Implementation of an Open-World Recognition-capable Network Intrusion Detection System

Project or

Version 0.1

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 20/Oct/23 | 0.1 | Created Document | Andrew Bajumpa Alexandre Broggi Joseph ODowd |

Table of Contents

1. Introduction 5

1.1 Purpose 5

1.2 Scope 5

1.3 Definitions, Acronyms and Abbreviations 5

1.4 References 5

1.5 Overview 6

2. Project Overview 6

2.1 Project purpose, scope and objectives 6

2.2 Assumptions and constraints 6

2.3 Project deliverables 6

2.4 Evolution of the Software Development Plan 6

3. Project Organization 6

3.1 Organizational Structure 6

3.2 External Interfaces 6

3.3 Roles and Responsibilities 6

4. Management Process 8

4.1 Project Estimates 8

4.2 Project Plan 8

4.2.1 Phase Plan 8

4.2.3 Project Schedule 8

4.2.4 Project Resourcing 8

4.2.5 Budget 9

4.3 Project Monitoring and control 9

4.3.1 Requirements management plan 9

4.3.2 Schedule control plan 9

4.3.3 Quality control plan 9

4.3.4 Reporting Plan 9

4.3.5 Measurement Plan 9

4.4 Risk Management plan 9

4.5 Close-out plan 9

5. Technical process plans 9

5.1 Methods, tools and techniques 9

6. Supporting process plans 10

6.1 Configuration management plan 10

6.2 Evaluation plan 10

6.3 Documentation plan 10

6.4 Quality assurance plan 10

7. Annexes 10

# 

# Introduction

The project IOWR-NIDS is an “Implementation of Open-World Recognition for a Network Intrusion Detection System”. The project will use prior research in the field of Open-World Recognition to create a Real-Time packet identification tool that is capable of identifying when it is not likely to have gotten the packet identification correct.

## Purpose

This Software Development Plan should specify our team’s direction of efforts while trying to accomplish our goals of having a minimum viable product of IOWR-NIDS created by the date of December 4th and a final product created by the end of the spring semester. This document should include our schedule and plans for achieving the first of those goals and outlines for achieving the second.

## Scope

This software should be a stand-alone product to be used by a network administrator for identifying packets.

## Definitions, Acronyms and Abbreviations

IOWR – Implementation of Open World Recognition

NIDS – Network Intrusion Detection System

Please refer to the Glossary in the references for complete definitions of terms.

## References

Vision document: 10/6/2023

Glossary: 10/13/2023

Software Development Plan: 10/20/2023

SRS: 11/3/2023

SAD: 11/10/2023

Risk assessment: 11/17/2023

Testing plan: 11/29/2023

MVP: 12/4/2023

Software Manual: TBD

Installation Manual: TBD

Assurance and Reliability Documentation: TBD

## Overview

The rest of this document is the plans created by our team for directing this project going forwards. This is section 1 which provides a meta-explanation of this document. We will start with an overview of what the project is and what we will be doing in the coming months, this is in section 2. We will then go on to say our team’s timetables for completing milestones, which will be section 3. Section 4 will cover our plans for when things go wrong or go overtime. Section 5 describes our core values with the development of the software. Section 6 defines our evaluation procedures. After that we have our additional information.

# Project Overview

## Project purpose, scope and objectives

This project aims to produce a front-end tool and implementation of an AI enabled, open world recognition, network intrusion detection system. The product will be a GUI tool that can be used by experienced technicians to detect various forms of malicious network traffic, respond appropriately, and retrain the model as needed. The deliverables also include assurance and reliability documentation and usage and installation guides.

## Assumptions and constraints

It is assumed that the final software will not be restricted by hardware limitations. The software is designed with a Linux system in mind. Considering the amount of time the existing research project has been going on for, our biggest constraint is the duration of the course.

## Project deliverables

|  |  |
| --- | --- |
| **Deliverable** | **Delivery Date** |
| Vision document | 10/6/2023 |
| Glossary | 10/13/2023 |
| Software Development Plan | 10/20/2023 |
| SRS | 11/3/2023 |
| SAD ☹ | 11/10/2023 |
| Risk assessment | 11/17/2023 |
| Testing plan | 11/29/2023 |
| MVP | 12/4/2023 |
| Software Manual | TBD |
| Installation Instruction | TBD |
| Assurance + Reliability Documentation | TBD |

## Evolution of the Software Development Plan

The software development plan will be updated to reflect changes to the project as unforeseen challenges, revisions, and deviations naturally arise.

# Project Organization

## Organizational Structure

The primary team consists of three members, all of whom participate in three scrum meetings a week. Alexandre acts as the scrum master. The project is overseen by the client, Nathanial Bastian, and a related project manager, Gokhan Kul; they review the progress and ensure everything is on-track. Professor Khatib acts as an intermediary and supervisor when necessary.

As a team we have a leader in Andrew, but we decide the majority of our decisions that do not affect one of the other stakeholders as a group vote. Because we have three team members votes will always break the group into a majority decision.

## External Interfaces

The team communicates with the client, Nathanial Bastian, and a related project manager, Gokhan Kul, via email and biweekly meetings on Teams. The team communicates with the professor, Dr. Khatib, briefly in-person after CIS498 lectures.

## Roles and Responsibilities

|  |  |
| --- | --- |
| **Person** | **Process Role** |
| Andrew Bajumpa, Team Lead | Software Engineer  Repo/CICD Admin  Code Reviewer |
| Alexandre Broggi, Scrum Manager | Machine Learning Engineer  Scrum Manager  Test Designer Tester |
| Joseph ODowd, Front-end Developer | User Interface Designer   Designer  Implementer |
| Nathanial Bastian | Requirements Specifier |
| Gokhan Kul | Project advisor |

| **Activities** | **Document name** | **Elaboration** | **Technical Control** | **Approval** |
| --- | --- | --- | --- | --- |
| Identify the project main idea | Vision Document | A document describing what the team will be doing for the project | Development Team | Project manager |
| Setting up the Development tools | Software Development Plan | Explains the requirements and plan for the project’s development at all stages | Development Team | Project manager |
| Software specifications | Software Requirements Specification | Lists the different requirements by the client for the end product | Development Team | Project manager |

# Management Process

## Project Estimates

Development will require no dedicated hardware or hosting, so expected costs are $0.00 USD.

## Project Plan

### Phase Plan

### Releases

Minimum Viable Product due December 4th

Final Product due Q1 2024

### Project Schedule

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Phase** | **Iteration** | **Primary Objective (risks/use cases addressed)** | **Scheduled Start/Stop** | **Effort Estimate (person days)** |
| Inception | I1 | Identify problem and determine the scope of the problem. | September 24th – October 6th | 10 days |
| Elaboration | E1 | Elaborate on requirements and develop schedule | 10/9/23 - 11/3/23 | 19 |
| Construction | C1 | Develop an MVP | 11/6/23 - 12/4/23 | 21 |
|  | C2 | Refine product to a fully featured and usable state | 1/1/24 - 3/22/24 | 60 |
| Transition | T1 | Develop user guides, installation manuals, and assurance documents. | 3/25/24 - 4/10/24 | 13 |

### Project Resourcing

#### Staffing Plan

For this project we require the following roles:

* Machine Learning Specialist
* Front-end Dash Developer
* Statistician
* Network Analyst
* Project Manager

### Budget

School project, N/A

## Project Monitoring and control

### Requirements management plan

The project will be directed by bi-weekly meetings with the client and weekly meetings with the project advisor. These meetings will ensure the project continues along a path that satisfies the requirements of the client, and that the project can be course corrected with minimal lost time or effort.

### Schedule control plan

Our team will monitor progress by triweekly scrum meetings to identify the achievement of goals or setbacks. This will allow our team to quickly change priorities to aid team members who are not able to get around setbacks without help. Our plan for realignment with our schedule is to take work from team members who are struggling. Should all team members be struggling to achieve weekly goals we will re-evaluate our timeline.

### Quality control plan

Our team will test how well the model is able to detect known data based on F1 score from the testing dataset. We will also test how well the model will be able to detect unknown data by using holdout data that the model has not seen an instance of and finding the F1 score of the binary classification problem between known and unknown data.

### Reporting Plan

Team and Individual Sprint Reports are due bi-weekly. Documentation will be generated throughout the course of MVP development and will be updated throughout the course of final project development and refinement.

### Measurement Plan

We plan to measure the performance of our model using F1 scores compared to the training data.

## Risk Management plan

## Close-out plan

Upon the orderly completion of the project, all members will work to polish the final product. Documentation will be rounded out for the end users. Project materials will be kept on GitHub, potentially transferred to a new maintainer. The final product and its documentation will be transferred to the client in a usable format. A review of the final product and a retrospective report will take place between the client and team.

# Technical process plans

## Methods, tools and techniques

Programming Guidelines:

• Modular Code

• Well Tested Code

• Well Documented Code

• Thoroughly linted Code

User Interface Guidelines:

• Verbose

• Real-Time

Test Guidelines:

• Thorough Tests

• Overarching Tests

• Use-Case Tests

Design Guidelines:

• Fast

• Modular

Manual Guidelines:

• Thorough

• Verbose

# Supporting process plans

## Configuration management plan

Configuration changes will be managed through GitHub issues tracker. Those will then be linked to the GitHub Project for the management of the project. The GitHub Project will be managed as a Kanban Board with items related to each section of the development plan.

The code history will also be kept using GitHub. This allows collaboration between different project branches while keeping those branches separate. Branches will be development of specific features that will be tied into the main code base when their development is complete.

## Evaluation plan

To complete the project, we must evaluate our work in a clean and scientific manner. We will run some of the model training data through and identify the percentage of the training data that was correctly identified. We will also send random packets through the model in the expectation that an “unknowns detected” alert will be sent to the user interface; we will then attempt to reidentify the random packets and run them back through the model with the expectation that they will have the new class identification. The final evaluation will be the percent of random packets that the model catches as unknowns and the number it reclassifies as the new class.

## Documentation plan

The team will document the code after a working version of the code item has been completed. This is so that code documentation will not bottleneck work as one person waits for another to finish working to start. This will create complications while working as a team but due to the small nature of the team it is unlikely to cause long term problems. Once a code item has been completed it will be documented with pydocs describing the input types, output types, and a description of the item’s uses.

## Quality assurance plan

Our team will test how well the model is able to detect known data based on F1 score from the testing dataset. We will also test how well the model will be able to detect unknown data by using holdout data that the model has not seen an instance of and finding the F1 score of the binary classification problem between known and unknown data.

# Annexes

IOWR-NIDS: Vision Document, Software Requirements Specification