

Softwareprojekt 2016

Reviewdokument: PECTO

Rapid Layer-2 Encryption Framework

Yannic Faulwetter
Daniel Scheliga

Michael Fuchs
Tobias Schubert

Milan Haverkock
Jörn Weisensee

Felix Seidel
Nils Winkelbach

letzte Änderung: 22. April 2017

Reviewdokument

Das Reviewdokument ist in zwei Teile aufgeteilt. Zum einen enthält es die Ergebnisse der Planung, zum anderen werden hier auch die des Entwurfes festgehalten. Die Ergebnisse der Planung beinhalten das Vorgehensmodell, Aufwands- und Risikoabschätzung, Meilensteine und diverse organisatorische Vereinbarungen. Im Entwurf werden die Resultate der aktuellen Iteration, Festlegungen für die nächste Iteration und die verwendeten Werkzeuge aufgezeigt.

1 Ergebnisse der Planung

Ein großer Teil der Planung eines Projektes spiegelt sich im Pflichtenheft wieder. Dort wird die Anforderungsanalyse festgehalten, Ziele deklariert, sowie Entscheidungen und Produktinformationen niedergeschrieben.

1.1 Vorgehensmodell

Diese Modellart ist eine Beschreibung des Vorgehens, welches passend zu den Anforderungen des Projektes gewählt werden sollte. Bei der Auswahl und Entscheidung für ein spezielles Vorgehen spielen Anforderungen, Richtlinien und Kenntnisse über den Ablauf eine Rolle.

Gewähltes Vorgehen: Nach dem Vorbild von Scrum wurde ein agiles Vorgehen, in insgesamt 3 Sprints aufgeteilt, beschlossen. Jeder Sprint endet mit einem der Vorstellungstermine, die uns durch das Softwareprojekt vorgegeben sind.

Projektspezifische Anpassungen: Einzelpersonen haben mehrere Rollen, da der Aufwand für jede Rolle sehr unterschiedlich ist.

1.1.1 Vorgehensspezifische Inhalte

Da für dieses Projekt ein agiles Vorgehen gewählt wurde, mussten hierbei die Meilensteine, welche das Ziel am Ende jeder Iteration sind, definiert werden. Diese geben einen Ausblick auf den Leistungsumfang des Systems, welchen es bis zum Ende der jeweiligen Iteration abdecken soll.

1.2 Aufwandsabschätzung

Mit der Aufwandsabschätzung wird versucht, die verschiedenen Teile des Projektes anhand von Aufwands- und Komplexitätskriterien einzuteilen. (siehe Abbildung 1)

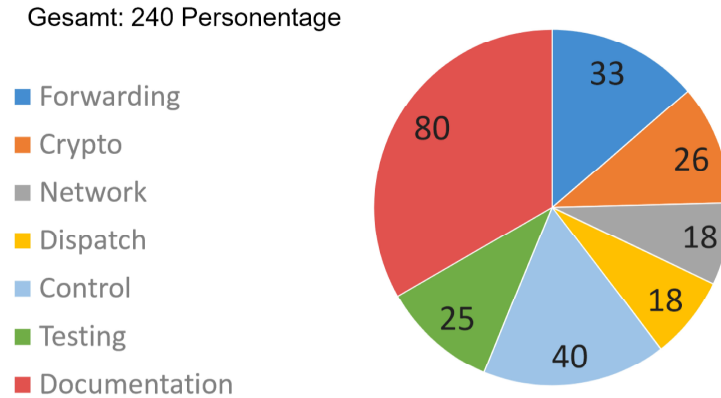


Abbildung 1: Aufwandsabschätzung

1.3 Risikoabschätzung

Bei der Risikoabschätzung wird versucht eine Vorhersage zu treffen, mit welcher Wahrscheinlichkeit bestimmte Risiken eintreffen können und wenn diese eintreffen, welchen Schaden sie im Projekt verursachen. (siehe Abbildung 2)

R1: Kommunikationsprobleme im Team

R2: Umfang zu groß

R3: Technologie(Rechner) erfüllt nicht die Anforderungen

R4: Framework bietet nicht die benötigte Funktionalität

R5: Ausfall von Mitgliedern

R6: Anforderungsänderungen durch den Auftraggeber aufgrund eines Kommunikationsproblems im Pflichtenheft

R7: falsche Umsetzungsstruktur, gewählte Struktur lässt sich nicht umsetzen

R8: Rechtliche Vorgaben

R9: versteckte Komplexität

| | | | | | | |
|------------------------|--------------|-----------|------------------|-----------|----------------|-------------|
| Auswirkung des Risikos | katastrophal | R4 | | | | |
| | wesentlich | | | R2 | | |
| | moderat | | | | R7, R9 | |
| | gering | R5 | R3 | | R1 | |
| | unwesentlich | R8 | R6 | | | |
| | | gering | unwahrscheinlich | möglich | wahrscheinlich | fast sicher |
| Eintreffen des Risikos | | | | | | |

Abbildung 2: Risikoabschätzung

1.4 Meilensteine

Erster Meilenstein: Pflichtenheft, Grobentwurf, lauffähige Implementierung (einfaches Forwarding)

Zweiter Meilenstein: lauffähige Implementierung (Verschlüsselung mit statischem Schlüssel)

Dritter Meilenstein: lauffähige Implementierung (Schlüsselaustausch, Gruppenschlüssel), Validierung

1.5 Organisation

Die Organisation betrifft alle Regelungen, Vereinbarungen und Aufteilungen der am Projekt beteiligten Personen um ein geordnetes und effizientes Arbeiten zu ermöglichen.

1.5.1 Kommunikationswege

Zulip: Dient der schnellen Gruppenkommunikation um effektiv Missverständnisse zu lösen und für eine direkte Kommunikation unter Gruppenmitgliedern.

Email-Verteilerliste: Dient zur Planung von Treffen und wird für Mitteilungen, die an alle gesendet werden, verwendet.

Gruppentreffen: Dient zur Diskussion und Besprechung von Problemen.

Phabricator: Dient der Zuweisung von Aufgaben an einzelne Gruppenmitglieder und der direkten Kommentierung.

1.5.2 Zusätzliche Vereinbarungen

- Gruppentreffen jeden Donnerstag von 9:00 Uhr - 12:30 Uhr
- Kleingruppentreffen nach Absprache und Bedarf

1.5.3 Rollenverteilung im Agilen Vorgehen

Produkt Owner: Michael Rossberg

Scrum Master: Felix Seidel, Tobias Schubert

Entwicklungsteam: Felix Seidel, Milan Haverkock, Michael Fuchs, Daniel Scheliga, Yannic Faulwetter, Nils Winkelbach, Jörn Weisensee, Tobias Schubert

Kunde, Anwender: Telematik Lehrstuhl

Management: Tobias Schubert

1.5.4 Organisatorische Rollenverteilung

Betreuer: Michael Rossberg

Teamleiter: Tobias Schubert

Code: Felix Seidel

Präsentation: Daniel Scheliga

Grafik: Nils Winkelbach

Build-Master: Milan Haverkock

Dokumentation: Michael Fuchs

Test: Jörn Weisensee

Web-Master: Yannic Faulwetter

1.5.5 Programmieraufteilung

Forwarding : Milan Haverkock, Daniel Scheliga, Nils Winkelbach

Network : Felix Seidel, Michael Fuchs, Yannic Faulwetter

Crypto : Tobias Schubert, Jörn Weisensee

Dispatch : noch keine genaue Einteilung getroffen

Control : noch keine genaue Einteilung getroffen

Testing : noch keine genaue Einteilung getroffen

2 Ergebnisse des Entwurfs

Die Ergebnisse des Entwurfs sind zum größten Teil in der Entwurfsdokumentation nachzulesen. Dort sind die Zusammenhänge der verschiedenen Pakete, Komponenten und Klassen aufgezeigt und anschaulich mit Hilfe von UML-Diagrammen dargestellt.

2.1 Werkzeuge

Die verwendeten Werkzeuge sind Softwarelösungen, die die einzelnen Bereiche der Organisation und Entwicklung ermöglichen bzw. erleichtern.

2.1.1 Organisatorische Werkzeuge

Phabrikator: Zuweisung einzelner, zu bearbeitender Tasks und deren Verwaltung, sowie die Bereitstellung des Wiki's.

Quellcodeverwaltung: Hierfür wird *Subversion* benutzt um einen einheitliche Arbeitsgrundlage an den Dateien zu gewährleisten.

Arbeitszeiterfassung: Hierfür wird *Kimai* verwendet, dort werden alle Zeiten eingetragen und verschiedenen Themen/Bereichen zugeteilt.

LaTeX: Hierbei handelt es sich um eine Textbeschreibungssprache, mit der verschiedenste Dokumente erstellt werden können. Diese stehen nach der Erstellung in verschiedenen Formaten zur Verfügung. Im speziellen wird es in diesem Projekt für die Erstellung des Pflichtenheftes, den Reviewdokumenten und dem Entwurfsdokument verwendet.

Doxygen: Mit diesem Programm werden die Kommentare aus dem erstellten Code gezogen, um daraus eine Dokumentation der Implementierung zu erzeugen.

PlantUML: Erzeugung von UML-Diagrammen, die Aktivitäten, Aufbau und Funktionen des Systems grafisch darstellen.

2.1.2 Entwicklungswerkzeuge

Buildsystem: Es wird *scons* verwendet um eine einheitliche und angepasste Kompilierung von C++-Dateien zu gewährleisten. Hierbei werden sogenannte *sconstruct* erstellt, mit denen verschiedene Vorbedingungen, wie das Einbinden von Bibliotheken und die Auswahl des Compilers erfüllt werden können.

Entwicklungsumgebung: Ein beliebiger *Texteditor*, der Syntaxhighlighting für die Programmiersprache C++ unterstützt, wird zum Verfassen des Codes verwendet.

Programmiersprache: Es wird die hardwarenahe Programmiersprache *C++* verwendet, da diese alle, für das Projekt erforderlichen, Anforderungen erfüllt.

Betriebssystem: Für den Betrieb der Software ist *Linux* unabdingbar, da das verwendete DPDK-Framework nur auf dessen Grundlage funktioniert.

Bibliotheken: In diesem Projekt werden viele verschiedene Standardbibliotheken von C++ und vom Auftraggeber bereitgestellte Bibliotheken verwendet. Zusätzlich hierzu wird als spezielles Framework (Ansammlung von Bibliotheken) das DPDK genutzt. Dieses liefert mit seiner umfangreichen Klassensammlung die Grundlage (Verschlüsselung, Forwarding, ...) für das System.

2.2 Ergebnisse des Entwurfs für die erste Iteration:

Die Ergebnisse der ersten Iteration ergeben sich aus dem aktuell zu erreichenden Meilenstein.

Pflichtenheft: Das Lastenheft wurde vollständig in das Pflichtenheft überführt und um weitere Punkte, im Dialog mit dem Auftraggeber, ergänzt.

Grobentwurf: Der Grobentwurf umfasst eine erste Übersicht über die Funktions- und Arbeitsweise des Systems.

Implementierung: Eine erste Lauffähige Implementierung wurde umgesetzt und umfasst die Funktionen eines einfachen Forwardings ohne Verschlüsselung.

Planung: Es wurden Festlegungen für die weiteren Iterationen getroffen, hierbei wurden insbesondere die Meilensteine für die nächste Iteration festgelegt und das weitere Vorgehen innerhalb des Teams besprochen.

2.3 Festlegungen für die nächste Iteration:

Die Festlegungen für die jeweilige nächste Iteration spiegeln sich in den Meilensteinen wieder.

Grobentwurf verfeinern: Der Grobentwurf wird um spezifischere Diagramme und Beschreibungen zum Feinentwurf ergänzt. Dies geschieht jeweils parallel zu den Meilensteinen, da automatisch jede Änderung dort festgehalten wird.

Implementierung erweitern: Die aktuelle Implementierung wird um einen ersten Entwurf der Verschlüsselung mit statischem Schlüssel erweitert.

Hardware konfigurieren: Die zur Verfügung gestellte Hardware wird initial konfiguriert um die Voraussetzungen für das System zu gewährleisten und um die Implementierung ersten Tests unterziehen zu können.