Protocol Formatter System

Version <1.0>

Document Control

Approval

The Guidance Team and the customer shall approve this document.

Document Change Control

|  |  |
| --- | --- |
| Initial Release: | 0.1 |
| Current Release: | 1.0 |
| Indicator of Last Page in Document: | & |
| Date of Last Review: | TBA |
| Date of Next Review: | TBA |
| Target Date for Next Update: | TBA |

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

Elsa Tai

Customer: Dr. Jaime Acosta

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

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Modifier | Description |
| 1.0 | 2/2/2016 | Elsa Tai | Created a draft document |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

[Document Control ii](#_Toc473808083)

[Approval ii](#_Toc473808084)

[Document Change Control ii](#_Toc473808085)

[Distribution List ii](#_Toc473808086)

[Change Summary ii](#_Toc473808087)

[1. Introduction 4](#_Toc473808088)

[1.1. Purpose and Intended Audience 4](#_Toc473808089)

[1.2. Scope of Product 4](#_Toc473808090)

[1.3. Definitions, Acronyms, and Abbreviations 4](#_Toc473808091)

[1.3.1. Definitions 4](#_Toc473808092)

[1.3.2. Acronyms 4](#_Toc473808093)

[1.3.3. Abbreviations 5](#_Toc473808094)

[1.4. Overview 5](#_Toc473808095)

[1.5. References 5](#_Toc473808096)

[2. General Description 6](#_Toc473808097)

[2.1. Product Perspective 6](#_Toc473808098)

[2.2. Product Features 6](#_Toc473808099)

[2.2.1. Actors Descriptions 6](#_Toc473808100)

[2.2.2. Use Case Descriptions 6](#_Toc473808101)

[2.3. User Characteristics 6](#_Toc473808102)

[2.4. General Constraints 7](#_Toc473808103)

[2.5. Assumptions and Dependencies 7](#_Toc473808104)

[3. Specific Requirements 8](#_Toc473808105)

[3.1. External Interface Requirements 8](#_Toc473808106)

[3.1.1. User Interfaces 8](#_Toc473808107)

[3.1.2. Hardware Interfaces 13](#_Toc473808108)

[3.1.3. Software Interfaces 13](#_Toc473808109)

[3.1.4. Communications Interfaces 13](#_Toc473808110)

[3.2. Behavioral Requirements 13](#_Toc473808111)

[3.2.1. Same Class of User 14](#_Toc473808112)

[3.2.2. Related Real-world Objects 14](#_Toc473808113)

[3.2.3. Stimulus 16](#_Toc473808114)

[3.3. Non-behavioral Requirements 17](#_Toc473808115)

# Introduction

## 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 Protocol Formatter 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, efficiency, reliability, maintainability, and portability of the product. The intended audience of the SRS is Dr. Jaime Acosta and Baltazar Santaella from the Army Research Lab (ARL) and the Software Engineering teams. This document serves as an agreement between both parties regarding the product to be developed.

## Scope of Product

Network systems communicate and pass information among nodes using the layered architecture. Data at each layer must follow a protocol specification so that nodes involved will be able to send, receive, and interpret information. Network security analysts conduct assessments of network systems by attempting to identify vulnerabilities in protocols and their implementations. Additionally, analysts may need to uncover the communication methods and data among malware. At the present time, inspection, analysis, and modification of network data is done manually and one file at a time. There is no mechanism or system to capture and facilitate reuse of analyzed data.

The University of Texas at El Paso (UTEP) and ARL are collaborating to develop the Protocol Formatter system that will provide the following services:

* Display packet data in a manner that would facilitate various types of analysis;
* Capture analysis result; and
* Facilitate reuse of analyzed data.

## Definitions, Acronyms, and Abbreviations

### 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 1: Definition of terms used in the report

|  |  |
| --- | --- |
| **TERM** | **DEFINITION** |
| Actor | A representation in the use case diagram denoting external entities that interact with a system being modeled, e.g., the student success system. |
| Extend Relationship | Denotes insertion of optional behavior of another use case into the primary use case. |
| Generalization Relationship | Denotes a relationship between a general use case and a specific use case. |
| Include Relationship | Denotes the inclusion of behavior of another use case into the primary use case. |
| Use Case | A modeling technique that presents the basic functionality of a system and the actors that interact with each function. |

### Acronyms

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

Table 2: Acronyms

|  |  |
| --- | --- |
| **TERM** | **DEFINITION** |
| SRS | Software Requirements Specification |
| UTEP | The University of Texas at El Paso |
| ARL | Army Research Lab |
| BPF | Berkeley Packet Filter |

### Abbreviations

This section provides a list of used abbreviations and their associated definitions.

Table 3: Abbreviations

|  |  |
| --- | --- |
| **TERM** | **DEFINITION** |
| e.g. | For example |
| i.e. | That is |
| TBD | To be determined |

## 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 Protocol Formatter 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 Protocol Formatter.

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.

## References

[1] J. Acosta and B. Santaella, Requirements Definition Document, 2016.

# General Description

## Product Perspective

Protocol Formatter system is management system that captures and facilitates reuse of analyzed data.

## Product Features

*Figure 1* presents a level 1 use case diagram that provides an overview of the main functionalities provided by the system and the interactions between actors and the system. Figure 3 presents the notations used in a use case diagram. The actors, represented by stick figures, are external entities that interact with the system. The use case, represented by ovals, elucidate the actors’ interactions with the system. *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 the 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.

TBA

Figure 1: Level 1 Use Case Diagram

TBA

Figure 2: Level 2 Use Case Diagram



Figure 3: Use Case Diagram Notation

### Actors Descriptions

Protocol Formatter system classifies actors into the following groups:

* The Analyst class represents network security analysts, who conducts assessments of network systems.

### Use Case Descriptions

Protocol Formatter system supports the following primary use cases:

* Manage Formatter
* Manage Script
* Analyze PDML

ACCESS supports the following include, extend, and generalized use cases:

* Manage Historical PDML
* Convert PCAP to PDML
* Filter

## User Characteristics

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

## General Constraints

The general constraints on the development of the system are as follows:

* The system will be completed by the end of May 2017.

## Assumptions and Dependencies

The assumptions and dependencies of the system are as follows:

* TBA

# Specific Requirements

## External Interface Requirements

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

### User Interfaces

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

* General
* Window
  + Command Line window
  + Historical window
  + Formatter window
  + Packet window
  + Editor window
  + Filter window
  + Script window
  + Hook window

#### General

1. The system shall include the components as presented in Figure 4.
   1. Menu Bar
   2. Icon Bar
   3. Content Area.



Figure : Layout

1. The Menu Bar component shall include the following as presented in Figure 5.



Figure : Menu Bar Component

1. The Icon Bar component shall include the following:
   1. Icon labeled as “Open”
   2. Icon labeled as “Save”
   3. Icon labeled as “Filter”
   4. Icon labeled as “Undo”
   5. Icon labeled as “Redo”
2. The Content Area component shall display the default layout as presented in Figure 6:



Figure : Default Setting for Content Area Component

1. The layout of the Content Area component shall be customizable:
   1. The user shall be able to add new window to Content Area.
   2. The user shall be able to remove existing window from Content Area.
   3. The user shall be able to reposition windows in the Content Area.
   4. The user shall be able to resize windows within the Content Area.
   5. The user shall be able to maximize windows within the Content Area.
   6. The user shall be able to minimize windows within the Content Area.

#### Command Line Window

1. Command Line window shall include the components as presented in Figure 7:
   1. Title labeled as “Command Line Window”
   2. Text labeled as “Command available: Apply <formatter> <pdml> [<outputfilename>]”
   3. Command line text box.



Figure : Command Line Window

#### Historical Window

1. Historical window shall include the components as presented in Figure 8.



Figure : Historical Window

#### Formatter Window

1. Formatter window shall include the components as presented in Figure 9.



Figure : Formatter Window

#### Packet Window

1. Packet window shall include the components as presented in Figure 10.



Figure : Packet Window

#### Editor Window

1. Editor window shall include the components as presented in Figure 11 .



Figure : Editor Window

#### Filter Window

1. Filter window shall include the components as presented in Figure 12.



Figure : Filter Window

#### Script Window

1. Script window shall include the components as presented in Figure 13.



Figure : Script Window

#### Hook Window

1. Hook window shall include the components as presented in Figure 14.



Figure : Hook Window

### Hardware Interfaces

There are no hardware interface requirements specified at this time.

### Software Interfaces

This section describes the characteristics of each interface between other application systems and the system.

1. The system shall interface with TShark 2.0 to convert PCAP to PDML format using the following command.

* $ tshark –r <name of the capture file in PCAP format> -T pdml.

### Communications Interfaces

There are no communication interface requirements specified at this time.

## Behavioral Requirements

This section describes the behavioral requirements of the system.

### Same Class of User

This section describes requirements associated with a particular class of user.

1. The system shall have one class of user – Analyst.
2. The Analyst class shall have access to all features supported by the system.

### Related Real-world Objects

This section describes related real-world object requirements of the system.

#### PDML

1. Structure of a PDML shall consist of the following:
2. <pdml> and </pdml> tags
3. <pdml> element
4. At least one <packet> element.
5. Structure of the PDML shall be preserved at all times.
6. A <packet> element shall consist of at least one <proto> element and the following attributes presented in Table 4.

Table : Attributes of <Packet> Element

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Description** | **Data Type** |
| Num | Required; Number of current packet within the capture. | Integer |
| Len | Required; Total length of the packet | Bytes |
| Caplen | Required; Length of portion of the packet that has been captured. | Bytes |
| Timestamp | Required | Float |

1. A <proto> element shall consist of at least one <field> element and the following attributes presented in Table 5.

Table : Attributes of <Proto> Element

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Constraints/Description** | **Data Type** |
| Name | Required; Name of the current protocol | String |
| Longname | Optional; Long name of the current protocol | String |
| Size | Required; Size of the headers of the current protocol | Bytes |
| Pos | Required; Starting position of the headers of the current protocol | Bytes |

1. A <field> element shall consist of the following attributes presented in Table 6.

Table : Attributes of <Field> Element

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Constraints/Description** | **Data Type** |
| Hide | Optional | Boolean |
| Longname | Required | String |
| Mask | Optional | Bytes |
| Name | Required | String |
| Pos | Required | Bytes |
| Show | Required |  |
| Showdtl | Optional | String |
| Showmap | Optional | String |
| Showname | Required | String |
| Showvalue | Optional; Generate from the value attribute. | Integer |
| Size | Required | Bytes |
| Unmaskedvalue | Required only if the length of the current field is not an integer number of bytes. |  |
| Value | Required | Hex String |

1. A PDML shall have at most one historical copy of itself.
2. A historical copy of the PDML shall be created only upon initial entry to the system.

#### Formatter

1. A formatter shall consist of rules that are specific to one protocol.
2. A formatter shall be associated with at most one protocol.
3. A rule shall consist of two parts:
   1. Filter
   2. Action.
4. A filter shall consist of an expression in the Berkeley Packet Filter (BPF) syntax.
5. An expression shall consist of at least one primitive as defined by the BPF.
6. An action shall consist of the steps an analyst takes to perform one of the following:
7. Hiding
8. Renaming
9. Annotating
10. Creating a hook.
11. A hiding action shall conceal one of the following:
12. A tag
13. A field and its associated value.
14. A renaming action shall rename a value associated with a field.
15. A hook shall consist of the following:
    1. File path to script
    2. Input arguments required by the script: The arguments are associated values of specific attributes defined in the <field> element.

#### Protocol

1. A protocol shall have at most one formatter.

#### Script

1. A script shall be specified in Python.

### Stimulus

This section describes the stimulus requirements of the system.

#### General

1. When the user opens a capture file and the capture file is of type PCAP, the system shall covert the capture file from PCAP to PDML.
2. When the user opens a capture file and the capture file is of type PDML, the system shall check to see if a historical copy of the capture file exists.
3. If the historical copy of the PDML capture file does not exist and it is the file’s initial entry to the system, the system shall create a historical copy of the capture file.
4. When the user clicks on the “x” on the data block, the system shall remove that data entry.

#### Filtering

1. When the capture file is loaded and displayed in the Packet Window, the system shall scan for protocols and display a list of formatters for the existing protocols in the capture file.
2. If the user selects one or more formatters from the displayed list of formatters, the system shall apply the rules to all applicable packets in the capture file.
3. When the formatters have been applied, the system shall display a list of existing protocols in the capture file.
4. If the user selects a protocol from the displayed list of protocols, the system shall isolate the packet data that uses the selected protocol and display the result in the Packet window.
5. When the user clicks on the apply filter button in the Filter window, the system shall isolate the packet data based on the filter criteria specified and display the result in the Packet window.
6. When the user clicks on the reset filter button in the Filter window, the system shall remove all the filter criteria and display the complete set of packet data.

#### Editing

1. When the user selects a packet data and clicks on the Editor window, the system shall enter into an edit mode and disable the filtering mechanism.
2. When the user is in edit mode and clicks on the ‘hide” check box next to the field name, the system shall record the action of hiding a field and its associated value and display it as an action entry in the Formatter window.
3. When the user is in edit mode and clicks on the value of a field, the system shall record the action of renaming a value and display it as an action entry in the Formatter window.
4. When the user is in edit mode and clicks on annotate symbol, the system shall record the action of annotating a field name and its associated value and display it as an action entry in the Formatter window.
5. When the user is in edit mode and right click on a value, the system shall launch the Hook window.
6. When the user clicks on the add rule button in the Hook window, the system shall record the action of creating a hook and display it as an action entry in the Formatter window.
7. If the user clicks on add rule in the Formatter window, the system shall form a new rule with the recorded actions and filter and add it to the formatter.
8. When the user clicks on the Filter window and the recorded actions haven’t been saved as a rule, the system shall exit out of the edit mode and deleted all the recorded actions.

#### Restoring (Incomplete – TBA)

#### Undo (Incomplete – TBA)

#### Deletion (Incomplete – TBA)

## Non-behavioral Requirements

TBA.

&