Project Management Plan



In conjunction with



# ABSTRACT

* This project plan for Network Discovery and Java Deep Packet Payload Collection will encompass the entire Software Development Life Cycle. During the requirements phase we will meet with our stakeholders: Matilda Cloud Solutions and The Fellows Consulting Group. The scope of the project includes deployment of Matilda Cloud, the use of automated on-premise infrastructure discovery to log packet header data and payload, and analyzation of the payload in order to find vulnerabilities in the system. Additionally, there will be a simple GUI that will allow one to execute a security scan of the network processes and display the results.

**TABLE OF CONTENTS**

[**ABSTRACT**](#_px71mrbibg6g) **0**

[**Figure 1: Identify Potential Automation, Simplification in a Process, and Security Vulnerabilities**](#_vmgujavajqjc) **2**

[**Figure 2: Add Network Packet Deep Payload Collection to Matilda Cloud Solutions Discovery**](#_n6jai11v7xx7) **2**

[**INTRODUCTION**](#_vwccrwwaehf8) **3**

[**PROJECT ORGANIZATION**](#_nsyel244kh8t) **3**

[**LIFE CYCLE MODEL USED**](#_i12tvkg32g8z) **3**

[**RISK ANALYSIS**](#_rwpb1xfvsvfv) **4**

[**HARDWARE AND SOFTWARE RESOURCE REQUIREMENTS**](#_9avflueb4awe) **5**

[**DELIVERABLES, SCHEDULE**](#_l4e09qr3znil) **5**

[**MONITORING, REPORTING, AND CONTROLLING MECHANISMS**](#_q44nojf24bze) **7**

[**PROFESSIONAL STANDARDS**](#_cv05f1kt7e9h) **7**

[**EVIDENCE THE DOCUMENT HAS BEEN PLACED UNDER CONFIGURATION MANAGEMENT**](#_x9silmws8p81) **7**

[**REFERENCES**](#_uupgiac5cq24) **7**

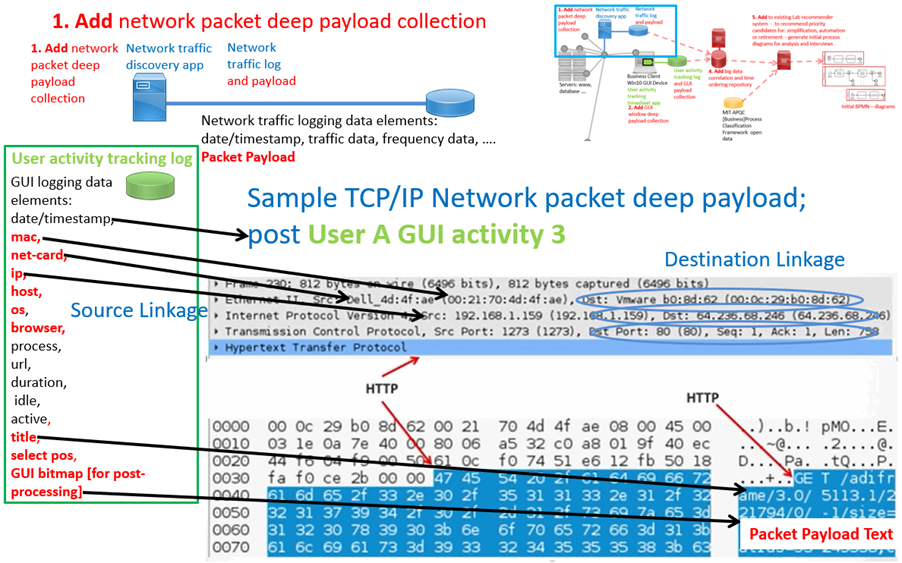
[**Appendix**](#_i0k1zdkjnj4y) **8**

[Appendix A.](#_lp1251gl6f09) 8

# **Figure 1: Identify Potential Automation, Simplification in a Process, and Security Vulnerabilities**

# 

# **Figure 2: Add Network Packet Deep Payload Collection to Matilda Cloud Solutions Discovery**



# INTRODUCTION

* This project shall result in the development of a standalone application that will add a network packet deep payload collection functionality. While network discovery applications are available, this deeper discovery of vulnerabilities will allow for enhanced monitoring and understanding of underlying network processes.
* This project plan will henceforth outline important points that will be relevant to the project’s development, and will include information on organization, life cycle models, risk analysis, hardware and software, deliverables, scheduling, and reporting.

# PROJECT ORGANIZATION

* The development team consists of three individuals that will work concurrently on each feature release, and this development team will be mentored by Dr. Hill and the sponsors of the project from Matilda Cloud Solutions.
* Brad Mickow will perform the role of project manager and will facilitate agile method duties, such as weekly stand-ups and task management via a kanban board.
* Alex will perform the role of requirements engineer, and will generate and maintain the quality of requirements and documentation related to the project.
* Kevin Dang will perform the role of dev lead, and will facilitate the general direction of development a well as collaborate with members to assign tasks efficiently.

# LIFE CYCLE MODEL USED

* The lifecycle model we are going to use is Agile lifecycle. In Agile, the tasks are divided into time boxes (small time frames) to deliver specific features for a release. The overall goal of each Agile method is to adapt to change and deliver working software as quickly as possible.

# RISK ANALYSIS

|  |  |  |
| --- | --- | --- |
| **Risk** | **Likelihood of Event** | **Mitigation Strategy** |
| Lack of knowledge of the specific area | **Likely** | Researching, reading, watching videos related to the topic |
| Delay in Schedule | **Certainty** | Created comprehensive project timeline with frequent baseline reviews |
| Failure to follow project plan | **Certainty** | Reduce continuously update project plan |
| Team does not agree on proposed concept | **Likely** | Compromise |
| Group member gets sick/injured | **UnLikely** | Have group members share knowledge |
| Unable to contact customer or guide | **Unlikely** | Multiple means of contacting customers |

# 

# HARDWARE AND SOFTWARE RESOURCE REQUIREMENTS

* describe the hardware and software required to carry out the development. Both hardware and software must be available in the lab include the rationale.

HARDWARE

This will be supplied through the UTD Lab. The full network includes the following:

* Computer Network (Computers, Servers, Routers, Switches)
* Network Connectivity

SOFTWARE

* Matilda Cloud
* Intellij and/or Eclipse as the choices of IDE
* Java for general language development
* Pcap4j, a Java-based packet capturing library
* Java Swing for applications GUI development
* React/Angular and HTML, CSS, Javascript for web development

# DELIVERABLES, SCHEDULE

* **Project Management Plan:**

Uses to document planning assumptions and decisions, facilitate communication among team members and also it summarized or detailed. Since planning is the initial step, there is no dependency. In addition, all team members will work on completing the project management plan document.

Due date: Friday, 09/06

* **Requirements Documentation:**

A document containing all the requirements of the project and It is written to allow people to understand what the project should do. In order to start working on requirements documentation, we need to have the project plan. All team members going to work on identifying all needed requirements.

Due date: Friday, 09/20

* **Architecture Documentation:**

software architecture document is a map of the software. We use it to see, at a glance, how the software is structured. It helps to understand the software’s modules and components without digging into the code. It’s a tool to communicate with others, developers and non-developers about the software. The architecture documentation dependence on requirement documentation. All team members will work on architecture documentation.

Due date: Friday, 10/04

* **Detailed Design Documentation**

The Software Design Document is a document to provide documentation which will be used to aid in software development by providing the details for how the software should be built. Within the Software Design Document are narrative and graphical documentation of the software design for the project including Graphical User Interface design, class diagram, sequence diagrams and other supporting requirement information. Design documentation dependence on requirements documentation. All team members will work on completing the design documentation.

Due date: Friday, 10/25

* **Testing Plan:**

A test plan is a detailed document that outlines the test strategy, [Testing](https://www.guru99.com/software-testing.html) objectives, resources (manpower, software, hardware) required for testing and test schedule. Testing plan dependence on requirements documentation, architecture documentation and design documentation of the project. All team members will participate on completing the testing plan document.

Due date: Friday, 11/15

* **Final Project Report:**

The Final Project Report is intended to concisely summarize the outcomes of a project and is the final document of the project. A Final Project Report is used to document project successes, lessons learned and performance in order to signal improvement in project delivery for the future. Final project report dependendence on all the documentations (requirements, architecture, design, and testing). All team members will work equally on completing the report.

Due date: Friday, 12/06

* **Final Project Demonstration:**

The Final Project Demonstration helps to showcase the possible applications, feasibility, performance and method of the project. It can be used as a demonstration to the sponsors, UTD staff members, and/or students. The Final Project Demonstration dependence on all the documentations and the actual code. All team members are going to present.

Due date: Thursday (Tentative)

# MONITORING, REPORTING, AND CONTROLLING MECHANISMS

* Management reports shall consist of Gantt charts detailing the scheduling of tasks and the work done to transform them into deliverables, updated at every team meeting. By tracking the start, end, and work done for tasks, trends can be analyzed and work can be optimized in order to make the team more efficient.
* Project work will be controlled by using the agile method, in which weekly stand ups will be performed to view work trajectories and reorganize, if need be. A kanban board will also be used to assign and organize current tasks. In doing so, member responsibilities will be more explicitly defined and tracked, allowing for better workflows.

# PROFESSIONAL STANDARDS

* All team members will display academic honesty with all work done towards the completion of the project. As such, heavily referenced code will be documented and cited as necessary as to prevent plagiarism, and written code will be documented, along with a header signifying which members developed it.
* The members will meet every Wednesday at 4:30PM in the ECSS Lab 3.415 to develop a good history of communication and tasks.
* All tasks and deliverables are to be developed and presented with professional standards of organization and structure. Tasks will be properly documented and versioned with a repository of choice. Deliverables will be properly formatted with good grammar and be submitted before the deadline. Doing so will maintain a high quality within work done for tasks and deliverables.

# EVIDENCE THE DOCUMENT HAS BEEN PLACED UNDER CONFIGURATION MANAGEMENT

# REFERENCES

* Complete, correctly formatted using IEEE standard in order to remain transparent and achieve academic honesty.

# 

# Appendix

## Appendix A.

The following provides a professional standards guideline for the teams. This guideline may be

tailored. The professional standards must be agreed upon by each member in the team.

Guideline:

On the first occurrence of unacceptable behavior, determine the circumstances involved, resolve

the problem, and document the event in the meeting minutes.

On a second occurrence, notify the instructor of the problem. A meeting will be set up to evaluate the situation and resolve the problem.

On a third occurrence, again notify the instructor of the problem. A meeting will be set up to

evaluate the situation and resolve the problem. At this point, the team will have the \*option\* of

removing the team member. If removed, then the team member receives a pro-rated grade based

on the number of weeks they have participated in the group.

Examples of unacceptable behavior may include not delivering on time, delivering poor quality

work, missing team meetings, being unprepared for team meetings, disrespectful or rude behavior, etc. Reasons such as "too busy" or "I forgot", or "my dog ate my design model" are unacceptable.

Valid reasons that must be considered include those listed for obtaining an incomplete standing in a course (illness, death in the family, travel for business or academic reasons, etc.)