

HOCHSCHULE DER MEDIEN STUTTGART

BACHELORARBEIT IM STUDIENGANG

MEDIENINFORMATIK

**Analyse und Ausarbeitung neuer Ansätze
zur Verhinderung von Schadsoftware im
E-Mailverkehr**

Vorgelegt von Stephan Traub

Matrikelnummer 26813

zur Erlangung des akademischen Grades eines
Bachelor of Science

am 17.07.2016

Erstbetreuer:

Prof. Dr. Martin Goik
Hochschule der Medien

Zweitbetreuer:

Dipl.-Ing. Benjamin Kenner
audius GmbH

Eidesstattliche Erklärung

„Hiermit versichere ich, Stephan Traub, ehrenwörtlich, dass ich die vorliegende Bachelorarbeit mit dem Titel: „Analyse und Ausarbeitung neuer Ansätze zur Verhinderung von Schadsoftware im E-Mailverkehr“ selbstständig und ohne fremde Hilfe verfasst und keine anderen als die angegebenen Hilfsmittel benutzt habe. Die Stellen der Arbeit, die dem Wortlaut oder dem Sinn nach anderen Werken entnommen wurden, sind in jedem Fall unter Angabe der Quelle kenntlich gemacht. Die Arbeit ist noch nicht veröffentlicht oder in anderer Form als Prüfungsleistung vorgelegt worden. Ich habe die Bedeutung der ehrenwörtlichen Versicherung und die prüfungsrechtlichen Folgen (§ 26 Abs. 2 Bachelor-SPO (6 Semester), § 24 Abs. 2 Bachelor-SPO (7 Semester), § 23 Abs. 2 Master-SPO (3 Semester) bzw. § 19 Abs. 2 Master-SPO (4 Semester und berufsbegleitend) der HdM) einer unrichtigen oder unvollständigen ehrenwörtlichen Versicherung zur Kenntnis genommen.“

Stephan Traub

Datum

Abstract

Over 90 per cent of assaults on computer users issue from a simple email. Our strong addiction to this communication system, in business environment as well as in private shpere, provides a huge surface for malware attacks. Current anti-malware filters can't provide a satisfactory to detection rate of these malware. This paper describes new possibilities for a dynamic recognition of the growing threat in the email traffic. Owing to these new approaches, they significantly increase the detection rates of the email filters of new forms of malware. The enhancement for this new approaches of both efficiency and the improved detection is currently outlined against filter technologies by measurements. This work focuses on prevailing existing threats relating to small to midsize email infrastructures.

Kurzfassung

Über 90% der Angriffe auf Computerbenutzer können auf eine E-Mail zurückgeführt werden. Durch die starke Abhängigkeit von diesem Kommunikationsmittel, sowohl im betrieblichen als auch im privaten Umfeld, bietet dies ein effektives Einfallstor für Schadsoftware. Aktuelle Anti-Malware-Filter können in vielen Fällen keine zufriedenstellende Erkennungsrate dieser Schadsoftware bieten. Diese Arbeit zeigt neue Möglichkeiten zur dynamischen Erkennung der stetig wachsenden Bedrohung im E-Mailverkehr auf. Durch diese neuen Ansätze werden die Erkennungsraten der E-Mailfilter signifikant gesteigert und so neue Formen von Schadsoftware verhindert. Die Steigerung der Effizienz und die verbesserte Erkennung werden gegen aktuell eingesetzte Filtertechnologien mittels Messungen verglichen. Im Fokus stehen hier aktuell bestehende Bedrohungen für kleine bis mittelgroße E-Mail-Infrastrukturen.

Inhaltsverzeichnis

Abbildungsverzeichnis	v
Tabellenverzeichnis	vii
Abkürzungsverzeichnis	viii
1 Übersicht	1
1.1 Zielsetzung	1
1.2 Struktur	2
1.3 Stil der Arbeit	2
2 Einführung	3
2.1 Definitionen	3
2.2 E-Mail	5
2.3 Angriffsvektor E-Mail	7
2.4 Beispiel eines Angriffsablaufs	9
2.4.1 Praktische Bezugnahme	12
3 Ausgangslage	14
3.1 Testumfeld	14
3.2 E-Mail-Filterung	16
3.2.1 Methoden	16
3.2.2 Vorgang	17
3.2.3 Filtertechnologien	20
3.3 Messungen	24
3.3.1 Ergebnisse	24

3.3.2	Zusammenfassung der Messungen	28
3.4	Darstellung des Defizits	28
4	Neue Ansätze	31
4.1	Rückschlüsse aus den Messergebnissen	31
4.2	Neue Wege der Erkennung	31
4.2.1	Verhinderung von Spam durch bekannte Absender	32
4.2.2	Verhinderung von Makro-Malware	35
5	Schlussbetrachtung	44
5.1	Vergleich zu anderen Ansätzen	44
5.2	Zusammenfassung	46
5.3	Diskussion	47
5.4	Weiterführende Arbeit	49
5.5	Danksagung	49
	Glossar	50
	Literatur	53

Abbildungsverzeichnis

1	Teilmengendiagramm von Spam. Quelle: Autor	4
2	Phishing E-Mail, bei der versucht wird, die Anmeldedaten für einen Bankzugang zu erhalten. Quelle: Wikipedia [52]	8
3	Spam E-Mail mit Malware im Anhang. Quelle: Autor	9
4	Domain-Abfrage beim Domain-Registrar. Quelle: Autor	10
5	Angriffskette der Malware. Quelle: Autor	12
6	Eine Aufforderung für die Lösegeldzahlung nach der erfolgreichen Verschlüsselung. Quelle: Wikipedia [51]	13
7	Meldung auf der Firmenwebseite, nachdem Malware von dessen Absenderadresse versendet wurde. Quelle: Autor	13
8	Übersicht der MTA Struktur des Messsystems. Quelle: Autor	17
9	Filterstruktur innerhalb des Messsystems. Quelle: Autor	18
10	Spam-Kampagnen mit Malware als Anhang - Zeitraum 23.04-23.05.2016. Quelle: Autor	22
11	Anzahl der E-Mails am Messsystem mit entsprechenden Dateiendungen. Quelle: Autor	26
12	Filterung des Spamassassin. Quelle: Autor	27
13	Spam-Report von Symantec. Quelle: Symantec [41]	28
14	Für den Benutzer offensichtliche Spam-Nachricht. Quelle: Autor	32
15	Konzept für die Einbindung der Absenderdatenbank. Quelle: Autor	35
16	VBA-Code einer Malware. Quelle: Autor	37
17	Position des <i>MacroMilters</i> im E-Mailfluss. Quelle: Autor	40
18	Position des <i>MacroMilters</i> in der Filterkette. Quelle: Autor	41

19	Kombination des <i>Amavisd-New</i> und <i>MacroMilters</i> für die zurück- gewiesenen E-Mails mit den entsprechenden Dateiendungen. Quelle: Autor	42
20	Anzahl der eindeutigen Malware-Dokumente am Honeypot-System im Monat Mai 2016. Quelle: Autor	43
21	Anteil der vom MacroMilter erkannten Malware. Quelle: Autor	46

Tabellenverzeichnis

1	Übersicht der Messsysteme	15
2	Empfangene Nachrichten im Messzeitraum.	25
3	Aufteilung der zurückgewiesenen Nachrichten	27
4	Anzahl der legitimen und Malware Dokumenten mit Auftreten der Funktionen	38
5	Ansatz zur Gewichtung der Funktionsaufrufe im Makro-Code	39

Abkürzungsverzeichnis

FBI Federal Bureau of Investigation

FQDN Fully Qualified Domain Name

HTTP Hypertext Transfer Protocol

IT Informationstechnik

MDA Mail Delivery Agent

MIME Multipurpose Internet Mail Extensions

MTA Mail Transfer Agent

RBL Real-time Blackhole List

RFC Request For Comments

SMTP Simple Mail Transfer Protocol

URI Uniform Resource Identifier

VBA Visual Basic Access

1 Übersicht

Seit der Erfindung der E-Mail wächst die Anzahl an E-Mailadressen und täglich versendeten Nachrichten stetig. Heutzutage ist es nicht mehr unüblich, dass eine einzelne Person mehrere E-Mailadressen besitzt. Fast schon selbstverständlich hat jeder zumeist zwei für die jeweils geschäftliche und private Kommunikation.

Diese Beliebtheit und ebenso weite Verbreitung lässt sie in den Fokus für missbräuchliche Verwendung rücken. Häufig wird Schadsoftware als Anhang oder Link getarnt einer E-Mail angefügt, um diese so an möglichst viele Benutzer zu verteilen. Durch die hohe Anzahl an Nachrichten, die jeder Benutzer täglich erhält, ist eine automatisierte Filterung unerlässlich. Diese Filter müssen maschinell bewerten, welche dieser Nachrichten der Benutzer erhalten möchte und welche nicht. Um diese feine Abstimmung zu finden, werden unterschiedliche Filtermethoden wie Spam- oder Anti-Malware-Filter eingesetzt. Trotz dem Einsatz von solchen Filter kommt es dennoch zu Infektionen mit Schadsoftware, die sich als E-Mailanhang getarnt hatte.

1.1 Zielsetzung

Diese Arbeit soll aufzeigen, welche Defizite im Hinblick auf die Verhinderung von Schadsoftware innerhalb E-Mailinfrastrukturen mit bis zu 1000 Benutzern existieren. Genauer werden hierbei die beiden Filtermethoden der weitverbreiteten E-Mail-

Filter *Spamassassin*¹ und *Amavisd-New*² sowie dem E-Mailserver *Postfix*³ betrachtet. Durch das Ausarbeiten neuer Ansätze, soll die Erkennungsrate der Filter signifikant gesteigert und so neue Formen von Schadsoftware innerhalb des E-Mailverkehrs verhindert werden. Der Fokus wird auf die Erkennung von Spam sowie die Analyse von Anhängen gelegt.

An einem produktiven System soll anhand eines direkten Vergleichs von Messungen die Effizienz der Ansätze aufgezeigt werden.

1.2 Struktur

Zunächst soll eine Basis an Verständnis geschaffen und der Problembetrachtung vermittelt werden. Im nächsten Schritt werden die bestehenden Angriffsvektoren der E-Mail, die für diese Arbeit als relevant betrachtet werden, bestimmt. Zusätzlich wird in Kapitel 3 ein Standard für ein Messverfahren definiert, der für die Bestimmung der Ausgangssituation herangezogen wird. Durch die Analyse und Aufarbeitung der Messergebnisse können bestehende Defizite erkannt und evaluiert werden. Mit Hilfe dieser Ergebnisse soll durch die Optimierung und Anwendung neuer Ansätze in Kapitel 4 eine Verbesserung der Erkennungsrate von Spam und Schadsoftware im E-Mailverkehr erreicht werden.

1.3 Stil der Arbeit

Viele der aufgeführten Quellen sind in englischer Sprache verfasst. Beruhend auf der Etablierung von englischen Begriffen in der IT wurden diese in der Arbeit nicht übersetzt. Häufig würde hier eine Übersetzung nicht zur Verständlichkeit des Textes beitragen. Entsprechende Erläuterungen sind für themenspezifische Begriffe im Glossar zu finden. Firmen-, Produkt- sowie Softwarenamen werden *kursiv* kenntlich gemacht. Zusätzlich wird bei der Nennung von Software auch eine Fußnote mit der entsprechenden Webseite angegeben. Programmcodes, Konfigurationsparameter oder Dateipfade werden in **mono-type** kenntlich gemacht.

¹<https://spamassassin.apache.org/>

²<https://www.ijs.si/software/amavisd/>

³<http://www.postfix.org/>

2 Einführung

Grundlegend müssen zunächst Begrifflichkeiten, die eine zentrale Rolle in dieser Arbeit spielen, beschrieben werden. Nur so kann ein eingehendes Verständnis für die Thematik vermittelt werden. Zusätzlich werden die existierenden Angriffsvektoren innerhalb eines E-Mailverkehrs aufgezeigt. Hierzu sind in diesem Kapitel mehrere reale und genutzte Angriffsarten via E-Mail dargestellt.

2.1 Definitionen

Schadsoftware: Hierbei handelt es sich um Programme, die meist mit einem kriminellen Hintergrund einwickelt wurden. Innerhalb der Arbeit wird der Fokus auf die Schadsoftware gelegt, die via E-Mail verteilt wird.

Im Umfeld der Arbeit wird das englische Akronym Malware (*malicious software*) als Synonym für Schadsoftware verwendet. Schadsoftware und Malware sind gleichbedeutend zu betrachten.

Spam: Eine genaue Definition von Spam ist schwierig zu erfassen. Verallgemeinert wird Spam als unerwünschte und ungewollte E-Mail bezeichnet. Zumeist handelt es sich um Nachrichten, die vom Benutzer nicht erwartet oder von ihm nicht in Verbindung mit einem Vorgang gebracht werden können [7, S. 8]. Ein Beispiel wäre hier Werbung eines Online-Shops, bei dem der Benutzer jedoch noch nie eingekauft hat. Des weiteren kann Spam auch einen pornographischen oder kriminellen Hintergrund haben.

Innerhalb der Arbeit wird der Verbreitungsweg von Schadsoftware durch E-Mails be-

trachtet. Hierbei wird Spam als Transportmittel der Schadsoftware definiert. Somit ergibt sich eine Verkettung zwischen der Verhinderung von Spam und Malware. Wird eine Nachricht aufgrund ihrer Eigenschaften schon als Spam erkannt, ist eine Klassifizierung des Anhangs durch einen Malware-Scanner nicht mehr notwendig.

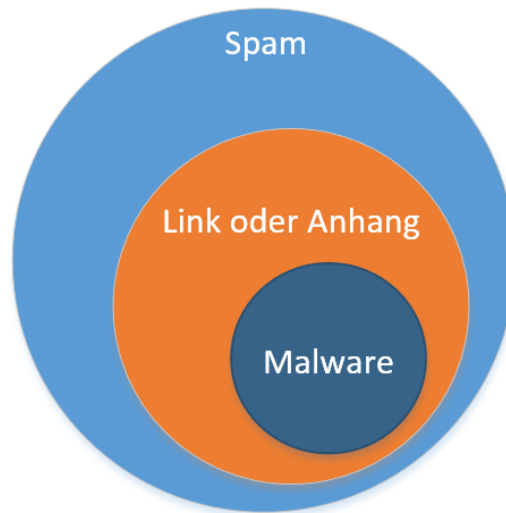


Abbildung 1: Teilmengendiagramm von Spam. Quelle: Autor

Wie in Abbildung 1 zu sehen ist, führt eine Erkennung von Spam die Vermeidung von Malware herbei, da Malware eine Teilmenge der Spam-Nachrichten ist.

Durch diese Kapselung der Inhalte, entsteht eine Art Kette mit folgenden Schritten:

1. Wird die E-Mail aufgrund der Verbindungsinformationen als Spam erkannt, ist eine genauere Analyse des Anhangs oder E-Mailtextes nicht mehr nötig.
2. Wird jedoch die Nachricht nicht im ersten Schritt erkannt, muss eine Analyse des Textes auf Spam-Muster durchgeführt werden.
3. Erbringen alle vorherigen Schritte keine Klassifizierung, muss der Anhang explizit auf Verhaltens- und Inhaltsmuster von Malware überprüft werden.

2.2 E-Mail

Im Jahr 2015 beliefen sich die an einem Tag versendeten E-Mails auf 205,6 Milliarden, wobei 112,5 Milliarden im Arbeitsumfeld empfangen und gesendet wurden [43, S. 4]. Das auf dem Simple Mail Transfer Protocol (SMTP) basierende Kommunikationsmittel ist mitunter ein des am meisten genutzten Mediums zur Übermittlung von Nachrichten.

Dadurch, dass die E-Mail eine zentrale Komponente der täglichen Kommunikation darstellt, sind die verarbeitenden Infrastrukturen mit einer hohen Anzahl an E-Mailobjekten konfrontiert. Hier stellt sich die Herausforderung, die legitimen Ham-E-Mails von ungewolltem Spam zu unterscheiden. Das Ziel muss sein, die Anzahl an Spam-Nachrichten, die fälschlicherweise als Ham erkannt wurden (False-Positiv), so gering wie möglich zu halten. Auf der anderen Seite darf die False-Negativ-Rate nicht steigen. Diese Rate repräsentiert die Anzahl der E-Mails, die Spam oder Malware enthalten aber dennoch an den Benutzer zugestellt wurden. Hierbei konnten diese nicht von den Filterinstanzen als solche erkannt werden.

Grundsätzlich wird eine höhere False-Negativ-Rate in Kauf genommen. Der Benutzer erhält zwar die Spam-E-Mail in seinem Postfach, kann sie dann aber bei Bedarf löschen. Wenn die False-Positiv-Rate aber zu hoch ist, bekommt der Benutzer legitime E-Mails nicht mehr. Bei den täglich empfangenen Mengen an Nachrichten, könnte hier bereits eine leichte Erhöhung der beiden Raten schon tausende E-Mails betreffen.

Bezug genommen auf diese Arbeit, dass jede Spam-Nachricht auch Malware enthalten kann, müssen die Raten weiter aufgeschlüsselt werden. Wird eine höhere False-Negativ-Rate bei der Malware-Filterung eingesetzt, besteht eine höhere Wahrscheinlichkeit, dass der Benutzer Malware innerhalb der E-Mail erhält. In einer Betrachtung auf die Verhinderung von Malware, würde eine höhere False-Positiv-Rate einen besseren Effekt erzielen, da durch strikteres Zurückweisen von Nachrichten das Gefahrenpotential jeder E-Mail sinkt.

Die E-Mail selbst enthält neben dem Absender, Empfänger und dem E-Mailtext auch weitere für die Filterung wichtige Metainformationen. Alle Informationen werden innerhalb des MIME-Baums der Nachricht gespeichert. Die RFC 2822 bestimmt hierbei, in welcher Form diese Informationen angegeben werden müssen [37, S. 5ff].

Eine Möglichkeit, diesen Baum mit eigenen Knoten zu erweitern um weitere Informationen über die E-Mail zu hinterlegen, ist gegeben. Diese Knoten werden innerhalb einer Nachricht als MIME-Header oder E-Mail-Header bezeichnet. Im Falle der E-Mail-Filter wird die Erweiterungsmöglichkeit der E-Mail-Header gerne für die Angabe der Klassifizierung einer E-Mail verwendet. Somit kann z.B. eine Spam-E-Mail über einen Eintrag in den MIME-Baum als solche klassifiziert werden. Alle späteren Filterinstanzen und Benutzeranwendungen können diese Werte auslesen und eventuell neue zu setzen.

Listing 2.1: E-Mailheader einer legitimen E-Mail mit Metainformationen.

```
Received: from <entfernt>
Received: by <entfernt>
X-Spam-Flag: NO
X-Spam-Status: No, score=-0.5 required=5.0
X-Spam-Checker-Version: SpamAssassin 3.4.0 (2014-02-07) on <entfernt>
X-Spam-Tests: BAYES_00=-1.9,DKIM_SIGNED=0.1,DKIM_VALID=-0.1,DKIM_VALID_AU=-0.1,
RCVD_IN_MSPIKE_H3=-1.5,RCVD_IN_MSPIKE_WL=-0.01
X-Spam-RBL: <dns:49.210.146.62.wl.mailspike.net> [127.0.0.18]
X-Spam-Timing: total 388 ms - read_scoreonly_config: 0.97 (0.3%),
signal_user_changed: 1.77 (0.5%), parse: 0.72 (0.2%),
extract_message_metadata: 13 (3.4%), get_uri_detail_list: 4.9 (1.3%)
X-Spam-RemoteHelo: <entfernt>
X-Spam-RelaysUntrusted:<entfernt>
X-Virus-Scanned: <entfernt>
Received: <entfernt>
From: <entfernt>
To: <entfernt>
Subject: <entfernt>
Date: <entfernt>
Message-ID: <entfernt>
References: <entfernt>
In-Reply-To: <entfernt>
Accept-Language: en-GB, de-DE, en-US
Content-Language: en-US
Content-Type: text/plain; charset="iso-8859-1"
Content-Transfer-Encoding: quoted-printable
MIME-Version: 1.0
Return-Path: <entfernt>
```

Im Listing 2.1 ist ein solcher MIME-Baum mit den entsprechenden Metainformationen zu sehen. Hier sind alle, vom Spam- oder Malware-Filter hinzugefügte Felder mit einem **X-** gekennzeichnet. Diese beschreiben die erfolgenden Filtermethoden mit ihren Ergebnissen für diese Nachricht. So kann z.B. über die Auswertung des Feldes **X-Spam-Status** ein E-Mailprogramm diese sofort als Spam erkennen.

2.3 Angriffsvektor E-Mail

Die E-Mail bildet in 91% aller Angriffe den erste Kontaktpunkt zum Opfer [21, S. 3]. 30% aller Empfänger lesen E-Mails mit angehängter Schadsoftware oder einer Verlinkung auf diese. Hiervon öffnen 12% die angehängten oder verlinkten Malware-Programme und führen diese aus [49, S. 18].

Wie beschrieben, werden Links auch gerne für die Verbreitung von Malware genutzt. Der Benutzer wird auch hier dazu verleitet, auf diesen zu klicken um auf eine Webseite mit integriertem Malware-Download geleitet zu werden [35, S. 5f]. Im schlechtesten Fall wird die Malware durch eine bestehende Sicherheitslücke im Browser auf dem System installiert. Der Benutzer bekommt hiervon nichts mit. Um eine höhere Wahrscheinlichkeit für eine erfolgreiche Infektion zu bekommen, werden häufig Links und Anhänger kombiniert in eine Spam-E-Mail integriert.

Der Telekommunikationskonzern Verizon hat bei einer Analyse im Jahr 2015 festgestellt, dass die E-Mail eines der am häufigsten verwendeten Transportmittel ist, um Malware dem User zu übermitteln [49, S. 46]. Hieraus resultiert, dass die E-Mail eine der wichtigsten Verbreitungswege für Schadcode ist. Dies unterstreicht nochmals die Wichtigkeit der Filtermethoden innerhalb der E-Mailinfrastrukturen, um solche Angriffe zu unterbinden.

Dieser Trend kann schon seit einiger Zeit beobachtet werden. Gründe für die wachsende Zahl an Angriffen via E-Mail sind, dass Schwachstellen in einen Computer einzudringen, nicht mehr so einfach auszunutzen sind. Durch schnellere Update-Zyklen für Sicherheitspatches und stärkere Verschlüsselungen innerhalb von Netzwerkprotokollen rückt der Mensch immer weiter in den Fokus, als Schwachstelle missbraucht zu werden [15, S. 2]. Die Möglichkeiten, Malware in Anhängen und Links zu integrieren, bietet die E-Mail eine interessante Plattform zur Verbreitung. Durch trickreiches Formulieren und Formatieren einer Nachricht, kann der Benutzer zusätzlich zu einer ungewollten Aktion, das Ausführen der Malware, gedrängt werden.

Über die E-Mail wird nicht nur Schadcode verteilt, sondern auch versucht, vom Benutzer vertrauliche Daten abzugreifen. Hierbei handelt es sich zumeist um Zugangsdaten für Online-Shops, E-Mailpostfächer oder wie in Abbildung 2 gezeigt zu einer Bank. Gibt der Benutzer auf der gefälschten Webseite seine Zugangsdaten ein und versucht sich so anzumelden, erhält der Angreifer alle eingetragenen Daten wie

Passwörter oder Benutzernamen. Diese Unterform von Spam wird Phishing genannt aber innerhalb dieser Arbeit nicht weiter beleuchtet, da hierdurch keine Malware verbreitet wird.



Abbildung 2: Phishing E-Mail, bei der versucht wird, die Anmeldedaten für einen Bankzugang zu erhalten. Quelle: Wikipedia [52]

Ein weiterer Punkt ist, dass es innerhalb einer E-Mail keinerlei Integritätssicherheit gibt. Das heißt, dass ein Empfänger zu keiner Zeit sichergehen kann, dass der angezeigte Absender auch der ist, der die E-Mail versendet hat [33, S. 9f]. In diesem Fall kann nur der Empfänger selbst evaluieren, ob er den Inhalt der E-Mail mit dem Absender in Verbindung bringen kann. Eine serverseitige Evaluierung des Absenders wird nur rund von der Hälfte der *Alexa Top Million Domains*¹ durchgeführt [12, S. 1]. Im Falle von Kleinstsystemen im mittelständischen oder privaten Bereich ist die Zahl noch geringer einzuschätzen.

Grundlegend bildet die E-Mail durch die fehlende Integritätssicherheit und die weite Verbreitung eine sehr große Angriffsfläche. Die Flexibilität und der weitreichende Funktionsumfang des Inhalts einer E-Mail machen es schwer für Filtersysteme, Spam und Malware von legitimen Nachrichten zu unterscheiden.

¹<http://www.alex.com/topsites>

2.4 Beispiel eines Angriffsablaufs

In vielen Fällen beginnt der Angriff mit einer vermeintlich legitimen E-Mail an den Empfänger. Häufig werden gängige Themen wie eine Rechnungsstellung oder Mahnungen als Vorwand verwendet. Diese Inhalte sind nicht nur für jeden Empfänger aus erster Sicht legitim, sondern vermitteln auch eine gewisse Brisanz. Dieser digitalisierte Vorgang ist heutzutage zudem bei einer papierlosen Rechnungsstellung vollkommen üblich.

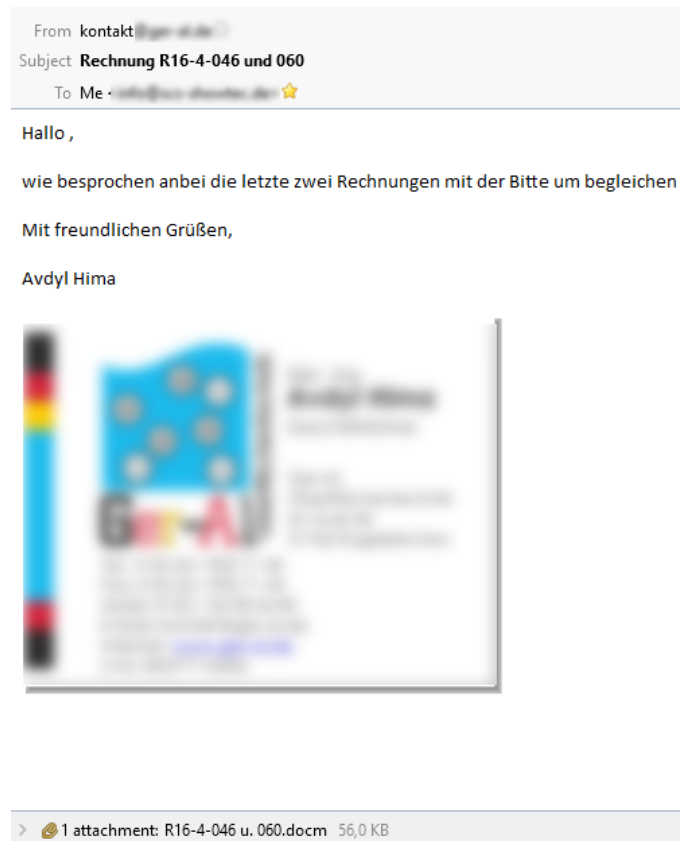



Abbildung 3: Spam E-Mail mit Malware im Anhang. Quelle: Autor

Abbildung 3 zeigt eine solche E-Mail mit einem Malware-Anhang. Für einen unbedarften Benutzer ist es hier nicht ersichtlich, dass es sich bei dem Anhang um Schadsoftware handeln muss. Dieser Anhang wird häufig durch ein *Microsoft Office* Dokument repräsentiert. Es stellen für die meisten Benutzer einen vertrauten Dateityp dar. Häufig werden hier *Microsoft Word* oder *Microsoft Excel* Dokumente verwendet [34, S. 19].

Um die Vertraulichkeit zu erhöhen, wird in vielen Fällen der Absender der E-Mail vom Malware-Versender gefälscht. Somit kann eine Betrachtung des Absenders keinen Rückschluss auf die Echtheit der Nachricht geben. Selbst für einen erfahrenen Benutzer ist es fast unmöglich zu ermitteln, ob es sich wirklich um eine Fälschung handelt. Beispielsweise zeigt die Abfrage des Domain-Inhabers in diesem Fall auch kein verdächtiges Ergebnis, wie in Abbildung 4 zu sehen ist. Der Inhaber ist hierbei wirklich eine legitime Firma.

Domaindaten

Domain  de
Latest update 08.07.2011

Domain holder

The domain holder is DENIC's contractual partner and hence holds the material rights to the domain.




Domain holder 
Address 
Postal code 
City Engelskirchen
Country DE

Abbildung 4: Domain-Abfrage beim Domain-Registrar. Quelle: Autor

Dem rein textlichen Inhalt der E-Mail ist somit keine bösartige Eigenschaft zuzuschreiben. Im Fall des Anhangs oder Links wird die Schädlichkeit erst nach genauerer Analyse sichtbar.

Im Optimalfall wird die Datei beim Abspeichern auf dem Rechner des Benutzers schon durch den lokalen Virenschanner erkannt. Durch die schnellen Abwandlungen der Schadsoftware durch die Entwickler und die zumeist nur einmal täglich aktualisierten Malware-Scanner, kann eine Erkennung des Scanners ausbleiben. Im schlimmsten Fall ist kein Malware-Scanner auf dem Zielrechner installiert. Eine serverseitige Überprüfung schlägt in vielen Fällen aus denselben Gründen fehl (siehe Kapitel 3.2.3).

Die eigentliche Gefahr geht im Falle der *Microsoft Office* Dokumente nicht vom Dokument selbst aus, sondern von der Möglichkeit, ein Makro-Code hinzuzufügen. Ursprünglich waren und sind diese Makros in den Dokumente für Erweiterungen gedacht, die vom Verarbeitungsprogramm selbst nicht bereitgestellt werden. So kann mit Visual Basic Access (VBA) ein Programmcode angefügt werden und so erwei-

terte Funktionen innerhalb des Dokuments verwendet werden. Durch die großen Funktionsumfang von VBA bietet dies natürlich auch einen großen Angriffsvektor und Möglichkeiten für Malware-Entwickler, Schadcode direkt in das Dokument einzubetten.

Durch Analysewerkzeuge wie dem *Office Malware Scanner*² von Frank Boldewin, kann das Dokument in seine Bestandteile zerlegt werden. Durch diese Verarbeitung wird bei einem *Microsoft Office* Dokument der VBA-Code vom restlichen Text getrennt. Somit wird die Möglichkeit geschaffen, den Code zu analysieren, ohne das Dokument öffnen zu müssen.

Die eigentliche Analyse der Code-Funktionen stellt sich in vielen Fällen als sehr schwer heraus. Grundsätzlich wird von den Malware-Entwicklern versucht, die eigentliche Funktion stark zu verschleiern [8]. Durch Konvertierung sowie Verschachtelungen von Funktionen und Code-Teilen wird hier versucht, einen Höchstgrad an Verschleierung zu erzeugen [26] und [30, S.6 f]. Dieses Vorgehen soll die Erkennung bei einem auf Pattern oder Heuristiken basierenden Anti-Malware-Filter unterbinden.

Der Makro-Code selbst bildet nur eine Art Zwischenhändler oder auch Dropper genannt [18, S. 82]. Dieser infiziert das System und lädt danach von einem anderen, im Internet stehenden System, eine weitere Schadsoftware nach [30, S. 8]. Diese wird dann auf dem Rechner ausgeführt und stellt somit den schädlichen Teil der Malware dar. Falls dieser Vorgang Erfolg hat, ist der Rechner mit Schadsoftware infiziert. Abbildung 5 zeigt den gesamten Angriffsverlauf in seinen einzelnen Schritten.

In den Jahren 2015 und 2016 wurden auf diesem Weg häufig Erpresser-Software, auch Ransomware genannt, verteilt. Hierbei handelt es sich um Malware, die Daten auf einem infizierten Rechner mit einer starken Verschlüsselung verschlüsselt und sie so zunächst unbrauchbar macht [2, S. 7]. Das Opfer wird nach der erfolgreichen Verschlüsselung seiner Daten dann zur Bezahlung eines Geldbetrags aufgefordert. Nachdem der Erpresser diesen Betrag vom Opfer erhalten hat, wird der Entschlüsselungsvorgang vom Malware-Entwickler eingeleitet und das System durch diesen Vorgang wieder verwendbar gemacht [14, S. 87].

²<http://www.reconstructor.org/code.html>

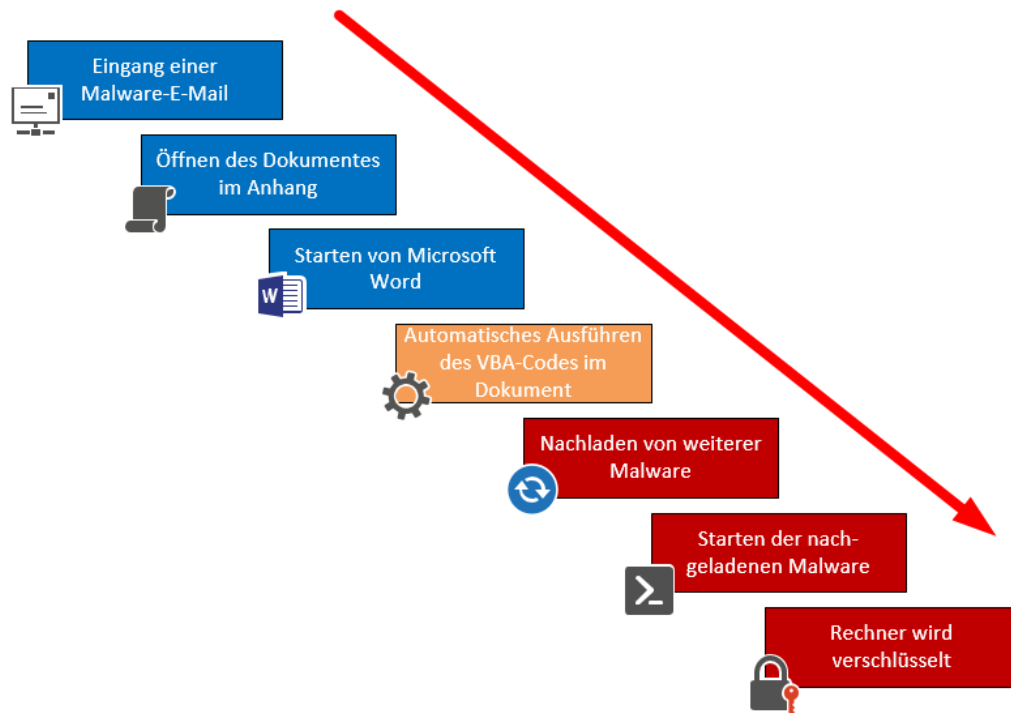


Abbildung 5: Angriffskette der Malware. Quelle: Autor

2.4.1 Praktische Bezugnahme

In vielen Fällen hat eine Infizierung mit Ransomware verheerende Folgen. In Berichten aus Medien, sind hierbei durch die Kampagnen der Malware-Entwickler einige öffentlichen Stellen infiziert worden [22]. Dadurch, dass die Ransomware in vielen Fällen für den Arbeitsablauf wichtige Daten verschlüsselte, wurde auch häufig das vom Malware-Entwickler geforderte Lösegeld bezahlt [6].

Der Einsatz einer Lösegeldforderung macht diese Art der Kriminalität sehr lukrativ. Durch die einfache Verbreitungsmöglichkeit via E-Mail und der nahezu anonymen Zahlweise via BitCoin bietet sich hier eine ideale Plattform. Laut einer Hochrechnung des Forbes [17] und des Cisco Tales Security Teams [27], kann eine Kampagnen innerhalb eines Tages bis zu 1.093.590 \$ (1 BitCoin = 419 \$ - Kurs am 02.03.2015) erwirtschaften. Das FBI ermittelte von April 2014 bis Juni 2015, dass Erpresser Werte von rund 18 Millionen US-Dollar eingenommen haben [16].

Durch die Möglichkeit, solch immense Summen erreichen zu können, hat sich ein neues Geschäftsmodell entwickelt, womit auch die Anzahl und Professionalität

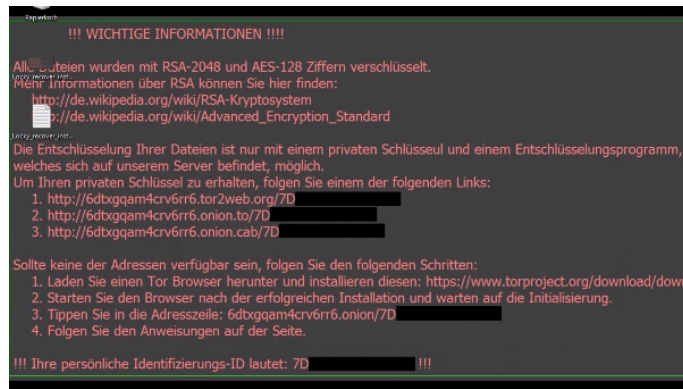


Abbildung 6: Eine Aufforderung für die Lösegeldzahlung nach der erfolgreichen Verschlüsselung. Quelle: Wikipedia [51]

der Kampagnen steigt. Diese Entwicklung zeigt sich auch in Deutschland, bei der nicht nur im Namen von legitimen und existierenden Firmen die Erpresser-Malware versendet, sondern auch ein Imageschaden für die Firma selbst entsteht (siehe Abbildung 7).



Abbildung 7: Meldung auf der Firmenwebseite, nachdem Malware von dessen Absenderadresse versendet wurde. Quelle: Autor

Nicht nur der Schaden für die Reputation der Firma ist enorm, sondern auch die rückläufigen Antworten von Empfängern sind hier nicht zu verachten. Innerhalb einer Spam-Kampagnen können einige tausend E-Mails versendet werden. Dies führte bereits in einem Fall zu unzähligen Anrufe bei der Firmenzentrale [13]. So kann ein Angreifer nicht nur seine Malware verteilen, sondern auch den vermeintlichen Absender empfindlich schädigen. Gegenmaßnahmen sind hier nahezu unmöglich.

3 Ausgangslage

3.1 Testumfeld

Innerhalb der Arbeit wird regelmäßig Bezug auf Zahlen aus dem Betriebsalltag eines E-Mailservers genommen. Hierbei handelt es sich um ein produktiven Mailserver, der täglich für den Nachrichteneingang verwendet wird.

Zu beachten ist, dass die Eigenschaften des E-Mailverkehrs stark von unterschiedlichsten Faktoren abhängen können und einer starken Varianz ausgesetzt sind. Die ermittelten Zahlen stellen also einen speziellen Fall dar. Sie lassen jedoch Tendenzen und Rückschlüsse zu, bestimmen aber keine allgemeingültigen Verhaltensmuster.

Alle Messwerte sind im Bezug auf das genannten Messsystem zu betrachten. Eine großflächige Analyse und ein Vergleich der Zahlen mit anderen Systemen ist in dieser Arbeit nicht angedacht. Um eine repräsentative Ergebnis zu erhalten, werden die ermittelten Zahlen gegen verfügbare und vergleichbare Statistiken geprüft. Diese werden an den entsprechenden Stellen referenziert.

Produktives System

Bei dem verwendeten Messsystem handelt es sich um ein aktuelles Linux System der Distribution *Ubuntu*¹ in der Version 14.04 LTS, das die in Tabelle 1 gelisteten Services und Filter installiert hat. Das System bedient rund 753 Empfänger für 16

¹<http://www.ubuntu.com/>

Domains. Im Monat werden hier eingehend rund 107.000 E-Mails verarbeitet, was ein Jahresvolumen von ca. 1.284.000 darstellt.

Service	Version	Beschreibung
postfix	2.11.0	Ein frei verfügbarer Mail Server, der unter Linux/Unix erhältlich ist.
amavisd-new	2.7.1 (20120429)	Dieser Service bietet eine Schnittstelle für weitere Filtersysteme an. Im Messsystem wird hierdurch der Virenschanner <i>ClamAV</i> eingebunden. Zusätzlich bietet er noch die Möglichkeit, ohne eine Inhaltsüberprüfung durchzuführen, gewisse Dateiendung zu blockieren. Der Service wurde als Milter implementiert.
ClamAV	0.89.1	Als Modul am <i>Amavisd-New</i> angehängte Virenschanner. Dieser prüft die Anhänge auf bekannte Viren-Pattern und Heuristiken.
Spamassassin	3.4.0	Hierbei handelt es sich um einen weit verbreiteten Spam-Filter, der durch diverse Filter- und Erkennungsmethoden Spam von Ham unterscheiden kann.

Tabelle 1: Übersicht der Messsysteme

Honeypot-System

Zusätzlich zum produktiven System, wird zur Ermittlung von verschiedenen Metainformationen noch ein Honeypot verwendet. Dieses System nimmt jede an ihn gesendete Nachricht an. Filtermethoden werden hierbei nicht angewendet. Das Honeypot-System dient nur zur Erfassung von Spam und potentieller Malware-Dateien. Diese Dateien werden für spätere Analysen herangezogen.

Bei der Installation dieses Systems, wurde auf die gleiche *Ubuntu* Version gesetzt, wie die des produktiven. Das Volumen am Honeypot, mit einer konfigurierten Test-Domain, liegt im Durchschnitt bei 7.800 E-Mails am Tag, was ca. 93.600 im Jahr entspricht.

3.2 E-Mail-Filterung

Um eine Aussage über angewendeten Messungen zu geben, ist es zunächst notwendig den Messaufbau zu beschreiben. Dieser bildet die Grundlage für die später durchgeführten Analysen und soll als Basis hierfür definiert werden. Zunächst müssen die etablierten und empfohlenen Maßnahmen zur Verhinderung von Spam und Malware im E-Mailverkehr betrachtet werden. Im Detail werden die Funktionen der freien Anti-Viren-Software *ClamAV*² sowie des frei verfügbaren Anti-Spam-Filters *Spamassassin* in Verbindung mit dem ebenfalls freien Mail Transfer Agent (MTA) *Postfix* betrachtet. Alle Komponenten werden häufig eingesetzt und bilden im Verbund ein weitverbreitetes E-Mail- und Filtersystem.

3.2.1 Methoden

Um die unterschiedlichen Schritte der Annahme einer E-Mails besser darzustellen, wurden diese hier in Abschnitte unterteilt. Jeder stellt die Möglichkeit des Einsatzes eines entsprechenden Filters dar. In welcher Ausprägung und Restriktive gefiltert wird, entscheidet der Serverbetreiber selbst. Hier gilt jedoch auch der Grundsatz, dass so wenig Spam wie möglich und so viele legitime E-Mails wie möglich, übermittelt werden.

- **Verbindungsaufbau:** Innerhalb diese Schrittes wird geprüft, ob die ankommende Verbindung eine valide Absender-Domain besitzt. Valide heißt hier z.B., dass der Domainname richtig aufgelöst werden kann. Zusätzlich wird eine Überprüfung der Absender-Domain gegen die global im Internet verfügbare Real-time Blackhole List (RBL) durchgeführt. Falls die Domain auf dieser Liste zu finden ist, wird die E-Mail zurückgewiesen.

²<https://www.clamav.net/>

- **Überprüfung des Inhaltes:** Nachdem der erste Schritt durchgeführt worden ist, wird der Inhalt der E-Mail betrachtet. Hier kommen Technologien wie Malware-Scanner für Anhänge, sowie Spam-Filter für den textlichen Inhalt der Nachricht zum Einsatz. So wird überprüft, ob sich im Anhang oder im Text unerwünschte Inhalte befinden. Die Definition von unerwünscht, muss im Falle des Spam-Filters der Administrator oder das Unternehmen selbst definieren. Diese Einstellung gelten global für alle E-Mails, die am System eintreffen.
- **Überprüfung am Rechner des Endanwenders:** In letzter Instanz wird auf dem Zielrechner des Benutzers die E-Mail überprüft. Lokal installierte E-Mail-Programme wie *Microsoft Outlook* oder *Mozilla Thunderbird* können nochmals mit eigenen Malware- oder Spam-Erkennungssoftware die E-Mail überprüfen. Diese Filterinstanz stellt eine bessere Individualität als die oben genannte globale Filterrichtlinie dar.

3.2.2 Vorgang

Bei dem Messsystem handelt es sich um ein produktives System, das tagtäglich E-Mails im Unternehmensumfeld empfängt. Dieses arbeitet als MTA in erster Instanz, was eine Verarbeitung gleichermaßen von Spam sowie für den Geschäftsablauf wichtigen E-Mails bedeutet.

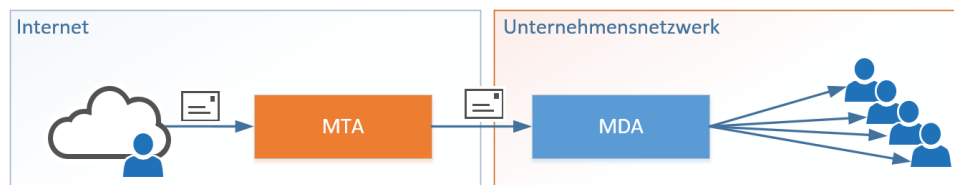


Abbildung 8: Übersicht der MTA Struktur des Messsystems. Quelle: Autor

Wie in Abbildung 8 zu sehen ist, durchläuft jede eingehende E-Mail in jedem Fall zuerst den MTA. Dieser wendet die konfigurierten Filtermethoden auf jede E-Mail an. Nach der Filterung werden die legitimen Nachrichten an den Mail Delivery Agent (MDA) übergeben, der die Zustellung an die entsprechende Mailbox des Benutzers übernimmt. Somit bildet diese Infrastruktur eine Art Kette, die zum Vorteil hat, dass der MDA mit den sensiblen Benutzerdaten nicht direkt im Internet verfügbar ist.

Innerhalb des MTA ist es möglich, die Filter in einer entsprechende Reihe anzuordnen. Diese Anordnung ist für die Verarbeitungsgeschwindigkeit und Effizienz nötig. Abbildung 9 zeigt die im Messsystem eingesetzte Anordnung.

1. Im ersten Schritt steht hier der *Postfix* selbst als Filter. Dieser bietet rudimentäre Filtermethoden und kann so schon eine schnelle grobe Vorfilterung vornehmen.
2. Der daran anschließende Filter ist der *Amavisd-New*. Dieser stellt eine granuläre Filterung dar, jedoch setzt dieser zumeist an Metainformationen der E-Mail und nicht am Inhalt an. Im Falle dieser Implementierung, wird über den *Amavisd-New* noch der Virens Scanner *ClamAV* bedient. Dieser überprüft dann erst den Anhang der Nachricht auf Schadsoftware.
3. Vor der Übergabe an den MDA, wird der Text der E-Mail noch vom *Spamassassin* überprüft. Dieser erstellt einen individuellen Spam-Level für jede Nachricht und fügt diesen dem E-Mail-Header an. Eine Auswertung dieser Level geschieht am MDA.

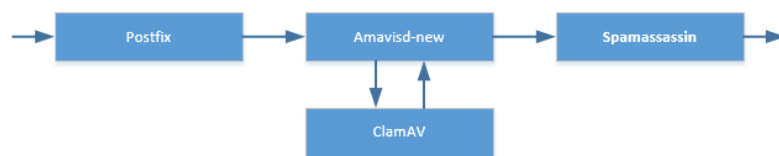


Abbildung 9: Filterstruktur innerhalb des Messsystems. Quelle: Autor

Zurückweisung

Im Falle, dass eine E-Mail vom MTA zurückgewiesen wird, erhält der sendende MTA eine Fehlermeldung vom empfangenden. Mit dieser Fehlermeldung wird dem sendenden System signalisiert, dass die Nachricht nicht erfolgreich vom Empfangssystem angenommen worden ist. Sie beinhaltet in den meisten Fällen eine Fehlermeldung und eine Fehlernummer, den sogenannten Reject-Code. Ähnliche Verfahren werden im HTTP angewendet.

Die Fehlernummern wurde innerhalb der RFC 3463 standardisiert und stellen unterschiedliche Fehlerverhalten oder Stadien dar [47, S. 2ff]. Die Text der Fehlermeldung ist hingegen frei wählbar.

Filterregeln

Im Detail werden E-Mails mit unterschiedlichen Eigenschaften zurückgewiesen, also die Annahme der Nachricht verweigert oder mit dem entsprechenden Spam-Level versehen. In welchen Fällen dies geschieht, ist in den folgenden Auflistungen dargestellt.

Zurückweisung der E-Mail durch den *Postfix* an den sendenden MTA:

- Die Empfängerdomain ist dem MTA nicht bekannt.
- Die Empfängeradresse wird nicht vom MTA verwaltet.
- Die Nachrichtengröße übersteigt 52MB.
- Der Servername des Senders ist nicht valide nach RFC 952 [20, S. 1].
- Der Servername des Senders ist kein FQDN.
- Der Sender ist keine valide Domain, also nicht zu einer IP-Adresse auflösbar.
- Der sendende Server ist als Spam-Versender bekannt und steht auf einer RBL.
- Der sendende Server hat eine dynamische IP-Adresse.

In folgenden Fällen wird die E-Mail an den sendenden MTA vom *Amavisd-new* zurückgewiesen:

- Wenn sich innerhalb der E-Mail ein Anhang mit entsprechender Dateiendung³ befindet.
- Wenn an die E-Mail eine Datei angehängt wurde, die vom Virens Scanner als Malware klassifiziert wurde.

Nachdem die eingehende E-Mail diese beiden Stationen passiert hat, wird im Messsystem noch eine Analyse des textuellen Inhaltes durchgeführt. Hierzu wird mit Hilfe des *Spamassassin* der Inhalt auf folgende Merkmale untersucht:

- Enthält bekannte Spam-URIs, die in einer im Internet verfügbaren Datenbank⁴

³ Endungen im Detail: .ade, .adp, .app, .asp, .bas, .bat, .cer, .chm, .cmd, .cnt, .com, .cpl, .crt, .csh, .der, .exe, .fxp, .gadget, .grp, .hta, .inf, .ins, .isp, .jar, .jse, .js, .ksh, .lnk, lib, .mad, .ma, .mam, .mar, .mat, .mav, .mcf, .mda, .mdb, .mde, .mdt, .mdw, .mdz, .msc, .msh, .msh1, .msh2, .mshxml, .msh1xml, .msh2xml, .msi, .msp, .mst, .ops, .osd, .pcd, .pif, .pl, .plg, .prf, .prg, .ps1, .ps1xml, .ps2xml, .psc1, .psc2, .pst, .reg, .scf, .scr, .set, .shb, .shs, .sys, .tmp, .url, .vb, .vbe, .vbp, .vbs, .vsmacros, .vsw, .vxd, .ws, .wsc, .wsf, .wsh, .xbap, .xnk.

⁴<http://uribl.com>

stehen.

- Enthält nicht-europäische Schriftzeichen.
- Statistische Prüfung des Textes gegen eine ständig aktualisierte Datenbank mit Schlüsselwörtern für Spam (Vorgehen siehe Kapitel 3.2.3).
- Prüfen der Metainformationen und E-Mail-Header, wie Absender oder Empfänger, auf Unregelmäßigkeiten.

Nachdem auch diese Instanz der Filterung passiert wurde, wird der E-Mail der Spam-Level definiert. Dieser bestimmt ob es sich bei der Nachricht um Ham oder Spam handelt. Eine Zurückweisung der E-Mail wird in diesem Schritt nicht mehr unternommen. Der Spam-Level beschreibt, wie hoch die Wahrscheinlichkeit ist, dass es sich um Spam handelt. Die Skala reicht von < 5 = kein Spam, ≥ 5 = möglicherweise Spam, bis hin zu ≥ 15 = Spam. Durch jeden der oben genannten Tests wird der Spam-Level der E-Mail entsprechend erhöht. Desto mehr Tests positiv ausfallen, umso höher steigt der Spam-Level.

In jedem Fall wird die Nachricht dem Empfänger zugestellt. Der Spam-Level wird vom *Spamassassin* innerhalb der E-Mail-Header als **X-Spam-Score** benannt und kann somit von der nächsten Instanz, dem MDA, ausgewertet werden. Beim Standardverhalten werden E-Mails ab einem Spam-Level von ≥ 5 beim Benutzerprogramm in einen Spam- oder auch Junk-Order verschoben.

3.2.3 Filtertechnologien

Um die eingesetzten Filtermethoden besser zu beschreiben, werden diese im folgenden Abschnitt nochmals im Detail dargestellt. Hierbei werden die Funktionen und Detailmessungen beschrieben.

Anti-Malware-Filter

Der Open Source Anti-Malware-Filter *ClamAV* bietet durch seine kostenlose Nutzung und der einfachen Möglichkeit der Integration in unterschiedliche MTAs eine solide Anti-Malware Basis. Durch seine von Benutzern und Entwicklern ständig aktualisierte Malware-Datenbank und dem heuristischen Ansatz für unterschiedliche

Dateitypen, arbeitet er mit eher klassischen Technologien. Durch die einfache Integration und Robustheit ist dieser Malware-Scanner im Zusammenhang mit MTAs weitverbreitet.

Aus dem klassischen Ansatz des *ClamAV* bildet sich jedoch ein Nachteil in Bezug auf die sich ständig ändernde Malware-Bedrohung. In einigen Fällen führt dies dazu, dass eine Erkennung von Malware erst einige Tage verspätet oder sogar ganz ausbleibt [44]. Eine schnelle Anpassung und Änderung der Schadsoftware trägt zusätzlich zu einer schlechten Erkennung bei [50, S. 20].

Um auf eine aktuelle Bedrohung (siehe Kapitel 2.4) Bezug zu nehmen, wurden innerhalb des Honeypot-Systems alle *Microsoft Office* Dokumente mit enthaltenem VAB-Makro-Code extrahiert und diese Anhänge auf dem System abgespeichert. Eine eingehende Prüfung der Anhänge mit dem frei verfügbaren *oletool*⁵ konnte ein heuristisches Verhaltensmuster und Inhalte von Malware für jede abgelegte Datei bestätigen. Hieraus resultiert, dass jede dieser Dokumente potentielle Malware darstellt.

Im Test gegen den *ClamAV* wurden 191 dieser unterschiedlichen Malware-Dokumente in täglichen Durchläufen geprüft. Der Malware-Scanner wurde in der Standardeinstellung belassen und aktualisiert somit jeden Tag um 24 Uhr seine Definitionsdatenbank für Malware. Das Ergebnis einer viertägigen Beobachtung war, dass 51 (26,70%) Makro-Malware-Dokumente aus Anhängen am Tag des Auftretens nicht erkannt wurden. Sieben sind nach mehr als einem Tag nicht als Malware klassifiziert worden. Selbst die im *ClamAV* integrierte heuristische Erkennung von schädlichem VBA-Makro-Code in *Microsoft Office* Dokumenten schlug hierbei nicht an. Somit liegt die Wahrscheinlichkeit, dass eine Nachricht mit Makro-Malware als Anhang nicht am selben Tag des Auftretens erkannt wird bei 30,37% ($\frac{58}{191} = 0,3037$). 24 Stunden später verringert sich diese auf 3,7% ($\frac{7}{191} = 0,037$).

Abbildung 10 zeigt sehr anschaulich, in welchen Wellen die Spam-Kampagnen mit angehängter Malware auftreten können. Innerhalb dieser Spam-Wellen sind bis zu 44 eingehende Nachrichten pro Minute zu beobachten. Bei einer Wahrscheinlichkeit von 30,37% auf 44 Nachrichten angewendet, ergibt auf eine ganze Zahl gerundet 13 (13,36) vom *ClamAV* nicht erkannte Anhänge der Nachrichten pro Minute. Hierbei wird vom Fall ausgegangen, dass an jeder eingehenden E-Mail ein anderer Malware-

⁵<https://bitbucket.org/decalage/oletools/wiki/olevba>

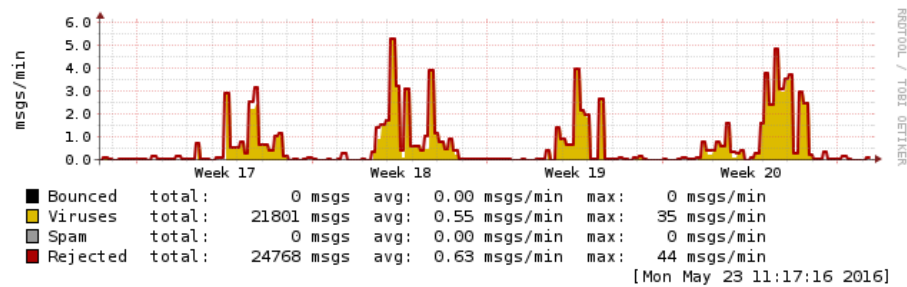


Abbildung 10: Spam-Kampagnen mit Malware als Anhang - Zeitraum 23.04-23.05.2016. Quelle: Autor

Anhang angehängt ist. In vielen vom Autor beobachteten Fällen werden jedoch bei Kampagnen immer derselbe Typ von Malware versendet, was die Wahrscheinlichkeit einer Infektion bei einer Nichterkennung dieser einzelnen Datei steigen lässt.

Aus diesen Zahlen ist zu erkennen, dass ein Anti-Malware-Scanner dazu neigt, Malware behaftete Dokumente nicht als Malware zu erkennen. Durch die hohe Anzahl der Malware-Nachrichten, die am E-Mailsystem in Wellen eingehen, erhöht sich so die Wahrscheinlichkeit einer Zustellung und damit auch die einer Infektion.

In der Praxis tendiert durch z.B. einen Absender-Filter die Anzahl der tatsächlich zugestellten Nachrichten nach unten, die Erkennungsrate des Virenschanners selbst beliebt jedoch unangetastet. Im Kapitel 4.2.2 wird hier noch genauer Bezug genommen.

Anti-Spam-Filter

Eine Basis für die ersten Anti-Spam-Filter ist in der RFC 2505 aus dem Jahre 1990 zu finden. Damals wurden erste Maßnahmen zur Verhinderung von Spam beschrieben, die sich grundsätzlich an den Metainformationen orientierten [28, S. 6]. Aus diesem Dokument wurden eine weitere Vorgehensweise entwickelt, um ein intelligenteres und mehr auf den Text fokussiertes Analysieren zu etabliert. Einer der Vorreiter auf diesem Gebiet war Paul Graham. Er entwickelte auf Basis des mathematischen Gesetzes des bayesschen Filters ein stochastisches Mittel gegen Spam [19].

Um Spam von Ham grundlegend zu unterscheiden, wird die Wahrscheinlichkeit mit 1 über 2 bezeichnet. Das bedeutet, dass die Wahrscheinlichkeit (P), dass eine E-Mail Spam (S) oder Ham (H) ist, gleich verteilt ist.

$$P(H) = P(S) = \frac{1}{2}$$

Diese Definition ist sehr statisch im Hinblick auf die variierende Verteilung zwischen dem Spam- und Ham-Anteil innerhalb des E-Mailverkehrs. Jedoch ist diese ein sehr guter Annäherung an einen praxisnahen Wert.

Eine zweite Annahme geht davon aus, dass die betrachteten Worte stochastisch unabhängig auftreten. Dies ist auf den ersten Blick eine eher paradoxe Annahmen, da gewisse Worte häufig in Zusammenhängen auftreten [39]. Aus dieser Verarbeitung des Textes wird eine Wortliste/Vektor erzeugt [4, S. 2].

Der *Spamassassin* arbeitet unter anderem mit dieser Technik, um eine E-Mail entsprechend als Ham oder Spam zu klassifizieren. Anhand der Wortliste und den E-Mail-Headern kann durch Abgleich von über 100 Testabläufen innerhalb des *Spamassassin* ein Spam-Level ermittelt werden [42].

Ein gezieltes Aushebeln der Tests kann unter anderem, durch einer Überhäufung des E-Mail-Textes mit legitimen Textbausteinen erreicht werden. Eine zielführende Bewertung der Tests kann somit nicht mehr stattfinden und der Filter entscheidet zugunsten der False-Positive-Rate, dass es sich um ein Ham E-Mail handelt [40, S. 6]. Im einfachsten Fall werden Buchstaben durch Sonderzeichen ersetzt oder gar Bilder als Text eingefügt [3, S. 7].

Zusätzlich zur dynamischen Analyse mit den Testabläufen kommen statische Listen mit Spam-URIs zum Einsatz. Es handelt sich dabei um im Internet zugängliche Listen mit URIs und Domainnamen, die von Spam-Versender häufig verwendet werden. Ist hier eine Verlinkung zu einer Domain oder URI aus dieser Liste innerhalb des E-Mailtextes zu finden, besteht eine sehr hohe Wahrscheinlichkeit, dass es sich um Spam handelt.

Spam-Kampagnen, die hinsichtlich der Wirkung gegen einen Anti-Spam-Filter optimiert wurde, besteht keine Möglichkeit mit den eingesetzten Techniken mehr, diese auf Basis der Texte zu erkennen.

3.3 Messungen

Für die Messung wurde der Messpunkt des eingehenden MTAs bestimmt. Hier konnte durch die Analyse der Log-Dateien ein genaues Bild der Verarbeitung der eingehenden Nachrichten erstellt werden. Alle Filterinstanzen dokumentierten jedes Filterevent in eine entsprechende Log-Datei. Die Einträge in diesen Log-Dateien beschreiben bei einem Malware- oder Spam-Befund den Grund für die Ausfilterung der Nachricht.

Um eine Langzeitanalyse zu ermöglichen, werden die Log-Dateien täglich archiviert und können so für spätere Auswertungen herangezogen werden. Die Auswertungen selbst wurden mit den Tools *pflogsumm*⁶ und *logwatch*⁷ durchgeführt. Für die graphische Aufarbeitung kam das Programm *Mailgraph*⁸ zum Einsatz.

Die Messungen selbst erstreckten sich über den Monat April im Jahr 2016. Innerhalb dieses Monats war kein Feiertag sowie keine sonstigen außergewöhnlichen Ereignisse, die den E-Mailverkehr beeinflusst hätten, vorhanden. Startzeitpunkt war der 01.04.2016 um 06:24 Uhr und der Endzeitpunkt der 30.04.2016 00:00 Uhr. Somit wurde ein Monat mit 25 Arbeitstagen und 5 Wochenendtagen abgebildet.

3.3.1 Ergebnisse

Im Folgenden werden alle ermittelten Messdaten in aufbereiteter Form dargestellt. Aufbereitet heißt hierbei, dass nicht ganzzahlige Daten auf zwei Nachkommastellen gerundet und personenbezogene Daten entfernt wurden. Alle Daten basieren auf den entsprechenden Log-Dateien und sind nicht aus einer Echtzeitanalyse entstanden. Bei dem System handelt es sich, wie erwähnt, um eine produktives und für die alltägliche Arbeitskommunikation verwendete System. Alle Werte stellen den eingehenden Nachrichtenfluss dar.

⁶<https://jimsun.linuxnet.com/>

⁷<https://sourceforge.net/projects/logwatch/files/>

⁸<http://mailgraph.schweikert.ch/>

Filterungsschritt 1: *Postfix* und *Amavisd-new*

Um einen Überblick über das System zu bekommen, sind in der Tabelle 2 die Messdaten der Filterinstanzen *Postfix* sowie *Amavisd-new* in einer Übersicht über den gesamten Messzeitraum hinweg.

An <i>Spamassassin</i>	74.059	68,91%
Zurückgewiesen	33.414	31,09%
<hr/>		
Gesamt	107.470	100%

Tabelle 2: Empfangene Nachrichten im Messzeitraum.

Aus der Tabelle 2 ist zu entnehmen, dass 31,09% der empfangenen Nachrichten schon in den ersten beiden Filterinstanzen direkt zurückgewiesen werden. Nur 68,91% der E-Mails wurden an den *Spamassassin* für eine textbasierte Spam-Analyse weitergeleitet.

Bei allen zurückgewiesenen E-Mails handelt es sich um E-Mails, die nicht an den Empfänger zugestellt wurden und zurückgewiesen worden. Der sendende Server erhält vom MTA eine entsprechende Nachricht, dass die Annahme verweigert wurde.

In der Tabelle 3 ist eine genauere Aufstellung der Gründe für die Zurückweisung zu sehen. Aufgeteilt sind diese in die folgenden Rubriken:

- **Verbindungsfehler:** Bei diesem Typ handelt es sich um Nachrichten, die schon beim Verbindungsversuch zurückgewiesen wurden. Zumeist sind hier falsche oder fehlende Servernamen des Senders oder Verbindungsabbrüche, als auch Port-Scans der Grund.
- **Adressüberprüfung:** Hierbei wurde die E-Mailadresse des Empfängers im Zielsystem nicht gefunden. Somit ist eine Zustellung an ein nicht vorhandenes Postfach nicht zielführend und daher wird die E-Mail zurückgewiesen.
- **RBL:** Falls eine E-Mail auf einer der im Internet verfügbaren RBLs steht, kann davon ausgegangen werden, dass es sich hierbei um Spam handelt. Die E-Mail wird daher zurückgewiesen.

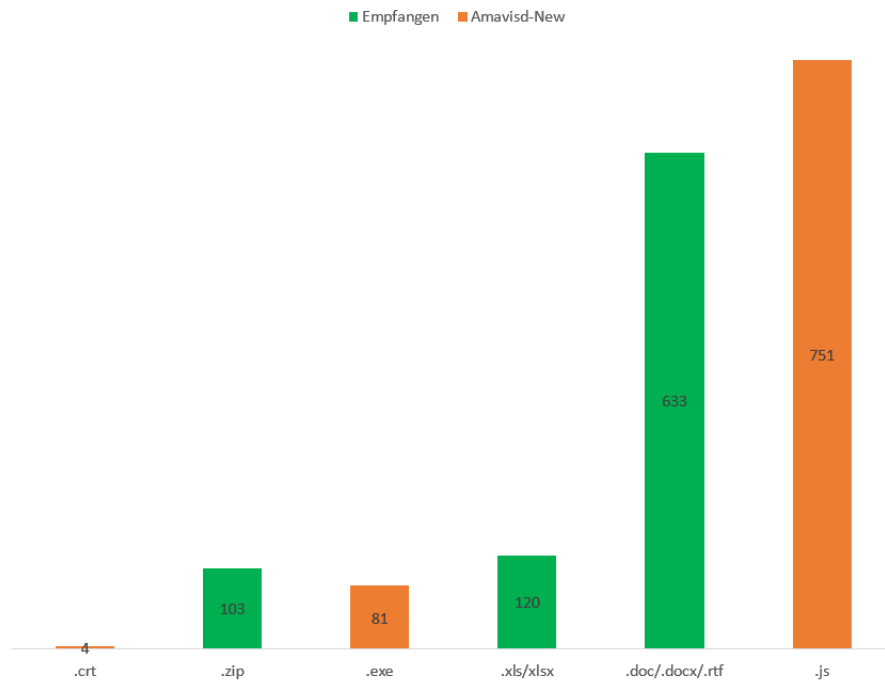


Abbildung 11: Anzahl der E-Mails am Messsystem mit entsprechenden Dateieindungen. Quelle: Autor

- **Blockierte Dateieindungen:** Der *Amavisd-new* Filter dient zur Filterung der Anhänge und verweigert die Annahme bei verbotenen Dateieindungen. In Abbildung 11 ist die Aufteilung der unterschiedlichen Dateieindungen zu sehen. Die Dateieindungen `.js` für Javascript-Dateien sowie `.exe` für ausführbare Programme wurden hierbei geschlossen zurückgewiesen. Beide sind mit hoher Wahrscheinlichkeit Malware [14]. Die `.crt` Endung steht für eine Zertifikatsdatei, die im Normalfall nicht via E-Mail versendet wird und somit auch zurückgewiesen wurde.
- **Malware:** Bei dieser Rubrik handelt es sich um Zurückweisungen, die vom *Amavisd-new* durchgeführt werden, wenn ein Virus vom *ClamAV* erkannt wurde.

Verbindungs- fehler	Adress- überprüfung	RBL	Blockierte Dateien- dungen	Malware	Summe
3.162	3.154	26.259	836	3	33.414
9,46%	9,44%	78,59%	2,50%	0,01%	100%

Tabelle 3: Aufteilung der zurückgewiesenen Nachrichten

Filterungsschritt 2: *Spamassassin*

Die nach dem Filterungsschritt 1 verbleibenden 74.059 E-Mails werden nun noch vom *Spamassassin* überprüft. Dieser hat nun die Aufgabe, die Nachrichten innerhalb ihrer Metainformation und textuellen Inhalten auf bekannte Spam-Eigenschaften zu untersuchen. Nach dieser Klassifizierung wurden 4.713 (9%) durch diesen Filter als Spam markiert, da sie entsprechende Merkmale einer Spam-E-Mail hatten. Die restlichen 69.346 (81%) von 74.059 wurden nun an die entsprechenden Empfänger ohne eine Spam-Markierung zugestellt (Abbildung 12).

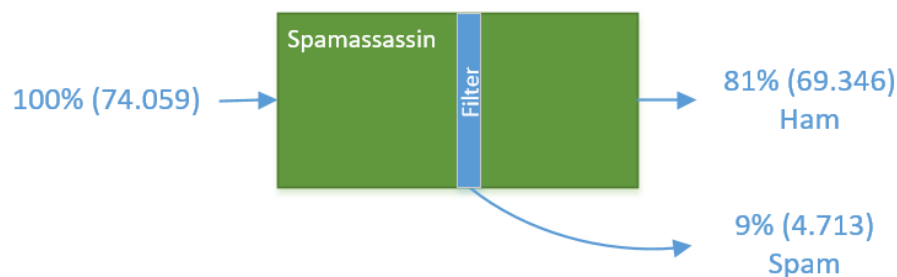


Abbildung 12: Filterung des Spamassassin. Quelle: Autor

Nimmt man alle Filterungsschritte zusammen, sind das 69.346 E-Mails im Monat April 2016 auf 753 Empfänger. Das bedeutet auf die Empfänger aufgeteilt rund 92 E-Mails im Monat.

3.3.2 Zusammenfassung der Messungen

In der Summe aller Filterinstanzen sind es somit 38.127 Nachrichten, die entweder als Spam deklariert oder direkt schon zurückgewiesen wurden. Prozentual bestätigt ist der Anteil der damit ungewollten E-Mails bei 107.473 Nachrichten 35,48%. Laut der Statistik von *Symantec* liegt der weltweite Mittelwert bei einer Firma mit 501-1000 Mitarbeitern bei rund 53,5% (Abbildung 13).

Company Size	Apr '16 (%)	Mar '16 (%)
1-250	53.2	53
251-500	53.3	53.5
501-1000	53.5	53.3
1001-1500	52.4	52.5
1501-2500	52.2	52.3
2501+	52.3	52.6

Spam by Organization Size

Source: Symantec

Abbildung 13: Spam-Report von Symantec. Quelle: Symantec [41]

Somit liegt das Messsystem mit 18,02% hinter den von Symantec herausgegeben Zahlen, was zu vernachlässigen ist, da diese nicht explizit ein Unternehmen widerspiegeln, sondern globale Zahlen für die Ermittlung heranziehen und daraus eine große Varianz entsteht. Der Umfang, wie oft eine E-Mail innerhalb der zugestellten 64,52% fälschlicherweise nicht als Spam markiert worden ist (False-Positive), bleibt hier offen.

3.4 Darstellung des Defizits

Die oben genannten Filtermethoden und Vorgänge zur Verhinderung von Spam und Malware zeigt im Vergleich zu den globalen Statistiken von Symantec, dass rund 50% der empfangenen E-Mails Spam enthalten. Dennoch kommt es immer wieder dazu, dass E-Mails von den Filterinstanzen nicht erkannt wurden.

Die aktuellen Filtermethoden bieten viele Möglichkeiten, E-Mails auf entsprechende Merkmale hin zu untersuchen, jedoch obliegt es immer dem Betreiber, wie restriktiv er diese einsetzt. Eine zu starke Einschränkung der Empfangsfilter führt unweigerlich zu einer Erhöhung der False-Positiv-Rate in Bezug auf Ham-E-Mails. Das heißt, dass

legitime E-Mails als Spam vom System erkannt werden, was aus Benutzersicht sehr negativ ist. Die gezogene Statistik aus dem Messsystem zeigt, dass der Mittelwert des Spam-Level, das der *Spamassassin* einer Ham E-Mail zuweist, bei einem Wert von 2,5 liegt. Wenn der Level ≥ 5 ist, wird die E-Mail als Spam markiert und landet somit im Spam-Ordner des Benutzers. 3.107 E-Mails haben im Messzeitraum ein Level von 4 erreicht. Würde man hier also ein restriktiveres Regelset anwenden und das Level für eine Spam-Nachricht auf ≥ 4 herabsetzen, würden diese E-Mails als Spam markiert an den Benutzer zugestellt werden.

Bei der Betrachtung des *Amavisd-new* und des *Postfix* sind die Folgen einer False-Positiv-Erkennung mit einer Zurückweisung verbunden und somit wird die E-Mail erst gar nicht angenommen. Der Sender erhält hierbei eine Fehlermeldung, dass die E-Mail nicht Zugestellt wurde. Das im Falle des Messsystems angewendete Blockieren bestimmter Anhänge (siehe Kapitel 3.1) hat auch einen Einfluss auf die Kommunikation z.B. mit Kunden. Durch ein grundsätzliche Blockieren von *Microsoft Office* Dokumenten mit einem aktiven VBA-Makro-Code, kann infolgedessen die E-Mailkommunikation mit Geschäftspartnern empfindlich gestört werden. Bei einer Erhebung in einem Kundensystem wurde ermittelt, dass von 464.150 Nachrichten mit *Microsoft Office* Dokumenten 9.863 ein legitimes Makro enthalten. Im Falle eines restriktiveren Ansatzes und dem Blockieren solcher Dokumente würden diese an den Sender zurückgewiesen werden, was den damit verbundenen Geschäftsprozess unterbinden würde.

Diese Methode, *Microsoft Office* Dokumenten mit einem aktiven VBA-Makro-Code zu blockieren, wird jedoch häufig als Gegenmaßnahme für Makro-Malware angegeben [9, S. 11] und [48, S. 5]. Anhand der ermittelten Zahlen scheint diese jedoch nicht zielführend für die Abbildung eines Geschäftsprozesses, der auf Dokumente mit VBA-Makros basiert.

Ein Freischalten der entsprechenden Kommunikationspartner von diesem Filter würde aufgrund der Möglichkeit, in E-Mails den Absender zu fälschen, eher kontraproduktiv sein. Schlimmer ist hier noch, dass Benutzer hierdurch eine nicht vorhandene Sicherheit vorgetäuscht wird und die Wahrscheinlichkeit für das Öffnen des Anhangs zunimmt. Falls jedoch diese Blockierregel nicht aktiviert wird, lässt man auch unweigerlich die in Kapitel 2.3 beschriebene Malware-Typen zu. So ergibt sich ein Dilemma zwischen dem Zulassen von potentiell gefährlichen Anhängen und der negativen Beeinflussung des täglichen Geschäftsablaufes.

Zusammengefasst wird ersichtlich, dass die etablierten Filtermethoden hier ein sehr starres, aber effizientes Konzept bieten [1, S. 4ff]. Sie bilden einen soliden Grundschutz können jedoch nicht flexibel und dynamisch auf neue oder sich ändernde Bedrohungen reagieren.

4 Neue Ansätze

4.1 Rückschlüsse aus den Messergebnissen

Die erbrachten Ergebnisse lassen darauf schließen, dass die klassischen Filtermethoden eine Basis bieten und auch bereits schon ein Großteil der Bedrohungen eliminieren können. Jedoch wandelt sich die Bedrohung, wie in Kapitel 2.4.1 beschrieben, ständig. Zudem etabliert sich während der Filterung zwangsläufig auch eine Art Qualitätsprüfung. Dies bedeutet bei einer Nichterkennung einer mit Spam oder Malware behafteten E-Mail, dass diese eine erhöhte Qualität aufweisen muss. Das muss nicht zwangsläufig zu einer erhöhten Wahrscheinlichkeit führen, dass eine Infektion erfolgt, jedoch wird es hierbei für den Benutzer auch schwerer diese Nachrichten als Spam zu klassifizieren. Bei einer zugestellten E-Mail mit Malware im Anhang besteht die letzte Filterinstanz darin, dass der Benutzer diese als solche erkennt und den Anhang nicht ausführt. Automatisierte Methoden haben in so einem Fall keine Anwendung gefunden.

4.2 Neue Wege der Erkennung

Aufbauend auf dem Fundament der existierenden Filtermethoden wurden sich in dieser Arbeit mit neuen Ansätzen auseinandergesetzt. Im Fokus steht hier die Implementierung einfacher Methoden, die zum existierenden System hinzugefügt werden könnten. Entstanden sind diese Methoden durch die Beobachtung und Analyse der ermittelten Messwerte. Zusätzlich wurden Problematiken und Erkenntnisse aus dem



Abbildung 14: Für den Benutzer offensichtliche Spam-Nachricht. Quelle: Autor

Betriebsalltag miteinbezogen. Ein Beispiel hierfür ist die Ausarbeitung des Konzeptes zur Verhinderung von Malware in *Microsoft Office* Dokumenten. Dieses wurde aus einer Problematik (siehe 3.4) aus dem Betriebsablauf heraus entworfen und stellt somit eine sehr nahe Bindung zu einem realen Problem dar.

4.2.1 Verhinderung von Spam durch bekannte Absender

Aus der Analyse des Spam-Verhaltens anhand des Messsystem konnte ermittelt werden, dass 35,48% der Nachrichten als Spam von konventionellen Filtereinrichtungen erkannt wurden. Jedoch besteht immer wieder das Problem, dass Spam-Nachrichten als False-Negativ an den Benutzer zugestellt werden.

In Abbildung 14 konnte die Nachricht nicht von den automatisierten Filterinstanzen auf dem Messsystem erfasst werden. Der Test, ob vom Absender die Absenderadresse vergessen worden ist, bewertet der *Spamassassin* die E-Mail hierbei positiv mit 2,095 und 1,552 (`FREEMAIL_FORGED_REPLYTO` und `REPLYTO_WITHOUT_TO_CC`) als Spam. Zusätzlich kommen noch eine fehlerhafte Auflösung der Absender-Domain mit 0.792 (`RDNS_NONE`). Daraus ergibt sich ein Spam-Level von 4,438. Der *Bayes_00-Test*¹ verringert hierbei jedoch den gesamten Spam-Level um -1,9 Punkte, da er aus dem Text keine Spam-Eigenschaften ermittelt worden. Diese Vorgehen wird zu Gunsten der False-Positiv-Rate durchgeführt. Es ergibt sich aus 4,438-1,9 ein Spam-Level von 2,538 was <5 ist und somit die Nachricht nicht als Spam klassifiziert wird, obwohl sie dies nachweislich sein sollte.

¹https://wiki.apache.org/spamassassin/Rules/BAYES_00

Listing 4.1: Positive Tests des *Spamassassin* mit Bewertung

```
X-Spam-Tests: BAYES_00=-1.9, FREEMAIL_FORGED_REPLYTO=2.095,  
RDNS_NONE=0.793, REPLYTO_WITHOUT_TO_CC=1.552
```

Für den Benutzer ist hier im Beispiel (Abbildung 14) klar erkennbar, dass es sich um Spam handeln muss. In vielen Fällen wird dies aus dem Kontext ersichtlich. Dieser entsteht aus dem Absender sowie Inhalt der E-Mail. Der Benutzer schließt aus diesen beiden Informationen, ob es sich um eine Ham- oder Spam-E-Mail handelt.

Diese einfache Auswertung ob, die der Benutzer meist unterbewusst für jede E-Mail durchführt, lässt sich in ein MTA-Filter-System überführen. Hierzu sind keine hoch komplexen Algorithmen nötig, sondern eine dynamisch wachsende Datenbank mit allen Empfängeradressen ausgehender E-Mails. Im Detail müsste diese Datenbank vom ausgehenden E-Mailsystem befüllt werden. Sobald eine E-Mail an einen externen Empfänger geht, wird diese E-Mailadresse als bekannt in der Datenbank hinterlegt. Hieraus erzeugt sich eine dynamisch wachsende Whitelist, die alle Kommunikationspartner auflistet. Diese Datenbank wird dem *Spamassassin* zur Verfügung gestellt, der dann somit ein weiteres Bewertungskriterium bekommt.

So kann durch die Analyse des Textes und der E-Mail-Header der Spam-Filter schon eine gute Einschätzung liefern. Sollte jedoch die Entscheidung zweifelhaft ausfallen, wie oben beschrieben (Spam-Level zwischen 0 und 5), könnte diese Datenbank eine weitere Entscheidungsgrundlage bieten. In einem ersten Test mit 35 False-Negativ Spam-Nachrichten wurde ermittelt, dass alle durch diese Methode erkannt werden könnten. Der Spam-Level lag bei jeder getesteten E-Mail zwischen 0 und 4,9.

Innerhalb des *Spamassassin* werden ähnliche Funktionen schon verwendet. Die Auto-Whitelist-Funktion merkt sich immer den letzten Spam-Level eines Absenders. Ein Beispiel wäre hier, wenn Bob eine Spam-E-Mail mit Level 10 sendet, wird dies so in der Datenbank vermerkt. Schickt Bob eine weitere E-Mail, die aber nur ein Level von 2 hat, wird dieser vom alten Wert abgezogen. Also $10-2=8$, wobei 8 das neue Level ist [46]. Bei der Beobachtung am Messsystem zeigt sich dennoch, dass im Messzeitraum 21.439 Absender eingehend vermerkt wurden. Hiervon haben jedoch nur 4.744 mehr als eine E-Mail gesendet haben. 16.695 Absender (77,87%) haben ausschließlich eine E-Mail übermittelt. Daraus ergibt sich, dass bei 4.744 (33,23%) lediglich ein Mehrwert durch das Whitelisting entsteht. Alle anderen Einträge würden in der Datenbank verweisen, da keine weitere E-Mail mit demselben Absender empfan-

gen wird. Häufig handelt es sich um System- oder Spam-E-Mails, die automatisiert generierte Absender aufweise.

Der beschriebene neue Ansatz invertiert dieses Vorgehen und könnte hierbei eine bessere Erkennung einer schon vorhandenen Kommunikation liefern. Die Wahrscheinlichkeit, dass eine Kommunikation aufgrund einer Spam-E-Mail mit dem Spamversender stattfindet, ist vernachlässigbar. Zudem würde diese Erweiterung der Erkennung erst eingesetzt werden, wenn der Spam-Level in einem Schwellenbereich zwischen 0 und 4,9 des Spam-Levels liegen würde. Oberhalb und unterhalb dieses Bereiches würde diese Datenbank nicht für eine weitere Bewertung der E-Mail herangezogen werden.

Im Beispiel würde Bob an Alice eine Nachricht senden. Diese Nachricht erhält am Spam-Filter eingehend den Spam-Level -1 und wird an Alice als legitime E-Mail zugestellt. Alice antwortet auf diese Nachricht und somit wird über den Ausgangs-server die Empfängeradresse in die Datenbank eingetragen. Sendet Bob nun eine weitere Nachricht mit dem Spam-Level von 4,8 an Alice, würde der Eintrag in der Datenbank die Nachricht als Ham erkennen, da schon eine ausgehende Kommunikation mit Bob und seiner E-Mailadresse stattfand. Falls jedoch eine Nachricht eines Spam-Versenders mit einem niedrigen Spam-Level von 3,2 vom *Spamassassin* erkannt wird, kann diese Nachricht, aufgrund des fehlenden Eintrages der Absenderadresse des Spam-Versender innerhalb der Datenbank, besser bewertet werden. Eine Erhöhung des Spam-Levels um 3 Punkte durch die zusätzliche Bewertung, würde in solch einem Fall die Nachricht als Spam markieren.

Generell ist dieser Ansatz an einem Greylisting (RFC 6647) angelehnt [24, S. 3ff]. Hierbei werden hier Nachrichten temporär beim ersten Senden vom empfangenen MTA zurückgewiesen. Der sendende MTA versucht nach einer Zeitspanne dann nochmals die Nachricht an den empfangenen MTA zu senden. Nach dem zweiten Versuch, wird die Nachricht vom empfangenen MTA dann endgültig angenommen und der Absender in einer Datenbank hinterlegt. Sendet dieser Absender nun nochmals eine Nachricht, wird diese aufgrund des Datenbankeintrags direkt angenommen. Dieses Vorgehen soll Spam-Versender abhalten, da diese eventuell nicht versuchen die Nachricht ein zweites mal zuzustellen. Problem ist, dass der gesamte Nachrichtenfluss bei unbekannten Absendern stark verzögert wird. Zusätzlich ist die Effizienz sehr gering einzuschätzen [11, S. 5] und Spam- und Malware-Versender häufig dieses Vorgehen vorhersehen und durch nochmaliges senden aushebeln [31, S. 6].

Bei dem ausgearbeiteten Ansatz wird im Detail nicht eine unbekannten Datenbasis verwendet, sondern die bekannten und legitimen Absender der ausgehenden Nachrichten. So entsteht eine schlechtere Bewertung einer Nachricht erst, wenn sie dem System nicht bekannt ist. Bei diesem Ansatz wird so auch die Möglichkeit genommen, das System durch senden von entsprechend präparierten E-Mails zu Manipulieren. Eine Erweiterung der Datenbank ist nur über das ausgehende System möglich.

Aus technischer Sicht, müsste der *Spamassassin* lediglich eine Schnittstelle bieten, worüber vertrauenswürdige Systeme Absender in der Datenbank bekanntgeben könnten.

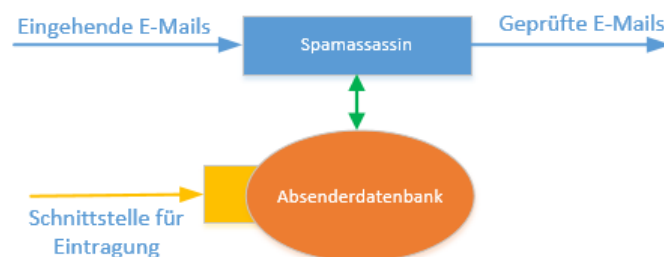


Abbildung 15: Konzept für die Einbindung der Absenderdatenbank. Quelle: Autor

Es wurde ermittelt, dass bei den ausgehenden E-Mails im Monat rund 4.280 Absender verwendet werden. Das heißt, dass 4.280 verschiedene Adressen innerhalb der Datenbank vermerkt wurden. Diese Zahl hält sich sehr konstant. Zwischen dem Monat April und Mai 2016 wurden in der E-Mailinfrastruktur lediglich nur 230 neue Adressen registriert. Im Jahr sind das hochgerechnet rund 2.760 bei der Annahme einer konstanten Wachstumsrate von 230 im Monat. Im besten Fall haben alle Einträge ein Verfallsdatum, dass den Eintrag invalidiert, falls er seit mehr als einem Jahr nicht mehr beobachtet werden konnte. Bei diesen Einträgen in der Datenbank, kann von einer einmaligen Kommunikation gesprochen werden.

4.2.2 Verhinderung von Makro-Malware

Die in Kapitel 2.3 beschriebene Wandlung vom Ausnutzen von Schwachstellen in Programmen oder Protokollen zieht sich hin zur Schwachstelle Mensch. Um diese Schwachstelle besser automatisiert zu schützen wurde ein Ansatz mithilfe von

Analysen der Dateiinhalte entwickelt.

Bei diesem Ansatz wird zusätzlich vom Virenschanner jeder ankommende Anhang einer E-Mail auf eventuelle Malware-Eigenschaften geprüft. Im Detail wird der Fokus auf die Dokumentdateitypen der *Microsoft Office* Programmfamilie gelegt, die durch `.doc/.docx`, `.xls/.xlsx` und `.rtf` beschrieben werden. Sie stellen mit 88% einen Großteil der legitimen Anhänge dar. Bei den restlichen 12% handelt es sich um komprimierter Archive im `.zip`-Format (siehe Abbildung 11).

Durch die Möglichkeit, VBA-Code in diesen Dokumenten zu implementieren stellen sie ein potentielle Gefahr dar. Die erkannten Defizite (Kapitel 3.4) der aktuelle eingesetzten Filtermethoden können diese Art von Bedrohung nachweislich (Kapitel 3.2.3) nicht eliminiert.

Im Fokus dieses Ansatzes, soll die dynamische und nicht auf Virus-Pattern basierende Analyse des VBA-Codes innerhalb der Dokumente stehen. Nur so kann eine zuverlässige Erkennung bei schnell wechselnden Makro-Inhalten geschehen. Die Erkennung selbst soll keine direkte Verhaltensanalyse oder Interpretierung des Codes sein. Vielmehr wird hierbei die Art des Quellcodes und die verwendeten Funktionsaufrufe betrachtet.

Alle Werte beruhen auf Analysen von 25 unterschiedlichen, für den Unternehmensablauf wichtigen *Microsoft Office* Dokumenten mit Makros. Ihnen gegenüber standen 526 mit Malware behaftete. Aus diesen wurde ermittelt, welche Arten von Funktionsaufrufen mit einer hohen Wahrscheinlichkeit nicht in legitimen Makros enthalten sind. Diese Auswertung wurde durch die Analysen von Alexander Gazet [18] und vom *Microsoft* Malware Protection Center [29] weiter optimiert. Die Gegenanalyse von gutartigen Makros ergab, dass diese in einigen Punkten differenzierbar sind. Abbildung 16 zeigt ein Beispiel eines in VBA geschriebenen Malware-Codes. Man kann hier erkennen, dass mit verschiedenen Methoden versucht wird, die eigentliche Funktion zu verschleiern. Ein Beispiel wäre hierbei die Verwendung von Hexadezimalzahlen und zufällig generierte Funktionsnamen im Code.

Aus den Analysen der Dateien konnte ein Ausgangsheuristik für Malware-Dokumente definiert werden. Für die Analyse der Dateien wurde das Toolset *python-oletools*² von Philippe Lagadec verwendet. Diese bietet eine gute Möglichkeit, Dokumente auf diese Art von Funktionsaufrufe zu untersuchen.

²<http://www.decorage.info/python/oletools>

```

1 Sub Jj4TaqwL2swp()
2 Dim PB8KB580Kil: Set PB8KB580Kil = CreateObject(RweYesW7TWLkL4L68BV(Chr(22)&Chr(67)&Chr(91)&Chr(34)&Chr(48)&Chr(39)&Chr(62)&Chr(127)&Chr(7)&Chr(4)&Chr(47)
3 zcOV38GkwE1 = "0xfc,0xe8,0x22,0x0,0x0,0x60,0x89,0xe5,0x31,0xc0,0x64,0x8b,0x50,0x30,0x8b,0x52,0xc,0x8b,0x52,0x14,0x8b,0x72,0x28,0xf,0xb7,0x4a,0x26,0x31,
4 "0xc1,0xc7,0xd,0x1,0xc7,0x38,0xe0,0x75,0xf6,0x3,0x7d,0xf8,0x3b,0x7d,0x24,0x75,0xe4,0x58,0x8b,0x58,0x24,0x1,0xd3,0x66,0x8b,0xc,0x4b,0x8b,0x58,0x1c,0x1,0xd3
5 "0x29,0xc4,0x54,0x50,0x68,0x29,0x80,0x6b,0x0,0xff,0xd5,0x50,0x50,0x50,0x40,0x50,0x68,0xea,0xf,0xdf,0xe0,0xff,0xd5,0x97,0x6a,0x5,0x68,0x89,0
6 "0xd5,0x8b,0x36,0x6a,0x40,0x68,0x0,0x10,0x0,0x0,0x56,0x6a,0x0,0x68,0x58,0xa4,0x53,0xe5,0xff,0xd5,0x93,0x53,0x6a,0x0,0x56,0x53,0x57,0x68,0x2,0xd9,0xc8,0x5f
7 faQ4lLYFulusZbQ = RweYesW7TWLkL4L68BV(Chr(33)&Chr(44)&Chr(25)&Chr(35)&Chr(30)&Chr(54)&Chr(50)&Chr(14)&Chr(36)&Chr(47)&Chr(99)&Chr(40)&Chr(12)&Chr(13)&Chr(
8 Chr(117)&Chr(23)&Chr(8)&Chr(53)&Chr(62)&Chr(16)&Chr(38)&Chr(72)&Chr(34)&Chr(44)&Chr(23)&Chr(100)&Chr(49)&Chr(6)&Chr(8)&Chr(55)&Chr(89)&Chr(36)&Chr(23)&Chr(63)
9 Chr(19)&Chr(61)&Chr(9)&Chr(19)&Chr(5)&Chr(7)&Chr(74)&Chr(78)&Chr(122)&Chr(13)&Chr(54)&Chr(22)&Chr(23)&Chr(37)&Chr(88)&Chr(86)&Chr(72)&Chr(102)&Chr(28)&Chr(10
10 Chr(90)&Chr(41)&Chr(23)&Chr(113)&Chr(125)&Chr(118)&Chr(31)&Chr(58)&Chr(54)&Chr(52)&Chr(99)&Chr(67)&Chr(21)&Chr(4)&Chr(32)&Chr(54)&Chr(7)&Chr(43)&Chr(44)&Chr(
11 PB8KB580Kil.Run faQ4lLYFulusZbQ, 0, False
12 End Sub
13 Sub AutoOpen(): Jj4TaqwL2swp: End Sub
14 Sub Auto_Open(): Jj4TaqwL2swp: End Sub
15 Sub Workbook_Open(): Jj4TaqwL2swp: End Sub
16
17 Private Function RweYesW7TWLkL4L68BV(ByVal HBueV8oGJCW6FYB0t As String, ByVal Uv10zeLIW4slueQ As String) As String
18 Dim dVYR44WepkmaeC As Integer: Dim qjERmEHn8y1404kY As Integer: Dim gjNt0TTWfNGq4vGLH As String
19 dVYR44WepkmaeC = Len(Uv10zeLIW4slueQ)
20 For qjERmEHn8y1404kY = 1 To Len(HBueV8oGJCW6FYB0t)
21 gjNt0TTWfNGq4vGLH = Asc(Mid$(Uv10zeLIW4slueQ$, (qjERmEHn8y1404kY Mod dVYR44WepkmaeC) - dVYR44WepkmaeC * ((qjERmEHn8y1404kY Mod dVYR44WepkmaeC) = 0), 1))
22 Mid$(HBueV8oGJCW6FYB0t, qjERmEHn8y1404kY, 1) = Chr$(Asc(Mid$(HBueV8oGJCW6FYB0t, qjERmEHn8y1404kY, 1)) Xor gjNt0TTWfNGq4vGLH)
23 Next
24 RweYesW7TWLkL4L68BV = HBueV8oGJCW6FYB0t
25 End Function
26

```

Abbildung 16: VBA-Code einer Malware. Quelle: Autor

Folgende Eigenschaften und Funktionsaufrufe können in einem Dokument mit Malware VBA-Code gefunden werden:

- Sobald das Dokument geöffnet wird, wird ein Funktionsaufruf durchgeführt.
- Herunterladen einer Datei von einer URL.
- Eine Datei wird gelöscht.
- Eine Datei wird angelegt.
- Versucht eine Datei oder Programm auf dem System auszuführen.
- Versucht eine Kommandozeile zu öffnen.
- Versucht ein Systemprogramm auszuführen.
- Eine Datei wird kopiert.
- Der Code beinhaltet Hexadezimal Text.
- Der Code beinhaltet Base64 Text.
- Verschleierungen mit der Funktion `Chr()`³.

Die gelisteten Funktionsaufrufe stellen die Ausgangsheuristik dar, sind jedoch in dieser Form nicht gewichtet. Zusätzlich müssen diese Funktionsaufrufe und Eigenschaften in der Kombination ihres Auftretens betrachtet werden. Die Klassifizierung ausschließlich auf einer dieser Funktionsaufrufe würde nicht zielführend sein, da diese im Einzelfall auch in legitimem Code enthalten sind.

³<https://msdn.microsoft.com/de-de/library/613dxh46%28v=vs.90%29.aspx>

Vorgang	legitimen Dokumente	Malware-Dokumente
Auto-Start des Makro	11	15
Datei löschen, kopieren, erstellen	12	28
Programmaufruf	2	27
Datei herunterladen	1	24
Kommandozeile öffnen	1	28
Systemprogrammaufruf	0	28
Hexadezimal Text	0	497
Base64 Text	0	494
Funktion Chr()	0	492

Tabelle 4: Anzahl der legitimen und Malware Dokumenten mit Auftreten der Funktionen

In Tabelle 4 ist eine Tendenz der 526 untersuchten Malware Dateien zu den unterschiedlichen Funktionen zu sehen. Rund 490 diese Dateien weisen die Verwendung von Hexadezimalzahlen, Base64, sowie mit der **Char()**-Funktion codierten Text auf. Innerhalb dieser 25 legitimen Dateien konnte keine dieser Funktionen gefunden werden. Jedoch gilt es hier zu beachten, dass diese Auswertung gegenüber einer geringen Anzahl legitimer Makro-Codeteile steht und somit einen speziellen Fall repräsentiert. Um diese Auswertung zu verallgemeinern wird ein vereinzelter Auftreten dieser Funktionsaufrufe noch nicht zur Klassifizierung des Dokuments als Malware verwendet. Funktionen, wie das Öffnen der im *Microsoft Windows* integrierten Kommandozeile sowie das Herunterladen von Dateien, kann in Einzelfällen auch im legitimen Code beobachtet werden. Betrachtet werden müssen diese verdächtigen Funktionsaufrufe in einer Häufung. Erst wenn im Code mehrfach z.B. Base64 oder Hexadezimalzahlen verwendet werden, steigt das Potential Malware zu sein, da hierdurch Code-Teile verschleiert werden können.

Grundlegend ergibt sich die Bewertung aus dem Gefahrenpotential der Funktionen. Das reine Erstellen z.B. einer Datei stellt keine direkte Gefahr dar und wird somit geringer bewertet. Eine mehrfache Verwendung der **Char()**-Funktion jedoch lässt die Vermutung steigen, dass hierdurch Funktionen und Code-Teile verschleiert worden sind. Funktionen wie das Aufrufen von Systemprogrammen hingegen werden bei einmaligem Auftreten schon als stark verdächtig angesehen.

Um diese Gefahreinstufung darzustellen, wurde eine Skala von 1 bis 10 definiert. Hierbei wird mit 1 eine geringe Gefahr sowie mit 10 eine hohe dargestellt. Hieraus ergibt sich die Bewertungstabelle 5 mit der entsprechenden Gefahrenbewertung. Tritt

somit eine Kombination dieser Funktionen auf, werden die Werte addiert. Enthält also ein *Microsoft Office* Dokument ein Makro, das eine Datei und die Kommandozeile öffnet sowie zusätzlich einen Download starten will, erhält es die Bewertung $2+8+5=15$.

Vorgang	Gewichtung (1-10)	Beschreibung
Auto-Start des Makro	2	Auch ein legitimer Code kann diese Funktion beinhalten.
Datei löschen, kopieren, erstellen	2	Auch ein legitimer Code kann diese Funktion beinhalten.
Programmaufruf	5	Kann in ein legitimen Code vorkommen, sollte es im besten Fall aber nicht.
Datei herunterladen	5	Ein Nachladen von Inhalten stellt ein Risiko dar.
Funktion <code>Chr()</code>	5	Dieser Aufruf kann ein- bis zweimal im legitimen Code Vorkommen.
Kommandozeile öffnen	8	Dieser Vorgang sollte selten in einem normalen Makro-Code vorhanden sein.
Systemprogrammaufruf	10	Dieser Aufruf kommt selten in einem legitimen Code vor.
Hexadezimal Text	10	Dieser Aufruf kommt selten in einem legitimen Code vor.
Base64 Text	10	Dieser Aufruf kommt selten in einem legitimen Code vor.

Tabelle 5: Ansatz zur Gewichtung der Funktionsaufrufe im Makro-Code

Nachdem die E-Mail einen entsprechende Bewertung erhalten hat, wird diese gegen einen global definierten Wert geprüft. Dieser Schwellenwert bestimmt, ab welcher Gefahrenbewertung des Markos, die E-Mail zurückgewiesen wird. Dieser Schwellenwert wurde aus der Analyse der 526 Malware und 25 legitimen Dokumenten erstellt. Hierbei wurde zunächst der Maximalwert der legitimen Dokumente ermittelt. Dieser lag bei 12 Punkten. Im Gegensatz zum Maximalwert wurde noch der Minimalwert der Malware behafteten Dokumente ermittelt. Dieser lag bei 14 Punkten.

Aus der Analyse der Maximal- und Minimalwerte ergibt sich ein Schwellenwert von 13 Punkten. Wird dieser von einem Dokument mit Makro-Code überschritten, wird die Nachricht an den Absender zurückgewiesen.

Nach der Analyse der Dateien und der Definition der Heuristik, als auch des Schwellenwertes, wurde die Implementierungsmöglichkeit am MTA betrachtet. Hierbei muss beachtet werden, dass diese in die aktuelle MTA-Infrastruktur und Filter-

methoden eingebunden werden muss. Für diese Aufgabe ist eine Verwendung der Milter-Schnittstelle im *Postfix* als zielführend anzusehen. Diese bietet die Möglichkeit, die E-Mail an einen Filter weiterzugeben, ohne die Nachricht vollständig anzunehmen zu müssen [53].

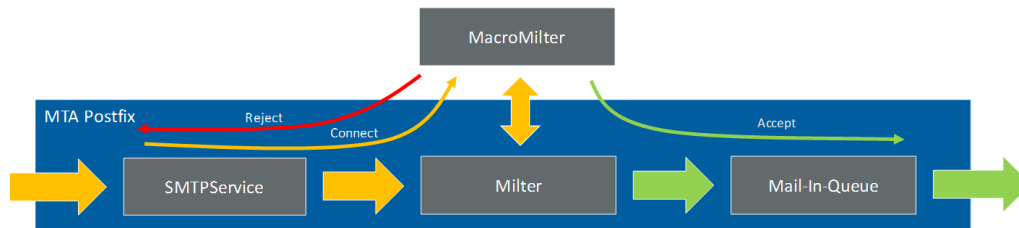


Abbildung 17: Position des *MacroMilters* im E-Mailfluss. Quelle: Autor

Wie in Abbildung 17 zu sehen ist, wird die Nachricht erst in die Empfangs-Queue für die Weiterverarbeitung gegeben, wenn die Analyse des Anhangs abgeschlossen ist. Um ein Anhang nicht doppelt analysieren zu müssen, ist der Einsatz einer Datenbank mit bekannten Malware-Anhängen als zielführend zu betrachten. Durch die Hinterlegung des Hash-Wertes des Malware-Dokuments, kann bei einem wiederholten Auftreten diese sofort als Malware klassifiziert und so die Durchlaufzeit verringern.

Folgende Schritte sind für die Analyse der Anhänge nötig:

1. E-Mail vom *Postfix* via Milter-Schnittstelle weiterleiten.
2. Annehmender Filter extrahiert den Anhang.
3. Prüfen ob, es sich um ein *Microsoft Office* Dokument handelt.
4. Prüfen ob, ein Makro-Code enthalten ist.
5. Dynamische Prüfung ob, Makro-Code auf das Malware-Heuristik passt.
6. E-Mail bei einem positiven Fund via *Postfix* an Sender zurückweisen und Hash-Wert in der Datenbank vermerken.

Für die Ermittlung der Effizienz und Anwendbarkeit dieses Ansatzes wurde hier vom Autor eine konkrete Implementierung basierend auf diesem Ansatz durchgeführt. Die Implementierung selbst wurde unter dem Namen *MacroMilter*⁴ erstellt. Grundlegend

⁴<http://sbdy.github.io/MacroMilter>

spiegelt dieser alle hier ermittelten Beschreibungen des Ansatzes wieder.

Messung

Der aus dem Ansatz resultierende *MacroMilter* wurde zusätzlich im Messzeitraum (April 2016) in das Messsystem implementiert. Somit kann eine exakte Analyse der Effizienz des Ansatzes gegenüber den klassischen Filterinstanzen des *Postfix* und *Amavisd-New* erstellt werden. Um das Defizit gegenüber diesen besser sichtbar zu machen, wurde der *MacroMilter* erst nach dem *Amavisd-New* eingebunden (siehe Abbildung 18). Somit erreichten den *MacroMilter* nur Nachrichten, die von diesem nicht gefiltert worden sind.

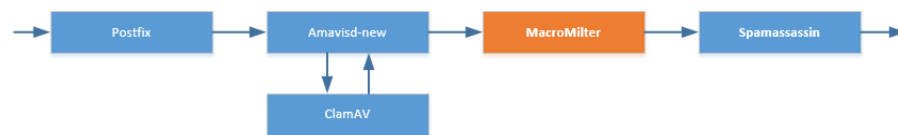


Abbildung 18: Position des *MacroMilters* in der Filterkette. Quelle: Autor

Die Analyse ergab, dass im Messzeitraum auf dem Messsystem 91 E-Mails mit einem potentiellen Malware-Code innerhalb eines *Microsoft Office* Dokuments vom *Postfix* und *Amavisd-New* nicht erkannt wurden. Hierbei konnte keiner der vorgelagerten Filter diese E-Mails als Malware oder Spam klassifizieren. Zusätzlich weisen 83 (75,70%) dieser Nachrichten einen Spam-Level von <5 auf und wären somit im Posteingang des Benutzers gelandet. Nur acht E-Mails haben einen Spam-Level ≥ 5 . Während des Messzeitraumes wurde nur eine Nachricht als False-Negativ vom *MacroMilter* erkannt. Daraus ergibt sich eine False-Negativ-Rate von 1,12%, was als sehr gering zu betrachten ist.

In der Abbildung 19 sind die erfassten Dateiendungen dargestellt. Hierbei wurden die Zurückweisungen des *Amavisd-New* und *MacroMilter* zusammen dargestellt. Im Falle der 11 *.zip* Zurückweisungen, wurde ein *Microsoft Office* Dokument mit Malware-Code innerhalb des komprimierten Ordners gefunden.

Auf den Gesamtanteil aller zurückgewiesenen Nachrichten gesehen ist der Anteil des *MacroMilters* geringe 0,27%. Jedoch geht von diesen 0,27% eine erhöhte Gefahr aus, da diese vom Virenschanner und Spam-Filter nicht erkannt wurden.

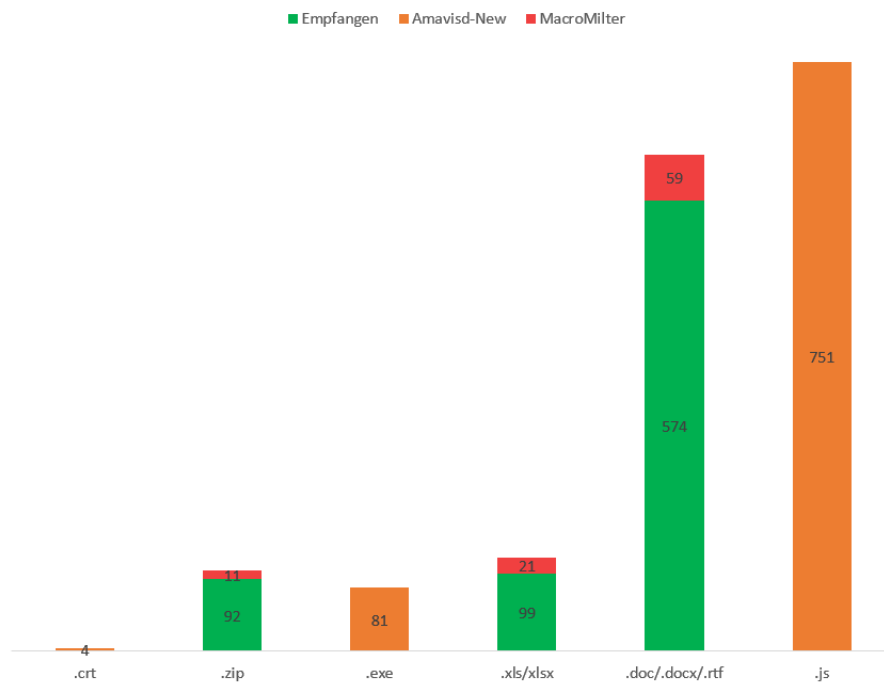


Abbildung 19: Kombination des *Amavisd-New* und *MacroMilters* für die zurückgewiesenen E-Mails mit den entsprechenden Dateiendungen. Quelle: Autor

Im weiteren Verlauf konnten so seit dem 01. Februar bis 01. Juni 2016 373 Dateien am produktiven E-Mailserver als Malware klassifiziert werden. Das macht im Monat 94 (93,25) erkannte Nachricht aus.

Zusätzlich konnte bei dieser Beobachtung festgestellt werden, dass die Angriffe immer in Wellen stattfinden. Die Auswertung der *MacroMilter* Statistik am Honeypot-System ergab, dass die meisten Angriffsversuche an Arbeitstagen von Montag bis Freitag zwischen 11:00 und 14:00 Uhr statt finden. Diese Werte könnten als zusätzliche Bewertungsgrundlage der Anhänge einfließen.

Trotz der Vorhaltung der eindeutigen Hash-Werte jeder schon überprüften Datei, kann eine starke Varianz innerhalb der Dateien bei den Angriffswellen festgestellt werden. Abbildung 20 zeigt die Anzahl der unterschiedlichen *Microsoft Office* Dokumente am Tag über einen längeren Zeitraum hinweg.

Innerhalb des Ansatzes wurde bewusst auf einen selbstlernenden Algorithmus verzichtet. Ein Trainieren dieses Algorithmus würde aufgrund der kleinen Datenbasis von legitimen Makro-Dokumenten erschwert werden. Zusätzlich ist durch den stati-

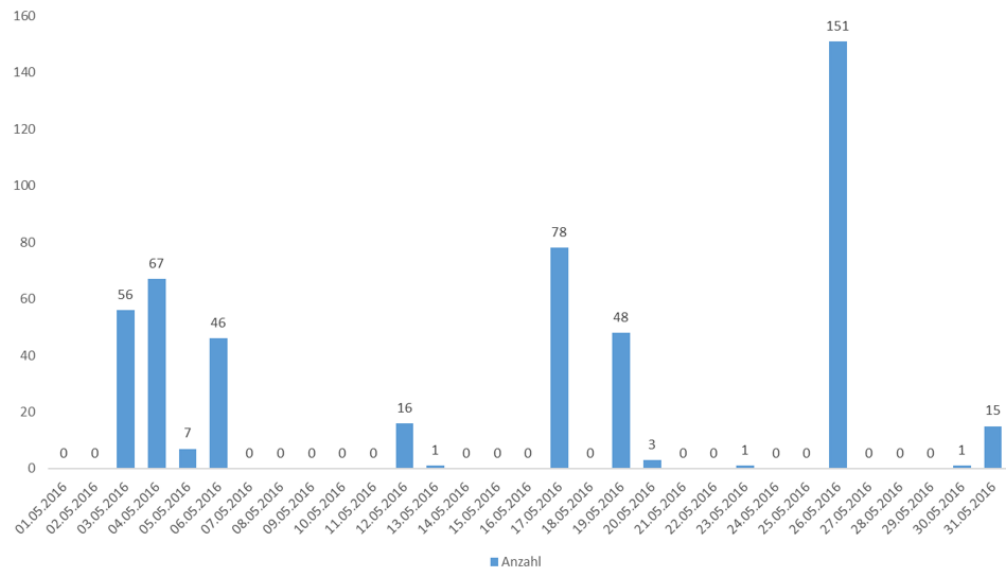


Abbildung 20: Anzahl der eindeutigen Malware-Dokumente am Honeypot-System im Monat Mai 2016. Quelle: Autor

schen Ansatz für die Bewertung auch eine höhere Sicherheit gegen Manipulationen gegeben.

5 Schlussbetrachtung

5.1 Vergleich zu anderen Ansätzen

Die genannten Ansätze sind sehr rudimentär und einfach in der Implementierung. Andere, meist kommerzielle, verfolgen andere Wege für die Verbesserung der Erkennung.

Einer dieser kommerziellen Ansätze ist es, den Fokus auf eine in der Cloud bereitgestellte Analyseplattform zu legen. Der Softwarehersteller *Avira*¹ hat sein Produktportfolio auf die neuen Bedrohungen via E-Mail angepasst und bietet neue Möglichkeiten zur Analyse von Dateien oder E-Mails in der von ihnen betriebenen Cloud. Hierbei werden bei einem negativen oder ausbleibenden Befund einer Datei durch den lokalen Erkennungsalgorithmus diese zur erweiterten Analyse in eine Cloud gesendet. Innerhalb dieser Cloud werden dann intelligentere und aufwändigere Analyseverfahren angewendet, um die Datei zu klassifizieren.

Andere Ansätze verfolgen eine Verhaltensanalyse der Malware-Programme durch das Ausführen in Sandboxen oder anderen abgeschlossenen Containerinfrastrukturen [32, S. 93] und [5, S. 4ff]. Der Hersteller von E-Mail-Filter *Proofpoint*² bietet diesen Dienst auch innerhalb einer Cloud-Lösung an. Hierbei wird der Anhang einer E-Mail noch zusätzlich auf unterschiedlichen Betriebssystemen virtuell in der Cloud ausgeführt und die Verhaltensweise der Schadsoftware analysiert. So kann eine Verhaltensanalyse durchgeführt werden und es wird dabei dynamisch bewertet, ob die Datei eine Gefahr darstellt oder nicht.

¹<https://www.avira.de>

²<https://www.proofpoint.com/>

Bei beiden Vorgehensweisen, kann bei der schnell wechselnde Malware-Bedrohung ein Laufzeitproblem entstehen. Eine Analyse der Daten muss in jedem unbekannten Fall durchgeführt werden. Das bedeutet, dass jede Datei, die den E-Mailserver passiert, potentiell als Bedrohung gesehen werden muss. Somit wird jede, dem Filter lokal unbekannte Datei in die Cloud gesendet und untersucht. Daraus kann eine Durchlaufzeit des Anhang von bis zu sechs Minuten (im Fall *Proofpoint*) entstehen, bis die Analyse ein Ergebnis ermittelt hat. Wie die Erhebung aus Kapitel 4.2.2 bestätigt, werden zu 88% nur Dokumente versendet. Dokumente haben durch einfaches Verändern des Textes eine sehr schnelle Änderungscharakteristik (siehe Abbildung 20). Seit Februar 2016 konnten im Honeypot-System rund 1.030 grundlegend unterschiedliche Makro behaftete Dokumente registriert werden. Wie gut ein Filter dieser Art mit der Masse an Dokumenten umgehen kann, gilt es noch zu analysieren. Fraglich ist, ob es in Bezug zum Datenschutz hierbei noch eine Fragestellung zu klären gibt.

Im kommerziellen Umfeld ist in den nächsten Jahren eine Fokussierung auf diese Cloud- und Sandbox-Lösungen zu erwarten. Marktführende Hersteller wie *Cisco*³, *Fireeye*⁴ und *Proofpoint*⁵ sind hierbei schon mit entsprechenden Produkten vertreten.

Im Gebiet der Spam-Verhinderung gibt es weitreichende Arbeiten im Bereich der künstlichen Intelligenz. Hier kann durch entsprechende textverarbeitende Algorithmen eine hohe und korrektere Erkennung erzeugt werden [10, S. 7]. Eine Durchdringen dieser Technologie in die Produkte für kleine bis mittlere Unternehmen ist erst in den kommenden Monaten zu erwarten. Zumeist werden diese noch aufgrund der Rechenkapazität in Großsystemen wie *Google Mail*⁶ oder *Microsoft Office 365*⁷ eingesetzt [45, S. 1ff].

Ein weiterer und schon aktiv genutzter Ansatz ist es, Links in Spam-E-Mails über eine zentrale Datenbank zu klassifizieren. Hierzu wird diese Datenbank von einem intelligenten Algorithmus befüllt, der dynamisch analysieren kann, ob es sich bei der Verlinkung um eine Malware- oder Phishing-Seite handelt. Eine bekannte Quelle für

³<https://www.cisco.com/>

⁴<https://www.fireeye.com/>

⁵<https://www.proofpoint.com/>

⁶<https://mail.google.com/>

⁷<https://products.office.com/en/business/explore-office-365-for-business>

solche Daten ist hier die *Google Safe Browsing*⁸ Datenbank.

5.2 Zusammenfassung

Diese Arbeit stellt einen repräsentativen Ausgangszustand von Mailinfrastrukturen in kleinen mit mittleren Unternehmen dar. Durch die Analyse der Messergebnisse konnten so in Bezug auf die neue Bedrohungslage neue Konzepte ausgebreitet werden. Die Projektion der neuen Ansätze auf die Ausgangsmessdaten stellt eine Aussicht auf die generelle Effizienz der Erkennung dar.

Die genannten 0,27% (siehe Abbildung 21), die nicht vom Malware-Filter erkannt wurden, sind trotz ihrer Unscheinbarkeit eine beachtliche Zahl. In Summe sind dies 91 Nachrichten, wovon in 83 Fällen auch nicht vom Spam-Filter klassifiziert worden sind, was 83 mal eine qualitativ hochwertige Bedrohung für die Anwender ausmacht. Das Infektionsrisiko, das von diesen Anhängen ausgeht ist, wie ermittelt, in den ersten 24 Stunden sehr hoch.

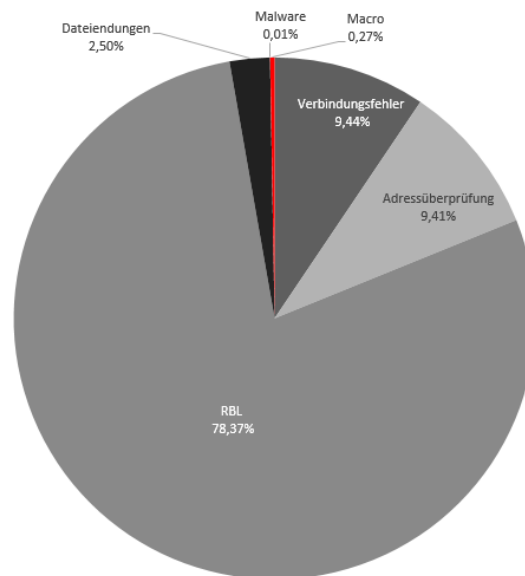


Abbildung 21: Anteil der vom MacroMilter erkannten Malware. Quelle: Autor

Im Bereich der Verhinderung von Spam konnte dies mit einfachen und vorhandenen Mitteln dargestellt werden. Es sind hierzu keine weiteren komplexen Implementierungen nötig. Durch die selbständige Erweiterung der Datenbank mit allen

⁸<https://developers.google.com/safe-browsing/>

ausgehenden E-Mailabsendern entsteht ein selbstlernendes System. So müssen keine speziellen Wartungen oder andere Aktualisierungen vorgenommen werden, um eine gleichbleibend effiziente Erkennung zu gewährleisten.

Auch große Internetfirmen wie *Google* haben angefangen, intelligentere Filtermethoden zu entwickeln und einzusetzen [38, S. 3]. Diese zeitgemäße Art, von zumeist kombinierten Filtermethoden, soll Malware und Spam auf eine neue Art erkennen und verhindern [36, S. 9]. In vielen Fällen sind bei dieser Vorgehensweise von Erkennungssystemen jedoch eine größere Rechenkapazität und laufende Optimierungen erforderlich, was bei einem kleineren Unternehmen aus Kosten- und KnowHow-Gründen nicht umsetzbar ist.

Sowohl die Erkenntnisse aus dieser Arbeit, als auch die von S. Kumar und S. Arumugam [25, S. 878] zeigen, dass klassischen Methoden der Analyse von Spam oder Malware keine ausreichende Sicherheit mehr bieten können. Neue Erkennungsmethode sind in naher Zukunft für eine weitere Sicherheitsoptimierung von E-Mailinfrastrukturen notwendig. Hierzu sind die unterschiedlichsten Ansätze denkbar. In vielen Fällen muss mit einer erhöhten Rechen- und Datenkapazität für die Filterung gerechnet werden. Diese Arbeit zeigt, dass auch mit einfachem und intelligentem Auswerten von schon vorhanden Informationen eine Erhöhung der Gesamtsicherheit erreicht werden kann. Um dies zu erreichen sind keine Großrechnersystem notwendig, sondern es reicht aus, einige Optimierungen der vorhandenen Infrastruktur vorzunehmen. Die Entwicklung und der Einsatz des *MacroMilter* zeigt diese Möglichkeit entsprechend deutlich auf.

5.3 Diskussion

Die erwähnten kommerziellen Anti-Malware-Produkte und Technologien sind im Bereich des Mittelstands und in Konzernen sicherlich eine gute Erweiterung der Sicherheitsinfrastruktur von E-Mailsystemen. Jedoch gilt es diesen Ansatz kritisch in Bezug auf die Effizienz und des Datenschutzes zu betrachten. Während der Erstellung der Arbeit wurde mit zwei führenden Herstellern von Anti-Malware- und Anti-Spam-Lösungen diskutiert. Im beiden Fällen, bei der Betrachtung des Cloud-, als auch Sandbox-Ansatzes, ist somit die Frage offen, ob die Erkennungsgeschwindigkeiten und Erkennungsraten der Malware und Spam-E-Mails im produktiven Alltag

überhaupt signifikant gesteigert werden können.

Eine grundlegende Verbesserung der beiden Werte wird hierbei nicht infrage gestellt, jedoch ob die Verhältnismäßigkeit des Vorganges in Bezug auf den Mehrwert zielführend ist. Zu beachten gilt es, dass in naher Zukunft auch andere Wege der Verteilung von Malware-Entwickler genutzt werden könnten. Kanäle für die Verbreitung könnten hier Messaging Produkte wie *WhatsApp*⁹ oder *Skype*¹⁰ sein. Beide verfügen über die Möglichkeiten, Links oder Dateien zu versenden. Derzeit sind mehrere dem Autor bekannte Konzern dabei, eine Evaluierung dieser neuen Kommunikationswege durchzuführen. Ziel soll es hier sein, diese in den produktiven Geschäftsablauf zu integrieren. Es stellt sich dann die Frage, wie innerhalb dieser zeitkritischen Bereiche die Filtermethoden die Zustellung verhindern können. Eine Durchlaufzeit der Filterung mit rund sechs Minuten pro Anhang ist hier nicht mehr möglich.

Eventuell ist das Blockieren von Anhängen, wie *Google* es bereits erkannt hat, die letzte Methode, Malware zu vermeiden. Links jedoch, wird man nicht so einfach verbieten können.

Offen bleibt auch die Frage nach dem Datenschutz bei einer Einsendung jedes Dokuments in eine Cloud des Anti-Malware Anbieters. Praktisch wird jeder Anhang durch eine, vom Benutzer nicht einsehbare, Infrastruktur geschleust.

Verschlüsselung im E-Mailverkehr

Eine fundamentale Verbesserung des gesamten E-Mailverkehrs durch das Konzept der objektbasierter Ende-zu-Ende-Verschlüsselung ist fraglich [23, S. 390ff]. In Bezug auf Spam und Malware würde dieses Vorgehen nur den Vorteil bieten, dass der Absender nicht mehr auf einfache Weise gefälscht und die Nachricht nur vom Empfänger gelesen werden kann. Hierdurch kann aber eine serverseitige Filterinstanz diese Nachricht nicht mehr untersuchen, da sie diese nur im verschlüsselten Zustand bereitgestellt bekommt. So kann ein Malware-Versender seine Nachricht mit einem ggf. gestohlenen Zertifikat verschlüsseln und keiner der Anti-Malware- oder Anti-Spam-Filter könnten sie auf diese Merkmale hin untersuchen. Zusätzlich wird dem Benutzer durch diese Verschlüsselung eine scheinbare Legitimität der Nachricht vorgetäuscht, was eine Infektion dramatisch erhöhen könnte.

⁹<https://www.whatsapp.com/>

¹⁰<https://www.skype.com/en/>

Mit dem Fokus auf Malware- und Spam-Bekämpfung sind hierbei noch einige Überlegung zu treffen, wie eine Verschlüsselung effektiv und richtig eingesetzt werden kann. Eine reine Signierung für den Schutz der Integrität wäre hier zumindest ein guter Einstiegspunkt.

5.4 Weiterführende Arbeit

Um eine Aussage über die Effizienz der beiden Ansätze zu erhalten, sind weiterführende Analysen und Auswertungen nötig. Durch die zeitliche Begrenzung dieser Arbeit war eine allgemeingültige und aussagekräftigere Bewertung nicht möglich. Dennoch wurde zwei Ansätze für eine schnelle und effiziente Filterung gezeigt.

Weiterführend ist der Ansatz zur Verhinderung von Spam in einer praktischen Anwendung zu betrachten. Eine konkrete Implementierung wäre hier als nächster Schritt zu sehen. Eine nachgelagerte Langzeitanalyse ist ebenso angedacht, um die Effizienz genauer zu betrachten.

Im Bereich der Verhinderung von Makro-Malware wurde schon eine konkrete Implementierung durchgeführt. Hierbei wurde der *MacroMilter* entwickelt, der sich an dem dargestellten Ansatz orientiert. Durch die Betrachtung der Messergebnisse des Filters soll dieser weiterhin optimiert werden.

5.5 Danksagung

An dieser Stelle möchte ich mich zunächst bei meinen beiden Betreuern Herren Prof. Dr. Martin Goik und Benjamin Kenner für das Vertrauen und die Möglichkeit für diese Arbeit danken. Zusätzlich möchte ich mein Kollege Peter Schiek für seinen Input und Gedankenstoß für dieses Thema danken.

Und zuletzt gilt der Dank meiner lieben Alex sowie Freunden und Familie für ihre Geduld und Hilfe. Ohne diese Unterstützung wäre ich nicht an dieser Zeile angekommen.

DANKE!

Glossar

Base64 ist eine 8-Bit Kodierung für Binärdaten.

BitCoin ist eine neuartige Währung und wurde 2008 das erste Mal erwähnt. Hierbei handelt es sich um eine selbstregulierende Währung, die nicht von Banken gesteuert wird. Angebot und Nachfrage beeinflussen hier den Kurs. Durch seine kryptographischen Eigenschaften ist die Zahlungsmethode bis zur Umwandlung in öffentlich gehandelten Währungen anonym.

Cloud steht für das englische Wort Wolke und wird in der IT als Synonym für verteiltes Rechnen von Daten verwendet. Anbieter solcher Rechnerverbände sind z.B. *Microsoft Azure*¹¹ oder *Amazon Web Services*¹².

Dropper Als Dropper wird eine Malware bezeichnet die nur die Aufgabe hat, andere Malware nachzuladen und auf dem System zu installieren.

False-Negativ stellt das Gegenteil von False-Positive dar. Eine Erkennung von Malware oder Spam bleibt hierbei durch den Filter aus und die Nachricht wird fälschlicher Weise zugestellt.

Fully Qualified Domain Name Diese Bezeichnung stellt einen vollständigen Name eines Systems oder Endpunktes im Netzwerk dar.

Ham Das englische Wort für Schinken wird sinnbildlich für eine gutartige und gewollte E-Mail verwendet. Ham stellt das Gegenteil von Spam dar.

¹¹<https://azure.microsoft.com/en-us/>

¹²<https://aws.amazon.com/>

Hash Ein Hash ist eine mathematische Einwegfunktion. Sie erstellt einen eindeutigen Wert über eine beliebige Menge dar. Die gleiche Menge ergibt immer denselben Hash-Wert. Ein Rückschluss vom Hash-Wert auf die Ursprungsmenge ist nicht möglich. Bekannte Algorithmen sind hier MD5, SHA-1 oder SHA-256.

Junk wird als Synonym für Spam verwendet.

Mail Delivery Agent Dieser Server-Typ ist für die Zustellung und Verwaltung der Benutzer-Mailboxen zuständig.

Mail Transfer Agent Ein Mail Transfer Agent ist eine E-Mailserver-Komponente, welcher für den Empfang, die Verarbeitung und anschließende Weiterleitung einer E-Mail zuständig ist. Häufig werden diese MTAs mit Filteraufgaben vertraut und sorgen als erste Instanz vor dem eigentlichen zu stellenden E-Mailserver, für eine Lastreduzierung. Er stellt keine E-Mails direkt an den Benutzer zu sondern übermittelt an den MDA, der dann diese Aufgabe übernimmt.

Milter steht für Mail-Filter und wird im Bezug auf den *Postfix* verwendet. Hierbei handelt es sich um eine vom *Postfix* bereitgestellte Schnittstelle für die Anbindung weiterer Mail-Filter.

Multipurpose Internet Mail Extensions Die *Multipurpose Internet Mail Extensions* bieten die Möglichkeit Kommunikationsdaten in Textform abzubilden. Sie werden innerhalb des *SMTP* verwendet. Diese textuelle Darstellung von E-Mails beinhaltet Anhänge, den Text und alle weiteren Metainformationen wie Absender, Empfänger, Datum und Uhrzeit usw..

Phishing Bei dieser Art von Angriff handelt es sich um die kriminelle Beschaffung von vertraulichen Informationen durch unterschiedlichste Methoden. Eine hiervon ist z.B. das Fälschen einer vertrauenswürdigen Webseite um das Opfer dazu zu bewegen, seine Anmeldedaten auf dieser anzugeben. So kann ein Angreifer entsprechende Informationen abgreifen.

Port-Scan Ein Port-Scan ist ein Vorgehen, bei dem ein Angreifer durch Probieren von Netzwerkports versucht, einen Überblick über das System zu erhalten.

Zumeist wird hier durch das Senden eines Netzwerkpaketes versucht herauszufinden, welchem Netzwerkdienst welcher Port zugeordnet ist.

Ransomware ist ein Akronym aus *Ransom* und *Software*, wobei Ransom das englische Wort für Erpressung ist. Somit ergibt sich aus Ransomware eine deutsche Übersetzung von Erpresser-Software.

Request For Comments oder RFC ist eine schriftliche Darstellung einer Diskussion über unterschiedliche Vorgehen in der IT. Hieraus entstehen im weiteren Verlauf unterschiedliche Standardisierungsvorlagen für z.B. Implementierungen von Internetprotokollen.

Spam ist ursprünglich die Abkürzung für das englische Dosenfleisch *Spiced Ham*. Dieses wurde durch seine Omnipräsenz im Krieg als Sinnbild für unnötiges und übertrieben häufiges Vorkommen genommen. In der Informationstechnik wird das Wort für unerwünschte E-Mails verwendet, die zumeist ungewollte Werbung oder Malware zum Inhalt haben.

Visual Basic Access ist eine Spezialisierung des Visual Basic. Hierbei handelt es sich um eine Programmiersprache, die von Microsoft entwickelt wurde. Bei der Spezialisierung handelt es sich um Anpassungen und Erweiterungen für *Microsoft Office* Dokumente. Mit dieser Sprache können komplexe Funktionen in ein Dokument integriert werden.

Literatur

- [1] *Alles über: Spam: E-Mail sicher und benutzbar halten*. Bd. 7. Linux-Magazin Technical review. München: Linux New Media, 2008. ISBN: 9783939551096.
- [2] Andrea Allievi und Holger Unterbrink. „CryptoWall4: The evolution continues“. In: (2016). URL: <http://blog.talosintel.com/2015/12/cryptowall-4.html>.
- [3] Ion Androutsopoulos, John Koutsias, Konstantinos V. Chandrinou, George Paliouras und Constantine D. Spyropoulos. *An Evaluation of Naive Bayesian Anti-Spam Filtering*. 2000.
- [4] Ion Androutsopoulos, John Koutsias, Konstantinos V. Chandrinou und Constantine D. Spyropoulos. *An Experimental Comparison of Naive Bayesian and Keyword-Based Anti-Spam Filtering with Personal E-mail Messages: Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. New York, N.Y.: Association for Computing Machinery, 2000. ISBN: 9781581132267.
- [5] Ulrich Bayer, Paolo Milani Comparetti, Clemens Hlauschek, Christopher Kruegel und Engin Kirda. *Scalable, Behavior-Based Malware Clustering*.
- [6] Bayerische Polizei, Hrsg. *Erpressungs-Trojaner auf Server einer Stadtverwaltung – Kripo ermittelt*. 2016. URL: <https://www.polizei.bayern.de/unterfranken/news/presse/aktuell/index.html/237562>.
- [7] Enrico Blanzieri und Anton Bryl. „A survey of learning-based techniques of email spam filtering“. In: *Artificial Intelligence Review* 29.1 (2008), S. 63–92. ISSN: 0269-2821. DOI: 10.1007/s10462-009-9109-6. URL: <http://eprints.biblio.unitn.it/1070/1/056.pdf>.

- [8] Andrew Brandt. *Anatomy of a Word Document Macro Malware Attack*. Hrsg. von Blue Coat Systems Inc. 2016. URL: <https://www.bluecoat.com/security-blog/2015-05-12/anatomy-word-document-macro-malware-attack>.
- [9] Bundesamt für Sicherheit in der Informationstechnik. *Ransomware - Bedrohungslage, Prävention & Reaktion*. Deutschland, 2016. URL: https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Cyber-Sicherheit/Themen/Ransomware.pdf?__blob=publicationFile&v=2.
- [10] Xavier Carreras und Lluís Màrquez. *Boosting Trees for Anti-Spam Email Filtering*. Barcelona, 2001.
- [11] Chiou Pin-Ren, Lin Po-Ching und Li Chun-Ta. „Blocking Spam Sessions with Greylisting and Block Listing based on Client Behavior“. In: (2012). URL: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.401.3018&rep=rep1&type=pdf>.
- [12] Zakir Durumeric, J. Alex Halderman, David Adrian, Ariana Mirian, James Kasten, Elie Bursztein, Nicolas Lidzborski, Kurt Thomas, Vijay Eranti und Michael Bailey. „Neither Snow Nor Rain Nor MITM“. In: *the 2015 ACM Conference*. Hrsg. von Kenjiro Cho, Kensuke Fukuda, Vivek Pai und Neil Spring, S. 27–39. DOI: 10.1145/2815675.2815695.
- [13] Ronald Eikenberg. *Neue Masche: Krypto-Trojaner Locky über Javascript-Dateien verbreitet*. Hrsg. von Heise Medien GmbH & Co. KG. 2016. URL: <http://heise.de/-3113689>.
- [14] Ronald Eikenberg und Schmidt Jürgen. „Tojaner auf Raubzug: Verschlüsselungs-Malware analysiert“. In: *c't magazin für computer technik* 7 (2016), S. 86–87.
- [15] Markus Engelberth. „Social Malcode: Benutzerabhängige Schadprogramme“. Dissertation. Mannheim: Universität Mannheim, 2013. URL: <http://ub-madoc.bib.uni-mannheim.de/34511/1/promotionsarbeit.pdf>.
- [16] Federal Bureau of Investigation, Hrsg. *Criminals Continue to Defraud and Extort Funds from Victims Using CryptoWall Ransomware Schemes*. 2015. URL: <https://www.ic3.gov/media/2015/150623.aspx>.
- [17] Thomas Fox-Brewster. *As Ransomware Crisis Explodes, Hollywood Hospital Coughs Up \$17,000 In Bitcoin*. Hrsg. von Forbes. 2016. URL: <http://www.forbes.com/sites/thomasbrewster/2016/02/18/ransomware-hollywood-payment-locky-menace>.

- [18] Alexandre Gazet. „Comparative analysis of various ransomware virii“. In: *Journal in Computer Virology* 6.1 (2010), S. 77–90. ISSN: 1772-9890. DOI: 10.1007/s11416-008-0092-2.
- [19] Paul Graham. *A Plan for Spam*. 2002. URL: <http://www.paulgraham.com/spam.html>.
- [20] K. Harrenstien, M. Stahl und E. Feinler. *RFC 952*. 1985. URL: <https://tools.ietf.org/html/rfc952>.
- [21] Hans Irlacher. *Advanced Threat Protection*. 31.03.2016.
- [22] Axel Kannenberg. *Ransomware: US-Krankenhaus zahlt 40 Bitcoins Lösegeld*. Hrsg. von Heise Medien GmbH & Co. KG. 2016. URL: <http://heise.de/-3109956>.
- [23] Walter Kriha und Roland Schmitz. *Sichere Systeme: Konzepte, Architekturen und Frameworks*. Xpert.press. Berlin: Springer, 2009. ISBN: 9783540789581. DOI: 10.1007/978-3-540-78959-8.
- [24] Murray S. Kucherawy und Dave Crocker. *RFC 6647: Email Greylisting: An Applicability Statement for SMTP*. 2012. URL: <https://tools.ietf.org/html/rfc6647.html>.
- [25] S. Kumar und S. Arumugam. „A Probabilistic Neural Network Based Classification of Spam Mails Using Particle Swarm Optimization Feature Selection“. In: *Middle-East Journal of Scientific Research* 23 (2015), S. 874–879. URL: <http://idosi.org/mejsr/mejsr23%285%2915/17.pdf>.
- [26] Philippe Lagadec. *8KB of malware crammed into a single command line in a macro*. 2016. URL: http://www.decorage.info/en/8KB_oneliner.
- [27] William Largent. *Ransomware: Past, Present, and Future*. Hrsg. von Cisco Systems. 2016. URL: <http://blog.talosintel.com/2016/04/ransomware.html>.
- [28] Gunnar Lindberg. *Anti-Spam Recommendations for SMTP MTAs*. Hrsg. von IETF. 1999. URL: <https://tools.ietf.org/html/rfc2505>.
- [29] Microsoft Malware Protection Center, Hrsg. *New feature in Office 2016 can block macros and help prevent infection*. 2016. URL: <https://blogs.technet.microsoft.com/mpmc/2016/03/22/new-feature-in-office-2016-can-block-macros-and-help-prevent-infection/?platform=hootsuite>.
- [30] Microsoft, Hrsg. *Malware Protection Center: Macro malware: Summary*. 2015. URL: <http://download.microsoft.com/download/A/E/5/AE546268->

- 519C-4A7E-8EC4-9EE29A80B79D/MMPC%20Threat%20Intelligence%20July%202015.pdf.
- [31] Fabio Pagani, Matteo De Astis, Mariano Graziano, Andrea Lanzi und Davide Balzarotti. *Measuring the Role of Greylisting and Nolisting in Fighting Spam*.
 - [32] Roberto Paleari. „Dealing with next-generation malware“. Dissertation. Mailand: Universität Mailand, 2011.
 - [33] Norbert Pohlmann. „Bedrohungen und Herausforderungen des E-Mail-Dienstes: Die Sicherheitsrisiken des E-Mail-Dienstes im Internet“. In: *Datenschutz und Datensicherheit - DuD* 34.9 (2010), S. 607–613. ISSN: 1614-0702. DOI: 10.1007/s11623-010-0145-9.
 - [34] Proofpoint, Hrsg. *Human Factor Report 2016*. 2016. URL: <https://www.proofpoint.com/us/human-factor-report-2016>.
 - [35] Niels Provos, Dean McNamee, Panayiotis Mavrommatis, Ke Wang und Nandendra Modadugu. *The Ghost In The Browser: Analysis of Web-based Malware*. Hrsg. von Google Inc. DOI: 10.1787/578054332028. URL: https://www.usenix.org/legacy/events/hotbots07/tech/full_papers/provos/provos.pdf.
 - [36] Moheeb Abu Rajab, Lucas Ballard, Noè Lutz, Panayiotis Mavrommatis und Niels Provos. *CAMP: Content-Agnostic Malware Protection*. Hrsg. von Google Inc.
 - [37] Peter W. Resnick. *Internet Message Format: RFC 2822*. 2001. URL: <https://www.ietf.org/rfc/rfc2822.txt>.
 - [38] Konrad Rieck, Philipp Trinius, Carsten Willems und Thorsten Holz. „Automatic Analysis of Malware Behavior using Machine Learning“. In: *IOS Press* (2011).
 - [39] S. Ritterbusch. *Die Mathematik des Bayes Spamfilters*. Hrsg. von Karlsruher Institut für Technologie. URL: <http://www.math.kit.edu/ianm4/~ritterbusch/seite/spam/de>.
 - [40] Henry Stern, Justin Mason und Shepherd Michael. *A Linguistics-Based Attack on Personalised Statistical E-mail Classifiers: Technical Report CS-2004-06*. 2004. URL: https://www.cs.dal.ca/sites/default/files/technical_reports/CS-2004-06.pdf.
 - [41] Symantec Corporation, Hrsg. *Security Response Publications*. 2016. URL: https://www.symantec.com/security_response/publications/monthlythreatreport.jsp#Spam.

- [42] The Apache Software Foundation, Hrsg. *The Apache SpamAssassin Project: Tests Performed: v3.2.x*. 2014. URL: https://spamassassin.apache.org/tests_3_2_x.html.
- [43] The Radicati Group Inc., Hrsg. *Email Statistics Report 2015-2019*. 2015. URL: <http://www.radicati.com/wp/wp-content/uploads/2015/02/Email-Statistics-Report-2015-2019-Executive-Summary.pdf>.
- [44] Jörg Thoma. "Antivirensoftware ist tot". Hrsg. von golem.de. 2014. URL: <http://www.golem.de/news/symantec-antivirensoftware-ist-tot-1405-106251.html>.
- [45] Kurt Thomas, Chris Grier, Justin Ma, Vern Paxson und Dawn Song. „Design and Evaluation of a Real-Time URL Spam Filtering Service“. In: *2011 IEEE Symposium on Security and Privacy (SP)*, S. 447–462. DOI: 10.1109/SP.2011.25.
- [46] Ivo Truxa. *The Auto-WhiteList*. 2014. URL: <https://wiki.apache.org/spamassassin/AutoWhitelist>.
- [47] Gregory M. Vaudreuil. *Enhanced Mail System Status Codes: RFC 3463*. 2003. URL: <https://tools.ietf.org/html/rfc3463>.
- [48] Michael Veit. *Sofortmaßnahmen gegen Krypto-Trojaner: Ein Sophos Whitepaper*. Hrsg. von Sophos Ltd. 2016. URL: <https://www.sophos.com/de-de/medialibrary/Gated%20Assets/white%20papers/Sophos-emergency-measures-against-crypto-Trojan-wp.pdf>.
- [49] Verizon, Hrsg. *Data Breach Investigations Report 2016*. 2016. URL: <http://www.verizonenterprise.com/verizon-insights-lab/dbir/2016/>.
- [50] Verizon, Hrsg. *Data Breach Investigations Rreport 2015*. 2015. URL: http://www.verizonenterprise.com/resources/reports/rp_data-breach-investigation-report_2015_en_xg.pdf.
- [51] Wikipedia, Hrsg. *Naive Bayes spam filtering*. URL: https://en.wikipedia.org/wiki/Naive_Bayes_spam_filtering.
- [52] Wikipedia, Hrsg. *Phishing*. URL: <https://de.wikipedia.org/wiki/Phishing>.
- [53] postfix.org, Hrsg. *Postfix before-queue Militer support*. URL: http://www.postfix.org/MILTER_README.html.