Email Authentication für Empfänger

Sebastiaan de Vos, Patrick Ben Koetter

Version 0.3, 31.03.2022

Inhaltsverzeichnis

Dokumentengeschichte

Titel	Datum	Verfasst durch	Kürzel
Email Authentication für Empfänger	31.03.2022	Sebastiaan de Vos	SdV

Version	Datum	Beschreibung	Kürzel
0.6	30.05.2022	Migration aus altem Repo nach Review von MW	PBK
0.5	15.04.2022	Termiologien (From-Header / Envelope)	SdV
0.4	14.04.2022	Detailkorrektur	MK, SdV
0.3	31.03.2022	Notizen ausformuliert	PBK
0.2	15.03.2022	Migration auf Asciidoc	PBK
0.1	14.03.2022	Erstes Draft	SdV

Das Fälschen der Absenderadressen und damit das Vorspiegeln einer falschen Identität ist eine der häufigsten Formen des Betrugs in Internet E-Mail. Indem Angreifer sich als jemand anderes ausgeben, wollen sie ihrem Opfer Informationen entlocken (z. B. Phishing) oder dieses dazu bewegen, eine für die Angreifer nützliche Handlung (z. B. CEO-Fraud) zu begehen. Dies führt auf Seiten der Opfer zu Misstrauen in E-Mail im Allgemeinen und es verursacht großen wirtschaftlichen Schaden für Privatpersonen wie auch für Unternehmen. In den vergangenen Jahren haben E-Mail Experten deshalb mehrere Methoden entwickelt, um diese Form des Missbrauchs einzudämmen.

Die drei wichtigsten Methoden werden kombiniert eingesetzt, um a) für eine Envelope Sender Domain sendende Systeme zu legitimieren (SPF), b) die Identität einer Domain zu verifizieren (DKIM) und c) um eine Richtlinie (DMARC) festzulegen, wie mit Nachrichten verfahren werden soll, welche SPF und DKIM nicht gerecht werden, sowie um Reports über den aktuellen Status möglichen Identitätsmissbrauchs zu erhalten. Die drei genannten Methoden werden unter dem Begriff „Email Authentication“ zusammengefasst.

Email Authentication

Email Authentication kombiniert die Methoden von SPF, DKIM und DMARC zu einem Mechanismus mit dem eingehende Nachrichten auf ihre Authentizität geprüft werden können. Die Methoden stellen für sich genommen die folgenden Möglichkeiten zur Verfügung:

SPF

SPF gestattet zu erkennen, ob senden wollende Systeme legitimiert sind im Namen einer Envelope Sender Domain zu senden oder nicht und auch wie mit denen, die nicht legitimiert sind, verfahren werden soll.

DKIM

DKIM gestattet zu erkennen, ob eine Nachricht von der im `DKIM-Signature:`-Header angegebenen Domain signiert wurde und ob der `body` oder ausgewählte Header der Nachricht verändert wurden.

DMARC

DMARC erfordert für eine Nachricht eine erfolgreiche Authentifizierung per SPF oder DKIM der `From:`-Header Domain. Darüber hinaus legt DMARC fest, welche Richtlinie bei Verletzungen von SPF und DKIM angewandt werden soll **und** ermöglicht durch Hinterlegung einer Kontaktadresse den Empfang von sog. Feedback Reports über die Authentifizierungsergebnisse einer Domain.

Dieses Dokument betrachtet Email Authentication aus Sicht eines empfangenden Mailsystems. Es nennt Software und Konfigurationsbeispiele für SPF, DKIM und DMARC, damit E-Mail vor der Annahme authentifiziert, Identitätsmissbrauch erkannt und die EmpfängerInnen vor missbräuchlichen Nachrichten geschützt werden können. Ziel ist, nur Nachrichten zuzustellen, die den senderseitigen Richtlinien für SPF, DKIM und DMARC gerecht werden.

Email Authentication für Sender?

Es ist ebenso wichtig die eigene(n) Senderdomain(s) mit den Methoden von SPF, DKIM und DMARC zu legitimieren und für andere verifizierbar zu machen. Bereits heute, in Zukunft aber noch viel mehr, wird die Zustellbarkeit eigener Nachrichten an fremde Systeme wesentlich von korrekter Email Authentication abhängen.

Was dazu getan werden muss, wird aber nicht in diesem Dokument behandelt. Informationen hierzu finden sich z. B. auf https://certified-senders.org oder https://dmarc.org. Sie richten sich besonders an Sender und beschränken sich vor allem darauf, wie DMARC konfiguriert werden sollte.

Die nachfolgenden Abschnitte zeigen, wie Sie mit Hilfe verschiedener Open Source Softwares die Richtlinien von SPF, DKIM und DMARC erkennen, auswerten und anwenden können.

Terminologien

Brief	**E-Mail Part**	**Bezeichnung laut RFC**	**Bezeichnung in dieses Dokument**
Absender am Briefumschlag	Message Envelope	RFC5321.MailFrom	Envelope Sender
Empfänger am Briefumschlag	Message Envelope	RFC5321.RcptTo	Empfänger
Absender auf Brief	Message Header	RFC5322.From	From-Header

1. Risikobetrachtung

Dieser Abschnitt behandelt das Risiko von SPF, DKIM und DMARC für eine verzögerte Zustellung sowie den Verlust legitimer Nachrichten.

Alle drei Methoden brauchen etwas Zeit, um ihre Aufgabe zu erfüllen, aber die Verzögerung, die dabei entsteht, bewegt sich im Bereich von Millisekunden. Der größte Zeitfaktor besteht bei SPF in (potentiell sequentiellen) DNS-Abfragen und bei DKIM in einer DNS-Abfrage sowie einer Signatur-Verifikation. DMARC besteht nur aus einer DNS-Abfrage und wurde so konzipiert, dass der Einfluss auf die Zustellung so gering wie möglich bleibt:

> Scalability is a major issue for systems that need to operate in a system as widely deployed as current SMTP email. For this reason, DMARC seeks to avoid the need for third parties or pre-sending agreements between senders and receivers. This preserves the positive aspects of the current email infrastructure.
>
> — RFC 7489, Abschnitt 2.3

Für das Risiko des Nachrichtenverlusts ist es wichtig, die Grundidee von DMARC zu verstehen: DMARC veröffentlicht Richtlinien für den Umgang von Verstößen gegen SPF und DKIM. Hierbei

erfordert DMARC, dass eine E-Mail mit mindestens einer der beiden Methoden, SPF oder DMARC, konform ist. Falls beide Methoden fehlschlagen, so gilt eine E-Mail als nicht authentisch.

Wenn ein Angreifer eine fremde Domain für einen illegitimen Nachrichtenversand missbraucht, wird dies im Regalfall dazu führen, dass bei der Nachricht sowohl die SPF- als auch die DKIM-Prüfung fehlschlägt. Die Frage auf Empfängerseite ist nun, wie mit diesen „Fehlern“ verfahren werden soll.

An dieser Stelle setzt die Policy an, die DMARC mit Hilfe des `p`-tags im DMARC-Eintrag im DNS der From-Header Domain veröffentlicht. Drei Werte sind für das `p`-tag zulässig:

`none`
: Ist `none` gesetzt, fordert die im `From:`-Header angegebene Senderdomain, dass nichts unternommen werden soll, wenn es zu Verstößen gegen SPF und DKIM kommt.

`quarantine`
: Ist `quarantine` gesetzt, fordert die im `From:`-Header angegebene Senderdomain, dass die Nachricht zwar angenommen, aber nicht direkt in die Mailbox zugestellt, sondern in Quarantäne, z. B. den SPAM-Ordner, gelegt werden soll.

`reject`
: Ist `reject` gesetzt, fordert die im `From:`-Header angegebene Senderdomain, dass die Annahme der Nachricht verweigert und diese nicht zugestellt werden soll.

Dieser Mechanismus ist einfach und er funktioniert schnell und zuverlässig. Er kann aber dazu führen, dass auch legitime Nachrichten abgelehnt würden, wenn eine Senderdomain (temporär) selbst ihre DMARC-Richtlinie nicht aufrecht erhält.

Friendly fire?

- Gerade in größeren Organisation kommt es immer wieder vor, dass Mailsysteme rechtmäßig E-Mails im Namen der Organisation versenden, diese dazu aber nicht per SPF legitimiert wurden. Dies ist dann der Fall, wenn die IP-Adressen des verwendeten Mailsystem nicht in dem SPF-Eintrag der Envelope Sender Domain vorkommen. [1].
- Auch kommt es vor, dass das öffentliche DKIM-Schlüsselmaterial im DNS der DKIM-signierenden Domain (entspricht im Regelfall der From:-Header Domain) falsch eingegeben oder übertragen wurde und in der Folge, obwohl der private Signierschlüssel valide ist, die Verifizierung der DKIM-Signatur fehlschlägt.
- Ihr Mailsystem sendet eine legitime Nachricht an eine Mailingliste, die diese Nachricht auf eine nicht mit DMARC kompatible Weise weiterverteilt. Ein typisches Problem ist es, wenn die Mailingliste den From:-Header unverändert beibehält, dabei jedoch die DKIM-Signatur entweder entfernt oder durch Modifikation der Nachricht invalidiert.

Diese drei beispielhaft genannten Szenarien zeigen, wie eine DMARC `reject`-Policy die Zustellung legitimer Nachrichten gefährdet.

In den ersten beiden Fällen wäre es Aufgabe der Senderdomain, die SPF- und DKIM-Konfiguration zu korrigieren und mit DMARC-Montoring darauf zu achten, dass keine eigenen Konfigurationsfehler für Zustellprobleme sorgen. Für den zuletzt genannten Fall ist es Aufgabe der Mailingliste, den

Nachrichtenversand DMARC-konform durchzuführen. Des Weiteren wurde ein Verfahren namens ARC entwickelt, um die Authentizität einer Nachricht über mehrere Instanzen hinweg zu transportieren. ARC ist jedoch in einem experimentellen Stadium und hat sich deshalb noch nicht etabliert.

Als Empfänger können Sie optional Ausnahmen konfigurieren, um aus Ihrer Sicht legitime und vertrauenswürdige Mailsysteme von der SPF-, DKIM- und DMARC-Prüfung auszunehmen. Des Weiteren sollten Sie DMARC-Reports versenden, damit Sender, deren Nachrichten gegen ihre eigene DMARC-Richtlinie verstoßen, davon erfahren und dies korrigieren können.

2. Die richtige Software

Um es vorweg zu sagen: Die eine Software, die alles perfekt kann, existiert nicht. Je nach Anwendungsfall passt sich aber die eine oder andere Software besser in Ihre Dienstarchitektur ein.

Wenn Sie eine modulare Architektur bevorzugen oder einen Maildienst betreiben, der auf mehrere Instanzen oder gar Maschinen verteilt ist, eignet sich Software, die auf einen Aspekt beschränkt ist wie z. B. SPF verifizieren, DKIM authentifizieren, DMARC-Richtlinien anwenden und Feedback Reports generieren, besser als eine monolitische Anwendung. Für diesen Anwendungsfall eignen sich die folgenden Software-Produkte:

Modulare Software

- OpenSPF
- OpenDKIM
- OpenDMARC

Wenn Sie hingegen eine All-in-One-Lösung wünschen oder einen Maildienst betreiben, der alles in einem Server vereint, eignet sich ein Monolith besser. Für diesen Anwendungsfall stehen die folgenden Software-Produkte zur Verfügung:

Monolitische Software

- rspamd
- Authentication Milter

Alle genannten Softwares unterliegen einer Open-Source-Lizenz und setzen ein Linux oder ein anderes Unix-oide Betriebssystem zur Ausführung voraus. Die nachfolgenden Abschnitte zeigen, wie sie für die verschiedenen Aufgabenstellungen konfiguriert werden.

3. DMARC prüfen

Die Prüfung einer eingehenden Nachricht gegen eine DMARC-Richtlinie besteht aus folgenden Prüfschritten:

1. Ob das sendende Systeme durch die SPF-Policy der Envelope Sender Domain legitimiert wurde.
2. Ob die Nachricht eine DKIM-Signatur in sich trägt und ob diese erfolgreich verifiziert werden kann.
3. Ob die From:-Header Domain eine DMARC-Policy veröffentlicht hat und ob die SPF- oder DKIM-

Prüfung für die From:-Header Domain erfolgreich war.

Ist dies der Fall, steht aus Sicht der DMARC-Policy einer Annahme der Nachricht nichts entgegen. Verletzen das sendende Systeme oder die Nachricht hingegen die DMARC-Policy, so soll das empfangende System die DMARC-Policy-Vorgaben bei Verletzungen umsetzen.

Diese drei Aufgabenkomplexe – SPF, DKIM und DMARC – können nacheinander von Software-Modulen oder innerhalb einer Software bearbeitet werden. Unabhängig davon welche Architektur Sie wählen, werden die Programme einen `Authentication-Results:`-Header in die geprüfte Nachricht eintragen – er listet die verschiedenen Prüfergebnisse:

mx.example.com dokumentiert die Prüfergebnisse von sender@example.net

```
Authentication-Results: mx.example.com;
    dkim=pass header.d=example.net header.s=202203-example.net header.b=dhpvJqM6;
    dmarc=pass (policy=reject) header.from=example.net;
    spf=pass (mx.example.com: domain of sender@example.net designates 192.2.0.1 as
permitted sender) smtp.mailfrom=sender@example.net
```

Prüfungsergebnisse fälschen

Ein Angreifer, der bewusst E-Mails mit gefälschten Absender-Adressen in Umlauf bringt, um Betrug zu begehen, wird auch versuchen SPF, DKIM und DMARC zu unterlaufen, indem er selbst in die E-Mails gefälschte „Prüfergebnisse“, in Form eines `Authentication-Results:`-Header einschleust.

Diese Betrugsversuche können unterbunden werden, indem für die eigenen Programme festgelegt wird, welchen `Authentication-Results:`-Headern sie trauen und welche sie ignorieren sollen. Dies geschieht indem ein Host oder eine Domain benannt wird, welchen die Programme vertrauen sollen.

In den nachfolgenden Beispielen wird immer `example.com` bzw. eine Subdomain dieser Domain verwendet. Passen Sie die Domain entsprechend der Domain Ihrer eigenen Mailplattform an.

3.1. Modulare Verarbeitung

Zur modularen Verarbeitung werden die Softwares OpenDKIM und OpenDMARC nacheinander über die MILTER-Schnittstelle des MTA in die Verarbeitung eingehender Nachrichten eingebunden. Dabei übernimmt OpenDMARC zwei Aufgaben – jene der SPF-Legitimation und jene der DMARC-Policy-Prüfung.

Der DMARC-Standard schreibt nur eine SPF-Prüfung zwingend vor und betrachtet das Anbringen gültiger DKIM-Signaturen als optional. Deshalb ist es praktisch, beide Aufgaben von einer Applikation (hier: OpenDMARC) abarbeiten zu lassen.

Aber die Bedeutung von IP-Adressen in Reputationssystemen lässt, nicht zuletzt wegen cloud-basierter Maildienste und deren wechselnden IP-Adressen, stetig nach und so empfiehlt die Kompetenzgruppe „E-Mail“ des eco-Verbandes in jedem Fall immer auch DKIM-Signaturen als „zweites Pfand“ mit anzubringen und auf diese bei der Annahme von Nachrichten, im vorliegenden Fall mit OpenDKIM, zu prüfen.

3.1.1. OpenDKIM

OpenDKIM kann die DKIM-Signaturen eingehender Nachrichten verifizieren und es kann DKIM-Signaturen auf ausgehende Nachrichten anbringen. Das Programm steht in allen gängigen Linux-Distributionen zur Verfügung und sein Verhalten wird für gewöhnlich über die Datei `/etc/opendkim.conf` gesteuert.

Das nachfolgende Beispiel konfiguriert OpenDKIM sich mit Hilfe des `Socket`-Parameters lokal an die IP-Adresse `127.0.0.1` auf dem TCP-Port `8892` zu binden, dort auf eingehende Nachrichten zu warten und deren DKIM-Signaturen - so vorhanden - zu verifizieren.

/etc/opendkim.conf

```
Syslog          true
Socket          inet:8892@127.0.0.1
AuthservID      mx.example.com          ①
Mode            v                       ②
```

① Der Parameter `AuthservID` legt fest, welche Identität OpenDMARC verwendet, wenn es seine Prüfergebnisse in einen `Authentication-Results:`-Header einträgt.

② Der Parameter `Mode` legt mit der Option `v`, dass OpenDKIM E-Mails verifizieren soll.

3.1.2. OpenDMARC

OpenDMARC prüft ob eingehende Nachrichten konform mit der DMARC-Policy der From-Header Domain sind, generiert optional DMARC-Reports und prüft (ebenso optional) ob sendende Server entsprechend der SPF-Policy der Envelope Sender Domain zum Senden im Namen dieser Domain legitimiert sind. Eine detailierte Funktionsbeschreibung stellt das Trusted Domain Project zur Verfügung, welches OpenDMARC entwickelt und veröffentlicht.

OpenDMARC Milter muss nach OpenDKIM als MILTER in den MTA eingebunden werden. Die DKIM-Prüfung muss abgeschlossen und ein `Authentication-Results:`-Header eingetragen sein wenn OpenDMARC auf SPF und DMARC zu prüfen beginnt.

Das Verhalten der Applikation wird für gewöhnlich mit Hilfe der Konfigurationsdatei `/etc/opendmarc.conf` gesteuert. Im nachfolgenden Beispiel wird OpenDMARC konfiguriert, sich lokal an die IP-

Adresse `127.0.0.1` auf dem TCP-Port `8893` zu binden, dort auf eingehende Nachrichten zu warten, eine SPF-Prüfung vorzunehmen, den `Authentication-Results:`-Header, welchen das vorangeschaltete OpenDKIM bereits eingefügt hat, auszuwerten und anschließend die – so vorhanden – DMARC-Policy der From-Header Domain zu prüfen.

/etc/openmarc.conf

```
Syslog              true
Socket              inet:8893@127.0.0.1          ①
AuthservID          mx.example.com               ②
TrustedAuthservIDs  mx.example.com               ③
SPFSelfValidate     true                         ④
RejectFailure       yes                          ⑤
```

① OpenDMARC kann mittels `inet`- als auch über einen `local`-Socket vom MTA angesprochen werden.

② Die Option `AuthservID` legt fest welche Identität OpenDMARC verwendet, wenn es seine Prüfergebnisse in einen `Authentication-Results:`-Header einträgt.

③ Diese Option legt fest, welchen bereits vorhandenen `Authentication-Results:`-Headern OpenDMARC vertraut und welche es widerum ignoriert. Die Identität des in diesem Abschnitt vorgeschalteten OpenDKIM muss hier gelistet sein, damit OpenDMARC dessen Prüfergebnisse in seine DMARC-Policy-Prüfung mit einbezieht.

④ Mit `SPFIgnoreResults true` wird die SPF Prüfung immer von OpenDMARC vorgenommen. Mit `SPFSelfValidate true` wird SPF nur von OpenDMARC geprüft, wenn sonst keine SPF-Prüfung im `Authentication-Result:`-Header vorhanden ist.

⑤ Opendmarc lehnt ohne weitere Konfiguration gar keine Mails ab. Dazu muss man explizit `RejectFailure yes` setzen.

3.2. Monolithische Verarbeitung

Monolithische Verarbeitung bedeutet alle Verarbeitungschritte für SPF, DKIM und DMARC finden in einer Applikation statt. Die populärste Software dafür stellt das Programm `rspamd` dar.

3.2.1. rspamd

rspamd ist mehr als nur ein Programm, um E-Mail Authentication durchzuführen. Begonnen als hochperformanter Ersatz für die Software `SpamAssassin` wurde `rspamd` mit der Zeit zu einer All-In-One-Lösung erweitert. Es bietet viele Filter-Funktionen und wird laufend aktualisiert und erweitert.

Bereits im Auslieferungszustand prüft rspamd, ob eine From-Header Domain über eine DMARC-Policy verfügt und vermerkt die Prüfergebnisse im Header der entsprechenden E-Mail, **aber** es führt nicht die mit der DMARC-Policy verbundenen Aktionen aus. Aktivieren Sie diese Aktionen mit folgender Konfiguration:

/etc/rspamd/local.d/dmarc.conf

```
actions = {
    quarantine = "add_header";
    reject = "reject";
}
```

4. DMARC Feedback Reports versenden

E-Mail-Empfänger tun durch den Versand von DMARC Feedback Reports den Teilnehmern des DMARC-Ökosystems Gutes, indem sie Rückmeldung zur SPF- / DKIM- / DMARC-Konformität ihrer Maildomain geben. Feedback Reports melden möglichen Identitätsmissbrauch sowie mögliche DMARC-Konfigurationsfehler, wovon die Stabilität des gesamten Ökosystems profitiert.

Ziel der DMARC-Policy einer From-Header Domain ist letztlich immer, das Policy-Level auf entweder `quarantine` oder `reject` zu setzen. Nur dann ist ein Schutzniveau gegeben! Das niedrigste Policylevel `none` ist der ungefährlichen Erprobung der SPF- und DKIM-Einstellungen vorbehalten. Es wird für gewöhnlich solange gesetzt bis die From-Header Domain ein „strict alignment“ seiner E-Mails erreicht hat und dann iterativ restriktiver gefasst.

Besonders in dieser Erprobungsphase ist es für Versender wichtig, DMARC Feedback Reports zu erhalten. Die Außensicht auf das Sendeverhalten ihrer Domain macht für sie sichtbar, ob noch Anpassungen an ihrer SPF-Policy erforderlich sind und / oder ob DKIM-Signaturen sauber validieren.

DMARC Feedback Reports werden von der empfangenden Mailplattform an eine oder mehrere E-Mail-Adressen, die in der DMARC-Policy der From-Header Domain vermerkt sind, versendet. Dabei unterscheidet der DMARC-Standard zwei Arten von DMARC Feedback Reports:

"Forensic" / "Failure" Reports

„Forensic Reports“ sind ausführliche Reports, die je festgestelltem Verstoß gegen eine DMARC-Policy versendet werden. Aus Sicht des Datenschutzes kann es dabei passieren, dass ein solcher Report personenbezogene und damit schützenswerte Informationen weitergibt, weswegen Forensic Reports umstritten sind. Diese Art Report ist im sog. `AFRF`-Format verfasst und wird an die mit dem `ruf`-Tag im DMARC DNS TXT-Record angegebene(n) Adresse(n) gesendet.

"Aggregated" Reports

Ein „Aggregated Report“ fasst die Ereignisse, welche eine Senderdomain mit DMARC-Policy betreffen, in einem bestimmten Zeitintervall (empfohlenerweise täglich) in aggregierter Form zusammen. Der Report wird als komprimierte XML-Datei erstellt und an die in dem `rua`-Tag im DMARC DNS TXT-Record angegebene(n) Adresse(n) versendet.

Das Versenden von Aggregated Reports ist ein zweiteiliger Vorgang: Zuerst werden die DMARC-Prüfergebnisse gesammelt und dann, zu einem bestimmten Zeitpunkt, werden daraus Reports generiert und versendet. Die nachfolgenden „Best Practices“ für den Versand basieren auf Empfehlungen und Erfahrungen der KG „E-Mail”.

4.1. Best Practices für den Versand

Wenn Ihr Report-Dienst Benachrichtigungen über möglichen Missbrauch einer Senderdomain versendet, müssen Empfänger ihm vertrauen können und ihr Dienst muss mit den teils personenbezogenen Daten vertrauensvoll umgehen. Wir empfehlen deshalb die nachfolgenden Maßnahmen:

1. Vermeiden Sie, Failure Reports zu versenden. Unter Umstände kann das Versenden von Failure Reports sogar gesetzlich problematisch sein, da in Failure Reports personenbezogene Daten, wie zum Beispiel die Empfänger-E-Mail-Adresse oder der Betreff, genannt werden. Der Report on the compliance of DMARC with the EU GDPR, den die eco Kompetenzgruppe „E-Mail" als Rechtsgutachten in Auftrag gegeben hatte, zeigt im Detail an welchen Stellen Failure Reports DSGVO-konform sind und wo sie diese mit der DSGVO unvereinbar verletzen.
2. Verwenden Sie als Versanddomain für Reports immer eine separate Domain oder Subdomain, damit das Sendeverhalten dieser (Sub)Domain nicht die Reputation Ihrer Hauptdomain negativ beeinflusst (denn Reports enthalten IP-Adressen - oder bei Failure Reports sogar Inhalte - von Spammern/Phishern).
3. Verwenden Sie eine eigene, dedizierte IP-Adresse für den Versand der Reports, denn auch diese IP-Adresse könnte eine schlechte Reputation erhalten.
4. Versehen Sie die Reports mit einer DKIM-Signatur, damit Empfänger zweifelsfrei nachvollziehen können, dass der Report von Ihrem Dienst stammt und dieser so über Zeit eine gute Reputation aufbauen kann.
5. Erstellen Sie eine eigene DMARC-Policy für die Report-(Sub)Domain und fordern Sie für diese (Sub)domain weder „Failure Reports" noch „Aggregated Reports" an, damit zwischen Ihrer Report-Domain und anderen Mailplattformen keine Report-Endlosschleife entstehen kann.
6. Wenn Ihr Maildienst aus mehreren Mailservern besteht, die alle Reports senden sollen, dann setzen Sie nur einen Report-Dienst für alle Server ein. Sammeln Sie die Report-Daten zentral in einer Datenbank und lassen Sie den Report-Dienst die „Aggregated Reports" auf Grundlage aller dort abgelegten Informationen generieren.
7. Lassen Sie Ihren Report-Dienst nur einmal täglich „Aggregated Reports" versenden. Senden Sie nicht exakt um 00:00 Uhr, denn das weltweit entstehende Report-Aufkommen kommt für die empfangende Plattform sonst einer DDoS-Attacke gleich.
8. Rechnen Sie mit Backscatter! Manche Empfänger nehmen Reports an und reagieren auf diese schon wenige Sekunden später mit einer Bounce-Mail oder einer Zustellbenachrichtigung (DSN). Überlegen Sie wo diese vielen Mails landen sollen.
9. Ein merklicher Anteil an im `rua`-tag benannten Empfängeradressen benennt Mailboxen für die das Zielsystem keine Nachrichten annimmt. Die Reports werden in der Mail-Queue ihres MTA verbleiben bis sie bouncen. Reports, die nicht zugestellt werden können, sollten Sie löschen und die Empfängeradressen möglicherweise von der Zustellung ausnehmen.
10. Das sendende eigene Postfach sollte genug Ratelimit haben, damit die Reports alle versendet werden können. Oft werden Tausende Reports innerhalb von Sekunden versendet.

4.2. DMARC Feedback Reports mit OpenDMARC

OpenDMARC sammelt Daten, die es für Reports nutzen wird, in einer Datei. Der Pfad zu dieser

Datei wird mit Hilfe des Parameters `HistoryFile` spezifiziert. Die Datei selbst muss für den User, mit dem OpenDMARC betrieben wird, les- und schreibbar sein.

„Failure Reports“ leiten Sie am besten zu sich selbst um:

/etc/opendmarc.conf

```
CopyFailuresTo        postmaster@mx.example.com
FailureReportsSentBy  postmaster@mx.example.com
FailureReports        yes
FailureReportsOnNone  true
ReportCommand         /usr/sbin/sendmail userpart@sub.domain.tld    ①
FailureReportsBcc     localpart@example.net                         ②
```

① Geben Sie hier nicht `sendmail -t` an

② Senden Sie (dennoch) „Failure Reports“, können Sie mit `FailureReportsBcc` diese immer auch an sich selbst senden und so mitverfolgen was berichtet wird.

OpenDMARC Multi-Host Reporting

Wenn Ihre Mailplattform mehrere Server einsetzt, die jeweils separate OpenDMARC-Instanzen einbinden, sollten Sie deren Daten an zentraler Stelle sammeln und die Reports zentral generieren lassen. Dies erleichtert dem Report-Empfänger die Auswertung.

Erstellen Sie dazu einen cronjob / systemd timer und lassen Sie das Programm `opendmarc-import` die Daten jeder OpenDMARC-Instanz periodisch in eine zentrale Datenbank schreiben:

```
% opendmarc-import --dbhost=hostname --dbname=name --dbpasswd=password
--dbport=port
```

Erstellen Sie dann einen zweiten cronjob / systemd timer und lassen Sie das Programm `opendmarc-reports` zentral reports generieren und versenden. Das Intervall, in dem Reports generiert und versendet werden, steuern Sie mit `interval=secs`. Wenn Sie die Reports nicht lokal, sondern über einen bestimmten anderen Server versenden wollen, geben Sie diesen mit den Parametern `smtp-host` und `smtp-port` an.

4.3. DMARC Feedback Reports mit rspamd

rspamd sammelt Daten, die es für Reports nutzen wird, in einer oder mehreren redis-Datenbanken. Das Generieren und Versenden von Reports müssen Sie explizit aktivieren, indem Sie den Abschnitt `reporting` in der Datei `/etc/rspamd/local.d/dmarc.conf` aktivieren und konfigurieren:

/etc/rspamd/local.d/dmarc.conf

```
servers = "192.2.0.1:6379";                ①
reporting {
    enabled = true;
    email = 'dmarc_reports@sub.example.com';    ②
    domain = 'example.com';                ③
    org_name = 'Example organisation';     ④
    bcc_addrs = ["postmaster@example.com"]; ⑤
    smtp = '127.0.0.1';                    ⑥
    smtp_port = 25;                        ⑦
    from_name = 'example.com DMARC report'; ⑧
    helo = 'example.com';                  ⑨
    msgid_from = 'example.com';            ⑩
}
```

① Hier können Sie bei Bedarf, abweichend von der zenralen redis-Konfiguration für rspamd, festlegen in welche redis-DB DMARC-Report-Daten geschrieben werden sollen.

② Hiermit legen Sie fest mit welcher Envelope Sender Adresse rspamd die Reports versenden wird.

③ Die Empfängerdomain in deren Namen die Reports generiert werden.

④ Geben Sie hier den Namen ihrer Organisation an.

⑤ Wenn Sie die Reports zusätzlich immer auch an andere Adressen versenden wollen, können Sie hier eine Liste von Adressen spezifizieren

⑥ Dieser Parameter legt den SMTP-Server fest, der für den Versand kontaktiert werden soll

⑦ Dieser Parameter legt den TCP-Port des SMTP-Servers fest, der für den Versand kontaktiert werden soll

⑧ Dieser Parameter legt den Anzeigenamen des Absenders fest (Standard ist: "Rspamd")

⑨ HELO im SMTP Dialog

⑩ Message-Id Format

Sobald rspamd DMARC-Prüfergebnisse gesammelt hat, können Sie damit beginnen, diese in Reports zu versenden. Erstellen Sie dazu einen cronjob / systemd timer, welcher periodisch die Daten des Vortags (!) auswertet und als Reports versendet:

```
% 27 3 * * * rspamadm dmarc_report
```

rspamd Multi-Host Reporting

Wenn Ihre Mailplattform mehrere Server einsetzt, die jeweils eigene rspamd-Instanzen einbinden, sollten Sie deren Daten an zentraler Stelle sammeln und die Reports zentral generieren. Dies erleichtert dem Report-Empfänger die Auswertung.

Konfigurieren Sie alle rspamd-Instanzen so, dass diese ihre Report-Daten an die zentrale redis-Datenbank senden und lassen Sie nur auf einem Host den cronjob / systemd timer ausführen, der periodisch Reports generiert und versendet.

[1] Ein fachkundiger E-Mail-Dienstleister wird **vor** dem Newsletter-Versand prüfen, ob SPF, DKIM und DMARC stimmen, auf mögliche Probleme aufmerksam machen und Lösungswege aufzeigen.