Document Control

Approval

The Guidance Team and the customer shall approve this document.

Document Change Control

<table>
<thead>
<tr>
<th>Initial Release:</th>
<th>0.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Current Release:</td>
<td>1.0</td>
</tr>
<tr>
<td>Indicator of Last Page in Document:</td>
<td>&amp;</td>
</tr>
<tr>
<td>Date of Last Review:</td>
<td>1/30/2018</td>
</tr>
<tr>
<td>Date of Next Review:</td>
<td>2/28/2018</td>
</tr>
<tr>
<td>Target Date for Next Update:</td>
<td>2/28/2018</td>
</tr>
</tbody>
</table>

Distribution List

This following list of people shall receive a copy of this document every time a new version of this document becomes available:

Guidance Team Members:
- Dr. Ann Gates
- Dr. Salamah Salamah
- Eric Camacho
- Elsa Tai

Customers:
- Dr. Jaime Acosta
- Mr. Baltazar Santaella

Software Team Members:
- Team 1
- Team 2
- Team 3
- Team 4
- Team 5
- Team 6
- Team 7

Change Summary

The following table details changes made between versions of this document:

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Modifier</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>1/15/2018</td>
<td>Elsa Tai</td>
<td>Created a draft document</td>
</tr>
<tr>
<td>1.0</td>
<td>1/30/2018</td>
<td>Elsa Tai</td>
<td>Updated the interface and all the models.</td>
</tr>
</tbody>
</table>
## Table of Contents

**DOCUMENT CONTROL** ................................................................................................. II

- APPROVAL ..................................................................................................................... II
- DOCUMENT CHANGE CONTROL ................................................................................... II
- DISTRIBUTION LIST ........................................................................................................ II
- CHANGE SUMMARY ....................................................................................................... II

### 1. INTRODUCTION ........................................................................................................ 4

1.1. PURPOSE AND INTENDED AUDIENCE ..................................................................... 4
1.2. SCOPE OF PRODUCT ................................................................................................... 4
1.3. DEFINITIONS, ACRONYMS, AND ABBREVIATIONS .................................................. 4
   1.3.1. Definitions ............................................................................................................. 4
   1.3.2. Acronyms ............................................................................................................. 4
   1.3.3. Abbreviations ....................................................................................................... 5
1.4. OVERVIEW .................................................................................................................. 5
1.5. REFERENCES ............................................................................................................... 5

### 2. GENERAL DESCRIPTION .......................................................................................... 6

2.1. PRODUCT PERSPECTIVE .......................................................................................... 6
2.2. PRODUCT FEATURES ................................................................................................ 6
   2.2.1. Actors Descriptions ............................................................................................ 7
   2.2.2. Use Case Descriptions ....................................................................................... 7
2.3. USER CHARACTERISTICS ......................................................................................... 7
2.4. GENERAL CONSTRAINTS .......................................................................................... 7
2.5. ASSUMPTIONS AND DEPENDENCIES ..................................................................... 8

### 3. SPECIFIC REQUIREMENTS ......................................................................................... 9

3.1. EXTERNAL INTERFACE REQUIREMENTS .................................................................. 9
   3.1.1. User Interfaces .................................................................................................... 9
   3.1.2. Hardware Interfaces ........................................................................................... 17
   3.1.3. Software Interfaces ............................................................................................ 17
   3.1.4. Communications Interfaces ............................................................................... 18
3.2. BEHAVIORAL REQUIREMENTS ............................................................................... 18
   3.2.1. Same Class of User ............................................................................................ 18
   3.2.2. Related Real-world Objects ............................................................................... 18
   3.2.3. Functions .......................................................................................................... 23
   3.2.4. Stimulus ............................................................................................................ 25
3.3. NON-BEHAVIORAL REQUIREMENTS ....................................................................... 26
1. Introduction

1.1. Purpose and Intended Audience
The purpose of the Software Requirements Specification (SRS) is to give the customer a clear and precise description of the functionality of the Testbed Management System. The SRS divides the system requirements into two parts, behavioral and non-behavioral requirements. The behavioral requirements describe the interaction between the system and its environment. Non-behavioral requirements relate to the definition of the attributes of the product as it performs its functions. This includes the level of security and performance of the product. The intended audience of the SRS is Dr. Jaime Acosta and Mr. Baltazar Santaella, and the Software Engineering teams. This document serves as an agreement between both parties regarding the product to be developed.

1.2. Scope of Product
Army Research Lab (ARL) Center for Cyber Analysis and Assessment recognizes the lack of dissectors for custom or non-IP protocols. Developing custom dissectors are difficult and time consuming especially when the associated protocol specifications are not available. Analysts would have to infer the protocol specification through extensive trial and error. ARL wants a system that would allow analysts to create custom dissectors through a graphical workspace.

The University of Texas at El Paso (UTEP) and ARL are collaborating to develop a Protocol Dissector Generator System (PDGS) that will provide the following services: Creation of protocol structure graphically and Generation of a LUA dissector script.

1.3. Definitions, Acronyms, and Abbreviations

1.3.1. Definitions
The definitions in this section are given in the context of the product being developed. This intention is to assist the user in their understanding of the document.

<table>
<thead>
<tr>
<th>TERM</th>
<th>DEFINITION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actor</td>
<td>A representation in the use case diagram denoting external entities that interact with a system being modeled, e.g., the testbed management system.</td>
</tr>
<tr>
<td>Extend Relationship</td>
<td>Denotes insertion of optional behavior of another use case into the primary use case.</td>
</tr>
<tr>
<td>Generalization</td>
<td>Denotes a relationship between a general use case and a specific use case.</td>
</tr>
<tr>
<td>Relationship</td>
<td>Include Relationship</td>
</tr>
<tr>
<td>Dissector</td>
<td>A parser that transforms information encoded in a packet to human readable form.</td>
</tr>
<tr>
<td>Use Case</td>
<td>A modeling technique that presents the basic functionality of a system and the actors that interact with each function.</td>
</tr>
</tbody>
</table>

1.3.2. Acronyms
This section lists the acronyms used in this document and their associated definitions.

<table>
<thead>
<tr>
<th>TERM</th>
<th>DEFINITION</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARL</td>
<td>Army Research Lab</td>
</tr>
</tbody>
</table>
1.3.3. Abbreviations
This section provides a list of used abbreviations and their associated definitions.

<table>
<thead>
<tr>
<th>TERM</th>
<th>DEFINITION</th>
</tr>
</thead>
<tbody>
<tr>
<td>e.g.</td>
<td>For example</td>
</tr>
<tr>
<td>i.e.</td>
<td>That is</td>
</tr>
<tr>
<td>TBD</td>
<td>To be determined</td>
</tr>
</tbody>
</table>

1.4. Overview
The SRS is divided into three major sections: Introduction (Section 1), General Description (Section 2), and Specific Requirements (Section 3).

Section 1 includes five subsections. Section 1.1 provides the purpose and intended audience of the document. Section 1.2 describes the scope of the product. Section 1.3 provides the definitions, acronyms and abbreviations. Section 1.4 provides the organization of the document. Section 1.5 lists the references used in this document.

Section 2 includes five subsections. Section 2.1 contains a description of the product, its overall structure, and its functionality. Section 2.2 summarizes the main features of the Protocol Dissector Generator system. Section 2.3 identifies each type of users of the system. This is accomplished through a summary of actors and use-cases. Section 2.4 states existing general constraints. Section 2.5 gives the assumptions and dependencies of Testbed Management System.

Section 3 includes four major subsections. Section 3.1 contains requirements that are related to the external interface. Section 3.2 contains the functional requirements that are organized in the following categories: same class of user, related real-world objects, stimulus, related features, and limits and default settings. Section 3.3 contains non-behavioral requirements.

1.5. References
2. **General Description**

2.1. **Product Perspective**
Protocol Dissector Generator System (PDGS) is an interactive system that facilitates the construction of the structure and specification of a protocol and automates the creation of a dissector script.

2.2. **Product Features**
Figure 1 presents a level 1 use case diagram that provides an overview of the main functionalities provided by PDGS and the interactions between actors and PDGS. Figure 3 presents the notations used in a use case diagram. The actors, represented by stick figures, are external entities that interact with PDGS. The use case, represented by ovals, elucidate the actors’ interactions with PDGS. Figure 2 presents a level 2 use case diagram that provides extensions of the functionalities, in particular the include, extend, and generalization interactions between the actors and Testbed Management System. The *include* relationship denotes the inclusion of behavior of another use case into the primary use case. The *extend* relationship denotes insertion of optional behavior of another use case into the primary use case. The *generalization* relationship denotes a relationship between a general use case and a specific use case. These components are described next.

![Protocol-Dissector Generator System Diagram](image)

*Figure 1: Level 1 Use Case Diagram*
2.2.1. **Actors Descriptions**

- The Analyst class represents those who have an interest in understanding packet information in a human readable form.
- The T-Shark class represents a system that decodes packet information.

2.2.2. **Use Case Descriptions**

PDGS supports the following primary use cases:

- Create dissector
- Manage protocol specification
- View dissected packets

2.3. **User Characteristics**

The users of the system have a variety of computer usage skills and are immersed in the area of cybersecurity.

2.4. **General Constraints**

The general constraints on the development of Testbed Management System are as follows:

- The system will be completed by the end of May 2018.
• The dissector script will be developed in LUA.

2.5. Assumptions and Dependencies
The assumptions and dependencies of PDGS are as follows:
• The system will run on Windows and Linux operating systems.
3. Specific Requirements

3.1. External Interface Requirements

This section contains the specification of requirements for interfaces among different components and their external capabilities.

3.1.1. User Interfaces

This section describes the characteristics of each interface of PDGS. The main interfaces listed below will be described in the following sections:

- General
- Overlays
- Views.

3.1.1.1. General

[SRS 1] The system shall have the following layout components as shown in Figure 4:

- Heading
- Menu
- Project Navigator View
- Dissector Builder Area View
- Packet Stream Area View
- Dissected Stream Area View
- Raw Data Area View
- Console View.

![Protocol Dissector Generator System](image)

Figure 4: Layout
[SRS 2] The Heading section shall display the name of the system as “Protocol Dissector Generator System” as shown in Figure 4.

[SRS 3] The Menu section shall include the components as shown in Figure 5.

[SRS 4] The system shall display the components as shown in Figure 6:
- Minimize Button,
- Maximize Button, and
- Close Button.

3.1.1.2. Overlays

[SRS 5] The PDGS Launcher Overlay shall include the components as shown in Figure 7.

[SRS 6] The New Project Overlay shall include the components as shown in Figure 8.
[SRS 7] The Dissector Script Overlay shall include the components as shown in Figure 9.

![Dissector Script Diagram]

Figure 9: Dissector Script

[SRS 8] The Project Import Overlay shall include the components as shown in Figure 10.

![Project Import Diagram]

Figure 10: Project Import

[SRS 9] The Project Export Overlay shall include the components as shown in Figure 11.

![Project Export Diagram]

Figure 11: Project Export
[SRS 10] The Organize Views Overlay shall include the components as shown in Figure 12.

![Organize Views](image)

Figure 12: Organize Views

[SRS 11] The Open PCAP shall include the components as shown in Figure 13.

![Open PCAP](image)

Figure 13: Open PCAP
3.1.1.3. Views

3.1.1.3.1. Builder Dissector Area View

[SRS 12] The Builder Dissector Area View shall include the following sections as shown in Figure 14
- Drawing Canvas (left) and
- Palette View (right).

![Dissector Builder Area Diagram]

Figure 14: Dissector Builder Area View

[SRS 13] The Palette View shall include the following sections as shown in Figure 15
- Field section and
- Construct section.

![Palette View Diagram]

Figure 15: Palette View
[SRS 14] The Field section shall include the components as shown in Figure 16.

![Figure 16: Field Section](image)

[SRS 15] The Construct section shall include the components as shown in Figure 17.

![Figure 17: Construct Section](image)
[SRS 16] A Field shall include the component as shown in Figure 18.

![Field][Abbreviation]

<table>
<thead>
<tr>
<th>Field [Abbreviation]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
</tr>
<tr>
<td>Abbreviation</td>
</tr>
<tr>
<td>Description</td>
</tr>
<tr>
<td>Reference List</td>
</tr>
<tr>
<td>Data Type</td>
</tr>
<tr>
<td>Base</td>
</tr>
<tr>
<td>Mask</td>
</tr>
<tr>
<td>Value Constraint</td>
</tr>
<tr>
<td>Required</td>
</tr>
</tbody>
</table>

Figure 18: Field

[SRS 17] A Start Field shall include the component as shown in Figure 19.

![Start Field [Protocol Name]]

<table>
<thead>
<tr>
<th>Start Field [Protocol Name]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Protocol Name</td>
</tr>
<tr>
<td>Protocol Description</td>
</tr>
<tr>
<td>Dependent Protocol Name</td>
</tr>
<tr>
<td>Dependency Pattern</td>
</tr>
</tbody>
</table>

Figure 19: Start Field

[SRS 18] An End Field shall not include any components as shown in Figure 20.

![End Field]

Figure 20: End Field

[SRS 19] A Reference List Field shall include the component as shown in Figure 21.

![Reference List [Reference List Name]]

<table>
<thead>
<tr>
<th>Reference List [Reference List Name]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reference List Name</td>
</tr>
<tr>
<td>Value</td>
</tr>
<tr>
<td>Text Description</td>
</tr>
</tbody>
</table>

Figure 21: Reference List Field
A Packet Information Field shall include the components as shown in Figure 22.

3.1.1.3.2. Project Navigator View

The Project Navigator View shall include the component as shown in Figure 23.

3.1.1.3.3. Packet Stream Area View

The Packet Stream Area shall display information about each packet as specified by the packet information field. An example is shown in Figure 24.
3.1.1.3.4. Dissected Stream Area View

[SRS 23] The Dissected Stream Area View shall display dissected information about each packet in a tree view. An example is shown in Figure 25.

![Dissected Stream Area View](image)

Figure 25: Dissected Stream Area View

3.1.1.3.5. Raw Data Area View

[SRS 24] The Raw Data Area View shall display the content of an opened PCAP file. An example is shown in Figure 26.

![Raw Data Area View](image)

Figure 26: Raw Data Area View

3.1.1.3.6. Console Area View

[SRS 25] The Console Area View shall display warning and error messages generated by the system. An example is shown in Figure 27.

![Console Area View](image)

Figure 27: Console Area View

3.1.2. Hardware Interfaces

There are no hardware interface requirements specified at this time.

3.1.3. Software Interfaces

This section describes the characteristics of each interface between other application systems and the system.
The system shall interface with T-Shark to decode packet information using the command as specified in the man-page of T-Shark (https://www.wireshark.org/docs/man-pages/tshark.html).

### 3.1.4. Communications Interfaces
There are no communication interface requirements specified at this time.

### 3.2. Behavioral Requirements
This section describes the behavioral requirements of the system.

#### 3.2.1. Same Class of User
This section describes requirements associated with a particular class of user. There are no same class user requirements specified at this time.

#### 3.2.2. Related Real-world Objects
This section describes related real-world object requirements of the system. Figure 28 shows the class diagram of PDGS.

![Class Diagram](Figure 28: Class Diagram)
3.2.2.1. Workspace

[SRS 27] The system shall store the fields presented in Table 4 for a workspace.

Table 4: Workspace Fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Data Type</th>
<th>Values and Constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>String</td>
<td>Required</td>
</tr>
<tr>
<td>Path</td>
<td>String</td>
<td>Required</td>
</tr>
</tbody>
</table>

3.2.2.2. Project

[SRS 28] The system shall store the fields presented in Table 5 for a project.

Table 5: Project Fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Data Type</th>
<th>Values and Constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>String</td>
<td>Required</td>
</tr>
<tr>
<td>Description</td>
<td>String</td>
<td>Required</td>
</tr>
</tbody>
</table>

[SRS 29] A project shall comprise of one layout preference and one protocol decision tree.

[SRS 30] A project shall be stored in XML as its intermediate format.

3.2.2.3. Layout Preference

[SRS 31] The system shall store the fields presented in Table 6 for layout preference.

[SRS 32] Table 6 for layout preference.

Table 6: Layout Preference Fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Data Type</th>
<th>Values and Constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td>View</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>X Coordinate</td>
<td>Integer</td>
<td></td>
</tr>
<tr>
<td>Y Coordinate</td>
<td>Integer</td>
<td></td>
</tr>
<tr>
<td>Width</td>
<td>Integer</td>
<td></td>
</tr>
<tr>
<td>Height</td>
<td>Integer</td>
<td></td>
</tr>
</tbody>
</table>

[SRS 33] The system shall support the following operations for the layout of each view:

- Attach view
- Detach view
- Close view
- Minimize view
- Maximize view
- Move view from one coordinate to another coordinate.

[SRS 34] The system shall support the following types of views:

- Project Navigation
- Dissector Building Area
- Palette
- Packet Stream Area
- Dissected Stream Area
- Raw Data Area
- Console Area.
3.2.2.4. Protocol Decision Tree

The system shall store the fields presented in Table 7 for each protocol decision tree.

Table 7: Protocol Decision Tree Fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Data Type</th>
<th>Values and Constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>String</td>
<td>Required</td>
</tr>
<tr>
<td>Description</td>
<td>String</td>
<td>Required</td>
</tr>
</tbody>
</table>

A protocol decision tree shall comprise of the following:
- At least one field,
- One start node construct, and
- At least one end node construct.

A protocol decision tree shall start with a start node construct and end with at least one end node construct.

A protocol decision tree shall describe its dependency to another protocol within the OSI layer.

The system shall store the fields presented in Table 8 for each protocol dependency.

Table 8: Protocol Dependency

<table>
<thead>
<tr>
<th>Field</th>
<th>Data Type</th>
<th>Values and Constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td>Protocol Name</td>
<td>String</td>
<td>Required</td>
</tr>
<tr>
<td>Pattern</td>
<td>Object</td>
<td>Required, A pattern can be described as an Integer/String/Range.</td>
</tr>
</tbody>
</table>

A protocol decision tree shall describe the required packet information that is displayed in the Packet Stream Area View.

The system shall store the fields presented in Table 9 for the required packet information.

Table 9: Packet Information

<table>
<thead>
<tr>
<th>Field</th>
<th>Data Type</th>
<th>Values and Constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>String</td>
<td>Required</td>
</tr>
<tr>
<td>Description</td>
<td>String</td>
<td>Required</td>
</tr>
</tbody>
</table>

3.2.2.5. Field

The system shall store the fields presented in Table 10 for each field.

Table 10: Fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Data Type</th>
<th>Values and Constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>String</td>
<td>Required</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>String</td>
<td>Required</td>
</tr>
<tr>
<td>Description</td>
<td>String</td>
<td>Required</td>
</tr>
<tr>
<td>Mask</td>
<td>String</td>
<td>Optional</td>
</tr>
<tr>
<td>Size</td>
<td>Integer</td>
<td>Required</td>
</tr>
<tr>
<td>Data Type</td>
<td>List</td>
<td>Required</td>
</tr>
</tbody>
</table>
[SRS 43] A field shall reference at most one reference list.

3.2.2.6. Reference List and List

[SRS 44] A reference list shall be associated with at least one list.

[SRS 45] The system shall store the fields presented in Table 11 for each list.

Table 11: List Fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Data Type</th>
<th>Values and Constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value</td>
<td>Integer</td>
<td></td>
</tr>
<tr>
<td>Description</td>
<td>String</td>
<td></td>
</tr>
</tbody>
</table>

3.2.2.7. Protocol Decision Tree Construct

[SRS 46] The system shall store the field presented in Table 12 for each protocol decision tree construct.
Table 12: Protocol Decision Tree Construct Field

<table>
<thead>
<tr>
<th>Field</th>
<th>Data Type</th>
<th>Values and Constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>String</td>
<td></td>
</tr>
</tbody>
</table>

[SRS 47] The system shall support the following types of protocol decision tree construct:
- Decision Construct
- Start Node Construct
- End Node Construct
- Expression Construct
- Connector Construct.

[SRS 48] A Decision Construct shall contain at least one Expression Construct.

[SRS 49] A Decision Construct shall be followed by at least two Connector Constructs labeled with values resulted from the Expression Construct contained in the Decision Construct.

[SRS 50] An Expression Construct shall comprise of at least one operand and at least one operator.

[SRS 51] The system shall support the following types of relational operators:
- Less than “<”
- Greater than “>”
- Less than or equal to “<="
- Greater than or equal to “>="
- Equal to “==”
- Not equal to “~=".

[SRS 52] The system shall support the following types of logical operators:
- And
- Or
- Not.

[SRS 53] The system shall connect the following combination via a connector:
- Field to field (with the exception of reference list field and packet information field)
- Field to Decision
- Decision to Field
- Decision to Decision.

3.2.2.8. Dissector Script

[SRS 54] The system shall store the fields presented in Table 13 for a dissector script.

Table 13: Dissector Script Fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Data Type</th>
<th>Values and Constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>String</td>
<td>Required</td>
</tr>
<tr>
<td>Format</td>
<td>String</td>
<td>Required</td>
</tr>
</tbody>
</table>

[SRS 55] The system shall support LUA as the only available dissector script format.

3.2.2.9. Dissected Packet

[SRS 56] The system shall color code packets according to the rules presented in Table 14.
Table 14: Dissected Packet Rules

<table>
<thead>
<tr>
<th>Color</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Red</td>
<td>None of the fields in a dissected packet matches the protocol structure.</td>
</tr>
<tr>
<td>Yellow</td>
<td>At least one field in a dissected packet fails to match the protocol structure.</td>
</tr>
<tr>
<td>Green</td>
<td>All fields in a dissected packet matches the protocol structure.</td>
</tr>
</tbody>
</table>

3.2.3. Functions

This section describes the function requirements of the system via level 1 and level 2 data flow diagrams as shown in Figure 30, Figure 31, and Figure 32.
Figure 30: Level 2 of 1.0 Create Dissector Process

Figure 31: Level 2 of 2.0 Manage Protocol Specification
3.2.4. Stimulus

This section describes the stimulus requirements of the protocol decision tree via the state transition diagram as shown in Figure 33. (TBA - other stimulus requirements will be provided at a later time)

3.2.4.1. Interface Stimulus

[SRS 57] If any of the required fields are not filled, the system shall display an error message.

[SRS 58] When the user hits the “Delete” key on the keyboard, the system shall delete the selected field from the Drawing Canvas.

[SRS 59] When the user hits the “Delete” key on the keyboard, the system shall delete the selected connector from the Drawing Canvas.

[SRS 60] When the user hits the “Delete” key on the keyboard, the system shall delete the selected construct from the Drawing Canvas.

[SRS 61] When the user drags a field onto the Drawing Canvas, the system shall expand and show the detailed information required for the field.

[SRS 62] If there exists more than one Start Field in the Drawing Canvas, the system shall generate an error message.

3.2.4.2. Dissector Script Generation

[SRS 63] When generating a dissector script, if there exists no Start Field in the Protocol Decision Tree, the system shall generate an error message.

[SRS 64] When generating a dissector script, if the Protocol Decision Tree doesn’t have at least one End Field, the system shall generate an error message.

[SRS 65] When generating a dissector script, if the Protocol Decision Tree has dangling connections, the system shall generate an error message.
3.3. Non-behavioral Requirements

This section describes performance and security requirements of the system. There are no non-behavior requirements specified at this time.

&