HAW HAMBURG

Bachelorarbeit

Marvin Herrmann

# Konzipierung und Implementierung einer Erweiterung zum Erkennen von Mails mit gefälschtem Absender für den Mail User Agent Thunderbird

*Fakultät Technik und Informatik*
*Department Informatik*

*Faculty of Computer Science and Engineering*
*Department Computer Science*

Marvin Herrmann

# Konzipierung und Implementierung einer Erweiterung zum Erkennen von Mails mit gefälschtem Absender für den Mail User Agent Thunderbird

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung
im Studiengang Bachelor of Science Angewandte Informatik
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Dr. Klaus Peter Kossakowski
Zweitgutachter: Prof. Dr. Olaf Zukunft

Eingereicht am: 30. Juli 2020

**Marvin Herrmann**

**Thema der Arbeit**

Konzipierung und Implementierung einer Erweiterung zum Erkennen von Mails mit gefälschtem Absender für den Mail User Agent Thunderbird

**Stichworte**

Spamerkennung, Thunderbird, E-Mail, Spoofing, Spam

**Kurzzusammenfassung**

In zunehmenden Maße werden in betrügerischer Absicht Spam-E-Mails unter Angabe eines scheinbar bekannten Absenders versendet. Ein wachsender Teil davon ist das so genannte Spoofing. Spoofing, also das Ausgeben einer Identität, welche dem Empfänger bekannt ist, ist mittlerweile ein fester Bestandteil des Spams. Diese Bachelorarbeit befasst sich damit, eine Lösung zu konzipieren und entwickeln, die es Nutzern ermöglicht, eine Mail auf Spoofing zu überprüfen und dadurch die Spoofing Gefahr zu verringern.

**Marvin Herrmann**

**Title of Thesis**

Design and implementation of an extension to recognize mails with forged sender address for the mail user agent Thunderbird

**Keywords**

Spam detection, Thunderbird, E-Mail, Spoofing, Spam

**Abstract**

Spam is almost not to avoid in the communications through email. Spoofing, to assume an identity known to the recipient of a mail, is now an inherent part of Spam. This bachelor thesis will design a way to make it possible for users to check a mail for spoofing and therefore reducing the risk of spoofing.

# Inhaltsverzeichnis

# Abbildungsverzeichnis

# 1 Einleitung

Die Kommunikation über E-Mail ist in der heutigen Zeit nicht mehr wegzudenken. In dem privaten wie in dem geschäftlichen Bereich ist die E-Mail zu einem festen Bestandteil der zwischenmenschlichen Kommunikation geworden. Leider hat jede Technologie aber neben den Vorteilen des schnellen Informationsaustausches auch ihre ungewollten negativen Aspekte. Im Fall der E-Mail ist es das überflüssige Empfangen von ungewollten E-Mails (Spam), deren Existenzberechtigung allein darin liegt, Schaden anzurichten. Es gibt permanent neue Möglichkeiten, wie man Spam filtern und eliminieren kann. Gleichzeitig werden Mittel und Wege gesucht, das Versenden von Spam durch umgehen anderer Methoden zu ermöglichen. Eine der neueren Methoden ist das E-Mail Spoofing. Diese Art des Spams setzt darauf, eine vertraute Person oder einen Dienstleistungsservice vorzutäuschen, um an private Informationen zu kommen. „Untersuchungen ergeben, dass 60% bis 90% aller E-Mails Spam sind. Spam kostet den Privatanwender Zeit und Nerven, mittelständische Unternehmen Zehntausende, große InternetProvider viele Millionen Euro pro Jahr.“[17] Spam kann in mehrere Kategorien eingeteilt werden, jedoch wird in dieser Arbeit nur das sogenannte Spoofing betrachtet.
Die Idee für dieses Konzept entstand auf der jährlichen Sicherheitskonferenz des DFN-CERT in Hamburg bei einer Diskussion mit mehreren Fachleuten. Eine wirkungsvolle Gegenmaßnahme würde es zur Zeit nicht geben, jedoch mehrere unterschiedliche Ideen, wie man solche E-Mails filtern könnte. Eine gute Umsetzung dieser Ansätze könnte Firmen und Privatpersonen Probleme oder sogar teure wirtschaftliche Schäden ersparen. In dieser Arbeit werden mehrere Konzepte vorgestellt, welche solche Angriffe erkennen und das Risiko eines Schadens durch Spoofing verringern können. Diese werden dann als Erweiterung für Thunderbird implementiert.

## 1.1 Motivation

Vermutlich jeder kennt die folgende Situation: Es ist Montagmorgen, das E-Mailpostfach wird geöffnet und eine große Anzahl an E-Mails ist eingegangen. Ein Großteil der Nachrichten ist von Mitarbeitern, Kollegen oder Freunden. Jedoch beinhalten einige dieser E-Mails merkwürdige Anfragen und Hyperlinks, die Ihnen fragwürdig vorkommen. Sie sind sich unsicher über den Ursprung und überprüfen daher von welcher E-Mail-Adresse die Nachricht versendet worden ist. Da diese mit der Ihres Kollegen oder Bekannten übereinstimmt, klicken Sie sorglos auf den Link oder befolgen die Anweisung in der E-Mail. Was Sie jedoch nicht wussten, ist, dass der wirkliche Versender der E-Mail die angezeigte Absenderadresse manipuliert hat, um Sie in Sicherheit zu wiegen. Da es immer schwieriger wird, diese betrügerischen E-Mails zu erkennen, steigt somit auch die ausgehende Gefahr eines Schadens durch Spoofing ständig an.

## 1.2 Ziel und Abgrenzung

Das Ziel dieser Arbeit ist es, für den Mail User Agent Thunderbird eine Erweiterung zu entwickeln, die dem Benutzer bei Verdacht auf E-Mail-Spoofing eine visuelle Rückmeldung gibt. Dadurch kann der Nutzer auch bei gut gespooften E-Mails einen Hinweis erhalten, um diese E-Mail genauer zu überprüfen. Durch die visuelle Rückmeldung des Spoofing-Versuches soll die ausgehende Gefahr durch Spoofing für die Nutzer reduziert werden.
Es ist nicht Ziel dieser Arbeit den Besitzer der gespooften E-Mail-Adresse oder den Betreiber der Domain über den Missbrauch zu informieren. Daher wird das Protokoll DMARC, welches den Inhaber der gespooften E-Mail-Adresse bei Missbrauch benachrichtigen kann, in der Erweiterung nicht beachtet. Auch das Protokoll DKIM, welches mithilfe von einer Signatur im Header die Integrität schützen soll, wird nicht eingesetzt. Dieses hat den Grund, dass zu viele Abfragen an unterschiedliche E-Mail-Server die Zeit erhöhen, welche die Erweiterung benötigt um eine E-Mail zu überprüfen. Das Entdecken von E-Mails, welche im Transfer geändert oder manipuliert worden sind, ist nicht Bestandteil dieser Arbeit.

## 1.3 Zielgruppe der Arbeit

Diese Arbeit ist für Leser gedacht, die an diesem Thema interessiert sind oder den Mail User Agent Thunderbird nutzen und ein gewisses Maß an Grundwissen über Informatik besitzen. Die resultierende Erweiterung ist sowohl für jede private Person als auch Firmenangestellte nützlich, die den Mail User Agent Thunderbird aktiv nutzen.

## 1.4 Verwandte Arbeiten

**Developing a Model to Detect E-mail Address Spoofing using Biometrics Technique - A.S. Zadgaonkar, S. Kashyap, M. C. Patel**
Dieses Model wurde 2013 von den drei Entwicklern im International Journal of Science and Modern Engineering vorgeschlagen um den wachsenden Spam zu bekämpfen. Es beschreibt, wie die Spammer Schwachstellen des E-Mail Protokolls SMTP ausnutzen und schlagen als Gegenmaßnahme vor, eine Biometrische Komponente einzubauen. Sie gehen darauf ein, wie ein solches System mit Hilfe eines Fingerabdrucksensors umgesetzt werden könnte.[21]

**Malwareanalyse mit Cuckoo - Manuel Selmeier**
Diese Bachelorarbeit befasste sich mit dem Bewerten von Hyperlinks oder Dateien, welche per E-Mail bei den Nutzern eingegangen sind. Die Nutzer mussten dafür die E-Mail mit den fragwürdigen Inhalten an ein Postfach weiterleiten. Diese Hyperlinks oder Dateien wurden dann in einer Kontrollumgebung geöffnet und anschließend je nach Verhalten bewertet. Die Bewertung wurde dann per E-Mail an die weiterleitende Person zurück geschickt.[16]

**Analyse und Erkennung von Phishing-E-Mails mit Delphish – Robert Krzeminski**
Delphish ist eine Erweiterung für Microsoft Outlook, welches zur Erkennung von Phishing-E-Mails genutzt wird. Der Kern dieser Arbeit besteht darin, eine graphische Darstellung von dem Risiko einer E-Mail und der darin befindlichen Links zu geben, sowie eine Anzeige der Ergebnisse von Reputationstests und anderer relevanter Informationen, wie z.B. WHOIS-Datensätze.[6]

**SMTP Path Analysis – Barry Leiba, Joel Ossher, V.T. Rajan, Richard Segal, Mark Wegman**
Diese Arbeit klassifiziert Spam anhand eines Lern-Algorithmus, der den Pfad der E-Mail untersucht und somit die Aufgaben von Domain-Authentifikations Systemen, Blacklist- und Whitelistdiensten übernehmen kann.[7]

## 1.5 Struktur der Arbeit

E-Mail-Spam und die Unterkategorie E-Mail-Spoofing sind auch nach mehreren Jahrzehnten Nutzung noch ein aktuelles Thema in der digitalen Welt. Nach dem einleitenden Abschnitt, werden im zweiten Kapitel Grundlagen erörtert, um einen möglichst gleichen Wissensstand für Leser zu ermöglichen. Dieses beinhaltet beispielsweise, welche Informationen in einer E-Mail versendet werden, was für Spam-Arten es gibt und Begriffe die im Verlauf dieser Arbeit genutzt werden.
Das dritte Kapitel befasst sich mit der Konzeptidee und dem Erstellen einer Architektur für die Thunderbird Erweiterung. Insbesondere werden hier die einzelnen Tests erläutert, welche am Ende in der Erweiterung eingebaut werden. Das Kapitel widmet sich einer Definierung eines klaren Ablaufes für die Erweiterung. Eventuelle Implementierungsfehler würden dadurch frühzeitig entdeckt und vermieden werden können.
In dem folgenden vierten Kapitel wird der theoretische Ansatz aus dem vorherigen Kapitel umgesetzt und eingebaut. Dabei wird aufgezeigt, welche Probleme bei der Implementation entstanden sind und ob von der originalen Architektur abgewichen werden musste.
Im fünften und letzten Kapitel wird das Fazit dieser Arbeit gezogen und ein Ausblick auf mögliche Fortführungen und Ergänzungen der Erweiterung gegeben. Unter dem Titel der Würdigungen wird eine gravierende Änderung durch aktuelle Gesetzgebungen und deren Auswirkungen erläutert.

# 2 Grundlagen

## 2.1 Begriffserklärung

### 2.1.1 Spam

„Unerwünscht zugesandte E-Mails werden generell als Spam bezeichnet. Dieser lässt sich grob in drei Formen unterteilen:“[4]

#### 2.1.1.1 Klassischer Spam

„Der Begriff Spam bezeichnet unverlangt zugesandte Massen-E-Mail. Unverlangt ist eine E-Mail dann, wenn das Einverständnis des Empfängers zum Empfang der Nachricht nicht vorliegt und nicht zu erwarten ist. Massen-E-Mail bedeutet, dass der Empfänger die Nachricht nur als einer von vielen erhält. Auf englisch bezeichnet man das als UBE (Unsolicited Bulk Email, Unverlangte Massenmail). Es reicht nicht, dass eine E-Mail unverlangt oder Massenmail ist. Zur Definition von Spam gehören beide Aspekte: Unverlangte, aber persönliche E-Mail (z. B. von einem lange aus den Augen verlorenen Schulkameraden) ist nicht als Spam einzustufen, wie auch Massenmail dann kein Spam ist, wenn man mit dem Empfang einverstanden ist (z. B. bei einem explizit abonnierten Newsletter).
Ein Newsletter, den ein Empfänger abonniert hat, wird nicht zu unverlangter E-Mail, nur weil der Empfänger es sich inzwischen anders überlegt hat und er den Newsletter nicht mehr erhalten möchte. Erst wenn der Empfänger seinen Wunsch, den Newsletter nicht mehr zu erhalten, gegenüber dem Absender ausgedrückt hat, wird der Newsletter bei weiterem Versand zu Spam.“[17]

#### 2.1.1.2 Schadprogramm-Spam (Malware-Spam)

"Mit Schadprogramm-Spam (Malware-Spam) wollen Angreifer Systeme der Empfänger mit Schadprogrammen infizieren. Dies kann direkt durch ein Schadprogramm im E-Mail-Anhang oder indirekt durch einen Link im E-Mail-Text oder im Anhang erfolgen, der auf ein Schadprogramm oder eine Webseite mit Drive-byExploits verweist."[4]

#### 2.1.1.3 Phishing

Phishing ist ein skalierbarer Betrug, bei dem durch Vortäuschung einer Identität versucht wird, sensible Informationen von dem Ziel des Angriffes zu erhalten.[1] Das Wort setzt sich aus password und fishing zusammen, also das Angeln nach Passwörtern. Meistens als Dienstleister ausgebend, wird der Empfänger mit einem passenden Grund dazu aufgefordert, seine persönlichen Daten zu aktualisieren. Auf der verlinkten Seite sieht das Opfer dann eine gute Kopie der echten Webseite, welche unter der Kontrolle der Spammer ist.

### 2.1.2 Spoofing

Übersetzt bedeutet Spoofing Manipulation, Verschleierung oder Täuschung. Es gibt unterschiedliche Arten des Spoofing in dem Bezug auf die Informatik. „Als Spoofing bezeichnet man im Kontext der Netzwerk- und Computersicherheit das Vortäuschen einer fremden Identität."[14]

#### 2.1.2.1 GPS-Spoofing

Das GPS-Spoofing befasst sich damit, die Standorterkennung zu überlisten oder direkt zu umgehen, um dem System einen anderen Standort vorzutäuschen. Im Gegensatz zu GPS-Jammern erzeugen und übertragen GPS-Spoofer formal gültige, jedoch falsche Positionsdaten. Der GPS-Jammer hingegen stört den GPS-Empfang oder kann diesen sogar zum Ausfall bringen.

[1] Frei übersetzt aus dem Artikel „Achieving a consensual definition of phishing based on a systematic review of the literature" von Dr.ir. Elmer Lastdrager

#### 2.1.2.2 Biometrisches-Spoofing

Das biometrische Spoofing ist darauf ausgerichtet, eingebaute Sicherheitsmaßnahmen wie Gesichtserkennung oder den Fingerabdrucksensor auszutricksen. Die meisten Methoden dieser Spoofing-Art basieren darauf, Gesichtszüge, Fingerabdrücke oder sogar die Venenanordnung zu fälschen. Dadurch erkennen die Sicherheitsmaßnahmen den Nutzer, welcher aber nicht derjenige ist, der den Zugang verlangt.

#### 2.1.2.3 E-Mail-Spoofing

Bei E-Mail-Spoofing werden in den Headern die Informationen so verändert, dass als Absender eine vertraute Person, Dienstleister oder E-Mail Adresse eines Bekannten angezeigt wird. Dieses wird bei dem Absenden der Nachricht festgelegt und daher brauchen die Betrüger keinen Zugriff auf die Komponenten. Jeder Mail Transfer Agent leitet die an ihn gesendete Nachricht, meist ohne Überprüfung, weiter zum angegebenen Ziel. Dadurch kann der Ersteller von Spoofing E-Mails sich mit vorgefertigten E-Mail Headern einklinken und somit vortäuschen, dass er selber nur weiterleiten würde.[2]

#### 2.1.2.4 IP-Spoofing

Das Angeben einer anderen IP als Quelle einer Anfrage, auch als IP-Spoofing bezeichnet, wird oft für Distributed Denial of Service (DDoS) Angriffe genutzt. Der Grund dafür ist, dass bei solchen Angriffen mit Hilfe des IP-Spoofings der Ursprung der Anfragen verschleiert werden kann. Dadurch sind Anfragen, die Teil des Angriffes sind, nicht zu unterscheiden von normalen Anfragen.

#### 2.1.2.5 ARP-Spoofing

Das ARP-Spoofing ist eine Möglichkeit, einen Man-in-the-Middle Angriff in einem lokalen Netzwerk auszuüben. Dabei werden die entstandenen Adress Resolution Protocol (ARP) Tabellen durch ein vermeintliches Update manipuliert. Der Datenverkehr zwischen zwei Hosts wird dadurch über einen dritten Punkt umgeleitet und kann abgehört oder auch

[2] Darstellung des Abschnitts auf Basis der Publikation [11, Email Authentication Mechanisms: DMARC, SPF and DKIM] in freier Übersetzung.

manipuliert werden. [3]
Bis auf E-Mail-Spoofing und IP-Spoofing benötigen alle aufgelisteten Spoofing-Arten Zugriff auf Komponenten des betroffenen Systems. Wenn in dieser Arbeit sich auf Spoofing bezogen wird, ist damit das E-Mail-Spoofing gemeint. Zu beachten ist, dass Spoofing bei jeder der drei in 2.1.1 genannten Formen des Spams auftreten kann.

### 2.1.3 Header

Wenn in dieser Arbeit ein „Header“ erwähnt wird, ist damit die Information im Kopf einer E-Mail gemeint. Diese Felder haben unterschiedliche Informationen. Die wichtigsten und bekanntesten sind „Received“, „from“ und „to“. Diese Informationen beinhalten von welchem Absender an welchen Empfänger die E-Mail gesendet worden ist und welche Mail Transfer Agenten diese E-Mail weitergeleitet haben.

### 2.1.4 SpamAssassin

„Beim SpamAssassin-System handelt es sich um eine Software zur Analyse von E-Mail-Nachrichten, die feststellt, wie wahrscheinlich es ist, dass es sich dabei um Spam handelt, und die über ihre Ergebnisse Bericht erstattet. Es ist ein regelbasiertes System, das verschiedene Bestandteile der E-Mail-Nachricht mit einem großem Satz an Regeln vergleicht. Jede Regel fügt an dem Spam-Punktestand einer Nachricht Punkte hinzu oder entfernt diese davon. Eine Nachricht mit entsprechend hohem Punktestand wird als Spam gemeldet.“[15] Unterschiedliche Methoden, wie zum Beispiel Heuristiken und statistische Analysen, überprüfen die E-Mail-Header und den Text des Bodies. Dadurch wird eine hohe Erkennungsrate erlangt. Aus diesem Grund ist es ein sehr beliebtes Werkzeug für E-Mail-Administratoren und durch die Möglichkeit der kostenlosen Nutzung eines der meistgenutzten und weitverbreitetsten Spamfilterprogramme.

### 2.1.5 SMTP

SMTP ist die Abkürzung für Simple Mail Transfer Protokoll. Es definiert, wie E-Mails in Computernetzen ausgetauscht werden. Wichtig ist hierbei, dass SMTP nur für die Übertragung zuständig ist. Für die Verarbeitung und das Abrufen sind andere Protokolle, wie

[3] Darstellung des Absatzes frei übersetzt nach [19, An Introduction to ARP Spoofing]

zum Beispiel POP3 und IMAP zuständig. Das Senden einer E-Mail kann entweder in einer einzelnen Verbindung zwischen Sender und Empfänger geschehen oder durch mehrere Verbindungen über zwischengeschaltete Systeme.
In beiden Fällen erfolgt eine formelle Übergabe der Verantwortung für die Nachricht, sobald der Server am Ende der E-Mail Daten eine Bestätigung („success response“) ausgegeben hat. Ein Server, der die Verantwortung hat, ist dafür zuständig, entweder die E-Mail weiterzugeben, oder den entsprechenden Fehler zurückzusenden. Jeder Server, der die E-Mail weiterleitet, erweitert den Received-Header um eine Zeile und fügt sie vor den anderen Received-Zeilen ein. Das bedeutet, dass die Received-Zeilen wie ein Stack aufgebaut sind. Das letzte Element in diesem Stack wird von dem eigenen E-Mail Anbieter eingefügt.[4]
Es ist mittlerweile einfach geworden, E-Mails mit Hilfe eines Scripts über SMTP zu verschicken. Es gibt Anleitungen und Beispiele im Internet, die einem erklären, wie man mit kleinen Veränderungen Newsletter oder Servicenachrichten per Script versenden kann. Dieses machen aber nicht nur Internetportale und Versender von Newslettern, sondern auch Spammer und Spoofer, welche mit Hilfe von editierten Scripts mehrere Millionen E-Mails am Tag verschicken.

### 2.1.6 SPF

Das Sender Policy Framework wurde 2006 als experimentelles Protokoll vorgestellt. Es ist ein Domain Name Server Resource Record, welcher spezifiziert, welche Hosts autorisiert sind, den Domainnamen für das „HELO“ und „MAIL FROM“ einer E-Mail zu nutzen und welche nicht. Sollte ein E-Mailserver dieses Protokoll einsetzen, wird in den E-Mails der Header „Received-SPF“ eingefügt. Dieser kann insgesamt die folgenden sieben unterschiedlichen Ergebnisse beinhalten.

- „None“ - Die Domain hat keinen Resource Record veröffentlicht oder die Sendedomain der Identität konnte nicht ermittelt werden.
- „Neutral“ - Der Domaininhaber hat angegeben, dass er nicht sagen kann, ob eine IP autorisiert ist, oder es einfach nicht kann. In beiden Fällen wird Neutral so behandelt wie „None“.
- „Pass“ - Der Client ist autorisiert, E-Mails von der Domain mit der Identität die er angibt zu senden.

[4] Darstellung des Absatzes frei übersetzt nach [5, RFC 5321]

- „Fail“ - Dieses Resultat ist eine eindeutige Erklärung, dass der Client nicht dazu berechtigt ist, E-Mails mit dieser Identität von der Domain zu senden.
- „SoftFail“ - Ein SoftFail liegt zwischen Fail und Neutral. Die Domain glaubt, dass der Host nicht autorisiert ist, möchte aber in diesem Fall kein Fail eintragen. Diese E-Mail sollte nicht verworfen, sondern genauer überprüft werden.
- „TempError“ - Während des Prüfens tritt ein Fehler auf der verhindert, dass ein Ergebnis eingetragen werden kann.
- „PermError“ - Ein PermError entsteht dann, wenn der veröffentlichte Resource Record nicht korrekt interpretiert werden kann. Dieses kann an einer falschen Formatierung des Veröffentlichten oder der einzelnen Identität liegen.

Die Umsetzungsrate des Sender Policy Framework ist relativ gering, weswegen der Großteil der Mail Transfer Agenten diesen Header nicht überprüfen oder sogar ignorieren.[5]

### 2.1.7 DKIM

„Bei DKIM (DomainKeys Identified Mail,[2]) wiederum signiert der für eine Mail verantwortliche Mailserver eine ausgehende Mail kryptographisch, wobei die Signatur mit einem im DNS hinterlegten Public Key verifiziert werden kann. Die Signatur schließt dabei wesentliche Bestandteile des Mail-Headers wie auch den Inhalt der Mail ein und ermöglicht dadurch zu erkennen, wenn eine Mail signifikant modifiziert wurde.“[18]

### 2.1.8 DMARC

„Sowohl SPF wie auch DKIM betrachten dabei nicht den im Mail-Client dargestellten Absender: bei SPF wird der beim Mailtransport über SMTP angegebenen Absender (SMTP.MAILFROM) benutzt, bei DKIM hingegen die mit der Signatur verbundene Domain. Erst mit DMARC (Domain-based Message Authentication, Reporting and Conformance, [3]) wird überprüft, ob die von SPF bzw. DKIM gelieferte Domain zu dem im Mail-Client sichtbaren Absender aus dem Mail-Header (RFC822.From) passt. Dabei ist es ausreichend, dass entweder ein erfolgreiches SPF oder eine der gültigen DKIM-Signaturen die passende Domain haben. Gibt es keine Übereinstimmung, so wird je nach der im DNS hinterlegten DMARC-Policy die Mail abgelehnt, in Quarantäne verschoben

[5] Frei übersetzt nach [20, RFC 4408]

oder trotzdem angenommen. Zusätzlich kann in der Policy auch ein Reporting erbeten werden, über welches Domaininhaber Verletzungen der DMARC-Policy detailliert nachvollziehen können."[18]

### 2.1.9 WhoIs

WhoIs ist ein in 1982 erstelltes und in 1985 sowie 2004 verbessertes Protokoll der Internet Engineering Task Force. Mit diesem Protokoll ist es möglich, den oder die Inhaber oder die IP-Adresse einer jeweiligen Domain zu erfragen. Das endgültige Protokoll wurde im RFC 3912 festgelegt und standardisiert.[6]

### 2.1.10 RDAP

Das Registry Data Access Protokoll (RDAP) ist ein von der Internet Engineering Task Force neu entwickeltes Protokoll, welches das WhoIs Protokoll ablösen soll. Es wurde im März 2015 im RFC 7483 als Standard vorgestellt.[7]

### 2.1.11 Promise (-Programming)

Ein Promise ist ein Objekt, welches als Platzhalter für einen Wert dient. Dieser Wert ist normalerweise ein Resultat einer asynchronen Operation wie einem HTTP-Request oder das lesen einer Datei von der Festplatte. Wenn eine asynchrone Funktion aufgerufen wird, wird sofort ein Promise-Objekt zurückgegeben. Auf diesem Objekt können dann Rückruffunktionen registriert werden, die erst ausgeführt werden, wenn die Operation erfolgreich war oder einen Fehler aufgetreten ist.[8]
Ein Promise kann nur einen von drei Zuständen haben. „pending" - der Ausgangszustand, weder erfüllt noch verworfen. „fulfilled" - die Operation wurde erfolgreich durchgeführt. „rejected" - die Operation ist fehlgeschlagen. Wenn ein Promise von „pending" auf entweder „fulfilled" mit einem Wert oder „rejected" mit einem Fehler übergeht, werden die Rückruffunktionen aufgerufen.

[6] Frei übersetzt nach [1, RFC 3912]
[7] Frei übersetzt nach [10, RFC 7483]
[8] Frei Übersetzt nach [12, JavaScript with Promises: Managing Asynchronous Code]

## 2.2 IP Vergabe und Verwaltung

Um die Verwaltung und Verteilung von IP Adressen zu vereinfachen, wurde 1992 im RFC 1366 festgelegt, dass IP Adressen in Zukunft durch regionale Organisationen kontinental vergeben werden sollen. Dieser RFC wurde in 1993 und 1996 überholt, bis dieser schließlich im August 2013 von dem RFC 7020 abgelöst worden ist. Zur Zeit existieren fünf regionale Organisationen, die für die jeweiligen Kontinente zuständig sind:

- AfriNIC, zuständig für Afrika
- APNIC, zuständig für Asien und die pazifische Region
- ARIN, zuständig für Nordamerika und Teile der Karibik
- RIPE NCC, zuständig für Europa, Teile von Asien und den Mittleren Osten
- LACNIC, zuständig für Latein Amerika und Teile der Karibik

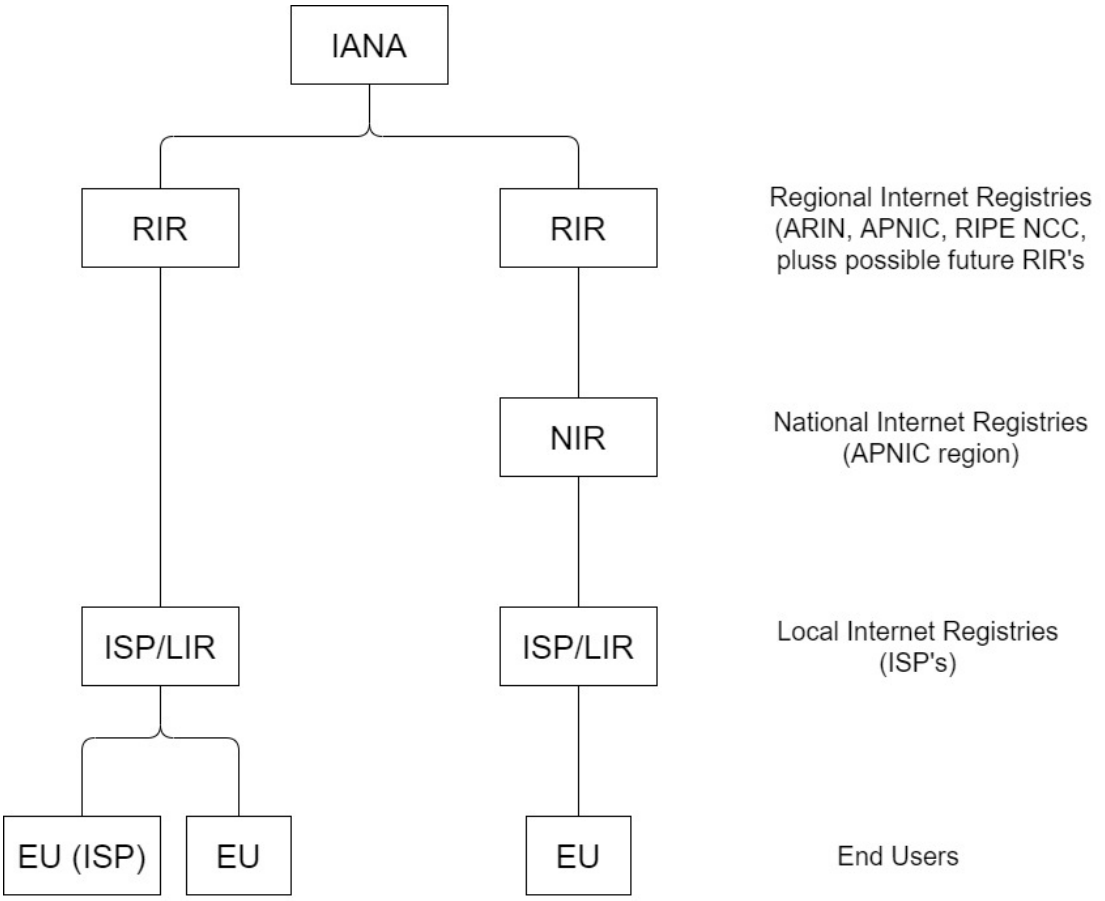

Quelle: [8]

Abbildung 2.1: Aufteilung der Regional Internet Registries.

Mit dem Erstellen der LACNIC wurde 2001 die Kontrolle der zu vergebenden IP Adressen von Süd Amerika und Teilen der Karibik von ARIN zu LACNIC übergeben. Ähnliches geschah 2004 durch die Entstehung der AfriNIC, welche die Verwaltung von RIPE NCC

und APNIC übertragen bekam. Den regionalen Registraturen unterstellt sind die nationalen Registraturen, welche hauptsächlich im asiatischen Raum, also verwaltet von APNIC, im Einsatz sind.

Die fünf regionalen Registraturen sind jedoch der IANA unterstellt. IANA ist die Abkürzung für Internet Assigned Numbers Authority. Diese vergibt selber keine IP Adressen, sondern veröffentlicht nur den Bestand der IP Adressblöcke, welche jede Regional Internet Registry besitzt. Das tut sie aus Gründen der Koordination zwischen den Registraturen. Ein weiterer wichtiger Dienst, den IANA ausübt, ist das Verwalten und Pflegen des DNS-Root Servers und der .int und .arpa Domains.[9]




Quelle: [9]

Abbildung 2.2: Darstellung der IP Verwaltung und Vergabe.

[9] Frei Übersetzt nach [3, Selbstdarstellung von IANA]

Wie in Abbildung 2.2 zu erkennen ist gibt es eine klare Hierarchie für die Vergabe der IP-Adressen. Die Regional Internet Registry, welche die von ihr verwalteten IP-Adressen entweder an nationale Registratoren oder direkt an Internet Service Provider vergibt. Diese wiederum vergeben die erhaltenen Blöcke an Endnutzer oder geben die Adressen an andere Internet Service Provider weiter.

## 2.3 Grundsätzlicher Aufbau von E-Mails

Die E-Mail wird im RFC 5322 in zwei Teile, den Header und Body, unterteilt. Der Header enthält Informationen über den Absender und Sendeverlauf. Der Body hingegen enthält die eigentliche Nachricht. Der E-Mail Header wird in kleine Felder unterteilt, die Informationen enthalten. Wie häufig ein Feld vorkommen kann, ist im RFC 5322, Absatz 3.6. Field Definitions, spezifiziert. Zum Beispiel darf das Feld „subject“ im Header höchstens einmal auftreten, während es für das Feld „received“ keine Begrenzung gibt. Daher kann das Feld so oft wie nötig vorkommen. Im RFC 5322 Absatz 3.6 ist definiert, dass nur die beiden Felder „orig-date“ und „from“ Pflichtfelder sind. Alle anderen Felder im Header sind daher optional und müssen nicht auftauchen.[10] Eine E-Mail ohne Body ist ebenfalls möglich, da der Body optional ist. Das bedeutet: Die kleinstmögliche E-Mail wäre eine Nachricht nur mit dem Header, welcher die beiden Felder „orig-date“ und „from“ beinhaltet, aber keinen Body.

[10] Frei übersetzt nach [13, RFC 5322]

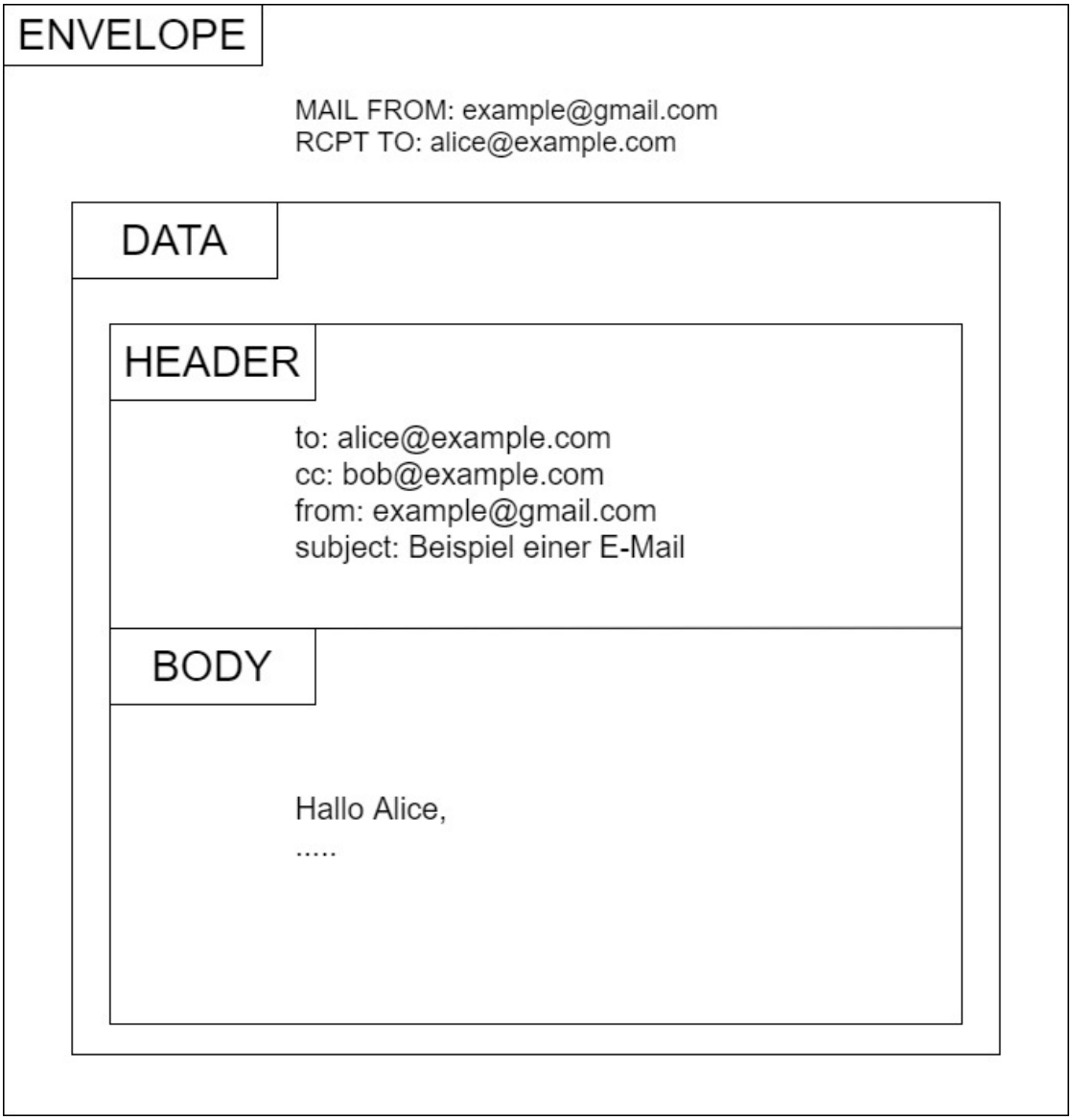



Abbildung 2.3: Konstrukt einer E-Mail

Umhüllt wird dann dieses Konstrukt von einem „Envelope“ also einem Umschlag. Dieser „Envelope“ beinhaltet nur Informationen, die zum Versenden benötigt werden. Diese bestehen aus dem Aufbau der Verbindung zwischen den zwei Mail Transfer Agenten, „MAIL FROM“ und „RCPT TO“. „MAIL FROM“ repräsentiert dabei die E-Mail Adresse des Absenders. Sollte es nicht möglich sein die E-Mail zuzustellen, wird an die Absenderadresse eine Rückmeldung der Unzustellbarkeit gesendet. Es ist jedoch einfach, die In-

formation, welche in „MAIL FROM“ hinterlegt ist, zu fälschen. „RCPT TO“ hingegen enthält die Information, an wen die E-Mail zugestellt werden soll.

## 2.4 Technischer Aufbau einer Received-Header-Zeile

Da sich diese Arbeit besonders mit dem Received-Header auseinandersetzt, wird an dieser Stelle ein Beispiel einer zuletzt eingetragenen Received-Zeile aus einem E-Mail Header genauer erläutert. Dieser Header kann mehrfach in der E-Mail vorkommen. Die Reihenfolge ist dabei invertiert. Der Unterste ist dementsprechend der Header, der zuerst und der Oberste derjenige, der als letztes eingefügt worden ist.

```
Received: from smtprelay02.ispgateway.de ([80.67.31.40]) by mx-ha.gmx.net
(mxgmx114 [212..227.17.5]) with ESMTPS (Nemesis) id 1MNuro-1j2oSP2Yem-00OJYm
for <beispiel@gmx.de>; Wed, 15 Jan 2020 16:38:57 +0100
```

Abbildung 2.4: Vollständige Received-Zeile einer E-Mail

In Abbildung 2.4 ist zu sehen, wie eine vollständige Received-Header-Zeile einer E-Mail aussieht. Zur übersichtlicheren Darstellung, wird dieser wiederum in Einzelteile zerlegt und erklärt.

```
Received: from smtp02.ispgateway.de ([80.67.31.40])
```

Abbildung 2.5: Erster Teil der Received-Zeile, der einliefernde Mailserver

Obwohl es einen definierten Standard von der Internet Engineering Task Force gibt, ist dieser Teil der Zeile in Abbildung 2.5 nicht immer einheitlich. Der formale Aufbau dieses Teils ist folgendermaßen definiert:

smtprelay.example.de (smtprelay.example.de [11.22.33.44])

Grundsätzlich gilt, dass in den eckigen Klammern die einzigartige IP-Adresse des einliefernden Rechners steht, hier in Blau dargestellt. Die Information, wie der einliefernde Rechner sich bei dem Verbindungsaufbau vorgestellt hat(auch HELO genannt), hier dargestellt in Rot, wird zusammen mit der IP-Adresse in runden Klammern eingetragen. Der Vorstellungsname ist leicht fälschbar, weshalb viele der empfangenden Server oder Transfer Agenten eine DNS-Abfrage auf die einliefernde IP-Adresse stellen. Das Ergebnis aus dieser Abfrage wird dann vor der runden Klammer, hier in Grün, festgehalten.

Sollten die HELO-Informationen und das Ergebnis der DNS-Abfrage übereinstimmen, so kann es sein, dass die HELO-Information einfach weggelassen wird. Dann besteht die Zeile nur aus dem Namen des einliefernden Servers und der IP-Adresse. Jedoch überprüfen nicht alle Mail Transfer Agenten die eingehende IP-Adresse. In diesem Fall wird der möglicherweise gefälschte Name des einliefernden Servers und die IP-Adresse eingetragen und die E-Mail weitergeleitet. Auch das Festhalten des Ergebnisses ist nicht einheitlich. Möglicherweise kommen die HELO-Information und das DNS-Ergebnis vertauscht vor. Nicht zu vergessen: Es gibt noch wenige alte Mail Transfer Agenten, die außer den leicht zu fälschenden HELO-Angaben keine weiteren Informationen eintragen und die E-Mail weitersenden.

In Abbildung 2.5 ist zu erkennen, dass bei Übereinstimmung von DNS-Ergebnis und der IP-Adresse der HELO-Wert entfällt. Der einliefernde E-Mail Server stellt sich mit smtprelay02.ispgateway.de vor. Da dieser Name zu der IP-Adresse (80.67.31.40) gehört, wird die Information aus dem HELO nicht nochmal innerhalb der runden Klammern aufgeführt.

by mx-ha.gmx.net (mxgmx114 [212..227.17.5]) with ESMTPS (Nemesis)
id 1MNuro-1j2oSP2Yem-00OJYm

Abbildung 2.6: Zweiter Teil der Received-Zeile, der empfangende Mailserver

Der eigene Mailserver des Empfängers, in diesem Fall mx-ha.gmx.net, hat die E-Mail entgegengenommen. Dieser wiederum trägt ein, wie die E-Mail empfangen wurde und

vergibt einen internen Schlüssel als Identifikationsnummer. Diese dient den Mailserver-betreibern, die entsprechende E-Mail leichter in den Log Dateien zu finden.

```
for <beispiel@gmx.de>; Wed, 15 Jan 2020 16:38:57 +0100
```

Abbildung 2.7: Dritter Teil der Received-Zeile, der Empfänger, Einlieferungsdatum und Zeitstempel

An dieser Stelle der Received-Zeile wird wiederholt, wer der Empfänger und wann genau die E-Mail bei dem Mailserver eingegangen ist. Sollte die E-Mail an mehrere Adressaten auf dem gleichen Mailserver gehen, entfällt die Angabe der anderen Empfänger in diesem Header. Jedoch ist dieses nicht der Fall für den „To“-Header, welcher alle Empfänger auflistet. Die Empfänger einer Blindkopie (BCC) oder einer Kopie (CC) werden nicht im „To“-Header auftauchen. Diese haben ein eigenes Feld. Die Header „To“und „CC“sind, sofern diese benutzt werden, im E-Mail Kopf einzusehen, im Gegensatz zum „BCC“-Header. Es ist die besondere Eigenschaft des „BCC“Headers, dass er nicht im E-Mail Kopf einsehbar ist.

## 2.5 Gefälschte Received-Header

Der From Header ist leicht zu fälschen, gleichzeitig kann ein gefälschter Absender schwer erkennbar sein. Ein Merkmal für eine mögliche Fälschung des Absenders findet sich im Untersten, also dem zuerst eingefügten, Received-Header. Sollte der dort eingetragene empfangende oder weiterleitende Server nicht mit dem E-Mail Anbieter des Absenders übereinstimmen, kann das ein Merkmal für einen gefälschten Absender sein.

```
Received: from mail-portal (unknown [80.242.181.215])
        by smtp-auth.pop-interactive.de (Postfix) with ESMTPSA id 4D7EB7AC7
        for <example@gmx.net>; Thu, 27 Feb 2020 16:10:55 +0200 (CEST)
From: example@gmail.com
```

Abbildung 2.8: Beispiel eines gefälschten From Headers

Obwohl diese E-Mail von einer Gmail-Adresse, in Abbildung 2.8 Blau dargestellt, kommen soll, kann beobachtet werden, dass Google-Mail nicht als absendender Server eingetragen ist. Stattdessen handelt es sich anscheinend um einen deutschen Absender. In Abbildung 2.8 Rot dargestellt. Zusätzlich weist die in Grün dargestellte Zeitzone Central European Standard Time (CEST) daraufhin, dass diese E-Mail einen Europäischen Absender hat. Alle Server der Firma Google, welche gmail.com hostet, stehen ausschließlich in den USA. Infolge dessen ist die Zeitzone ein Indiz dafür, dass die E-Mail einen gefälschten „From"-Header hat.

## 2.6 Angriffsarten

In dem ursprünglichen Simple Mail Transfer Protokoll wurde der Sicherheitsaspekt vernachlässigt, da im Vergleich zur heutigen Zeit die E-Mails damals in einem kleinen Netz versendet wurden. Die Verschlüsselung der E-Mail bei der Übertragung, der Leitung oder der Authentifizierung des Absenders und Empfängers waren daher nicht notwendig. Durch diese Art der Entwicklung sind drei Angriffsmöglichkeiten besonders einfach umzusetzen: [11]

- „From Header Spoofing" - Da der E-Mail Header und Body nur Textzeilen sind, ist es einfach, für den Sender eine beliebige E-Mail-Adresse in den From-Header einzutragen.
- „Phishing" - Meist in Verbindung mit dem oben genannten From Header Spoofing, versuchen Angreifer sich als Bank oder Dienstleister auszugeben, um entweder an Geld oder/und sensible persönliche Daten zu gelangen.
- „Man in the Middle" - Bei einem „Man in the Middle" Angriff kann ein Angreifer eine unverschlüsselte oder authentifizierte E-Mail weiterleiten und dabei den Inhalt beliebig ändern.

[11] Darstellung des Abschnitts auf Basis von [11, Email Authentication Mechanisms: DMARC, SPF and DKIM] in freier Übersetzung

## 2.7 Gegenmaßnahmen

Die erste Maßnahme Spam und Spoofing zu bekämpfen, sollte die Sensibilisierung der Nutzer mit diesem Thema sein. Viele Firmen geben jährlich große Summen aus, um die Angestellten auf dieses Thema vorzubereiten. Da sich aber manche Spammer mit ihren Nachrichten mittlerweile kaum noch von legitimen E-Mails unterscheiden, wird die Sensibilisierung irgendwann nicht mehr ausreichend sein.
Als technische Maßnahmen gibt es das Sender Policy Framework (SPF), Domain Keys Identified Mail (DKIM) und Domain-based Message Authentication, Reporting and Conformace (DMARC). Jedoch werden diese Methoden relativ gering eingesetzt. Obwohl diese drei Methoden gute Maßnahmen sind, den Ursprung der E-Mail zu überprüfen, helfen sie jedoch nicht, wenn besagte Nachrichten während des Transit manipuliert werden.[12]
Das Verschlüsseln der E-Mails mit dem dazugehörigen Austausch des Public-Keys ist ein erster Schritt für Nutzer in die Richtung des sicheren E-Mail Versandes. Dieses Konzept wurde in „Developing a Model to Detect E-mail Address Spoofing using Biometrics Technique“ weiterentwickelt. Die Idee, eine biometrische Komponente als Authentifizierung hinzuzufügen, ist zwar ein guter Ansatz, aber nach Meinung des Autors dieser Arbeit nicht einfach umsetzbar. Zur Realisierung dieser Idee bräuchte der E-Mail Sender dann extra Hard- und Software, um die biometrische Authentifizierung durchzuführen. Sollten die Biometriedaten dann auf irgendeine Weise im Internet landen, könnte der Nutzer nie wieder seine E-Mails sicher mithilfe seiner Biometrie signieren, da andere Personen seine Signatur mit den biometrischen Daten fälschen könnten.

[12] Frei übersetzt nach [11, Email Authentication Mechanisms: DMARC, SPF and DKIM]

# 3 Konzeption und Architektur

## 3.1 Konzeptidee

Die Idee für diese Erweiterung war, den Ursprung der E-Mail durch das Abfragen der WhoIs-Daten zu überprüfen. Dafür wird der erste Eintrag aus dem Received Header extrahiert und die Ursprungs IP-Adresse als Anfrage an die WhoIs-Datenbank gesendet. Sollte der Ursprung der E-Mail nicht mit den Informationen aus den WhoIs-Daten übereinstimmen, wird diese E-Mail dann als Spoof deklariert werden. Nach der Entwicklung dieses Konzeptes wurde ein Dozent für Rechnernetze bezüglich der Idee befragt. Am Ende dieser Diskussion empfahl er, anstelle der WhoIs Informationen, die RDAP Informationen zu nutzen und weitere Tests einzubauen. Im weiteren Verlauf entwickelte sich das Konzept, dass nun die RDAP an Stelle von WhoIs Informationen abgerufen werden. Zusätzlich wurden vier weitere Tests zu dem vorhandenen eingebaut. Diese folgenden fünf Testmethoden untersuchen separat die E-Mail auf Spoofing. Jedes Testresultat bekommt dann eine Punktzahl zugewiesen, die in den Einstellungen geändert werden kann. Diese Punkte werden miteinander addiert. Sollte die Gesamtpunktzahl eine ebenfalls einstellbare Punktzahl übertreffen, wird die entsprechende E-Mail in Thunderbird als Spam markiert.
In 2.7 wurden verschiedene Mittel und Wege erwähnt, wie Spam und Spoofing vermindert oder sogar verhindert werden können. Dieses Konzept zielt wie bereits zuvor erwähnt nicht darauf ab, E-Mails sicher zu versenden oder dieses zu ermöglichen, sondern gespoofte E-Mails zu erkennen. Aus diesem Grund werden, wie in 1.2 erwähnt, nur Erkennungsmaßnahmen betrachtet und nicht Mittel um E-Mails sicher zu versenden.

#### 3.1.0.1 Ursprungsüberprüfung mit Hilfe von RDAP

Dieser Test überprüft, ob die eingetragene IP Adresse in dem ersten Received Header als Absender für die jeweilige Domain zugelassen ist. Es kann aus Endnutzersicht nicht

unterschieden werden, ob die E-Mail direkt von der Webseite oder von einem Mail User Agent wie Outlook oder Thunderbird gesendet worden ist. Diese Information ist wichtig, da der Mail User Agent im Gegensatz zu der Webseite erst die Mail zu dem E-Mail Anbieter senden muss. Der Mail User Agent kann daher an einem beliebigen Ort auf der Welt sein. Durch den Eintrag der Received-Zeile, welcher beim Senden von dem Mail User Agent zu dem Mailserver entsteht, wird das Ergebnis des Tests verfälscht. Um dieses zu verhindern, werden die ersten beiden Received-Zeilen überprüft. Die in diesen Zeilen vermerkten IP Adressen werden extrahiert und dann mithilfe der RDAP Datenbank jeweils ein Standort zugewiesen. Im Anschluss wird die Domain des angeblichen Senders aus dem Header entnommen und der MX-Record, also die Information einer Webseite welche IP-Adressen berechtigt sind E-Mails von der Domain zu senden, erfragt. Diese IP-Adressen bekommen wieder mit Hilfe der RDAP Datenbank einen Standort zugewiesen. Sollte der Standort aus den Received Header nicht mit dem Standort aus dem MX-Record übereinstimmen, klassifiziert dieser Test die E-Mail als Spoof.

#### 3.1.0.2 SPF-Header Überprüfung

Das Sender Policy Framework(SPF) testet beim Versenden einer Nachricht, ob der Absender berechtigt ist, unter dem Namen der Domain E-Mails zu verschicken. Das Ergebnis wird in Form eines SPF Headers festgehalten und in den E-Mail Header geschrieben. Dieser Test überprüft, ob der Header eingetragen worden ist und wenn ja, welches Ergebnis dieser beinhaltet. Entsprechend dem Resultat, welches im Header vermerkt worden ist, wird der Test den Inhalt bewerten.
Anmerkung des Autors: Dieser Test wäre obsolet, wenn alle E-Mail Anbieter und Betreiber das Sender Policy Framework nutzen und alle Mail Transfer Agenten diesen Header überprüfen würden.

#### 3.1.0.3 Blacklist Test

In dieser Überprüfung sollen alle IP-Adressen aus den Receivedzeilen extrahiert, in einer Liste festgehalten und anschließend mit einer Blacklist abgeglichen werden. Damit auch neue Spam Sender identifiziert werden können, soll die Blacklist jede Stunde aktualisiert werden. Sollte mindestens eine IP aus der Liste in der Blacklist vorhanden sein, wird die E-Mail von diesem Test als Spam beziehungsweise Spoof eingestuft.

#### 3.1.0.4 Zeitzonen Überprüfung

Die Zusammensetzung des Received Headers ist in diesem Test von besonderer Bedeutung. Am Ende jeder Received Zeile werden, wie in 2.4 beschrieben, Zeit und Datum mit der dazugehörigen UTC Zeitzone eingetragen. Bei diesem Test sollen die Zeitzonen aus den ersten zwei eingetragenen Receivedzeilen extrahiert werden. Ob in diesen beiden Zeitzonen Länder sind, in denen die angegebene Domain ihre E-Mail Server hostet, soll anschließend überprüft werden. Sollte dieses nicht der Fall sein, soll diese E-Mail als Spoof klassifiziert werden.

#### 3.1.0.5 SpamAssassin Test

Die E-Mail soll durch SpamAssassin auf verschiedene Methoden überprüft werden. Unter anderem wird mit Hilfe von statischen Regeln, die auf regulären Ausdrücken basieren, die E-Mail nach bestimmten Formulierungen durchsucht, die typischerweise in Spam vorkommen. Das Ergebnis wird in Form eines Spam-Scores, also eines Zahlenwertes, festgehalten. Je nachdem, wie hoch der Punktestand ist, wird die E-Mail als Spam klassifiziert oder nicht. Dieser Test kann lokalisiert werden, indem der oder die Nutzer eine lokale Version von SpamAssasin installiert. Die Prüfung kann dann zur lokalen Installation weitergeleitet werden. Da eine Lokalisierung aber nicht Teil der Arbeit ist, wurde nach einem einfachen Zugang zu den Funktionalitäten von SpamAssasin gesucht. Einen unbegrenzten Zugriff bietet die API von Postmark an. Um eine einsatzfähige Erweiterung als Resultat zu haben, wird für diesen Prototypen die API von Postmark genutzt. Um eventuelle Sicherheitsrisiken zu vermeiden wird dieser Test optional gemacht. In den Einstellungen soll daher einstellbar sein, ob dieser Test genutzt wird oder nicht.

### 3.1.1 Feste Integrierung in Thunderbird

Im Rahmen der Konzeptidee wurden einige Entwickler von Thunderbird angesprochen, unter welchen Bedingungen eine festen Integrierung in dem E-Mail Client durchführbar ist. Ein Produkt Manager von Mozilla stellte dieses in Aussicht, unter der Bedingung, dass die Erweiterung für alle Nutzer sinnvoll wäre und das Kernteam dem Hinzufügen sein Einverständnis gibt. Hinderlich ist zurzeit noch, dass die Spam Markierung von Thunderbird gleichzeitig einen Lernalgorithmus betreibt, um Spam E-Mails zu erkennen. Eine auf Tests basierte Methode, um die E-Mails zu überprüfen, könnte dem Lernalgorithmus

fälschlich erkannte Spam Nachrichten übergeben. Dadurch würde die Erkennungsrate des Lernalgorithmus von Thunderbird vermindert werden. Vorläufig bleibt das Konzept als Erweiterung bestehen, was sich nach Rückmeldung und Nutzung der Erweiterung in Zukunft noch ändern kann.

### 3.1.2 Server vs E-Mail Service System vs Client

Obwohl die Arbeit sich mit der Entwickelung einer Erweiterung auseinandersetzt, wird auch der Ansatz mit anderen Architekturoptionen verglichen. Das bedeutet, dass auch andere Möglichkeiten der Implementation abgewägt werden müssen. Ein Ansatz, die Erweiterung direkt in Thunderbird zu integrieren, wurde bereits in 3.1.1 erwähnt. Ein anderer Ansatz wäre, die Prüfung nicht im Client, sondern direkt auf dem E-Mailserver durchzuführen. Der Vorteil wäre, dass der Nutzer am Ende nur überprüfte E-Mails erhält, welche nicht als Spoof eingestuft worden sind. Durch eine einzelne Installation von SpamAssassin und Umleitung der DNS-Abfragen auf den E-Mailserver könnte eine vollständige Lokalisierung erreicht werden. Nachteilig wäre jedoch, dass die Individualität der Testwertigkeiten nicht mehr gewährleistet werden kann. Es würde nur eine Wertigkeit für alle Nutzer des E-Mailservers geben. Ebenfalls nachteilig wäre, dass der Betreiber oder Verantwortliche des E-Mailservers eine Schnittstelle von SpamAssassin bereitstellen oder sogar entwickeln muss, um den Test lokalisiert zu nutzen.
Eine weitere Möglichkeit wäre die Prüfung als Service bereitzustellen. In dieser würden Nutzer die E-Mails an ein E-Mail Postfach senden und dann eine Bewertung als Antwort erhalten. Dieser Ansatz ist für Inhaltskontrollen wie in der Bachelorarbeit von Manuel Selmeier (Malwareanalyse mit Cuckoo) ausreichend, jedoch verändert das Weiterleiten der E-Mail den Header. Eine Auswertung der E-Mail nach dem Weiterleiten ist daher nicht mehr möglich. Aus diesem Grund wird dieser Ansatz nicht weiter beachtet.
Die Überprüfung auf Client-Seite durch eine Erweiterung ermöglicht es den Nutzern, Einstellungen an den Bewertungen vorzunehmen oder Tests zu ignorieren. Jeder Nutzer könnte selbst entscheiden, ob er diese Erweiterung vollständig nutzen möchte oder nur Teile davon. Die visuelle Rückmeldung, die E-Mail als Spam zu markieren, könnte im Zweifel von Benutzern ignoriert werden. Dieses ist jedoch vorteilhafter im Vergleich zu einer E-Mail die den Nutzer über die Quarantäne Maßnahmen unterrichtet.
Es wurden die Vor- und Nachteile der Konzepte abgewogen. Um ein gewisses Maß an Individualität und Einstellbarkeit für den Endnutzer zu erhalten, wurde beschlossen, diese Tests vorerst als Erweiterung für E-Mail Clients zu entwickeln. Weiter spricht für

diese Entscheidung, dass eine spätere Integration in Thunderbird zu diesem Zeitpunkt nicht ausgeschlossen ist und diese Arbeit vor allem einen funktionierenden Prototypen hervorbringen soll.

### 3.1.3 Client Auswahl

Nachdem die Entscheidung gefallen war, dass die Erweiterung für E-Mail Clients entwickelt wird, musste noch der E-Mail Client ausgewählt werden. Neben den beiden bekannten E-Mail Programmen Outlook von Microsft und Thunderbird von Mozilla, gibt es noch viele weitere Programme, die es erlauben, E-Mails von Konten unterschiedlicher Anbieter zu empfangen, zu sortieren oder zu versenden. Aufgrund der schwankenden oder geringen Nutzerzahlen bei den anderen Produkten wurden diese jedoch ignoriert. Die Wahl musste daher zwischen dem zahlungspflichtigem Produkt Outlook und dem Open-Source Projekt Thunderbird getroffen werden.
Der offensichtliche Vorteil von Thunderbird ist, dass es kostenlos ist. Outlook muss entweder alleinstehend oder in den sogenannten Office Paketen gekauft werden. Thunderbirds Stärke ist die Individualisierung durch Add-Ons und die klar strukturierte Oberfläche für Privatanwender. Outlook hingegen bietet dafür mehr Funktionalitäten an. Das Lernen dieses Funktionsumfangs ist in Outlook zwar notwendig, jedoch nicht besonders Umfangreich. Nachteilig für Thunderbird ist die eher schleppende Entwicklung. Während bei Microsoft ein großes, fest angestelltes Entwicklerteam Outlook ständig weiterentwickelt, sind bei Thunderbird vergleichsweise wenig Entwickler fest angestellt. Diese Entwickler werden aus den Spenden für Thunderbird bezahlt. Ein Großteil der Arbeit bei Thunderbird wird durch Freiwillige weltweit über die Bugzilla Webseite erledigt. Bei beiden Programmen helfen bei Fragen und Problemen meistens die Nutzer in dem jeweiligen Foren aus.
Am Ende wurde beschlossen, dass die Vorteile von Thunderbird den Vorteilen von Outlook überwiegen. Das Angebot des kostenlosen Programms, welches sich besonders durch die Individualisierung mithilfe von Erweiterungen hervorhebt, ist eine gute Grundlage für das Erreichen des Zieles dieser Arbeit.

## 3.2 Diagramme für die Erweiterung

Es wurde ein Komponentendiagramm und ein Aktivitätsdiagramm erstellt, um das Vorgehen der Erweiterung zu visualisieren und eventuelle Schwachstellen vor der Entwicklung zu erkennen. Nach den folgenden Diagrammen wird die Erweiterung implementiert.
Die Wertung der einzelnen Tests soll am Ende über eine grafische Oberfläche einstellbar sein. Werden keine Einstellungen vorgenommen, nutzt die Erweiterung die Standardwerte, welche im Programmcode hinterlegt sind. Die Erweiterung fragt zyklisch bei Thunderbird an, ob es neue ungelesene E-Mails gibt, um diese dann zu überprüfen. Intern werden dann diese E-Mails an die einzelnen Tests übergeben. Wie in A.1 zu sehen ist, müssen der SpamAssassin Test, der Origin Test und der Blacklist Test externe Informationen erfragen. Es ist möglich die Abfragen zu umgehen. Dafür ist eine lokale Version von SpamAssassin notwendig, eine eigene Spam-Blacklist und eine eigene Art der DNS-Namensauflösung. Um Abhängigkeiten von bestimmten Versionen oder Frameworks zu verhindern, wird darauf verzichtet, fremde Frameworks zu nutzen. Einzige Ausnahme bildet das grundlegende Services Modul von Mozilla, welches von Thunderbird bereitgestellt wird. Um die Wartung und mögliche Erweiterungen zu ermöglichen, wird darauf geachtet, dass jede Funktion einen sogenannten Contract erhält und die Variablen im Code einen aussagekräftigen Namen bekommen.
Wenn die Erweiterung gestartet wird, soll als Erstes die neueste Version der Blacklist geladen werden. Das Aktualisieren oder das erneute Herunterladen der Blacklist sollte nach spätestens einer Stunde ausgeführt werden, um eine gewisse Aktualität zu gewährleisten. Während die neueste Version der Blacklist geladen wird, fragt die Erweiterung Thunderbird nach ungelesenen E-Mails. Sollte es keine geben, so wartet die Erweiterung darauf, dass der Zyklus abläuft und fragt nach der eingestellten Zeit wieder nach ungelesenen E-Mails. Wenn ungelesene Nachrichten existieren, so werden diese nacheinander den Tests übergeben. Das Ergebnis aus den Tests wird unterschiedlich mit Punkten bewertet. Die Punkte werden dann anschließend addiert. Sollte der eingestellte Schwellwert erreicht oder überschritten werden, so wird die E-Mail als Spam markiert. Wird dieser Wert nicht erreicht, bleibt die Nachricht im Originalzustand. Dieser Prozess wird zyklisch mit allen ungelesenen E-Mails wiederholt, bis Thunderbird geschlossen wird.

# 4 Implementation

## 4.1 Grundaufbau

Thunderbird erlaubt nur Erweiterungen oder Webextenstions, die mit JavaScript oder HTML geschrieben worden sind. Daher wurde beschlossen, dass die Hintergrundprozesse mit JavaScript eingebaut und die Einstellungsmöglichkeiten mit Hilfe von HTML visualisiert werden. Die Überprüfungen laufen im Hintergrund, sodass der Nutzer nur die Einstellungen sehen und gegebenenfalls ändern kann. Als visuelle Rückmeldung bei einem erkannten Spoofing-Versuch, wird die E-Mail mit dem Spam-Zeichen von Thunderbird belegt. Sollte der Test fälschlicherweise die E-Mail als Spoof einstufen, kann der Nutzer das Spam-Symbol entfernen und als normale E-Mail behandeln.

## 4.2 Dokumentation von Thunderbird

Vor der eigentlichen Implementierung wurde anfangs nach einer Dokumentation von Klassen und deren Funktionen gesucht. Als die erste Implementierung fertig war und getestet wurde, gab das Programm „Function not found“ Fehler aus. Die Recherche nach dem Fehler ergab leider keine hilfreichen Auskünfte, daher wurde nach einer direkten Hilfe von Thunderbird gesucht. Als Hilfestellung auf der Webseite von Thunderbird wurde ein IRC-Channel aufgelistet. In einem daraufhin geführten Gespräch mit den anwesenden Entwicklern erklärten diese, dass der Stand der Dokumentation aus dem Jahr 2007 sei und über die Versionen hinweg viele Funktionen und Klassen verändert worden sind oder nicht mehr existieren. Eine aktuelle Dokumentation wäre zwar im Aufbau, aber noch nicht vollständig. Für auftretende Fragen oder Probleme bei der Entwicklung erbaten sie eine E-Mail mit den entsprechenden Problemen an die Developer Mailingliste. Im Verlauf der folgenden Zeit waren einige Entwickler bereit, dem Autor dieser Arbeit zu helfen. So konnten im Austausch für diese Hilfe kleinere Bugs in Thunderbird von dem Autor behoben werden.

## 4.3 Fehlende Funktionalitäten von Thunderbird

Nach mehreren Gesprächen mit dem Entwicklerteam von Thunderbird wurde dem Autor dieser Arbeit auf Bugzilla ein Account erstellt und zur Verfügung gestellt. Auf dieser Seite, die Bugreports und Implementationswünsche auflistet, wurde dann vom Autor der Funktionalitätswunsch, aus einer Erweiterung nach E-Mails suchen zu können, eröffnet. Nachdem dieser jedoch aufgrund von niedriger Priorität nicht bearbeitet worden war, entwickelte der Autor diesen selbst. Zunächst als experimentelles Feature bereitgestellt, wurde diese Funktionalität später in Thunderbird fest integriert.
Dieser Ablauf wiederholte sich am Ende der Entwicklung als auffiel, dass das Markieren von E-Mails als Spam aus einer Erweiterung heraus nicht vorgesehen war. Mittlerweile sind beide Funktionalitäten fest in Thunderbird integriert und werden vermehrt von anderen Entwicklern genutzt. Nachdem diese beiden Funktionalitäten in der Daylie Version, also der Testversion von Thunderbird, veröffentlicht worden waren, konnte die Erweiterung fertiggestellt werden.

## 4.4 RDAP Request

Der Request nach Informationen zu einer bestimmten IP wurde zwischen den fünf globalen Verwaltungsstellen einheitlich gestaltet. Wird zum Beispiel die Europäische Datenbank nach Informationen zu einer IP aus den USA gefragt, dann wird die Anfrage direkt an die zuständige Stelle weitergeleitet. Das Weiterleiten hat seine Vor- aber auch Nachteile. So wurden während der Implementation bei dem Ausführen von Tests hin und wieder Fehler gefunden, die auf mindestens eine nicht ausgefüllte Variable hinwiesen. Nach genauerem Betrachten der Variablen ließ sich ein Muster erkennen. Die Antworten der RIPE Datenbank, welche für die Verwaltung von Europa zuständig ist, unterscheiden sich trotz des RFC Standards von den Antworten, die von ARIN, dem Nord Amerikanischen Gegenstück, kommt.

```
1 {
2 "handle": "80.242.161.208 - 80.242.161.223"
3 "startAddress": "80.242.161.208"
4 "endAddress": "80.242.161.223"
5 "ipVersion":  "v4"
6 "name": "CUSTOMER-NET-PS-10646"
7 "type": "ASSIGNED PA"
```

```
"country":  "DE"
"parentHandle": "80.242.160.0 - 80.242.191.255"
"entities": [...]
"remarks":  [...]
"links":  [...]
"events": [...]
"rdapConformance":  [...]
"notices":  [...]
"port43": "whois.ripe.net"
"objectClassName":  "ip network"
}
```

Listing 4.1: RDAP Antwort von RIPE

Die Antwort von RIPE beinhaltet die Information "country", also in welchem Land die Domain gehostet wird. Diese Information wird von LACNIC und APNIC nur teilweise gegeben. In APNIC wird diese Information nur konstant gegeben, wenn der Hoster in China gemeldet ist. Gleiches gilt für LACNIC in Verbindung mit Brasilien.

```
{
"rdapConformance":  [...]
"notices":  [...]
"handle": "NET-100-128-0-0-1"
"startAddress": "100.128.0.0"
"endAddress": "100.255.255.255"
"ipVersion":  "v4"
"name": "CORE2"
"type": "DIRECT ALLOCATION"
"parentHandle": "NET-100-0-0-0-0"
"events": [...]
"links":  [...]
"entities": [...]
"port43": "whois.arin.net"
"status": [...]
"objectClassName":  "ip network"
"cidr0_cidrs":  [...]
"arin_originas0_originautnums": []
}
```

Listing 4.2: RDAP Antwort von ARIN

Die RDAP Antwort von ARIN gibt die Information, in welchem Land diese IP-Adresse gehostet wird, nicht preis. Ebenfalls erkennbar ist, dass die Reihenfolge der Parameter

sich von einer regionalen Organisation zu anderen regionalen Organisationen unterscheidet.

### 4.4.1 RDAP Inkonsistenz

Die Inkonsistenz im Aufbau der RDAP Antwort ist für die Bearbeitung trivial, da diese im JSON-Format erfolgt. Das JSON-Objekt kann entsprechend nach dem Objekt „country“ gefragt werden. Es wurde jedoch eine Lösung benötigt für den Fall, dass dieses Objekt nicht im JSON-Objekt vorhanden ist. Zur Lösung dieses Problems wurde zuerst der Link genommen um den Ursprung der Antwort zu ermitteln. Es stellte sich aber nach kurzer Zeit wieder heraus, dass es auch in diesem Fall Unterschiede gab. Der Link zu der Antwort von ARIN (https://rdap.arin.net/registry/ip/1.1.1.1), im Vergleich zu einem Link einer Antwort von RIPE (https://rdap.db.ripe.net/ip/2.2.2.2), enthielt nicht den Textkürzel „db“.
Eine einheitliche Art der Identifizierung wurde in der JSON Antwort unter dem Objekt „port43“ gefunden. Das „whois“ und „net“ können aus „whois.arin.net“ entfernt werden und dadurch die antwortende Autorität enttarnen.

## 4.5 Warum SPF, aber kein DKIM und DMARC?

An dieser Stelle der Arbeit fällt dem Leser eventuell auf, dass die Erweiterung zwar den SPF Status erfragt, jedoch DKIM und DMARC ignoriert. DMARC ist zunächst einmal eine Möglichkeit den Spam einzudämmen, indem der Betreiber der E-Mail Adresse über den Missbrauch informiert wird. Das Informieren der Betreiber ist nicht das Ziel dieser Arbeit, weswegen DMARC in den Augen des Entwicklers eine gute Sache, jedoch für diese Arbeit nicht relevant ist.
Das DKIM Verfahren, also das Einfügen einer digitalen Signatur, die vom Empfänger überprüft wird, ist ebenfalls ein guter Weg, um E-Mail-Spoofing zu verhindern. Das Abfragen des öffentlichen Schlüssels einer Domain und mit jenem das Entschlüsseln der digitalen Signatur für jede E-Mail, verbraucht jedoch zu viel Ressourcen und Zeit. Eine Möglichkeit wäre es, eine Datenbank für die öffentlichen Schlüssel anzulegen. Diese Datenbank müsste jedoch regelmäßig den Status der Schlüssel überprüfen und im Zweifel neu erfragen. Damit die Erweiterung jedoch die E-Mails in einer akzeptablen Zeit

abarbeiten kann, wird diese Erweiterung zunächst als eine kleine, kompakte Version in Einsatz gehen. Aus diesem Grund wird das DKIM Verfahren vorerst ignoriert.

## 4.6 Komponenten

Um eine einfache Erweiterung und Austauschbarkeit zu erreichen, wurde eine lose Kopplung angestrebt. Das bedeutet, dass die Tests nicht unnötig miteinander verknüpft werden. Jeder der fünf Tests bekommt die E-Mail als Eingabeparameter und nach dem Testen geben sie entweder „fail“ oder „pass“ zurück. Ein „fail“ Rückgabewert bedeutet, dass der Test Anzeichen für Spoof oder Spam gefunden hat.
Gesteuert werden die Tests von der selbstgebauten Komponente „Handler“, welcher wie der Name schon sagt, die Aufgabe hat die Tests zu koordinieren. Die Komponente fragt regelmäßig Thunderbird nach ungelesenen E-Mails, um diese an die Tests weiterzuleiten. Weiterhin verwaltet der Handler die Wertigkeiten, welche über die Einstellungen gesetzt werden können. Wenn die Tests ihre Ergebnisse preisgeben, wendet der Handler die Wertigkeiten auf die Antworten an, addiert die Punkte und prüft, ob die getestete E-Mail den Schwellwert erreicht oder überschritten hat. Sollten die Testergebnisse den eingestellten Grenzwert erreichen oder übertreffen, so wird die E-Mail als Spam innerhalb von Thunderbird markiert.

## 4.7 Aufruf der Erweiterung

Sobald Thunderbird geladen wird, ruft die Erweiterung sich selber auf (Listing 4.3 Zeile acht). Als Erstes lädt sie sich die neueste Version der Blacklist in den Speicher (Listing 4.3 Zeile zwei) und führt danach die Tests innerhalb der Main-Funktion aus (Listing 4.3 Zeile drei). Anschließend setzt sie Intervalle, welche die Hauptfunktion nach der eingestellten Zeit erneut aufruft (Listing 4.3 Zeile fünf und sechs). Der Grund für den Aufruf nach der festgelegten Zeit ist, dass Thunderbird zu dem aktuellen Zeitpunkt noch kein Event eingebaut hat, welches sich aktiviert, wenn eine neue E-Mail eingetroffen ist.

```
1 function Init(){
2   BlacklistLoader();
3   Main();
4   let min_in_millisec = 60000;
5   let blacklistIntervalID = window.setInterval(BlacklistLoader, (60 *
      min_in_millisec));
6   let mainIntervalID = window.setInterval(Main, (3 * min_in_millisec));
7 }
8 addEventListener("load", Init(), true);
```

Listing 4.3: Initialisierung der Erweiterung

## 4.8 Ausführung der Erweiterung

Um zu garantieren, dass die Tests sich nicht untereinander stören oder Ergebnisse von vorherigen oder nachfolgenden E-Mails präsentieren, wurden die Komponenten mithilfe der Future (engl. „Zukunft“) oder auch Promise (engl. „Versprechen“) Programmierung erstellt. Dieses Konzept erlaubt es, durch die Hilfe von Platzhaltern mit Ergebnissen zu arbeiten, deren Berechnungen noch nicht abgeschlossen sind. Das Promise kann dabei drei Zustände haben. „erfüllt“, „gebrochen“ und „wartend“.
Der Handler erstellt ein Promise mit den fünf unterschiedlichen Überprüfungen und übergibt an jeden Test die gleiche E-Mail. Sollte vom Nutzer gewünscht sein, dass der SpamAssassin Test nicht genutzt werden soll, wird die E-Mail nur an die vier verbleibenden Tests übergeben.

```
1  await fullMsgPromise.then( async (fullMsg) => {
2    if (noAPI){
3      Promise.all([SPF_Check(fullMsg), Origin_Check(fullMsg), Blacklist_Check(
       fullMsg), TimeZoneCheck(fullMsg)]).then((resultarray) => {
4       resultarray.unshift(null);
5        EVAL_MAIL(msgID, resultarray, fullMsg);
6      }).catch(reason => {
7        console.log(reason)
8      });
9    }else{
10     Promise.all([API_Check(message), SPF_Check(fullMsg), Origin_Check(fullMsg
       ), Blacklist_Check(fullMsg), TimeZoneCheck(fullMsg)]).then((resultarray)
       => {
```

```
11        EVAL_MAIL(msgID, resultarray, fullMsg);
12      }).catch(reason => {
13        console.log(reason)
14      });
15    }
16  })
```

Listing 4.4: Promise der Überprüfungen

Die Erweiterung überprüft, ob der Nutzer den SpamAssassin Test anwenden möchte mit Hilfe des „noAPI“Parameters (Listing 4.4 Zeile zwei). Der Haken für diese Option kann in den Einstellungen der Erweiterung gesetzt oder entfernt werden. Wenn jeder der aufgerufenen Tests ein Ergebnis erreicht hat, werden die Resultate an die Auswertung übergeben. Falls der SpamAssassin Test nicht gewünscht ist, wird an erster Stelle der Antwortsammlung ein null Objekt eingefügt, um die Evaluation für beide Fälle gleich zu halten. Sollte einer der Tests einen Fehler auswerfen, ändert sich der Zustand auf „gebrochen“ und der Grund wird in der Konsole ausgegeben. Es kann nicht von einem Fehler auf einen misslungenen Test geschlossen werden, daher wird der Status der E-Mail nicht verändert, sondern nur der Fehler ausgegeben.

## 4.9 Auswertung der Erweiterung

Die Auswertung der Ergebnisse erfolgt durch die Addition der eingestellten Punktzahlen und, sofern ausgeführt, des Multiplikators für den SpamAssassin Test. Es wird hierbei beachtet, ob der Nutzer eine eigene, abgespeicherte Wertung hat.

```
1  browser.storage.local.get("Prefs").then((prefs) =>{
2    let settings = prefs.Prefs;
3
4    if(settings != null && settings != undefined){
5      spf_value = parseInt(settings.spf_slider);
6      api_value = parseInt(settings.api_slider);
7      tz_value = parseInt(settings.tz_slider);
8      bl_value = parseInt(settings.bl_slider);
9      origin_value = parseInt(settings.origin_slider);
10     needed_score = parseInt(settings.needed_score_slider);
11   }
12
13   if (api_result != null) {
14     resulting_score = resulting_score + (parseFloat(api_result)*api_value);
```

```
15    }
16
17    if (spf_result == "fail") {
18      resulting_score = resulting_score + spf_value;
19    }
20
21    if (origin_result == "fail") {
22      resulting_score = resulting_score + origin_value;
23    }
24
25    if (blacklist_result == "fail") {
26      resulting_score = resulting_score + bl_value;
27    }
28
29    if (timezone_result == "fail") {
30      resulting_score = resulting_score + tz_value;
31    }
32
33    if (resulting_score >= needed_score){
34      browser.messages.update(msgID, { junk: true });
35    }
36  })
```

Listing 4.5: Addition der Wertigkeiten

Die Begründung für den Multiplikator bei dem SpamAssassin Test ist, dass das Ergebnis der API bereits ein Dezimalwert ist. Das Prinzip hinter dem Wert ist vergleichbar mit dem selbsterstellten Punktesystem. Ein niedriger Wert deutet auf eine legitime E-Mail hin, während ein hoher Wert auf Spam schließen lässt. Der Dezimalwert der Antwort reicht dabei von 0,0 bis zu 15,0 Punkten. Statt den Wert durch einen eigenen zu ersetzen, ist es aussagekräftiger, diesen Wert mit einem Faktor zu multiplizieren. Dadurch ist die Punktzahl mit den anderen Tests vergleichbar. Ist am Ende der Evaluierung der Schwellwert erreicht oder überschritten, wird die Nachricht mit dem Spam-Symbol gekennzeichnet.

## 4.10 Optionsseite der Erweiterung

Wie im dritten Kapitel beschrieben, soll der Nutzer die Möglichkeit haben, die Tests nach seinen Vorstellungen zu bewerten. Um dieses zu erreichen, wurde eine Optionsseite

erstellt. Dort können die Nutzer die nötigen Einstellungen vornehmen, um Wertigkeiten zu ändern und den SpamAssassin Test nicht ausführen zu lassen.
Wie in A.3 zu sehen ist, wurde diese Seite schlicht gehalten um ein einfaches Verständnis zu fördern. Jeder Test hat eine kurze Beschreibung, was dieser Test überprüft. Mit einem Regler unter den Namen des Testes lassen sich die Punkte einstellen. Sollte der Nutzer die Überprüfung durch SpamAssassin nicht wünschen, kann dieser den Haken bei diesem Test setzen. Die Einstellungen werden erst beachtet, wenn diese abgespeichert worden sind. Vorher nutzt die Erweiterung die Standardwerte. Sollte der Nutzer mit seinen eingestellten Punktzahlen nicht das gewünschte Ergebnis erzielen, kann dieser über klicken des Standard Knopfes die ursprünglichen Werte wieder einstellen.
Um ein Wiedererkennungswert für Nutzer dieser Erweiterung zu erreichen, wurde ein Logo erstellt. Dieses ist auf der Optionsseite und Seite für Add-Ons bei Thunderbird hinterlegt.

# 5 Abschluss

## 5.1 Fazit

Der Autor dieser Bachelorarbeit hat viele Erkenntnisse bei der Bearbeitung gewonnen. Zunächst, dass Open-Source Projekte schlecht dokumentiert werden oder sind. Die Personen, die unterstützen wollen, haben meistens andere Arbeitszeiten. In internationalen Projekten arbeiten diese eventuell sogar in einer anderen Zeitzone. Die unterstützende Person für Fragen bezüglich Thunderbird arbeitete von Neuseeland aus.
Dokumentationen sind aufwendig zu erstellen, mühsam aktuell zu halten und mit hohen Kosten verbunden. Die Dokumentation ist jedoch unersetzlich für ein Softwareprojekt, dessen Ergebnis jahrelang eingesetzt und verbessert werden soll.
Die Existenz einer Dokumentation bedeutet nicht, dass diese aktuell ist. Funktionen und Methoden ändern sich und Datentypen werden ersetzt. Gleiches gilt für Tutorials zu der Programmierung von Erweiterungen.
Auch wenn eine bestimmte Funktionalität von Entwicklern als grundlegend angesehen wird, bedeutet das nicht, dass diese auch eingebaut ist. Deshalb muss im Zweifel selbst grundlegendes ein- beziehungsweise nachgebaut werden.
Eine Modifikation oder das Einführen von neuen Funktionalitäten in Open-Source Projekten wird am schnellsten durch den Austausch von Wissen, Fähigkeiten und Mitarbeit erreicht. Bei fehlenden Funktionalitäten ist eine Kooperation aus unterschiedlichsten Kompetenzen notwendig und Sinnvoll und ein Austausch zielführend. Dieses hat sich im Falle dieser Arbeit deutlich herausgestellt.

Das Ziel dieser Arbeit, den Nutzer durch eine Erweiterung vor Spoofing zu warnen, wurde erfüllt. Die Erweiterung erkennt Spoofing- und Spam E-Mails, die nicht von den herkömmlichen Mitteln erkannt worden sind, und warnt den Nutzer in Thunderbird durch das Markieren der E-Mail als Spam. Für alle Thunderbird Nutzer ist die vorliegende Erweiterung öffentlich zugänglich. Zu finden ist sie unter dem Namen SpoofDetection auf

der Seite für Erweiterungen von Thunderbird.
Als Abschluss sollte noch erwähnt werden, dass dem Autor bewusst ist, dass diese Art der Identifizierung möglicherweise nicht lange wirkungsvoll bleibt, da es jährlich mehrere neue Ansätze zum Versenden von Spam E-Mails gibt. Es existieren aber auch mehr und mehr Personen, die genau wie der Autor auch versuchen neue Ansätze zu entwickeln, um diese E-Mails zu erkennen und zu entfernen. Es bleibt also bei dem Spiel zwischen Spam-Versendern und den Entwicklern, welche versuchen diese E-Mails zu erkennen und zu filtern. Die Situation ist daher vergleichbar mit Jäger und Gejagten.

## 5.2 Ausblick

Diese WebExtension kann in Zukunft noch dadurch ergänzt werden, indem die beiden Protokolle DKIM und DMARC eingebaut werden. Die Benachrichtigung an die Betreiber über den Missbrauch ihrer Domain durch Nutzer oder Bots könnte zu einer Verminderung der gesendeten Spam und Spoofing Nachrichten führen.
Wie in 3.2 bereits erwähnt, arbeitet die Erweiterung noch mit externen Quellen, wie den SpamAssasin Test, um verschiedene Informationen zu erhalten. Eine Lokalisierung durch den verweis auf eine lokale Installation von SpamAssassin und Aufbau eines eigenen DNS Services, könnte die Notwendigkeit von externen Quellen minimalisieren.
Eine weitere Möglichkeit der Ergänzung ist die Prüfung, wie in 3.1.2 beschrieben, auf den E-Mail Server auszulagern. In Verbindung mit einer vollständigen Lokalisierung wäre dieses vermutlich eine gute Empfehlung für Firmen, die einen eigenen E-Mailserver betreiben.
Sollte Thunderbird in Zukunft die Funktionalität von Events erweitern, könnte auch der zyklische Aufruf entfernt werden. Dadurch können neue E-Mails zwecks Überprüfung direkt an die Erweiterung weitergeleitet werden und müssen nicht mehr auf den Ablauf des Zyklus warten.
In „Malwareanalyse mit Cuckoo“ von Manuel Selmeier wird ein System aufgebaut, welches weitergeleitete E-Mails auf den Inhalt überprüft. Ob es ein Hyperlink oder eine angehängte Datei ist, spielt dabei keine Rolle. Dieses System könnte die in dieser Arbeit vorgestellte Erweiterung sehr gut ergänzen. Während sich die Erweiterung dieser Arbeit nur auf das Überprüfen der Header beschränkt und dabei den Inhalt ignoriert, ist es bei dem System von Manuel Selmeier genau umgekehrt. Ein extra Knopf für das manuelle Weiterleiten der fragwürdigen E-Mail wäre leicht zu implementieren und sollte zusätzlichen Schutz bieten.

## 5.3 Würdigungen

Bisher konnten durch die WhoIs Informationen E-Mails präventiv als Spam gefiltert werden. Wenn eine Domain dafür bekannt geworden ist, nur Spam zu verschicken, wurde der Registrant dieser Domain vorgemerkt. Wenn dieser Registrant eine andere Domain erstellt oder kauft, würden die von der Domain verschickten E-Mails entweder einen höheren Spam-Score bekommen oder direkt als Spam eingestuft werden.
Durch die europäische Datenschutz-Grundverordnung (DSGVO) von Mai 2018, kann dieses nicht mehr durchgeführt werden. Die Daten des Registranten unterliegen dem Datenschutz und dürfen daher nicht mehr in den WhoIs Informationen abrufbar sein.[1]

An dieser Stelle möchte der Autor auch dem Thunderbird Entwicklungsteam und besonders Geoff Lankow und Philipp Kewisch für die bereitgestellten Hilfen und Erklärungen danken.

[1] Informationen aus diesem Absatz frei übersetzt nach [2, ICANN GDPR WhoIs policy eliminates preemptive protection of internet infrastructure abuse; obstructs routine forensics to cybercriminals' advantage]

# Literaturverzeichnis


[1] Daigle, L.: *WHOIS Protocol Specification*. 2004. – URL https://tools.ietf.org/html/rfc3912

[2] EU, APWG: *ICANN GDPR WhoIs policy eliminates pre-emptive protection of internet infrastructure abuse; obstructs routine forensics to cybercriminals' advantage*. 2020. – URL https://apwg.eu/icann-gdpr-whois-policy-eliminates-pre-emptive-protection-of-internet-infrastructure-abuse/. – Eingesehen am 20.07.2020

[3] IANA: *About us*. – URL https://www.iana.org/about. – Eingesehen am 15.04.2020

[4] Informationstechnik (BSI), Bundesamt für Sicherheit in der: Die Lage der IT-Sicherheit in Deutschland 2019 / Bundesamt für Sicherheit in der Informationstechnik. 2019. – Forschungsbericht

[5] Klensin, J.: *Simple Mail Transfer Protocol*. 2008. – URL https://tools.ietf.org/html/rfc5321

[6] Krzeminski, Robert: Analyse und Erkennung von Phishing-E-Mails mit Delphish.

[7] Leiba, Barry ; Ossher, Joel ; Rajan, VT ; Segal, Richard ; Wegman, Mark N.: SMTP Path Analysis. In: *CEAS* Citeseer (Veranst.), 2005

[8] NCC, RIPE: *The Internet Registry System*. 2016. – URL https://www.ripe.net/participate/internet-governance/internet-technical-community/the-rir-system. – Eingesehen am 15.04.2020

[9] NCC, RIPE: *The Internet Registry System*. 2020. – URL https://www.ripe.net/publications/docs/ripe-738#ir. – Eingesehen am 05.07.2020

[10] Newton, A. ; Hollenbeck, S.: *JSON Responses for the Registration Data Access Protocol (RDAP)*. 2015. – URL https://tools.ietf.org/html/rfc7483

[11] Nightingale, Stephen J. ; Nightingale, Stephen J.: *Email Authentication Mechanisms: DMARC, SPF and DKIM*. US Department of Commerce, National Institute of Standards and Technology, 2017

[12] Parker, Daniel: *JavaScript with Promises: Managing Asynchronous Code*. Ö'Reilly Media, Inc.", 2015

[13] Resnick, P.: *Internet Message Format*. 2008. – URL https://tools.ietf.org/html/rfc5322

[14] Russell, Ryan: *Die mitp-Hacker-Bibel*. mitp Verlag, 2002

[15] Schwartz, Alan: *SpamAssassin*. O'Reilly Germany, 2005. – ISBN 9783897213937

[16] Selmeier, Manuel ; Gerling, Rainer W.: Automatische Analyse von Dateien und URLs mit der Cuckoo Sandbox. In: *Sicherheit in vernetzten Systemen* 25, S. H

[17] Topf, Jochen ; Etrich, M. ; Heidrich, J. ; Romeo, L. ; Thorbrügge, M. ; Ungerer, B.: *Antispam - Strategien Unerwünschte E-Mails erkennen und abwehren*, 2005

[18] Ullrich, Steffen: DKIM–Kryptographie auf wackligem Boden.

[19] Whalen, Sean: An introduction to arp spoofing. In: *Node99 [Online Document], April* (2001)

[20] Wong, M. ; Schlitt, W.: *Sender Policy Framework (SPF) for Authorizing Use of Domains in E-Mail, Version 1*. 2006. – URL https://tools.ietf.org/html/rfc4408

[21] Zadgaonkar, AS ; Kashyap, Suresh ; Patel, Murari C.: Developing a model to detect e-mail address spoofing using biometrics technique. In: *Int J Sci Mod Eng (IJISME)* 1 (2013), Nr. 6, S. 63–65

# A Anhang

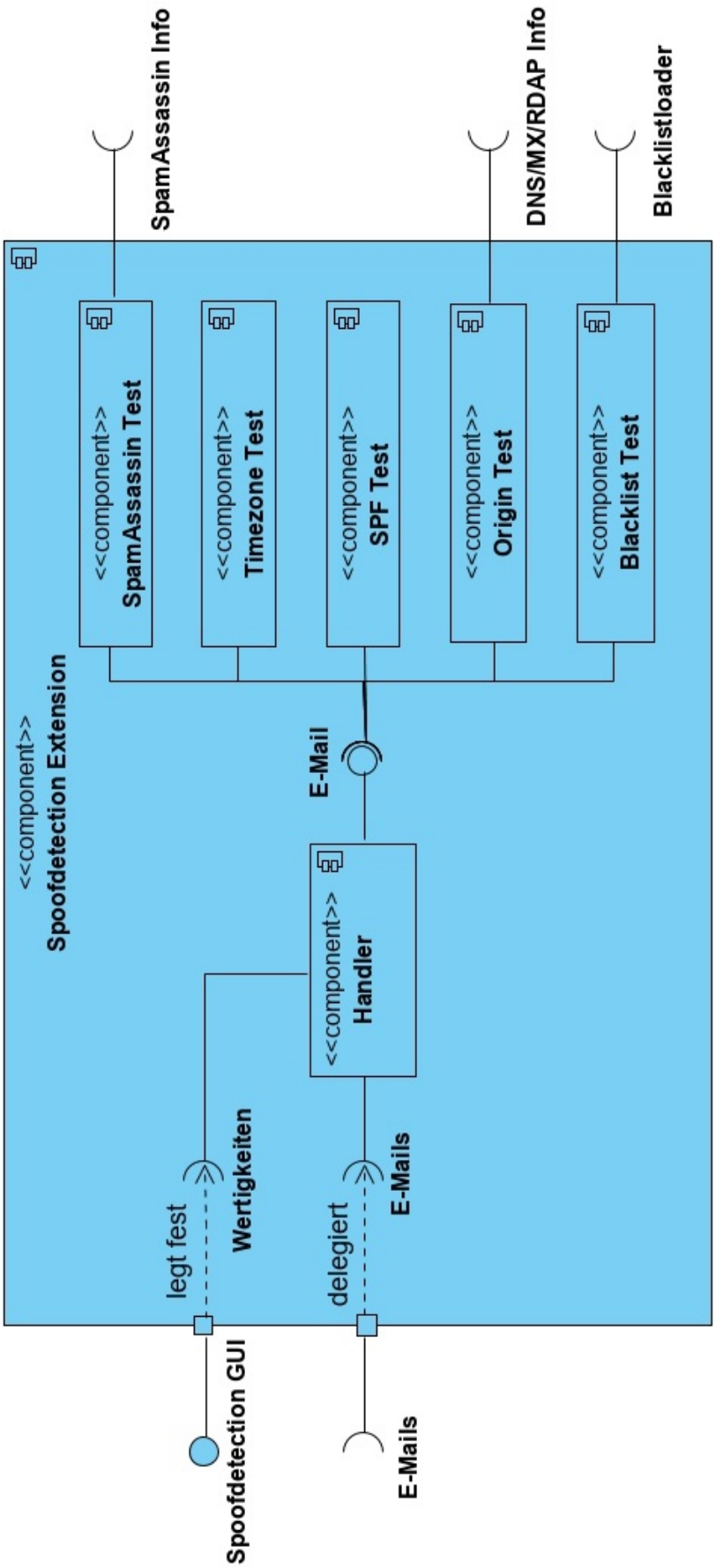



Abbildung A.1: Komponentendiagramm der Erweiterung

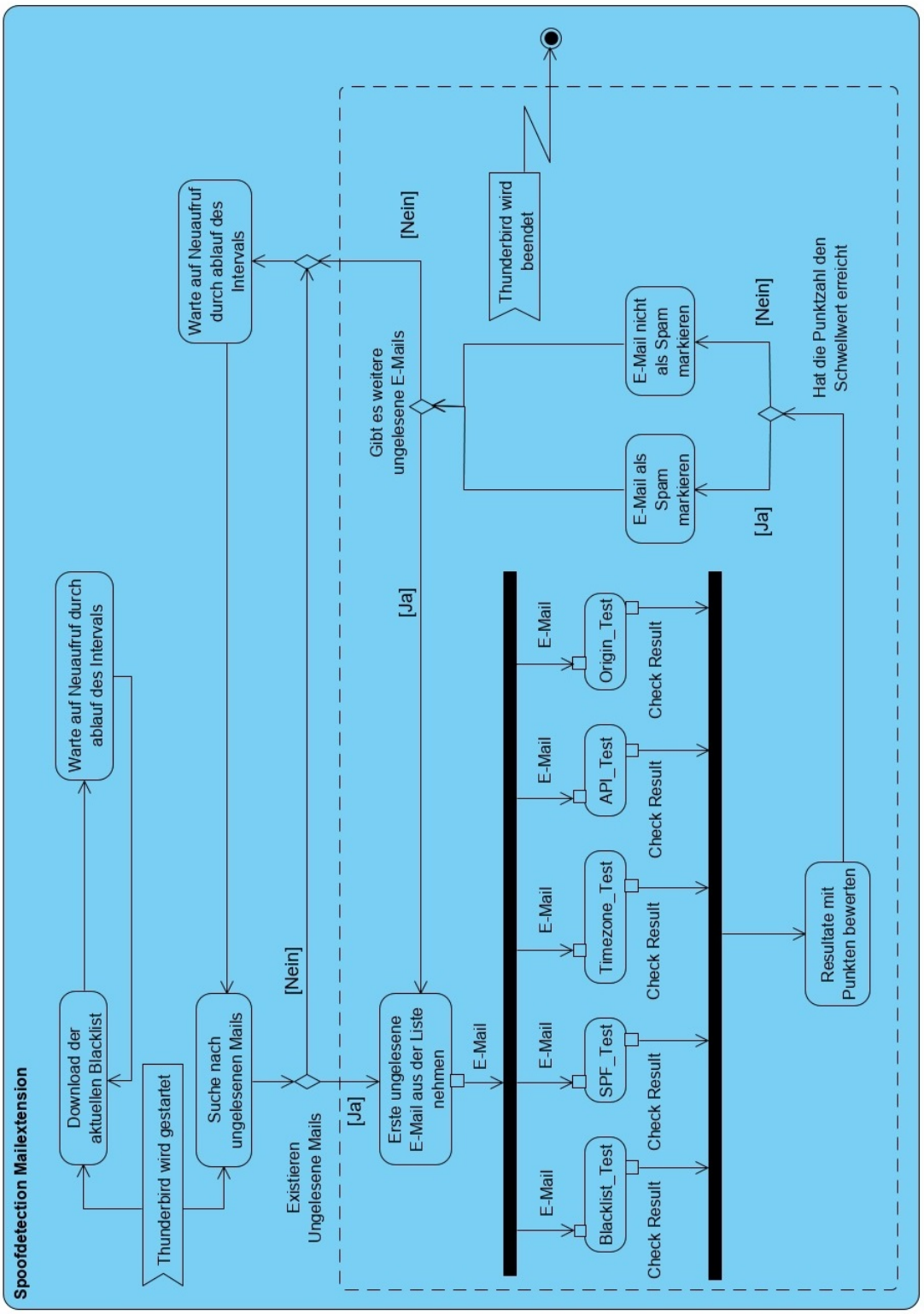



Abbildung A.2: Aktivitätsdiagramm der Erweiterung

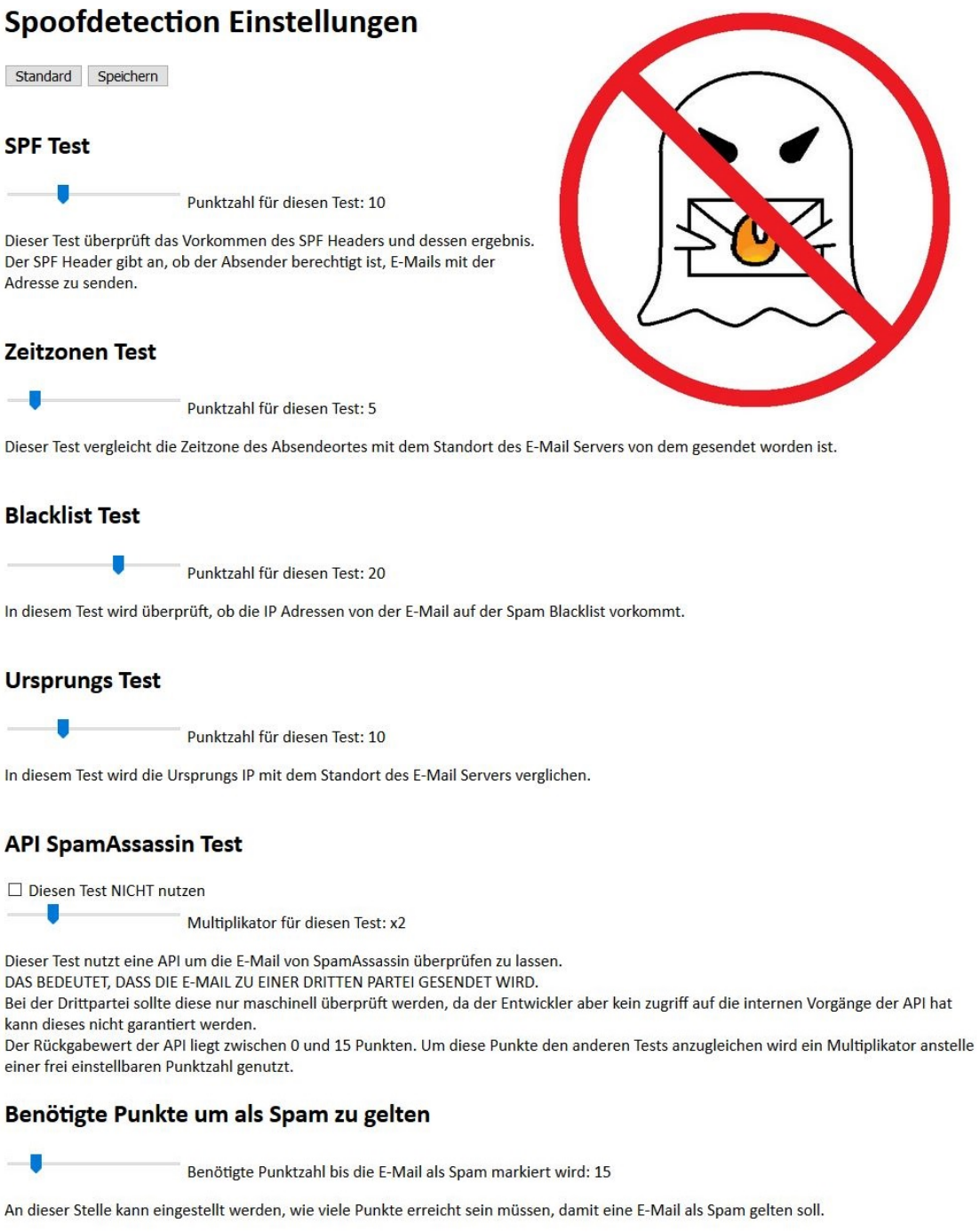



Abbildung A.3: Optionenseite der Thunderbird Erweiterung

## Erklärung zur selbstständigen Bearbeitung einer Abschlussarbeit

Gemäß der Allgemeinen Prüfungs- und Studienordnung ist zusammen mit der Abschlussarbeit eine schriftliche Erklärung abzugeben, in der der Studierende bestätigt, dass die Abschlussarbeit „— bei einer Gruppenarbeit die entsprechend gekennzeichneten Teile der Arbeit [(§ 18 Abs. 1 APSO-TI-BM bzw. § 21 Abs. 1 APSO-INGI)] — ohne fremde Hilfe selbständig verfasst und nur die angegebenen Quellen und Hilfsmittel benutzt wurden. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quellen kenntlich zu machen."

*Quelle: § 16 Abs. 5 APSO-TI-BM bzw. § 15 Abs. 6 APSO-INGI*

## Erklärung zur selbstständigen Bearbeitung der Arbeit

Hiermit versichere ich,

Name: _____

Vorname: _____

dass ich die vorliegende Bachelorarbeit – bzw. bei einer Gruppenarbeit die entsprechend gekennzeichneten Teile der Arbeit – mit dem Thema:

**Konzipierung und Implementierung einer Erweiterung zum Erkennen von Mails mit gefälschtem Absender für den Mail User Agent Thunderbird**

ohne fremde Hilfe selbständig verfasst und nur die angegebenen Quellen und Hilfsmittel benutzt habe. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quellen kenntlich gemacht.

Ort Datum Unterschrift im Original