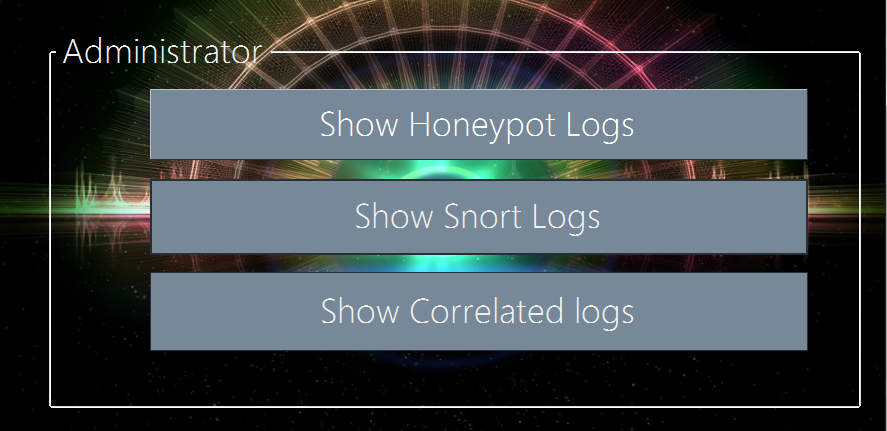


Register



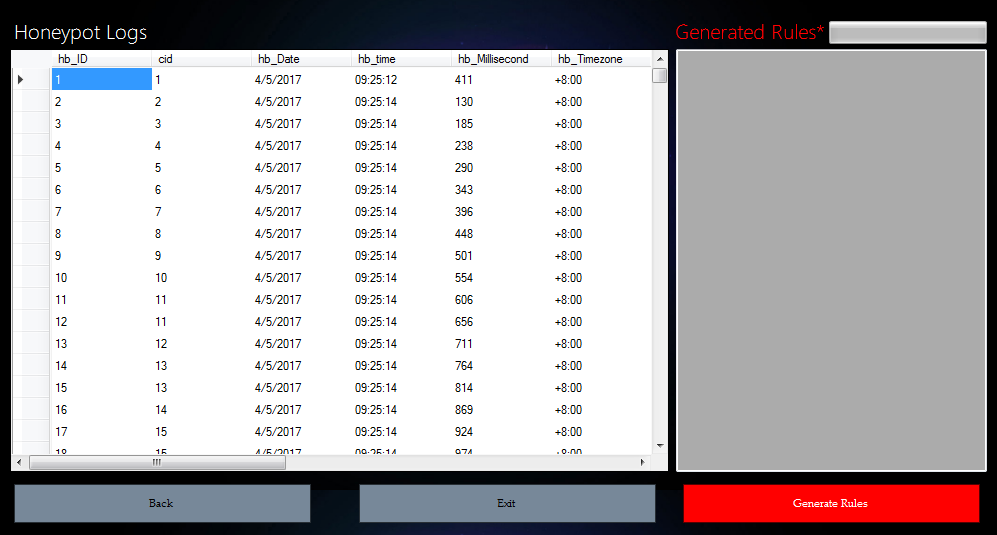
Admin

The Control center for the Administrator that shows the Raw logs, correlated logs, and the honeypot logs.



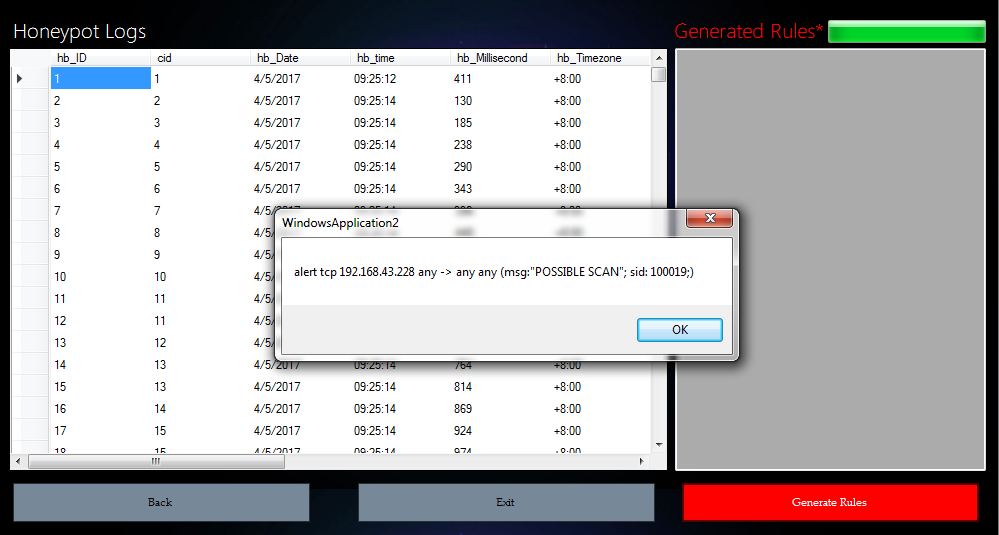
Raw Logs

Logs that came from the IDS to identify the type of attack and alert the admin.



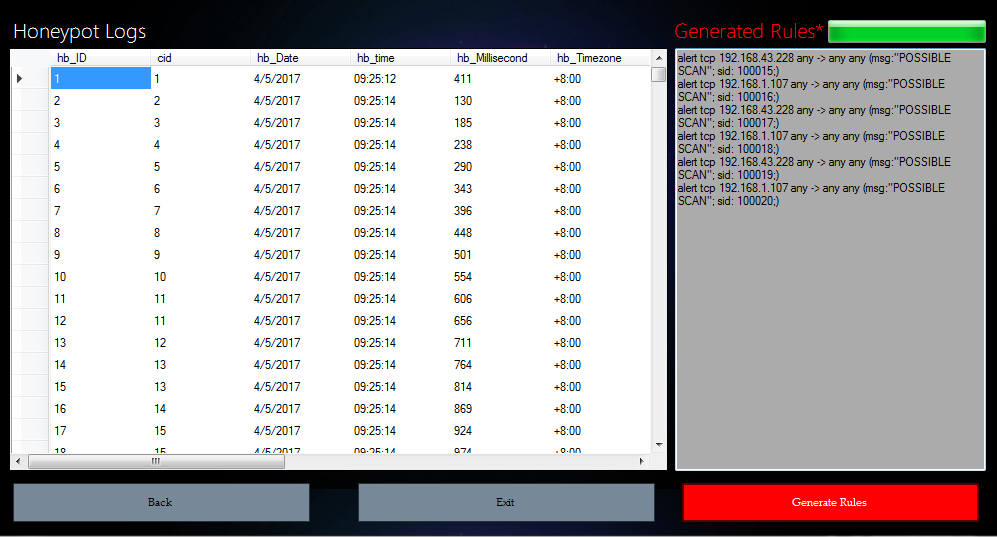
Honeypot Generate Rule

Automatically Generated rule based on the Security Analyst to identify the type of attack.

.

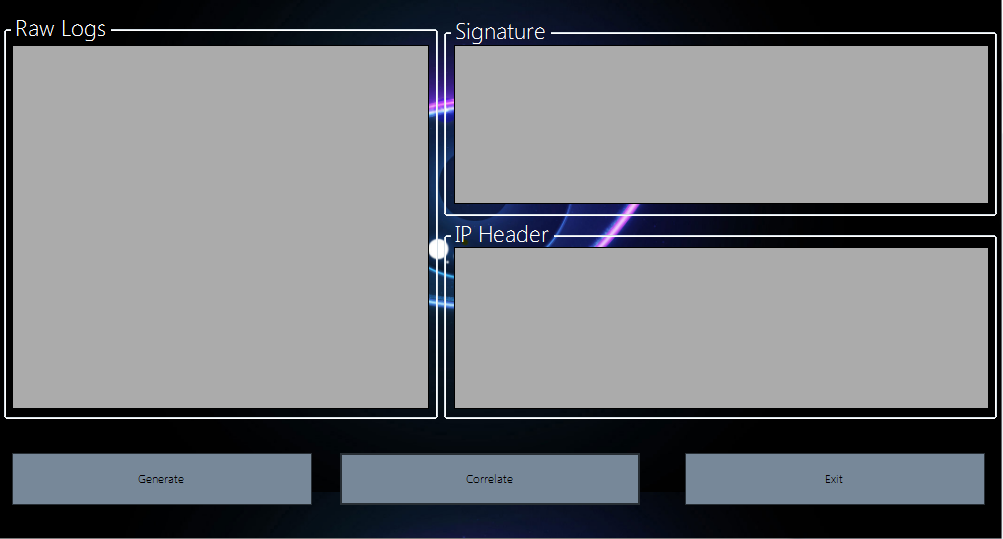
Honeypot Generated Rule

Rules Generated by the System



Snort Raw Logs

Logs that came from the ids to identify the type of attack and alert the admin.



## Development and Testing

The developers of this system have researched related literature and applications thoroughly to ensure that the product will surely work. The team has conducted several tests to make sure that different logs will be saved and that the system will not fail during the process of detecting logs.

## Description of Prototype

The prototype was built using Visual Studio and functions while Honeybot and SNORT is running. It simulates the process of identifying and recording logs from the honeypot server under the supervision of SNORT. The user must first log in or register to create a new user. When the user signs in with a admin account, they may choose to view raw logs from the honeypot server. Then the user may correlate the logs and review them.

## Implementation Plan

The system will be implemented to APC's Information Technology Resource Office where it can be managed and maintained by the staff. The authorizations and permissions will be given to the head director and faculties of his choosing.

## Implementation Results

After implementing, the system shall detect unidentified logs in APC's servers and determine if they are harmful or not. Thus, increasing the security measures while generating new rules to maximize the security capabilities of the establishment.

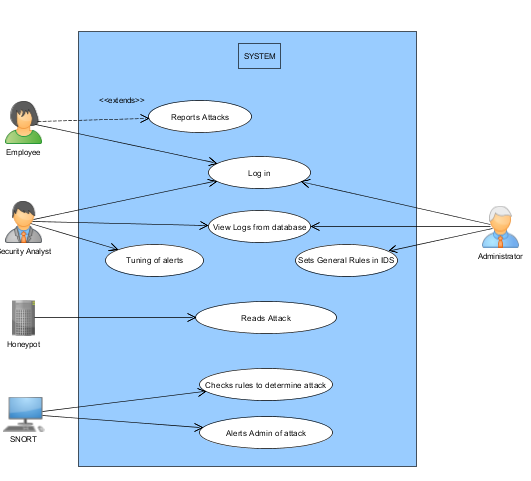
* 1. Conceptual design

Please refer to the diagrams shown on the Appendices.

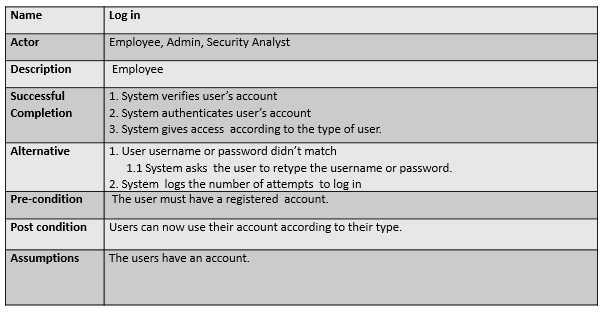
# Conclusions and Recommendation

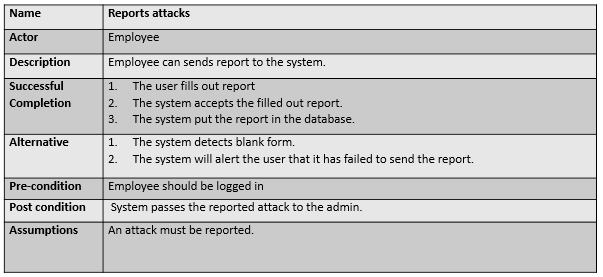
Many companies are at risk with zero-day attacks. Most companies would just put up a firewall and assume that everything is already safe on their network. But these days, a fire wall is not enough for securing an organization’s network. Organizations who want to enhance their network’s protection is this project’s target audience.

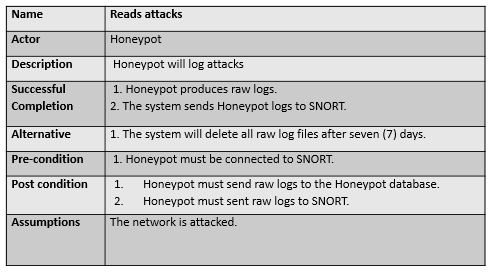
1. **Appendices**
   1. Diagrams
      1. Use case

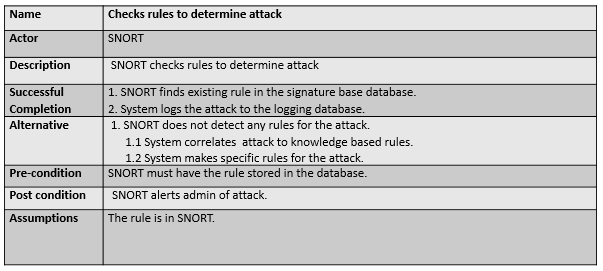


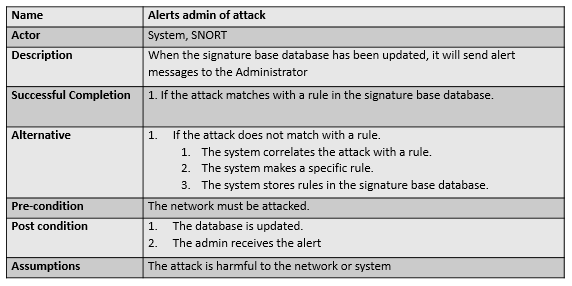
* + 1. Use Case Narrative

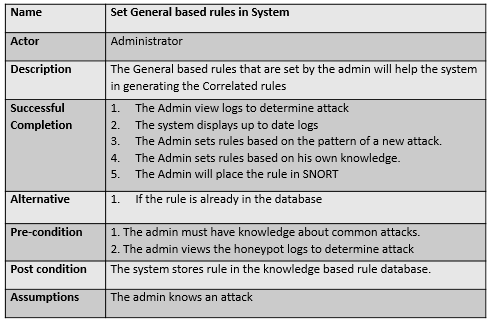


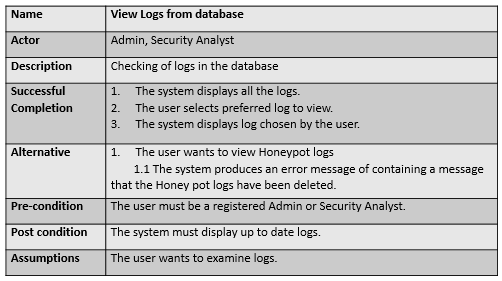


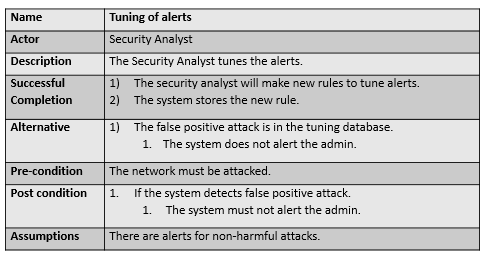




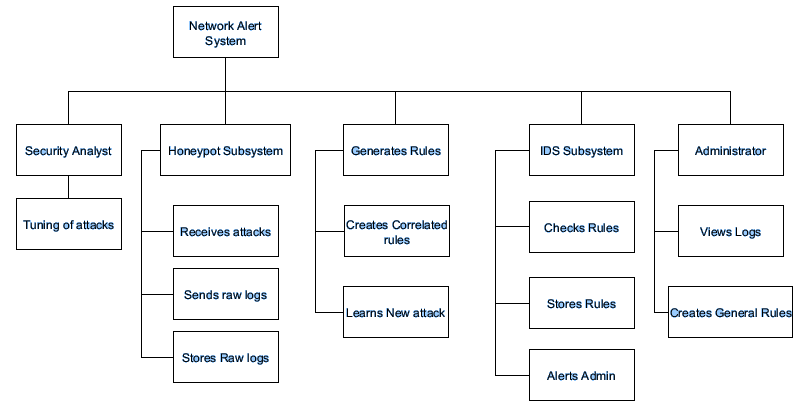




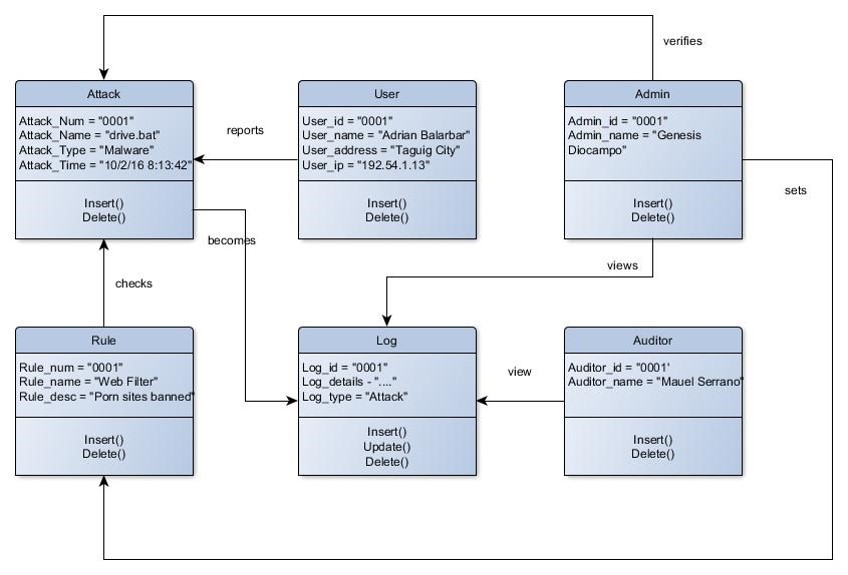




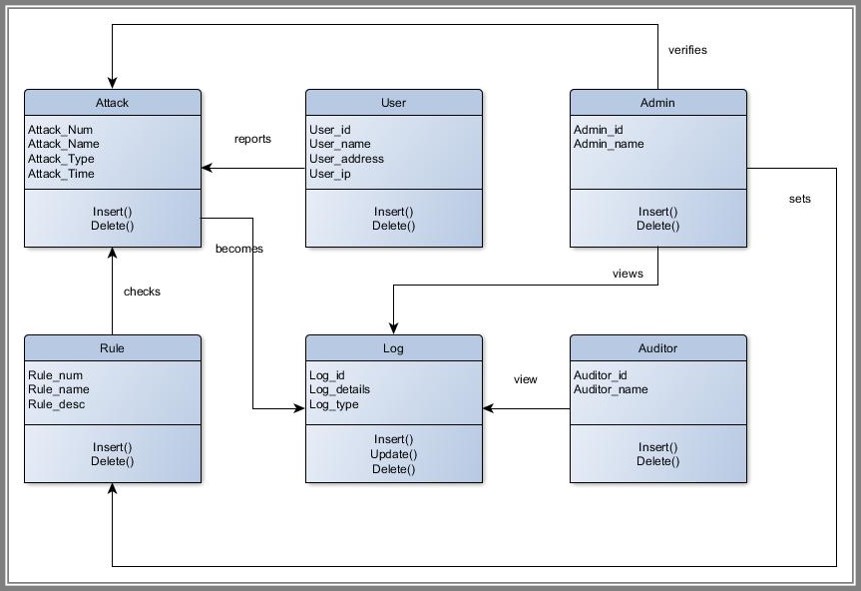
* + 1. Functional Decomposition Diagram



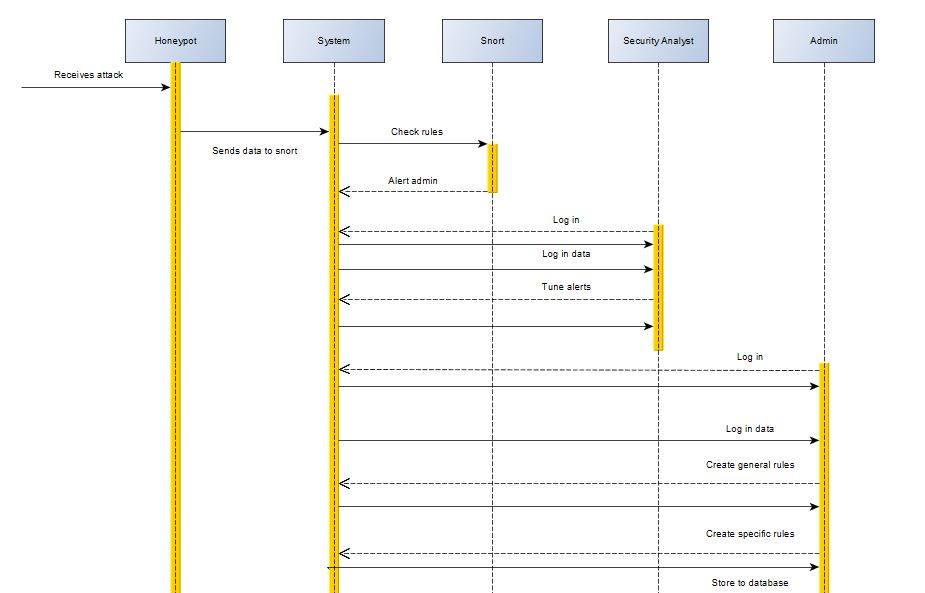
* + 1. Object Diagram



* + 1. Class Diagram

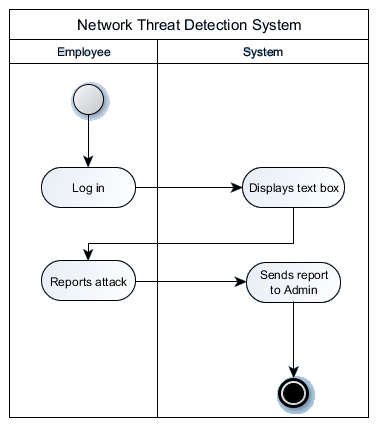


* + 1. Sequence Diagram

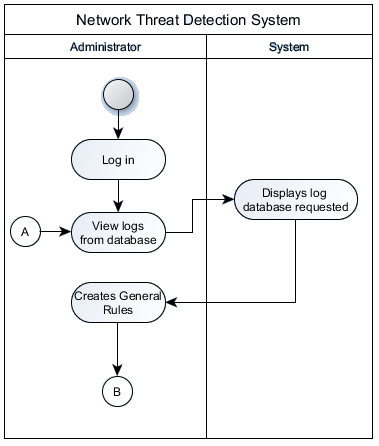


* + 1. Activity Diagrams

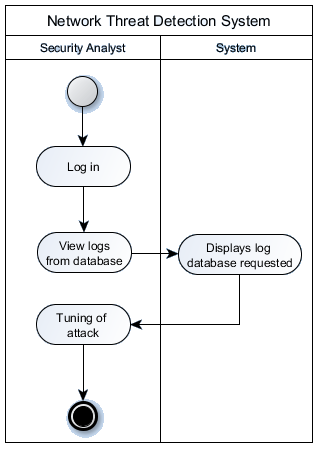
Employee



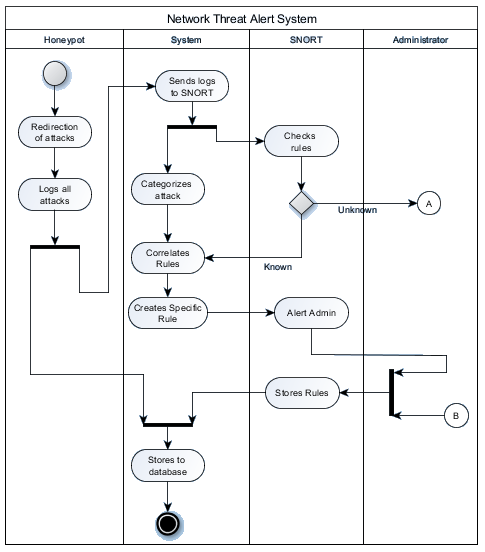
Administrator



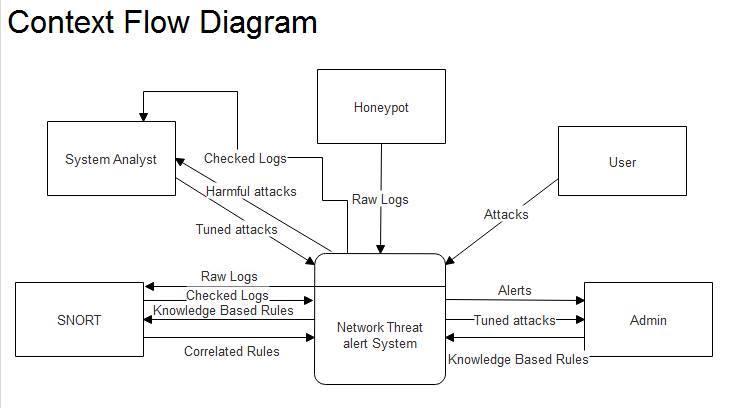
Security Analyst



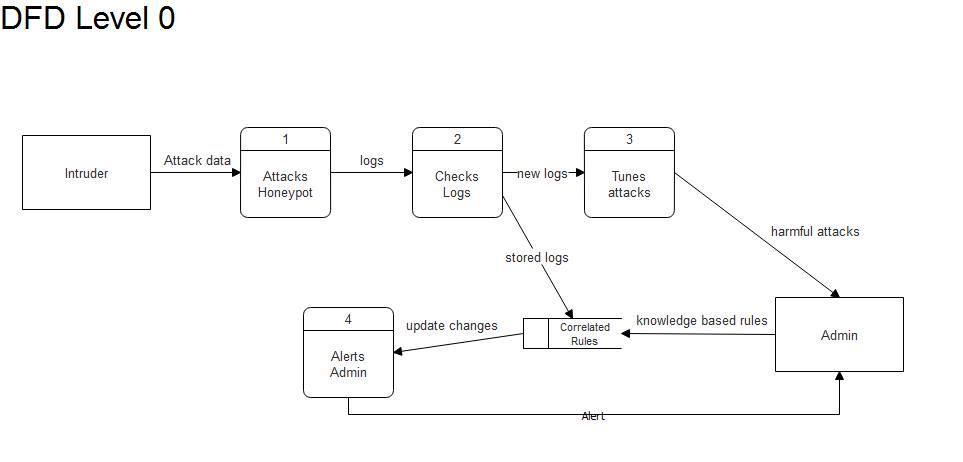
System

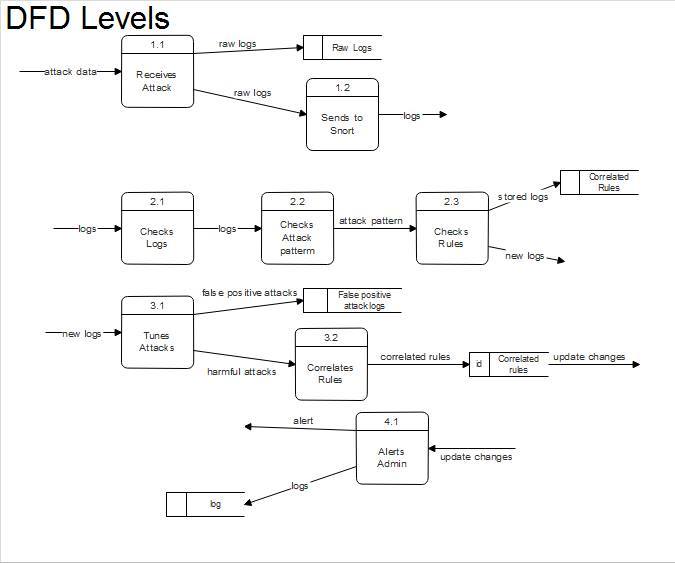


* + 1. Context Flow Diagram

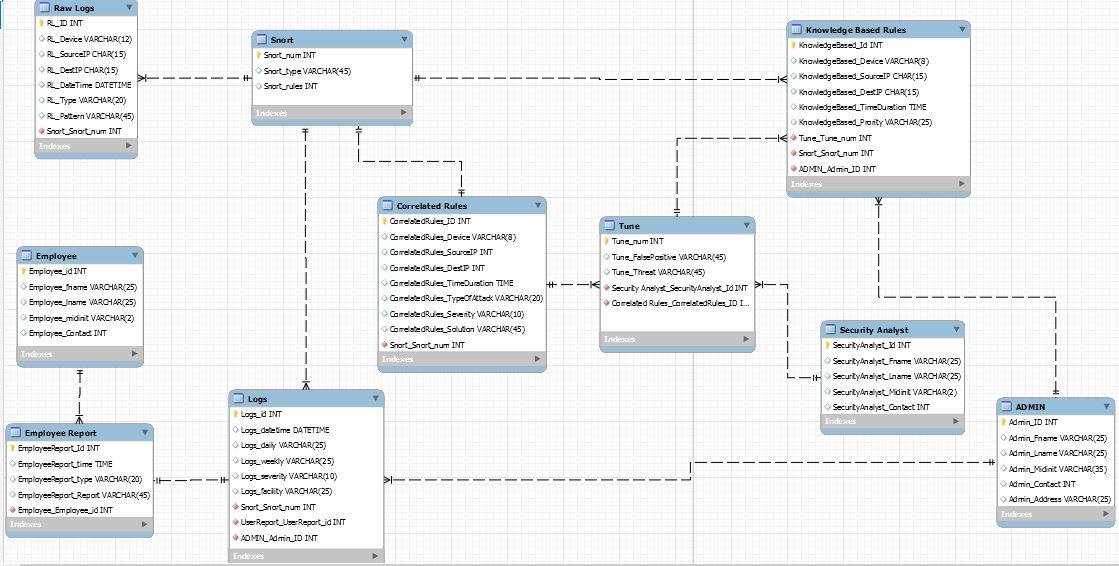


* + 1. Data Flow Diagram

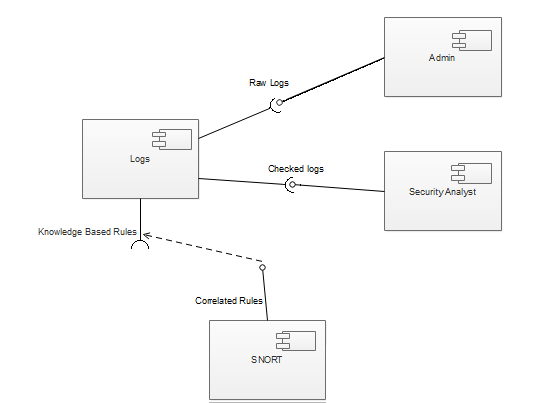




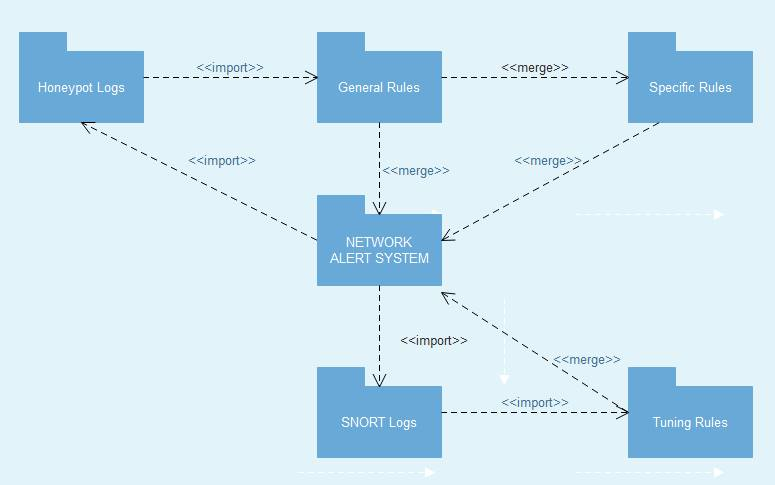
* + 1. Entity Relationship Diagram



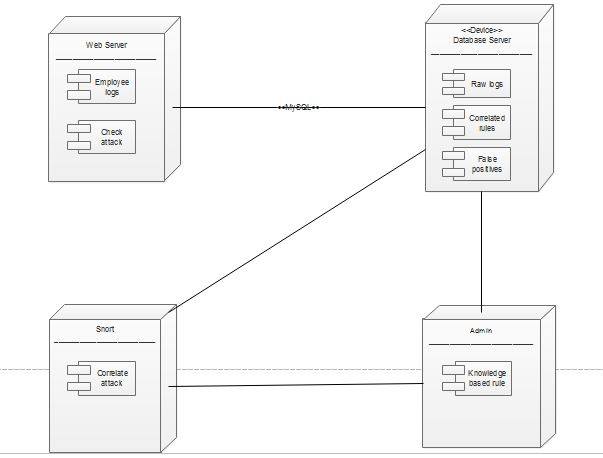
* + 1. Component Diagram



* + 1. Packaging Diagram



* + 1. Deployment Diagram



* 1. Project Vision and Scope

### Business Requirements

The members of the group, along with their adviser came up with the requirements of the system that they are going to implement. The following is the list of processes that the system must perform:

* Detection of attacks and vulnerabilities
* Logs attacks that occur in the system
* Logs the frequency of attacks
* Allows identification with detailed information about attack

### Business Opportunity

The potential market of this innovation is those companies that have trouble securing their network and other valuable assets. Especially nowadays where in new exploits and networks are made almost every day to gather sensitive information or destroy an infrastructure.

### Business Objectives and Success Criteria

The objective of this innovation is to provide a system that will be able to efficiently identify zero-day attacks, learn about zero-day attacks, and analyze zero-day attacks. The success measures of the system are more gathered information and new generated rules. Customer or Market Needs Network administrators have a difficulty in securing their network in an environment where in newly made attacks are made every day. The system will help network administrators lessen their workload. The system will also gather information from attacks, learn different patterns of attacks.

### Business Risks

One of the risk that the developers are trying to manage is the lack of knowledge since the area of network security is very broad. The researchers require vast knowledge on network security. Another risk is that the time constraint of the team developers.

### Vision of the Solution

The system will improve as more attacks and threats are logged and dealt with. This will allow the company to identify and handle the problem quicker and easier.

### Vision Statement

The team intends to provide a system that will help companies improve their security by identifying and logging unpredictable attacks, thus minimizing potential problems and decrease recurring attacks.

### Major Features

1. Detect known attacks base from the signature base engine
2. Record all attacks
3. Categorize the known attacks
4. Monitor the frequency of attacks
5. Alert admin/s of attacks.
6. Learning of zero day attacks.

Assumptions and Dependencies:

1. Database to log attacks
2. SNORT
3. A Honeypot server

Assumptions:

1. A basic knowledge of how to use SNORT
2. A basic knowledge of setting up a honeypot server
3. Analytical skills.
4. The Company has their own security countermeasure.

### Scope of Initial Release

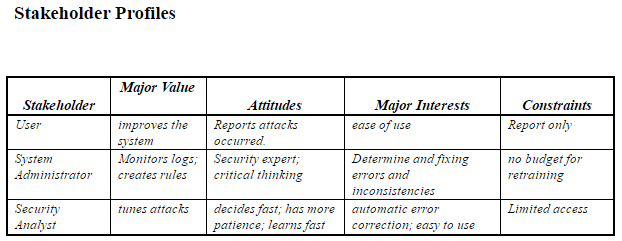
* The system will monitor any type of attack
* The system will be able to learn attacks
* The system practices the three types of correlation

### Business Context

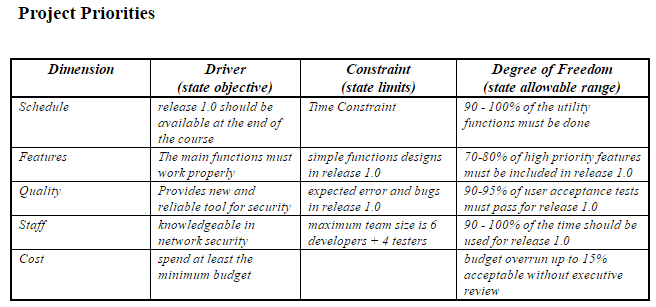
Since our team is making a system that is specialized in security, it could affect many companies. where in it includes:

* The security of the network will increase
* The security analyst doesn't need to monitor the network all the time because our system will alert whenever there are threats to the network.
* Reduce workload and errors.
* More accurate records.

### Stakeholder Profiles



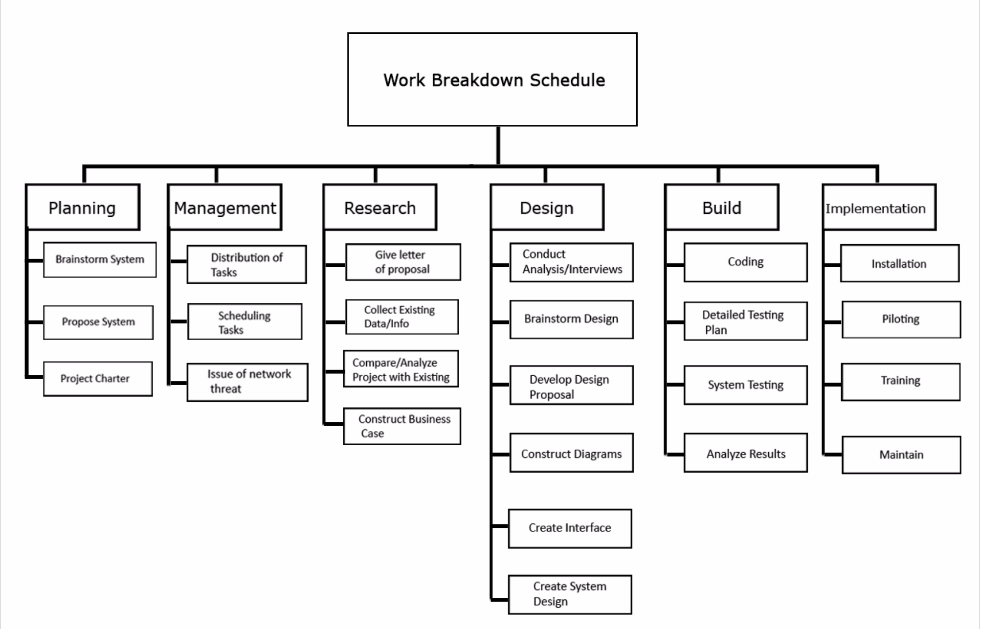
### Project Priorities



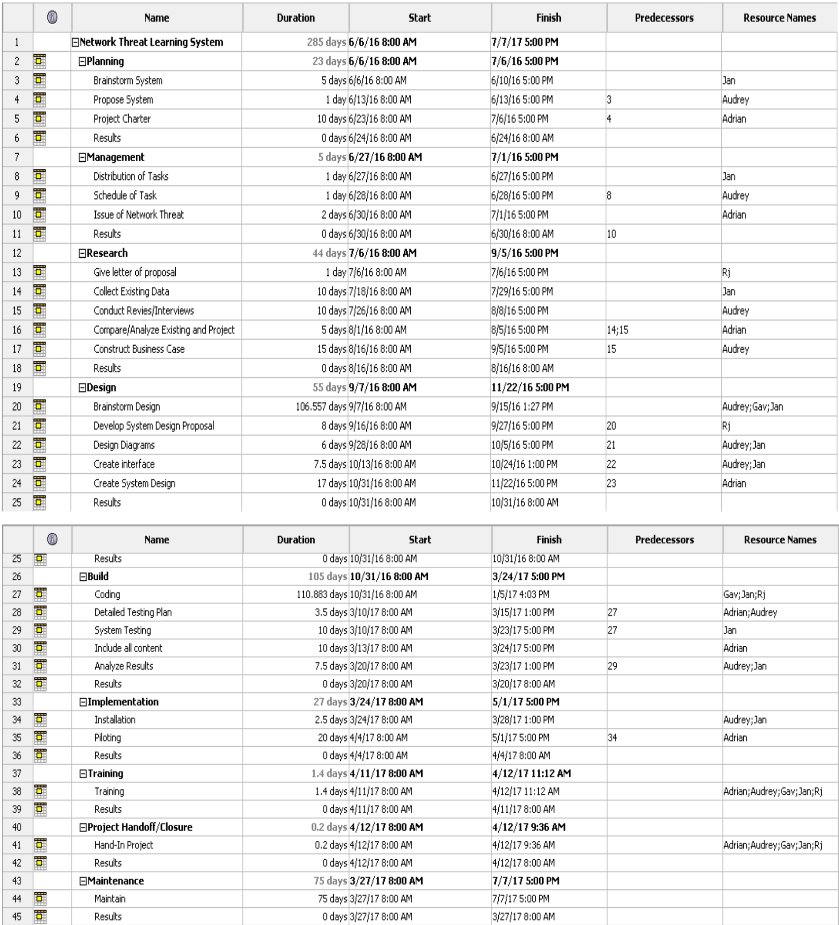
### Operating Environment

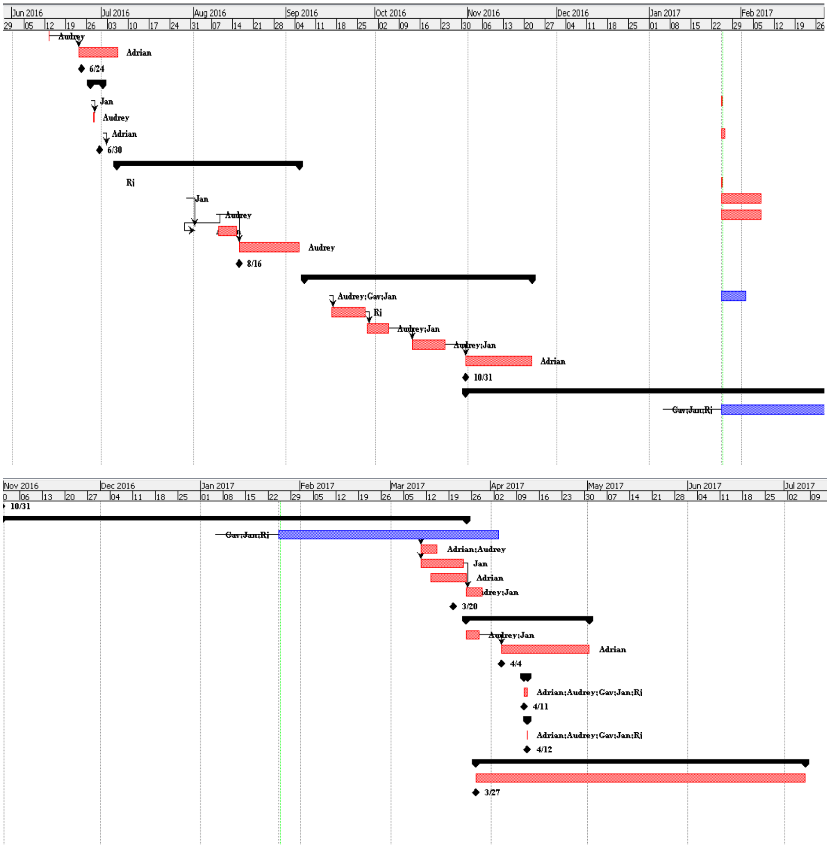
The system will be set up in the honeypot server. The users of the system will be the security administrators, security analyst, and the employees who use the computer. The system will function 24 hours to monitor the attacks happening in the network. The users will access the system when the system alerts the admin, when the admin checks and add rules, when the system analyst checks and tunes alerts, and when there are changes made.

## Work Breakdown Structure



## Gantt Chart





## Software Requirements Specification

### Introduction

1. Purpose

The purpose of this document is to define the components and functions of the system, Network Threat Alert System(NTAS). NTAS is to provide a system that will identify zero-day attacks and generate protocols for them.

1. Document Conventions

When reading this document, one might encounter technical terms, the terms that are to used will be defined as the following:

* Zero-day: it is a flaw of a software or a hardware that are exploited by attackers, before a developer has an opportunity to create a patch to fix the vulnerability.
* Honeypot: is a computer system that acts as a decoy to lure in attackers.
* Rules: it is a methodology that’s different for performing detection. It is based on detecting actual vulnerability.
* Tuning: controlling of alerts from false positive attacks.

1. Intended Audience and Reading Suggestions

The intended audience for this document are network supervisors and other audiences include the NTAS project manager and NTAS project developer. Inside the document includes descriptions, interfaces, and features of the system. The document is intended to be read in that order.

1. Product Scope

This project aims to help network administrators to efficiently secure their network. The project primarily focuses on detection of network attacks and correlation of attacks. The system can monitor any type of network attack, if it is in the parameter of the rules stored in the database.

1. References

This document is also referred to the wiki page of Network Threat Alert System:<http://projects2.apc.edu.ph/wiki/index.php/Project_-_Network_Threat_Alert_System_-108#Scope_and_Limitations>

### Overall Description

* + 1. Product Perspective

The system intends to provide a system that will help companies improve their security by identifying and logging unpredictable attacks and minimizing potential problems to decrease recurring attacks.

* + 1. Product Functions
* Detection of attacks and vulnerabilities
* Logs attacks that occur in the system
* Logs the frequency of attacks
* Allows identification with detailed information about attack
* Generates rules and protocols for attacks
  + 1. User Classes and Characteristics

This product’s users are Security Administrators, Security Analysts, and network supervisors. They will monitor and analyze what attacks are coming into the network. They will also examine the generated protocols for these attacks and may add their own.

There is also an added feature where in Employees may be able to report attacks by filling out a form provided by the system that will be sent to the Security Administrator.

* + 1. Operating Environment

The system will run on a Windows operating system and will be set up in the Honeypot server. The users of the system will be the Security administrators, Security analyst, and the Employees who use the computer. The system will function 24 hours to monitor the attacks happening in the network. The Admin will access the system when the system alerts the admin, while the Security Analyst tunes unwanted alerts.

* + 1. Design and Implementation Constraints

It is essential for the system to be connected to the network and depends heavily on Honeypot to consistently produce logs, and SNORT to generate rules for them. It is essential for the user to know how to operate SNORT in the computer terminal as well as how to configure Honeypot.

* + 1. User Documentation

SNORT 2.9.9 user manual, Honeypot user guide, and NTAS’s manual.

* + 1. Assumptions and Dependencies

Since the system will be implemented and tested in the Information Technology Resource Office, some future alterations will occur to improve the system’s scope. The system will be integrated into the network of APC and monitor network traffic. The system must be maintained to properly generate protocols and log attacks.

### External Interface Requirements

* + 1. User Interfaces

The web interface was created using Visual studios and the components like text boxes, tables, and button designed by visual studio. The logs and rules will be displayed in their respectable tables and text boxes. When the user wishes to correlate the data, the logs will simplify to similar signatures. The user also has the choice to save the data into a text file. When configuring SNORT, the user must use the computer terminal. The user will open Honeypot to configure the honeypot server and its interface.

* + 1. Hardware Interfaces

The system will be accessed through any computer inside the ITRO and connected to the internet.

* + 1. Software Interfaces

The system obtains data through SNORT and Honeybot. Honeybot will provide a honeypot server that produces logs for the system. While honeybot produces the logs, SNORT will be generating rules according to the logs that it reads and stores them. The signatures and ip of the logs will be stored into a database made with MySQL.

* + 1. Communications Interfaces

The user will interact with the system through a web browser and can only be accessible to the school’s network and computers. Any browser may use the system, from Firefox, Google Chrome, and Microsoft Edge.

### System Features

#### Records all logs

* Description and Priority

The recording of all attacks that is happening in the network is highly prioritized, since the best way to monitor the frequency of attack is to be able to recognize and analyze them so that it can be directly addressed in the future.

* Stimulus/Response Sequence

The system would log all on going activities in the network using the Honeypot. The Security Admin and Security Analyst would be able to see the logs.

* Functional Requirements

REQ-1: Honeypot must send the recorded logs to the database.

#### Correlation of attacks and rules

* Description and Priority

This feature would be able compare attacks that are similar, and if they are, there would only be one log for the said attack, and the same goes to rules. This would greatly help in the organization of the attacks and rules

* Stimulus/Response Sequence

There will be a button to click to be able to correlate

* Functional Requirements

REQ-1: The attack/rules have been recorded before.

REQ-2: The type of attack/rule has the same attributes.

#### Learning of zero day attacks

* Description and Priority

This feature would be able to recognize attacks based from their patterns and observe if the system is deviating from normal day-to-day activities. It would also make new rules that would accurately specify the attack.

* Stimulus/Response Sequence

The system will automatically produce rules based on the unusual behavior of the network.

* Functional Requirements

REQ-1: Recorded attacks.

REQ-2: Unusual network behavior.

### Other Nonfunctional Requirements

* + 1. Performance Requirements

Users must login with their user account to identify if he’s a Security Admin, Security Analyst or an Employee. The system must show generated rules base on the attacks from the honeypot as well as correlated logs.

Each rule must include Rule Number, date, time, and attack type.

* + 1. Safety Requirements

SNORT and Honeypot should be running in sync for the system to run effectively. The database must be maintained and organized to prevent any loss of data. Correlating logs and saving data to a text file regularly may minimize data loss. Only computers within the ITRO can access the system and the ability to configure the system will be given to with the most authority.

* + 1. Security Requirements

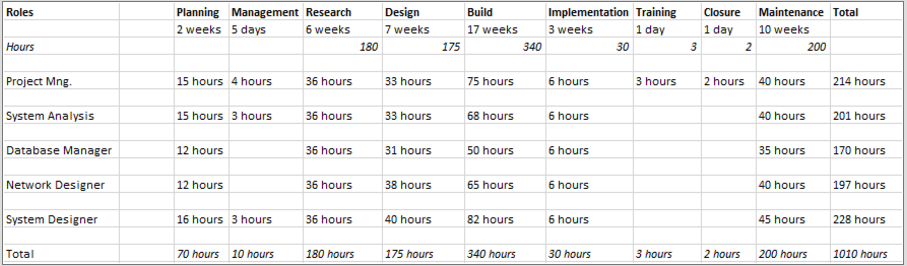
The system must provide logon security at the operating system level and at the application level. To avoid the System’s database to overload, the same attack must be tuned in the same Rule. The system must maintain separate levels of security for users and the system administrator. The Security Analyst must very if the attack is false positive or truly a threat

* + 1. Business Rules
* The Security Admin and Security Analyst can view the logs.
* The Security Admin can make new rules.
* The Security Analyst can tune attacks.
* The Employee can report suspicious computer activities.
* The Security Admin will verify the employee’s report.
* The system will correlate attacks/rules.
* The system will make specific rules.

### Other Requirements

There are currently no other requirements needed.

## Estimates



## Bibliography

(n.d.).

Amrita Anand, B. P. (2012, August). *An Overview on Intrusion Detection System and Types of Attacks It Can Detect Considering Different Protocols*. Retrieved from International Journal of Advanced Research in Computer Science and Software Engineering: https://pdfs.semanticscholar.org/72e2/b7045910bda635d0864117438553d13c45b6.pdf

G.V. Nadiamma, M. H. (2013, January 11). *Handling Intrusion Detection System using Snort Based Statistical Algorithm and Semi-supervised Approach*. Retrieved from Pak Academic Search: http://pakacademicsearch.com/pdf-files/eng/392/2914-2922%202013(Vol.%206,%20Issue%2016).pdf

Robin Sommer, V. P. (2010, May 16). *Outside the Closed World:*. Retrieved from UTDALLAS: https://www.utdallas.edu/~muratk/courses/dmsec\_files/oakland10-ml.pdf

S.Vijayarani, M. S. (2015, February). *INTRUSION DETECTION SYSTEM – A STUDY*. Retrieved from AIRCC Publishing Corporation: http://airccse.org/journal/ijsptm/papers/4115ijsptm04.pdf

Sagala, A. (2015, October). *Automatic SNORT IDS Rule Generation Based on Honeypot Log*. Retrieved from Research Gate: https://www.researchgate.net/publication/301325633\_Automatic\_SNORT\_IDS\_rule\_generation\_based\_on\_honeypot\_log

Shackleford, D. (2015, February). *Who's Using Cyberthreat Intelligence and How?* Retrieved from Sans: https://www.sans.org/reading-room/whitepapers/analyst/cyberthreat-intelligence-how-35767

Truhan, N. D. (2011, August). *INTRUSION DETECTION FOR 0-DAY VULNERABILITIES*. Retrieved from OhioLINK: https://etd.ohiolink.edu/rws\_etd/document/get/kent1310682482/inline