Arukgoda J.S.
100040X

jShark
Network Protocol Analyzer for Android

Software Requirement Specification

Version <1.0>

Revision History

Date	Version	Description	Author
31/07/13	1.0	Initial Software Requirement Specification	Arukgoda J.S.

Table of Contents

Software Requirements Specification

1. Introduction

1.1 Purpose

This document captures the initial version of software requirements for the system. This includes functional and non-functional requirements, documentation requirements, design constraints, interface specifications, licensing requirements and standards.

1.2 Scope

jShark is a free and open source NPA for Android platform. It is inspired by Wireshark which does not have an implementation to this date.

1.3 Definitions, Acronyms, and Abbreviations

[1] NPA : Network Protocol Analyzer

[2] API : Application Programmer Interface

[3] GUI : Graphical User Interface

1.4 Overview

The system requirements specification document for jShark describes what jShark is, and provides in detail the specific project requirements.

2. Overall Description

A network protocol analyzer captures network traffic to be viewed and to be analyzed so that network troubleshooting can be done. Wireshark is one of the most popular free and open source NPA for Windows and Unix platforms. But it is not available for Android platform. The need for a free and open source NPA arises with the increase of usage of mobile devices.

With jShark, the user can capture WLAN network traffics from his mobile device and view the captured packets, filter them according to source, destination and protocol type. A capture session can be stored in the mobile device for future references.

3. Specific Requirements

3.1 Functionality

The functional requirements identified are listed down and described below.

3.1.1 Capture IEEE 801.11 (WLAN) traffic

jShark should allow user to capture WLAN traffic when the device is in range with a wifi hotspot. The device need not be connected to the network to receive the network traffic.

3.1.2 Display captured network traffic real time

The captured packets should be decoded and displayed on the device screen real time. Whenever a new packet is captured, that should be added to the list of captured packets without any user involvement.

3.1.3 Allow user to filter the capture session

User should be able to filter the capture session so that only packets from a specific source, packets to a specific destination or packets using a specific protocol are displayed on the device screen.

3.1.4 Allow user to store a capture session on the device

jShark should allow user to store a capture session on the device memory as a separate file so that that session can be viewed later using jShark from the same device or from a different device.

3.2 Usability

3.2.1 No training should be needed for any type of user

The GUI of jShark should be descriptive, unambiguous and clear enough so that a normal user should be able to use all features of jShark without any prior training.

3.2.2 jShark should be responsive

The jShark should respond to the user command 'start capture session' within 1 second. A 'filter' command and 'save' command should take minimal time.

3.3 Reliability

3.3.1 jShark should be available at all times

Since jShark is a stand-alone software, it should be available on the installed device all the time with no allowance for system crashes / failures.

3.3.2 Bug rate

the goal of the project is to develop a bug free application but considering the experience and the technical knowledge of the developer, 1 bug per 1000 lines of code is accept for the first release of jShark

3.3.3 Accuracy level

The displayed capture session should be 100% accurate. In other words, the captured packet should be displayed without any changes to it and all packets captured should be displayed.

3.4 Performance

3.4.1 Real time capture and display

The captured packets should be processed and displayed real time. The latency between capture and display should be less than 500 ms.

3.5 Supportability

3.5.1 Coding practices

Since this is an open source project, the code is expected to be reviewed and improved by many different individuals and organizations. Therefore all good coding practices should be followed to ensure maximum convenience to those individuals and organizations.

3.6 Design Constraints

3.6.1 Existing libraries

An implementation of' pcap' or a similar API has to be used to capture network traffic. The library used will decide the programming language to be used.

3.7 On-line User Documentation and Help System Requirements

All documentation and the code will be made available in the Git Hub repository https://github.com/jseanm1/jShark

3.8 Purchased Components

The project will only use free and open source resources or trial versions of other licensed resources.

3.9 Interfaces

3.9.1 User Interfaces

A simple GUI complying with 3.2.2

3.9.2 Hardware Interfaces

jShark should import a library which is an implementation of 'pcap' to evoke the device's network interface.

3.9.3 Software Interfaces

jShark is a stand-alone application. Android SDK is used to interface with the Android operating system. No other software interface is needed at the moment.

3.9.4 Communications Interfaces

No communication interfaces.

3.10 Licensing Requirements

jShark will be published under GNU Generic Public License.