Software Requirement Specification

Version 1.0

<<Annotated Version>>

November 4, 2014

NetCaPy

Team Members

Srinivas Rishindra	2011A7PS091G
Rajat Jain	2011A7PS153G
Shubham Kankaria	2011A7PS063G
Jatin Parekh	2011A7PS006G
Anshuli Patil	2011A7PS071G

Submitted in partial fulfilment

of the requirements of

IS F341 Software Engineering project

Table of Contents

Table of Contents

1.0 Abstract

This project is aimed at developing a network data capturing software with a Graphical User Interface inspired from Wireshark. This tool should be able to scan the network interfaces for the network packets, capturing them, analysing them and saving them as pcap files on the secondary storage for offline analysis. The project provides the user to carry out live capturing of packets to do monitoring on the network.

2.0 Introduction

This section gives a scope description and overview of everything included in this SRS document. Also, the purpose for this document is described and a list of abbreviations and definitions is provided.

2.1 Purpose

The purpose of this document is to give a detailed description of the requirements for the “Network Data Capturing System” software. It will illustrate the purpose and complete declaration for the development of system. It will also explain system constraints, interface and interactions with other external applications. This document is primarily intended to be proposed to a customer (Here customer being Evaluator) for its approval and a reference for developing the first version of the system for the development team.

2.2 Scope

The “Network Data Capturing System” is an application that captures data on the user’s Ethernet/Wi-Fi Port. The user then has the option to use captured data for analysis using inbuilt functionalities or custom plugins.

The users of the software include users who want to know what his/her computer sends to the network and receives from it, engineers to debug their protocol implementations, network administrators to troubleshoot their networks and network security engineers to discover security breaches. The software allows the user to select the interface i.e. either WIFI or Ethernet. It will also be able to store data captured from the interface in a pcap file, which can be used for further offline analysis. The Software will also be able to filter the data it is capturing based on the BPF (Berkeley Packet Filters) queries made by the user, so that only interesting packets will be displayed to the user. Filtering can be based upon specific ports, IP address etc. The software will also be able to identify the various characteristics of data like packet size, packet type etc.

2.3 Definitions, acronyms, and abbreviations

Packet Analyser: A computer program or a piece of computer hardware that can intercept and log traffic passing over a digital network or part of a network.

Packet Capture: Packet capture is the process of intercepting and logging traffic.

Network Monitoring: The use of a system that constantly monitors a computer network for slow or failing components and that notifies the network administrator (via email, SMS or other alarms) in case of outages.

Promiscuous Mode: A mode for a wired network interface controller (NIC) or wireless network interface controller (WNIC) that causes the controller to pass all traffic it receives to the central processing unit (CPU) rather than passing only the frames that the controller is intended to receive.

Pcap (packet capture): pcap consists of an application programming interface (API) for capturing network traffic.

BPF: The Berkeley Packet Filter or BPF provides, on some Unix-like systems, a raw interface to data link layers, permitting raw link-layer packets to be sent and received.

2.4 Overview

The next chapter, the Overall Description section, of this document gives an overview of the functionality of the product. It describes the informal requirements and is used to establish a context for the technical requirements specification in the next chapter.

The third chapter, Requirements Specification section, of this document is written primarily for the developers and describes in technical terms the details of the functionality of the product.

Both sections of the document describe the same software product in its entirety, but are intended for different audiences and thus use different language.

3. Overall Description

This section will give an overview of the whole system. The system will be explained in its context to show how the system interacts with other systems and introduce the basic functionality of it. It will also describe what type of stakeholders that will use the system and what functionality is available for each type. At last, the constraints and assumptions for the system will be presented.

3.1 Product Perspective

The “Network Data Capturing System” is Linux platform based independent and self-contained tool.

3.2 Product functions

The system has two parts:

- Live packet capturing/ Sniffing
- Packet Analysis

The application collects and stores packets data from the network when used for live capturing. The application can stay online for further capturing of data or it can go offline to analyse the data stored by the application. The data is stored in pcap format. The application can load the data from the packets from pcap files stored on secondary storage which can be used for packet analysis. Various packet analysis options are available in the system such as Packet Count, Protocol Type Count etc.

3.3 User characteristics

There are four types of users that can primarily interact with the system: Regular users, Engineers, Network administrators, Network Security engineers.

● Regular user who wants to know what his/her computer sends to the network and receives from it. They will also be able to know from where a particular packet is coming from.

● Engineers to debug their protocol implementations. New protocol implementations can be tested by the engineers so as to compare the efficiencies of various protocols.

● Network administrators to troubleshoot their networks so that they can identify where the underlying problem lies.

● Network security engineers will be able to discover security breaches by observing the data.

3.4 General Constraints

1. Regulatory Policy: Use of open source libraries.
2. Hardware limitations: NIC card interface with proper drivers to interact with the application.
3. Parallel Operation: Live capturing and packet analysis should operate in parallel.
4. The application is constrained by the system interface to the Wi-Fi or Ethernet ports. Since there are multiple systems and multiple Ethernet and WI-FI manufacturers, the interface will most likely not be the same for every one of them. Also, there may be a difference between what features each of them provide.
5. Some of the systems may not have both WI-FI and Ethernet nevertheless the application should function the same irrespective of the type of network interface that is used by the system.
6. The application will also need enough capacity to store the data captured. Different platforms have different file systems, the application needs to take care of that.

3.5 Assumptions and dependencies

Factors that affect the requirements stated in this SRS:

1. Perspectives of different users about the application
2. Hardware Platform specifications

3. Operating System specifications

Assumptions:

1. If Network Capturing Data System runs out of memory it will crash.
2. One assumption about the application is that it will have enough performance.
3. Another assumption is that network interface components in all the systems work the same way.

3.6 Stakeholders

Customers (Regular users, admin), Project Manager, Team leader, Developers

4 Functionality of the Project

1. User need to start the application by entering root password.
2. He has two choices:
    a. He can start capturing packets by clicking start button
    b. He can open a file to view already/ previously captured packets.
3. If user starts new capturing then he will see current packets passing through the network and packet details on the UI designed.
4. If user selects opening a file option then that file is parsed and packets details are displayed on same UI.
5. By default when new packets are being captured same data is also moved to a log file.
6. When user clicks on save button, he can save that log as per specified location

4.1 Specific Requirements

4.1.0 Functional Requirements

1. Capturing Network Packets

Input: User can choose between different network interfaces on which to carry out capturing either Ethernet, Wi Fi or Loopback through the graphical user interface

Processing: Retrieval of packets from the kernel buffer space into the application and then parsing those packets to get all fields of them and displaying them on a user interface

Output: Display of packets with their fields in a table format with a console window to see the whole description and fields of the packet

2. Saving the Captured Packets

Input: Save button event generated by clicking on save button on the graphical user interface

Processing: Saving the packets on secondary storage in pcap format

Output: Successful creation and writing of packets in a .pcap file

3. Filtering

Input: BPF expression for filtering

Processing: Filtering performed on the list of packets to get only "interesting" packets

Output: Display of only packets that pass the filter in the table format

4. Analysis:

Input: Packets captured are input to the analysis functionality

Processing: Carrying out various analysis on packet length, routes of packets, etc. using different additional visualization tools available like gnu plot etc.

Output: Graphs, histograms

4.1.1External Interface Requirements

1. The user can interact with the system through a graphical user interface. To interact with the interface there is a need for a pointing device (e.g. mouse) to be installed.
2. Filtering requires bpf expressions to be made which requires a keyboard also to be installed.
3. User can shift window panels in the GUI as per his/her choice.
4. The system runs only on Linux platform and requires roots privileges to run it.
5. Presence of NIC card is most vital for the system to work.

4.1.2 Performance Requirements

1. Display of packets as they are captured should not be delayed by more than 50 milliseconds.
2. Responses to various events from the user (foe e.g., Start capture event) should not take much to respond back.

4.1.3 Design Constraints

1. Modular structure of the application so that new modules can be added to it to increase its functionality.

4.1.4 Quality Characteristics

1. Correctness: Application should satisfy all the specifications with good performance.
2. Efficiency: Implementation should be efficient.
3. Flexibility: Modular structure of program to add new functionalities later.
4. Security: Only system admin can use the application.
5. Portability: Application should work on most Linux distributions.
6. Reliability: Application should capture packets with 100% reliability.
7. Reusability: Application code should be reusable by another applications.
8. Usability: User Interface should be elegant and easy to use.

4.1.5 User Interfaces

The interface is composed of a single Main Window, consisting of a Menu Bar at the top, several tool bars which can be moved, a status bar and at the bottom and a central widget consisting of multiple tabs.

Main Window

Menu Bar consisting of the following Menus:

- File - Includes options to exit and open/save file
- Capture - Includes options to start/stop live capture of data
- Filter – Includes links to Pre Built filters (Ex: BPF) or custom filters
- Analysis – Includes links to Pre Built Analysis tools or custom analysis tools
- Help – It is supposed to link to help documents related to the Application

Tool Bars:

- File toolbar – Gives direct access to File Open/Save operations
- Capture Toolbar – Gives direct access to Start/Stop Live Capture
- Analysis Toolbar – Gives access to Analysis tools that support icons
- Filter Toolbar – Gives access to Filters that support tool bar feature

Status Bar:

- The status bar at the bottom should print appropriate messages for the user.

Central Widget:

The central tabbed widget should display multiple tabs as used in the application

The Default Tab:

- It should have option to display the currently active interfaces and its properties
- It should provide option to start live capture or open existing file
- Display common help tools and project homepage

The Packet Data Display Tab:

- This tab is to be displayed on Live capturing or opening a PCAP file
- It should display packet information in a tabular format
- It should also give detailed summary about a packet when selected

4.1.6Database Structure

1. To simplify the design we will currently use files as our database to store the packet data. We can use a relational database in future.

2. If user clicks save button to save the captured packets then packets should be captured in a pcap file.

5.0 Future Enhancements

1. Troubleshoot network problems in case data transmission is terminated prematurely/unexpectedly.
2. Run both processes (writing into and reading from log file) asynchronously using multithreading.
3. Try to capture other types of protocols also.