**Requirements Documentation**



In conjunction with



For Team 15

# 

# 

# 

# ABSTRACT

This document will describe the functional and non-functional requirements for the Deep Network Packet Discovery Project, using a use case diagram showcasing the linkages between users and use cases to explain the functional requirements.

**TABLE OF CONTENTS**

[**ABSTRACT**](#_ckcdhxrgemm1) **0**

[**LIST OF FIGURES**](#_v7uz64fw54nm) **2**

[**LIST OF TABLES**](#_yji1v6l5eug6) **2**

[**INTRODUCTION**](#_xu5cuxjw8skc) **2**

[**USE CASE MODEL FOR FUNCTIONAL REQUIREMENTS**](#_vxl3esgpyu6q) **3**

[USE CASES](#_lt5bdjmukpas) 3

[**RATIONALE FOR YOUR USE CASE MODEL**](#_e8ijhyj9unqp) **5**

[**NON-FUNCTIONAL REQUIREMENTS**](#_12tn53ij8f0p) **5**

[**EVIDENCE THE REQUIREMENTS HAVE BEEN PLACED UNDER CONFIGURATION MANAGEMENT**](#_57mwq3iiqby0) **6**

[**REFERENCES**](#_4yn1n3o26zz5) **6**

# LIST OF FIGURES

* Use Case Diagram Page 3

# 

# LIST OF TABLES

* Use Case Table Page 4

# 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

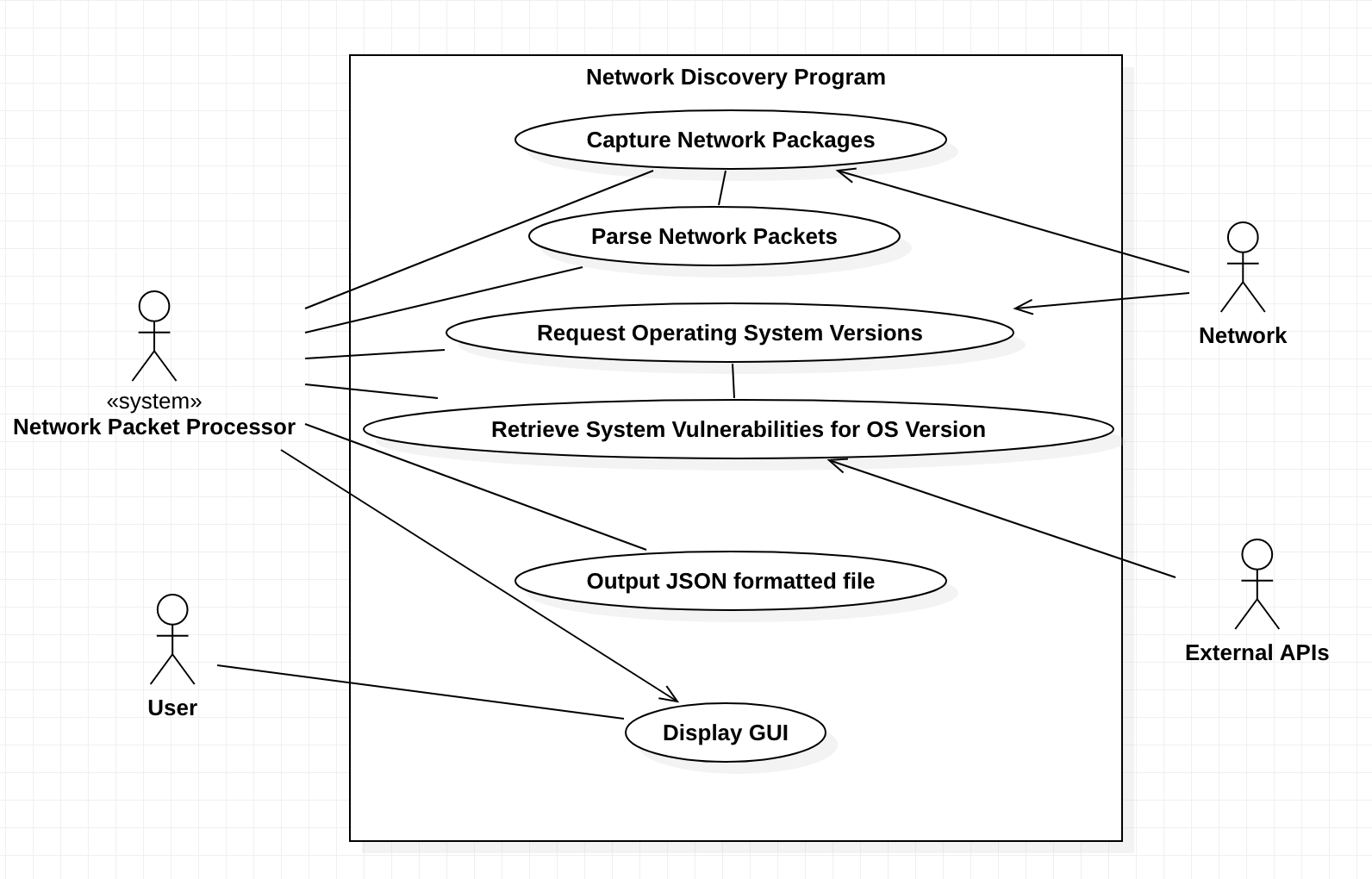
project’s development, and will include information on organization, life cycle models,

risk analysis, hardware and software, deliverables, scheduling, and reporting.

* This document is structured to present functional requirements with a use case diagram and a textual description, and non-functional requirements with a simple description.

# 

# USE CASE MODEL FOR FUNCTIONAL REQUIREMENTS



## 

## USE CASES

|  |
| --- |
| **Capture Network Packets**   * **Functional Requirement**: The program shall capture packets directly from the network it is connected to * **Actors**: Network Discovery Processor, Network * **Entry Condition**: Upon valid connection to a network * **Normal Flow of Events**: Program connects to network, then captures packets coming in through the network * **Exit Condition:** Program stops receiving unique packets, or network connection is disconnected. * **Exceptions**: N/A * **Special Requirements**: Requires a network connection. |
| **Parse Network Packet Payloads**   * **Functional Requirement**: The program shall parse the packet payloads for relevant information, such as computer metrics, host information, port information, network metrics, storage details, installed packages, users, and installed services. * **Actors**: Network Discovery Processor * **Entry Condition**: Upon capturing a packet from the network * **Normal Flow of Events**: Program captures a packet from the network, then extracts the payload, and finally parses it for information. * **Exit Condition:** Program stops receiving unique packets, or network connection is disconnected * **Exceptions**: N/A * **Special Requirements**: Requires a network connection. |
| **Request Operating System Versions**   * **Functional Requirement**: The program shall request operating system versions from network devices * **Actors**: Network Discovery Processor, Network Devices * **Entry Condition**: Upon valid connection to a network * **Normal Flow of Events**:Program connects to a network, gathers a list of network devices, then requests the operating system version from them. * **Exit Condition:** Program finishes requesting OS versions for all network devices, or network connection disconnects. * **Exceptions**: N/A * **Special Requirements**: Requires a network connection. |
| **Retrieve System Vulnerabilities for OS Version**   * **Functional Requirement**: The program shall compare operating system versions with external APIs in order to realize possible system vulnerabilities. * **Actors**: Network Discovery Processor, External APIs * **Entry Condition**: Upon retrieval of OS version from network device * **Normal Flow of Events**: Program retrieves OS version of a network device, then makes a call to an external API to gather security vulnerabilities related to the version. * **Exit Condition:** Program finishes requesting vulnerabilities for OS versions for all network devices, or network connection disconnects. * **Exceptions**: N/A * **Special Requirements**: Requires a network connection. |
| **Output JSON Formatted File**   * **Functional Requirement:** The program shall output a JSON formatted file of relevant information * **Actors**: Network Discovery Processor * **Entry Condition**: Upon completed discovery of network * **Normal Flow of Events**: Program finishes gathering unique data from the network, and then outputs relevant data gathered in a JSON file * **Exit Condition:** Program stops receiving unique packets, or network connection is disconnected. * **Exceptions**: N/A * **Special Requirements**: Requires a network connection. |
| **Display GUI**   * **Functional Requirement:** The program shall display a GUI that users can view summarized network information from * **Actors**: User, Network Discovery Processor * **Entry Condition**: Upon completed discovery of network * **Normal Flow of Events**: The program starts up, and displays a GUI which propagates with information in real-time to the information gathering process. * **Exit Condition:** The program is terminated. * **Exceptions**: N/A * **Special Requirements**: Requires a network connection. |

# RATIONALE FOR YOUR USE CASE MODEL

* We chose these use cases because they represent the basic functional requirements of the Network Discovery project.
* By linking these use cases to their actors, we can build a graphical view of the system in relation to how functional requirements and systems/users interact.

# NON-FUNCTIONAL REQUIREMENTS

* The GUI interface shall have 100% availability and reliability.
* The JSON formatted output shall process in less than 2 seconds for 99% of users considering their location, bandwidth and latency.
* The JSON formatted output shall secure its data according to industry practices
* The GUI interface shall be intuitive and easy to understand within half a minute of viewing it

# EVIDENCE THE REQUIREMENTS HAVE BEEN PLACED UNDER CONFIGURATION MANAGEMENT

* Link to the github hosting the document for the team to view: <https://github.com/Narthexes/matilda-senior-design-project/>

# REFERENCES

* N/A