The

Software Requirements Specification

for

abcdxyz Scanner

CS 4398 Software Engineering Project - Group 2

Prepared by:

Greg Pontejos

Donovan Wells

Jason Villegas

Kingsley Nyaos

Alex Gimson

February 4th, 2017

Version 1.0

Submitted to: Dr. R Podorozhny

Department of Computer Science

Texas State University

Copyright © 2017 Greg Pontejos, Donovan Wells, Jason Villegas, and Kingsley Nyaosi

All rights reserved.

No part of this publication may be reproduced in any form, in an electronic retrieval system or otherwise, without the prior written permission of the publisher.

# 1. Preface

## 1.1 Introduction

This section outlines the member contributions for this requirements document.

Greg: Preface, Overall Description, and Functional Requirements

Donovan: Preface, Overall Description, and Functional Requirements

Jason: Non-Functional Requirements and Appendix

Kingsley Nyaosi: Non-Functional Requirements

## 1.2 Purpose

This document details the layout and function of the abcdxyz scanner. This document will describe how the scanner should work and provide a thorough understanding of how different types of users can interact with the scanner as well as serve as a reference for the development team.

## 1.3 Document Conventions

This document uses bold headings to identify major sections of the Software Requirements Specifications document. The subcategories will follow a tiered numbering convention to help keep track of each subsection within a main section.

## 1.4 Intended Audience

This requirements document is written for developers and stakeholders to aid in the construction of this scanner.

## 1.5 Proposed Document Scope

This requirements document provides detail into the workings of the abxdxyz scanner.

All users who will use the scanner will interact with the scanner uniformly as there are no admin or members with limited abilities. The purpose of the scanner is to monitor network traffic, internal and external, for the purpose of troubleshooting network problems. Examples of its use include, but are not limited to, troubleshooting network problems between physical machines, virtual machines, and containers.

The scanner consists of a simple GUI contain a top menu, network window, monitor window, and a control window.. A user will be able to specify the interface in which the network traffic will be extracted from. The top menu will consist of basic program functionality, such as opening files, closing files, closing the program, and saving files. The monitor window simply outputs the desired network data, whether from a live or saved capture. The control window will display current metrics, such as total packets captured, and allow the User to start, stop, and clear a network packet capture.

## 1.6 Definitions and Abbreviations

Included are definitions of the relevant components contained within the abxdxyz scanner:

**ARP:** Address Resolution Protocol

**ICMP:** Internet Control Message Protocol

**JPCAP:** Java wrapper program for libpcap and Winpcap

**NIC:** Network Interface Controller

**Promiscuous Mode:** A computer networking operational mode that allows for access to all network data packets received by a NIC and/or WNIC.

**SDDP:** Simple Service Discovery Protocol

**TCP:** Transmission Control Protocol

**UDP:** User Datagram Protocol

**WNIC:** Wireless Network Interface Controller

# 2. Overall Description

This section will give a brief overview of the abxdxyz scanner. This scanner will allow users to scan network traffic on the network card of their choice. abxdxyz is intended to allow the user to scan ICMP, TCP, UDP, and ARP packets on a network.This report will also describe the types of users that can interact with the scanner and what functionality is available for each type of user. The main focus of this document will be the functional and non-functional requirements, where specific use cases will be explained.

## 2.1 Document Perspective

abxdxyz is a network protocol analyzer that will let users see what’s happening on their network at a microscopic level. The users can browse the network data via a GUI. From analyzing the data, the user will be able to examine in detail whats going on inside their network cable and thus troubleshoot network problems, examine security problems, debug protocol implementations and learn network protocol internals.

## 2.2 User Characteristics

All users that interact with the system will have the same use of the system thus the requirements for each user are all the same. It is assumed that the user is computer literate and will not be using the data for malicious purposes. All users will have the ability to run the scanner to capture packets in real time, import previous captures, view all packet information, and filter packet data as needed based on specified criteria.

## 2.3 Constraints

The scanner application is constrained by the availability of wired and wireless network interfaces -- access to a NIC or WNIC is required for the application to function properly, unless the user imports pcap files created by other programs. Additionally, the full functionality of the application can hindered by the permissions the user or system has set on the network interfaces. If promiscuous mode cannot be enabled, the user will not experience the full potential of the application.

## 2.4 Assumptions

It is assumed that the user is exercising ethical behavior when using this software. Additionally, all saved capture data will be unencrypted -- it is the responsibility of the User to encrypt their data.

# 3. Functional Requirements

This section includes the requirements that specify all of the fundamental actions of the abxdxyz scanner.

## 3.1 Scanner Settings

### 3.1.1 Use Case 1 - Start Live Capture

PRE: User must have specified a network interface for capture.

TRIGGER: The user will click the ‘Start’ button to begin capturing network packets.

OUTPUT: The program will display all packets that are captured in the center window.

### 3.1.2 Use Case 2 - Promiscuous Mode

PRE: User runs the program as super user and is not currently running a live capture.

TRIGGER: User will click the Promiscuous Mode checkbox to enable or disable promiscuous mode.

OUTPUT: All network packets will be captured if promiscuous mode is enabled, or only packets authorized to be seen will be captured if disabled.

### 3.1.3 Use Case 3 - Save Capture

PRE: User has completed a new packet capture.

TRIGGER: User will click File -> Save Capture.

OUTPUT: pcap file will be saved to the user’s desired location.

### 3.1.4 Use Case 4 - Open Capture

PRE: User has a previously saved capture, or pcap file available.

TRIGGER: User will click File -> Open, then select a valid pcap file.

OUTPUT: All captured packet data will populate in output terminal.

### 3.1.5 Use Case 5 - Filter Data - Network

PRE: User has completed all desired fields within the IP section of the network window -- at least one attribute must be completed.

TRIGGER: User will click the ‘Enable’ button to apply the filter.

OUTPUT: Only packets meeting the IP criteria will be displayed in the monitor window.

### 

### 3.1.6 Use Case 6 - Stop Live Capture

PRE: User currently has a live capture in progress.

TRIGGER: User will click the ‘Stop’ button in the control window.

OUTPUT: Packets will no longer be captured and all captured packets will be available to view in the monitor window.

### 3.1.7 Use Case 7 - Filter Data - Protocol

PRE: User has checked all desired protocols to be filtered, which are any combination of ARP, TCP, ICMP, UDP -- at least one attribute must be selected.

TRIGGER: User will click the ‘Enable’ button to begin filtering packets.

OUTPUT: User will only see all packets meeting the filter criteria in the monitor window.

### 3.1.8 Use Case 8 - Reset Capture Statistics

PRE: User has completed a new live capture has loaded a live capture via previously saved pcap file.

TRIGGER: User click the ‘Clear’ button in the control window

OUTPUT: All packets currently displayed in the monitor window will be removed.

### 3.1.9 Use Case 9 - View Payload

PRE: User will have a previous or recently run capture loaded, IP source, IP destination, protocol, and port selected.

TRIGGER: User will click the ‘View Payload’ button to view this feature.

OUTPUT: An output of the concatenated payloads for the network data stream will be displayed as cleartext.

### 3.1.10 Use Case 10 - View Conversations

PRE: User will have a complete previous or recently run capture.

TRIGGER: User will click the ‘View’ menu dropbox, then click ‘Conversations’

OUTPUT: A new window will appear displaying all source destination IP addresses and their corresponding destinations

### 

### 3.1.11 Use Case 11 - Follow Stream

PRE: User will have a populated current or filtered capture, ‘Packet Viewer’ window open, and a current packet displayed.

TRIGGER: User will click the ‘Follow Stream’ button

OUTPUT: A concatenation of the payloads for the current packet conversation will be displayed as ASCII output.

### 3.1.12 User Case 12 - View Interface Information

PRE: User will have interfaces connected and running on their current machine

TRIGGER: User will click the ‘Capture’ menu dropbox, then click ‘Interface Info’

OUTPUT: A new window will appear displaying all information for all active and inactive interfaces on their current machine.

# 4. Non-Functional Requirements

In this document, tasks, or use cases for this scanner will be evaluated in terms of how they satisfy Performance, Reliability, Robustness, Maintainability, Usability, Responsiveness, and Security requirements.

## 4.1 Performance

This scanner will need to perform consistently and be able to run in UNIX and Windows based systems. The program is lightweight and will not interfere with normal system functionality. All packet data will displayed in real time.

## 4.2 Reliability

This scanner will be able to effectively handle regular use by a user and function properly 95% of the time. If there are no bugs in the scanner, it will be reliable and able to process and handle all the incoming network data and all filters a user can pick from. This scanner will have an availability where 99% of data is captured. While, users are performing an action, a computer must not dismiss an engaged addition or change menu unless the user requests it. This scanner should not show the data in the GUI for any failed processes.

## 4.3 Robustness

This scanner will have multiple measures in place to ensure that it functions properly when an invalid input is entered. If a user picks an interface where no data is being transmitted, they will be prompted with an error message stating that no packets were found and to try another interface.

## 4.4 Maintainability

This scanner will be written in a way that is easy to maintain. All code will be written following a specific style guide with uniform naming conventions and will have comments indicating what each method is accomplishing. Software bugs arising from version updates will be kept to a minimum by practicing an agile software method and will be implemented incrementally.

## 4.5 Usability

This scanner will have a simple graphical user interface that will be easy for all users to use. There will be features the user can manipulate such as: filtering a specific type of packet, saving the network information from the current scan, loading the data from a previous scan, and selecting an interface to scan traffic. The menus of the system must be easily navigable by the users with buttons that are easy to to understand.

## 4.6 Responsiveness

abxdxyz is designed to have vast bounds of operation by conforming to a wide spectrum of operating conditions, such as a range of supported network bandwidth situations and network types. This scanner can only be accessed if the viewer or member has Wi-Fi, connected to the internet via wire, or has an internal network established.

## 4.7 Security

At abxdxyz, we recognize that our security threat model revolves around our user’s personal information. This scanner is designed to be safe and secure across our networks. We understand that identity security tactics revolve around the four key principles: detection, resistance, mitigation, and recovery. Since the scanner allows users to view non secure network data, It is the responsibility of the user to practice good ethics and ensure other user information is kept safe. abxdxyz is implemented with the following safety techniques:

* Access Control
* A user shall only be able to view one interface at a time
* The scanner will not manipulate things on the network as it will only measure activity from it
* The scanner will not send packets on the network or do other active things
* Encryption
* Wireless communication throughout the system will be encrypted

## 4.9 Maintenance and Future Development

There is no required maintenance for the program as long as there are no changes to the current network protocols. Future developments, if time and resources permit, include the addition of the following:

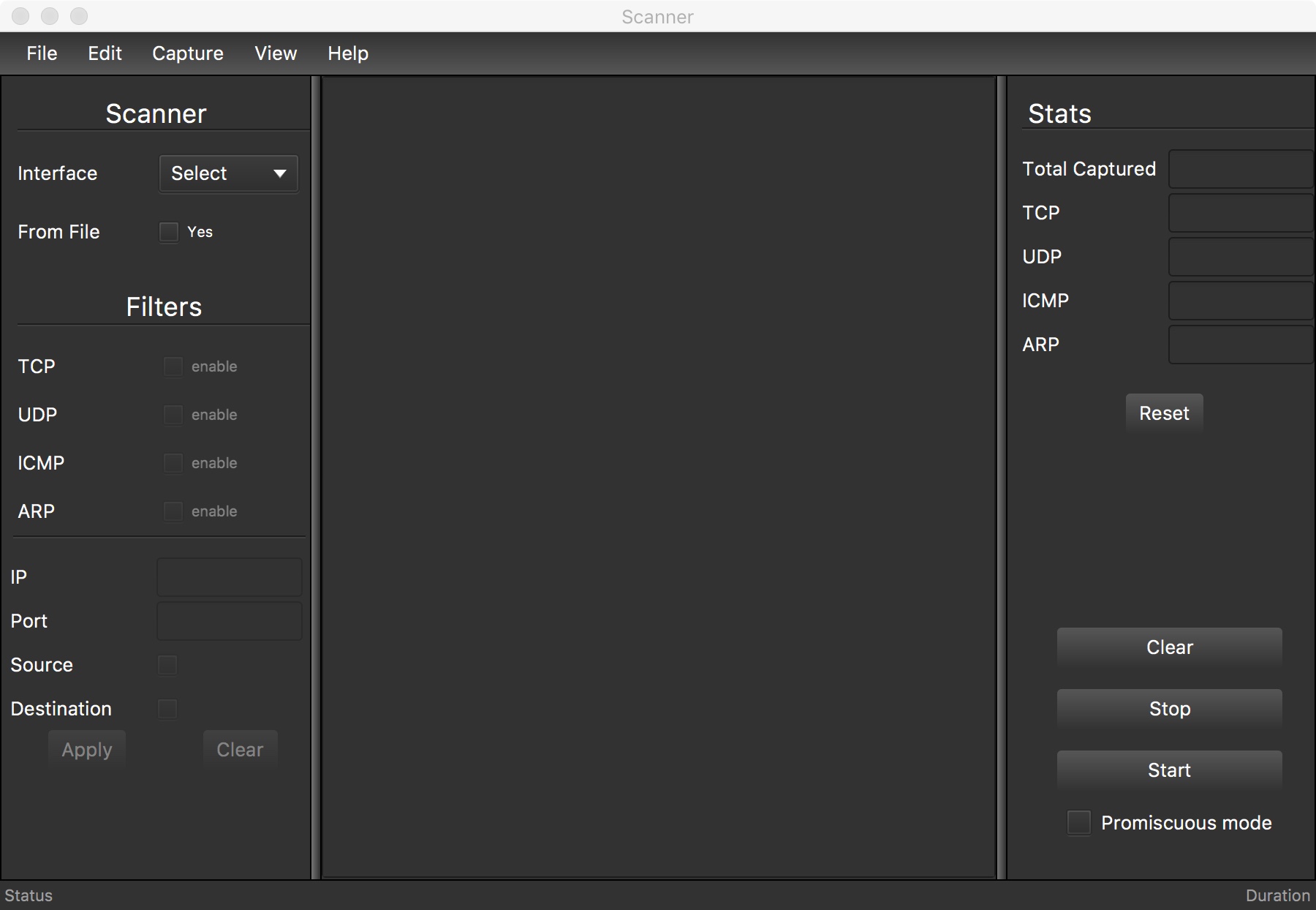
* Port scanning capabilities
* Data encryption and decryption for saved capture files.
* Custom TCP packet creation.
* Follow a TCP or UDP stream and display the payload in an easily readable format.

# 

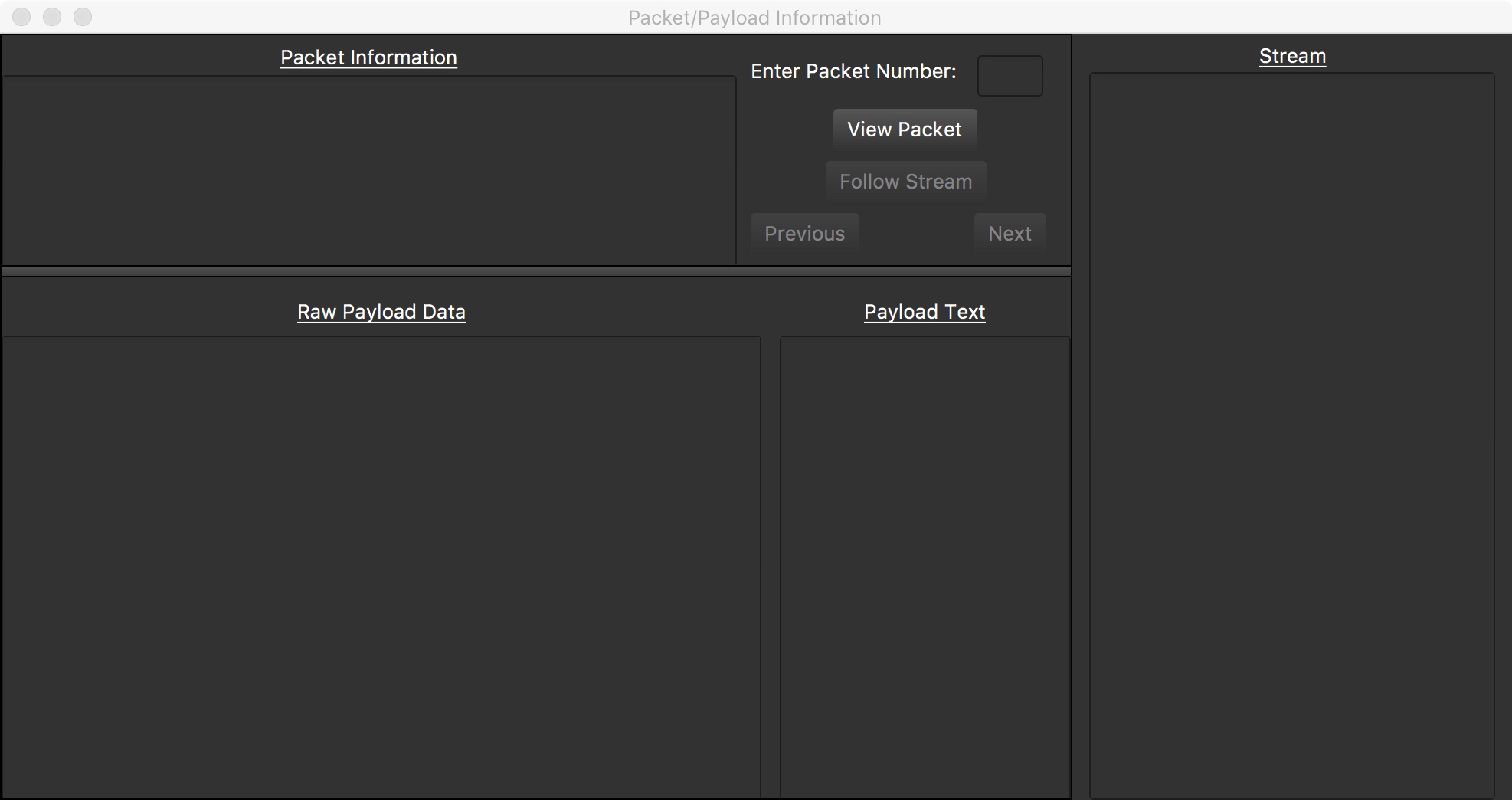
# 

# 5. Appendix

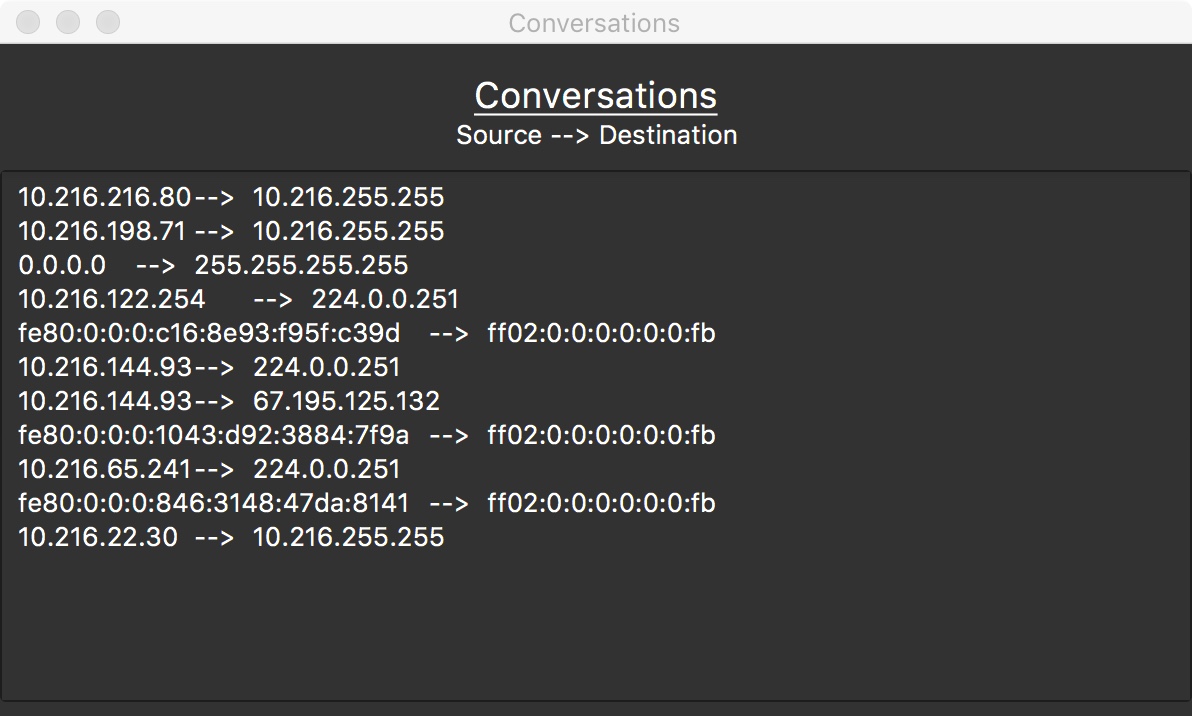
## 5.1 Main Window



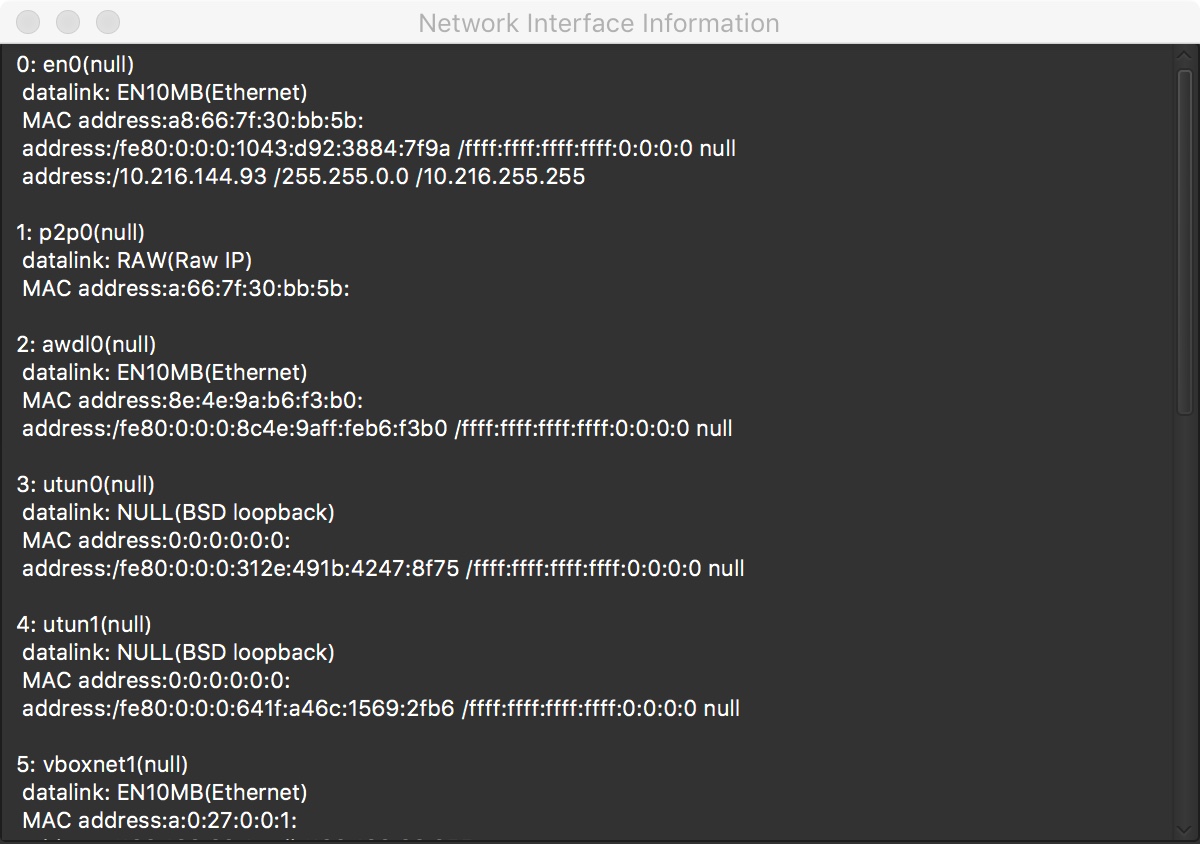
## 5.2 Packet Viewer Window



## 5.3 Conversations Window



## 5.4 Interfaces Window



## 5.5 About Window

