



sniffdatel

SW Engineering Projekt FS 2016

Anforderungsspezifikation

David Meister, Giorgio Vincenti, Samuel Krieg, Andreas Stalder

18. April 2016

1 Änderungsgeschichte

Datum	Version	Änderung	Autor
15.03.16	1.0	Erstellung erster Version	Giorgio Vincenti
21.03.16	1.1	Review ready	Alle
26.03.16	1.2	Review ready	Alle

Tabelle 1: **Änderungsgeschichte**

Inhaltsverzeichnis

1	Änderungsgeschichte	2
2	Einführung	5
2.1	Zweck	5
2.2	Gültigkeitsbereich	5
2.3	Referenzen	5
3	Allgemeine Beschreibung	5
3.1	Produkt Perspektive	5
3.2	Produkt Funktion	5
3.3	Benutzer Charakteristik	6
3.4	Einschränkungen	6
4	Use Cases	7
4.1	Use Case Diagramm	7
4.1.1	Bemerkung	7
4.2	Aktoren	7
4.3	Beschreibungen (Brief)	8
4.3.1	UC1: select interfaces	8
4.3.2	UC2: collecting sessions	8
4.3.3	UC3: listening conversations	8
4.3.4	UC4: save file	8
4.4	Beschreibungen (Fully Dressed)	9
4.4.1	UC1: select interfaces	9
4.4.2	UC2: collecting sessions	9
4.4.3	UC3: listening conversations	11
4.4.4	UC4: save file	11
5	Nichtfunktionale Anforderungen	12
5.1	Performance	12
5.1.1	Ladezeiten	12
5.1.2	Datenübertragungsraten	12
5.2	Qualität	12
5.2.1	Funktionalität	12

5.2.2	Zuverlässigkeit	12
5.2.3	Benutzbarkeit	13
5.2.4	Effizienz	13
5.2.5	Wartbarkeit	13
5.2.6	Übertragbarkeit	13
5.3	Datenschutz	13
5.4	Schnittstellen	14
5.4.1	Benutzerschnittstellen	14
5.4.2	Netzwerkschnittstellen	14

2 Einführung

2.1 Zweck

Dieses Dokument beschreibt die Anforderungen mittels Use Cases und nichtfunktionalen Anforderungen.

2.2 Gültigkeitsbereich

Dieses Dokument ist über die gesamte Projektdauer gültig. Es wird gegebenenfalls in späteren Iterationen angepasst. Somit ist jeweils nur die neuste Version des Dokuments gültig und alte Versionen sind obsolet.

2.3 Referenzen

Anforderungen

domainanalyse.pdf

3 Allgemeine Beschreibung

3.1 Produkt Perspektive

sniffdatel ermöglicht das Aufzeichnen und Abspielen von Voice over IP Paketen in Echtzeit. Wireshark bietet seit langem die Möglichkeit, den Netzwerkverkehr auf dem Interface aufzuzeichnen, RTP Streams zu filtern und diese abzuspielen. Wireshark ist jedoch nicht in der Lage, RTP Datenpakete in Echtzeit wiederzugeben. sniffdatel filtert RTP Streams aus dem mitgeschnittenen Netzwerkverkehr, stellt die Streams auf einem GUI dar und spielt sie nach Wunsch ab.

3.2 Produkt Funktion

sniffdatel scannt Netzwerkpakete auf einem Interface. Mittels Filtertechniken werden SIP,SDP und RTP Pakete gefiltert. Aufgrund der SIP SDP Pakete werden Sessions gefunden und aufgelistet. Zugehörige RTP Pakete werden gefiltert und deren Payload extrahiert, aufbereitet und auf einem Audio Player abgespielt.

3.3 Benutzer Charakteristik

Zielgruppe sind Informatikinteressierte, Studenten und Dozenten, welche sich mit der VoIP Thematik befassen möchten. Grundkenntnisse in IP-Netzwerken und VoIP sind hilfreich jedoch nicht zwingend notwendig.

3.4 Einschränkungen

sniffdatel wird nur auf Linux Plattformen mit installierter libpcap (Version 0.9) unterstützt. Ausserdem wird eine JRE (Java Runtime Environment) ab Version 8 vorausgesetzt. Es werden nur RTP Pakete, welche mit G.711 im PCMA Modus codiert wurden, unterstützt.

4 Use Cases

4.1 Use Case Diagramm

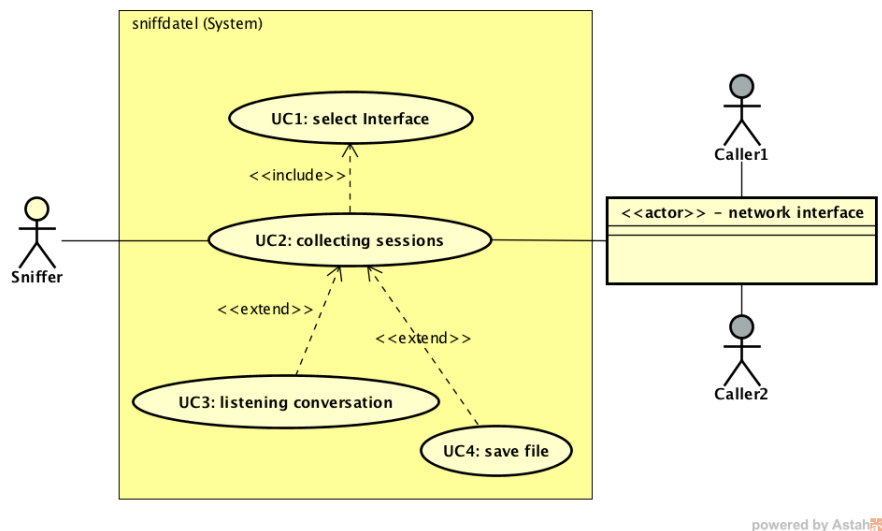


Abbildung 1: Use Case Diagramm

4.1.1 Bemerkung

Der Sniffer möchte gerne eine VOIP Session abhören (UC2). Damit dies möglich ist muss er zuerst ein Interface wählen auf dem er den Capture der Netzwerk Pakete durchführen möchte (UC1 - include). Falls Sessions gefunden wurden kann er diese abhören (UC3). Optional kann der Sniffer die Quelle als File abspeichern (UC4).

4.2 Aktoren

Actor	Beschreibung
Sniffer	Nutzer des Systems sniffdatel
network interface	Liefert die benötigten Pakete
Callers	Telefonierende Personen

Tabelle 2: Actors

4.3 Beschreibungen (Brief)

4.3.1 UC1: select interfaces

Der Sniffer hat die Möglichkeit Interfaces auszuwählen auf die er einen Capture starten möchte. Sniffdatel liefert dazu ein kleines GUI mit einer Auflistung der vorhandenen Netzwerk Interfaces. Der Sniffer wählt dazu eines aus und bestätigt die Auswahl. Der Sniffer kann das GUI auch jederzeit wieder verlassen mit cancel.

4.3.2 UC2: collecting sessions

Der Sniffer kann einen Capture Vorgang starten um eingehende Datenpakete zu analysieren. Sniffdatel listet dabei vorhandene VOIP Sessions im Netzwerk auf in einer Liste. Der Sniffer hat die Möglichkeit eine Session auszuwählen. Der Sniffer hat ebenso die Möglichkeit einen Capture Restart durchzuführen (starten Vorgang neu), oder den Capture Vorgang abubrechen.

4.3.3 UC3: listening conversations

Der Sniffer hat die Möglichkeit die von Sniffdatel aufgelistete Sessions abzuspielen mittels Play (trigger). Sniffdatel startet jetzt eine Audio Wiedergabe der ausgewählten Session. Der Sniffer hat die Möglichkeit die Wiedergabe zu stoppen, oder eine andere Session abzuhören. Sniffdatel beendet die Wiedergabe automatisch sobald keine weiteren eingehende Pakete zu einer Session gefunden wurden.

4.3.4 UC4: save file

Der Sniffer hat die Möglichkeit eine laufende Session als Datei lokal abzuspeichern. Für diesen Vorgang muss der Sniffer nicht die Quelle abhören, sondern kann statt Play den Button Save drücken. Es handelt sich hierbei um ein optionales Feature, welches nur bei ausreichender Zeit berücksichtigt wird.

4.4 Beschreibungen (Fully Dressed)

4.4.1 UC1: select interfaces

Primary Actor: Sniffer

Postcondition: Interface ausgewählt und bestätigt

Stakeholder and Interests:

- Sniffer: will Gespräch von Teilnehmer abhören

Goal: Gewünschtes Interfaces für Sniffing ausgewählt

Trigger: Clickevent auf "capture interface"

Main Success Scenario (or Basic Flow):

1. Snifferdatel liefert Interfaceauswahl
2. Sniffer wählt gewünschte Interfaces aus
3. Sniffdatel markiert gewünschtes Interfaces
4. Sniffer bestätigt Auswahl
5. Sniffdatel aktiviert markierte Interfaces (falls welche ausgewählt)

Extensions (or Alternative Flows):

- *a. jederzeit - Applikation stürzt ab
 1. Sniffer startet Applikation neu
- 1a. Sniffdatel liefert keine Interfaces
 1. Sniffer startet Applikation neu
- 1-3a. Sniffer bricht Vorgang ab
 1. Sniffdatel schliesst Interfaceauswahl

4.4.2 UC2: collecting sessions

Primary Actor: Sniffer

Precondition: Sniffdatel gestartet

Stakeholder and Interests:

- Sniffer: will VOIP-Sessions sammeln
- VOIP Teilnehmer: sind unwissend über Vorgänge des Tools

Goal: Analyse von VOIP Sessions

Trigger: Sniffer möchte das Programm verwenden

Main Success Scenario (or Basic Flow):

1. Sniffdatel zeigt GUI an
2. Sniffer Clickt "capture interface"
Siehe Sub-UC "select interface"
3. Sniffer Klickt "start capture"
4. Sniffdatel Listet Sessions auf
5. Sniffer wählt Session aus
6. Sniffdatel markiert gewünschte Session
7. Sniffer klickt "start audio"
Siehe Sub-UC "listening conversation"
8. Sniffer beendet Sniffdatel

Extensions (or Alternative Flows):

- *a. jederzeit - VOIP Verbindungsaufbau findet statt
 1. Sniffdatel nimmt neue Verbindung in Liste auf
 - *b. jederzeit - VOIP Verbindung wird beendet
 1. Sniffdatel kennzeichnet beendete Verbindung
 - *c. jederzeit - Sniffdatel absturz
 1. Sniffer muss Software neustarten
- 4-8a. Sniffer klickt "restart capture"
1. Sniffdatel startet Capture neu
- 4-8b. Sniffer klickt "stop capture"
1. Sniffdatel beendet Capture

4.4.3 UC3: listening conversations

Primary Actor: Sniffer

Postcondition: Wiedergabe ist beendet

Stakeholder and Interests:

- Sniffer: will Gespräch von Teilnehmer abhören

Goal: Gespräch erfolgreich abgehört

Trigger: Clickevent auf “start audio“

Main Success Scenario (or Basic Flow):

1. Sniffdatel startet Wiedergabe der ausgewählte Session
2. Sniffer beendet Wiedergabe

Extensions (or Alternative Flows):

- *a. jederzeit - Applikation stürzt ab
 1. Sniffer startet Applikation neu
- *b. jederzeit - Sniffer klickt “stop capture“
 1. Sniffdatel beendet Wiedergabe
- 1a. Sniffdatel startet Wiedergabe nicht
 1. Sniffer versucht Wiedergabe erneut zu starten
- 1b. Sniffer wählt andere Session
 1. Sniffdatel startet Wiedergabe der ausgewählte Session
- 2a. Sniffdatel beendet Wiedergabe

4.4.4 UC4: save file

Da es sich bei UC4 um ein optionales Feature handelt, wurde auf eine Fully Dressed Beschreibung verzichtet. Siehe Kapitel 4.3.4 UC4: save file für eine Kurzbeschreibung.

5 Nichtfunktionale Anforderungen

In diesem Kapitel behandeln wir die nichtfunktionalen Anforderungen an das Projekt. Wir behandeln Aspekte und Anforderungen aus den Bereichen Performance, Qualität, Schnittstellen und Randbedingungen.

5.1 Performance

5.1.1 Ladezeiten

Die Zeit zum Aufstarten des Programms darf 10 Sekunden nicht überschreiten. Beim Abspielen des aufgezeichneten VoIP Verkehrs darf die Verzögerung nicht länger als 1 Sekunde betragen (Real-Time).

5.1.2 Datenübertragungsraten

Sniffdatel ist ein lokal installiertes Tool zum Aufzeichnen und Abspielen von VoIP Verkehr. Das Tool soll bei Datenübertragungsraten von und bis 100Mbit/s stabil funktionieren.

5.2 Qualität

Bei der Softwarequalität stützen wir uns auf die ISO/IEC 9126 Norm. Es werden die Merkmale Funktionalität, Zuverlässigkeit, Benutzbarkeit, Effizienz, Wartbarkeit und Übertragbarkeit aufgeführt.

5.2.1 Funktionalität

Durch die Java Technologie wird sichergestellt, dass Sniffdatel auf allen gängigen Betriebssystemen funktioniert. Vorgesehen sind vorerst jedoch nur Clientgeräte (PC/Notebook) mit gängigen und aktuellen UNIX Betriebssystemen.

5.2.2 Zuverlässigkeit

Da es sich bei sniffdatel um ein lokales Programm handelt, welches in keinsten Weise businesskritisch ist, sind keine erhöhten Anforderungen an die Zuverlässigkeit des Systems gegeben. Eine Stabilität unter den gegebenen Rahmenbedingungen ist trotzdem Pflicht. Besonderes Augenmerk ist dem Buffer-Mechanismus zu schenken.

Bei fehlenden RTP Paketen, oder bei zu hoher Netzwerklast darf das Programm nicht abstürzen. Out-of-Order RTP Pakete oder Verzögerungen sollen durch einen guten Buffer-Mechanismus korrigiert werden.

5.2.3 Benutzbarkeit

Ein einfaches und intuitives Bedienen des Programms ist sehr wichtig für uns. Wir verzichten auf aufwändige GUI Elemente und beschränken uns auf wesentliche Funktionen und Anzeigen. Unser Ziel ist, dass weniger versierte Benutzer das Programm ohne Einarbeitungszeit bedienen können.

5.2.4 Effizienz

Siehe Kapitel Performance.

5.2.5 Wartbarkeit

Momentan sind keine Erweiterungen für sniffdatel geplant. Zusätzliche Features sind jedoch gut denkbar. Man könnte z.B. eine Entschlüsselungsfunktion für verschlüsselten RTP Payload einbauen oder zusätzliche Funktionen für die Audiowiedergabe (stop, zurück, etc.) einbauen. Unser Ziel ist es, solche Zusatzfunktionen für spätere Anpassungen möglich zu machen.

5.2.6 Übertragbarkeit

Dank der Java Technologie sind die Programme grundsätzlich Plattformunabhängig. Wir unterstützen vorerst jedoch nur UNIX Betriebssysteme, damit wir Probleme diesbezüglich vermeiden möchten.

5.3 Datenschutz

Sniffdatel speichert keine personenbezogenen Daten. Die abgehörten Gespräche werden nicht dauerhaft gespeichert. Es kann jedoch nicht verhindert werden, dass der Benutzer des Systems die Software missbräuchlich verwendet (z.B. Abhören von Gesprächen Dritter) oder die abgespielten Gespräche mit einem anderen Werkzeug abspeichert.

5.4 Schnittstellen

5.4.1 Benutzerschnittstellen

Die Steuerung des Programms ist nur mit der Maus vorgesehen.

5.4.2 Netzwerkschnittstellen

Im Programm können alle vom Betriebssystem erkannten Netzwerkinterfaces verwendet werden.