Kein Einfallstor für Phishing: SPF & DMARC für Subdomains

Regelung auch für gar nicht vorhandene Subdomains festlegen

Maurice-Jörg Nießen

2025-11-03

Inhaltsverzeichnis

Vorwort zum Thema “E-Mail-Sicherheit” 2
Stärkung der E-Mail-Sicherheit mit bewährten Mitteln 3
Regelung von Subdomains mittels SPF 5
DMARC Policy setzen für Subdomains 8
Überprüfen der eigenen Einträge 11
Technische Verfügbarkeit 12
Syntaktische Überprüfung 14
Überprüfung der Funktion 16
Erklärung 19

Vorwort zum Thema “E-Mail-Sicherheit”

Das immaterielle Gut “*Sicherheit*” wird besonders in Branchen, die damit nicht direkt zu tun haben, gerne mal mittels einer zu kurz gedachten *Kosten-Nutzen-Analyse* beiseitegeschoben.

Es wird auch nicht wirklich einfacher, wenn es sich um ein IT-Thema hinsichtlich der schon totgeglaubten E-Mail handelt und man sich Technologien bedient, die bereits seit Jahrzehnten im Einsatz sind.

Streng genommen profitieren von den im Folgenden thematisierten Technologien eher nur die E-Mail-Empfänger direkt und man selbst bestenfalls indirekt oder erst dann, wenn andere E-Mail-Absender dies ebenfalls umsetzen - idealerweise dann auch noch weitestgehend flächendeckend, gewissenhaft und frei von Fehlern.

Warum sollte ein Entscheider also Zeit, Personal und damit auch finanzielle Mittel für etwas einsetzen, wovon in erster Linie erstmal die “*Anderen*” profitieren?

Mal davon abgesehen, dass die Umsetzung der hier behandelten, technischen Anpassungen durch ein mehrköpfiges, direkt verfügbares, kompetentes IT-Team wohl kaum mehr Zeit in Anspruch nehmen sollte als es dauert, dieses Dokument zu lesen, steigen in den letzten Jahren zunehmen Schadenersatzklagen, die sich zunehmend auf immateriellen Schaden beziehen und aus dem Datenschutz ergeben[1] [2] [3].

Ungeachtet des jeweils dahinterstehenden geschäftlichen Handelns entstehen mehrstellige Forderungen von verschiedensten Gerichten dahingehend begründet, dass bestimmte, mittlerweile als üblich geltende “*technisch-organisatorische Maßnahmen*” (*TOM*) seitens des Beklagten bisher nicht oder nicht wirksam umgesetzt wurden.

Die im Folgenden thematisierten Anpassungen zur Stärkung der E-Mail-Sicherheit finden direkt auf einer Ebene statt, für die sie als Domain- oder sogar Markeninhaber in der Regel alleinig verantwortlich sind. Solange die entsprechenden *TOM*s nicht direkter Bestandteil einer vorangegangenen Beauftragung waren, dürfte sich das Abdrücken der Verantwortung an einen gegebenenfalls vorhandenen externen IT-Dienstleister bestenfalls “*schwierig*” gestalten[4].

[1] “*Non-material damage claims for data protection violations - The current state of play in Germany*” (Noerr, 06.03.2025)
[2] “*DLA Piper Studie: DSGVO-Strafen und Datenschutzverletzungen*” (DLA Piper, 25.03.2024)
[3] “*Groundbreaking Decision in Europe: CJEU Denies a Minimum Threshold for Raising Non-Monetary GDPR Damage Claims*” (Orrick, 05.05.2023)
[4] Anmerkung: *Ich bin weder Jurist, noch ist dies eine juristische Beratung. Es handelt sich lediglich um meine persönliche Meinung, die sich aus zahlreichen, mir zugetragenen Ereignissen und Erfahrungen innerhalb meines beruflichen Umfeldes, gebildet hat.*

Stärkung der E-Mail-Sicherheit mit bewährten Mitteln

Auch wenn das im Jahr 2006[5] entstandene *Sender Policy Framework* (*SPF*)[6] gerne mal als "*veraltet*" oder von "*DKIM & DMARC technisch überholt*" bezeichnet wird, hat es immer noch seine Daseinsberechtigung. *SPF* prüft die Herkunft von E-Mails alleinig anhand der IP-Adressen ihrer Versender (Schutzziel: *Authentizität*). Hingegen prüft *DomainKeys Identified Mail* (*DKIM*)[7], dessen erste offizielle Erwähnungen auf das Jahr 2007[8] zurückzuführen sind, mit digitalen Signaturen und kryptografischen Algorithmen, ob die E-Mails nach bzw. während dem Versenden verändert wurden (Schutzziel: *Integrität*). Mit *Domain-based Message Authentication, Reporting and Conformance* (*DMARC*)[9] werden seit 2015 die Vorzüge von *SPF* und *DKIM* vereint, das obligatorische Vorhandensein eines *DKIM*-Schlüssels sowie ein Berichtswesen (Schutzziel: *Nicht-Abstreitbarkeit*) hinzugefügt. Also statt sich zu ersetzen, ergänzen sich die Technologien gegenseitig.

> Für IT-Verantwortliche sind *Request for Comments* (*RFCs*) eine Mischung aus technischer Richtlinie, Referenzwerk und Planungsinstrument. Wer sie ignoriert, riskiert Inkompatibilitäten, Sicherheitslücken oder ineffiziente Infrastrukturen. Mit ihnen können fundierte Entscheidungen getroffen, Standards einhalten und die IT-Strategie zukunftssicher gestalten werden.

Wie eine Studie der *TU Berlin* aus Februar 2025[10] leider zeigt, werden jedoch häufig die erforderlichen *SPF*-Einträge derart grob getroffen, dass E-Mails auf jeden Fall zugestellt werden und so die gewünschte Schutzwirkung nur noch ausbleiben kann.

So autorisieren 34,7 % der analysierten Domains über 100.000 IP-Adressen in ihrem Namen E-Mails verschicken zu dürfen. Trotz des hohen Alters von *SPF* und der daraus resultierenden hohen Anzahl von *Best-Practice*-Ansätzen, haben sich (IT-) Verantwortliche dann doch für eher laxe Konfigurationen entschieden. Im Rahmen meiner Tätigkeiten habe ich eine Domain einer namhaften Unternehmung entdeckt, bei der die Anzahl der definierten IP-Adressen, die Anzahl der in dem Land ihrer Zentrale lebenden Einwohner um den Faktor größer 1000 übersteigt. Und nein, es war keine in Luxemburg ansässige Zentrale.

[5] `RFC4408` - "*Sender Policy Framework (SPF) for Authorizing Use of Domains in E-Mail, Version 1*" (2006)
[6] `RFC7208` - "*Sender Policy Framework (SPF) for Authorizing Use of Domains in E-Mail, Version 1*" (2014)
[7] `RFC6376` - "*DomainKeys Identified Mail (DKIM) Signatures*" (2011)
[8] `RFC4870` - "*Domain-Based Email Authentication Using Public Keys Advertised in the DNS (DomainKeys)*" (2007)
[9] `RFC7489` - "*Domain-based Message Authentication, Reporting, and Conformance (DMARC)*" (2015)
[10] "*Lazy Gatekeepers: Large-Scale Study on SPF Configuration in the Wild*" (TU Berlin, 12.02.2025)

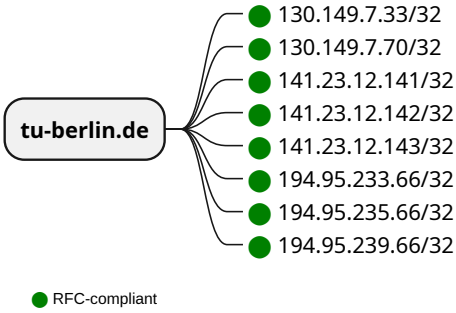


Abbildung 1: SPF-Eintrag der TU Berlin

Mit ihrem eigenen *SPF*-Eintrag zeigt die *TU Berlin* wie sich eine wohl immense E-Mail-Flut auch mit nur 8 autorisierten E-Mail-Servern bewältigen lässt.

Eine von der Firma *Validity* im Januar 2025 erstellte Studie[11] weisst für *DMARC* ebenfalls alarmierende Zustände aus. Danach haben weltweit über 84% erst gar keinen *DMARC*-Eintrag gesetzt und bei über der Hälfte der verbleibenden 16% sind die hinterlegten Einträge syntaktisch falsch. Gerade einmal 7,64% der weltweiten Domains haben einen gültigen *DMARC*-Eintrag, wobei hiervon 68% jegliche Schutzwirkung deaktiviert (**p=none**) haben.

> Auch wenn Deutschland im europäischen Vergleich eher das Schlusslicht bildet, so haben der Studie zufolge immerhin über 29% der **.de**-Domains einen gültigen *DMARC*-Eintrag. Bei mehr als 10% der **.de**-Domains ist der hinterlegte Eintrag aber auch fehlerhaft.

Viele Einträge scheinen gemacht worden zu sein, damit man überhaupt einen Eintrag hat. Mit dem Potenzial von *SPF* oder *DMARC* scheint man sich aber eher wenig auseinander gesetzt zu haben oder gar bewusst ignoriert zu haben, damit man die bisherige Zustellung von E-Mails - ob nun gewollt oder nicht - in keinerlei Weise beeinflusst. Mit der Beibehaltung des Status quo lässt sich schwer die E-Mail-Sicherheit verbessern.

Um dieser "*verdrehten*" Intention bei der Gestaltung der Einträge entgegenwirken, beginnen wir doch mal mit ihrem Status Quo:

Als verantwortlicher Domaininhaber haben sie bereits entschieden, ob und von welcher Domain E-Mail-Kommunikation stattfinden soll. Idealerweise haben sie diese Entscheidung auch bereits in einem *SPF*-Eintrag hinterlegt, indem sie die IP-Adressen[12] der Server gelistet haben, welche im Namen

[11] "*DMARC Adoption: A Deep Dive into the Current State of Email Authentication*" (Validity, 29.01.2025)
[12] Anmerkung: *In der `RFC7208` wird nur von Hosts und **nicht** von Netzwerken gesprochen, in denen sich möglicherweise legitime E-Mail-Server befinden könnten.*

der jeweiligen (Sub-)Domain E-Mails versenden dürfen. Aber haben sie auch hinterlegt, von welchen Subdomains keinerlei E-Mails zu erwarten sind?

Obwohl die Antwort "*Na ja, von allen Anderen!*" berechtigterweise auf der Hand liegen mag, so werde ich noch darauf eingehen, dass *SPF* keine Überprüfung übergeordneter Domains vornimmt und die in *DMARC* vorhandene Option, in diesem Fall eine abweichende Policy anzuwenden, in ihrer Konfiguration möglicherweise fehlt.

> Die E-Mail-Kommunikation über beliebige Subdomains ihrer Domain ist aller Wahrscheinlichkeit nach **ungeregelt** oder sogar aufgrund von Standardeinstellungen **erlaubt**.

Im Folgenden zeige ich, wie sich mit kleinen Anpassungen an der *SPF*- und *DMARC*-Konfiguration eine Regelung auf Subdomains ausweiten lässt.

Regelung von Subdomains mittels SPF

Für Domains oder Subdomains von denen keinerlei E-Mail-Versand ausgeht, sollte ein Eintrag für das *Sender Policy Framework* [13] in folgender Form hinterlegt sein:

```
hohlhupe.de IN TXT "v=spf1 -all"
```

Durch **`-all`**, auch als "*hard failure*"[14] bezeichnete Einstellung, wird mittels *SPF* den Empfängern empfohlen E-Mails der entsprechenden Domain abzulehnen. Die Absender erhalten dann eine Benachrichtigung per E-Mail von dem versendenden Server ähnlich der folgenden.

```
Message delivery failed.
Recipient Name: test@hohlhupe.de
Recipient Domain: hohlhupe.de
Error Message: 550 5.7.1 Relaying denied

SMTP Error Code 550:
Relaying denied
```

[13] `RFC7208` - "*Sender Policy Framework (SPF) for Authorizing Use of Domains in Email, Version 1*" (2014)

[14] Anmerkung: *Die Einstellung für ein "soft failure" unterscheidet sich nur durch eine Tilde ~ statt dem Minus −, was gerne mal übersehen wird.*

Je nach getroffener Einstellung wird die Zustellung mehrfach über eine gewisse Zeit lang versucht, so dass man nicht im unmittelbaren Zusammenhang mit dem Versenden der E-Mail eine Benachrichtigung erhält. Es können gerne dazwischen auch mal 6 bis 48 Stunden vergehen.

Doch reicht die oben getroffene *SPF*-Regel um die Verwendung der Domain **`hohlhupe.de`** als E-Mail-Absender gänzlich zu untersagen? Wie sieht beispielsweise mit einer Subdomain wie **`vip.hohlhupe.de`** aus? Gerne werden **`hr`**, **`external`**, **`internal`** oder ähnliche Subdomains zur Darstellung einer eher organisatorischen Trennung verwendet.

Die derzeit noch gültige Spezifikation für *SPF* sieht ein **none** als sofortiges Ergebnis einer Überprüfung vor [15], wenn die zu prüfende Domain syntaktisch fehlerhaft oder gar nicht erst vorhanden ist (**`NXDOMAIN`**)[16].

Somit ist das weitere Vorgehen mit dem eines fehlenden *SPF*-Eintrages gleichzusetzen. Eine Abfrage einer übergeordneten oder gar der Hauptdomain ist durch *SPF* nicht vorgesehen. Einzig das Vorhandensein eines entsprechenden *SPF*-Eintrages für die jeweilige Subdomain ist für eine Validierung einer damit versehenen Absender-Adresse entscheidend.

Es empfiehlt sich daher für jegliche Subdomain auch eine *SPF*-Regel zu erstellen - insbesondere auch für Subdomains, die weder vorhanden noch vorgesehen sind. Was im ersten Moment nach einer Sisyphusarbeit oder gänzlich unmöglich klingt, lässt sich mit Hilfe des Stern **`*`** (aka *"wildcard"*) jedoch leicht umsetzen:

```
; SPF for root domain
@    IN  TXT "v=spf1 mx -all"

; SPF for hr subdomain
hr   IN  TXT "v=spf1 include:hohlhupe.de -all"

; SPF wildcard for (all other) one-level subdomains
*    IN  TXT "v=spf1 -all"

; SPF wildcard for two-level subdomains
*.*  IN  TXT "v=spf1 -all"

; SPF wildcard for three-level subdomains
*.*.*  IN TXT "v=spf1 -all"
```

[15] `RFC7208` - Kapitel 4.3 - "*Initial Processing*"
[16] `RFC8020` - "*NXDOMAIN: There Really Is Nothing Underneath*"

Eine weniger restriktive Einstellung als **`-all`** zu treffen, ist schlichtweg kontraproduktiv. E-Mails die von Domains oder Subdomains stammen, die von ihren Inhabern gar nicht für den E-Mail-Versand vorgesehen sind, können auch keinerlei relevante Inhalte enthalten.

Abweichend von dem vorangegangenen Beispiel, wird über das Tag **`mx`** nun der E-Mail-Versand über die im *DNS* hinterlegten "*mail exchanger*" als für die Hauptdomain üblich deklariert. Diese Regelung wird im nächsten Eintrag mittels **`include:hohlhupe.de`** exemplarisch für die Subdomain **`hr`** übernommen. Sollte hier dieselbe E-Mail-Infrastruktur wie für die Hauptdomain zuständig sein, müssen Anpassungen am *SPF*-Eintrag nur dort vorgenommen werden. Die genaue inhaltliche Gestaltung der *SPF*-Einträge selbst, ist jedoch nicht Gegenstand dieses Dokumentes, daher sind die hier gezeigten Einträge als Beispiele zu verstehen.

Ab dem dritten Eintrag nimmt der besagte Stern `*` die Rolle eines Platzhalters für alle denkbaren, übrigen Subdomains ein. So wird jeder ausgedachten, nicht vorgesehen Absenderadresse dieser Subdomain-Ebene eine ablehnende *SPF*-Regel zugewiesen. Aufgrund der Spezifikation zur Nutzung des `*` innerhalb von *DNS*[17] gilt diese Regelung jedoch nur für die jeweilige, eine Ebene.

Für weitere Subdomain-Ebenen müssen somit auch weitere *SPF*-Einträge erstellt werden. Wie bei Domains üblich, werden die Ebenen durch einen Punkt `.` getrennt, so dass `*.*` stellvertretend für beispielsweise **`irgendwas.ausgedachtes.hohlhupe.de`** steht.

Dieses von einigen Administratoren als "*Hack*" bezeichnete und damit als nicht korrekt empfundenes Vorgehen, wird im Kapitel "*Wildcard Record*" des "*Sender Policy Framework*"[18] seit 2014 beschrieben. Wahrscheinlich haben aber die einleitenden Sätze in diesem Kapitel jahrelang eine sinnvolle Anwendung verhindert. Diese sagen nämlich, dass bei dem "*nicht empfohlen Einsatz*" eine "*besondere Vorsicht geboten*" sei, da "*jede Deklaration für jeden Host, der irgendeinen Resource Record (RR) hat, wiederholt werden muss*".

Für gewollte Domains (**`hohlhupe.de`**) und Subdomains (**`hr.hohlhupe.de`**) werden in dem oben stehenden Beispiel keinerlei *Wildcard Records* verwendet. Ihre *RR* werden explizit gesetzt. Angewandt werden sie ausschließlich nur für ungewollte Subdomains, die gar nicht in der *DNS-Zone* definiert sind und es somit für sie auch keine weiteren *RR* gibt, die wiederholt werden müssten. Statt mit dem `*` mehrere *RR* zusammenzufassen (was möglicherweise die ursprüngliche Intention der *Wildcard Records* war), ist ihre Verwendung hier im Sinne von "*für Alles andere*" zu verstehen.

Explizite *Resource Records* (*RR*) haben im *DNS* stets Vorrang vor *wildcard records*. Die Reihenfolge innerhalb der *Zone*-Datei ist diesbezüglich irrelevant.

[17] `RFC4592` - "*The Role of Wildcards in the Domain Name System*"
[18] `RFC7208` - Kapitel 3.5 - "*Wildcard Records*"

Für wie viele Ebenen man einen entsprechenden *Wildcard Record* definiert, sei jedem Domaininhaber selber überlassen. Der Aufwand pro Ebene ist allerdings mehr als überschaubar. Spätestens ab der 8. Ebene sollten E-Mail-Empfänger die Legitimität der Absenderadresse anzweifeln und mögliche Angreifer die Lust verlieren. Hinsichtlich der besseren Lesbarkeit sollten in der *Zone*-Datei die *Wildcard Records* allerdings hinter den explizit gesetzten *Resource Records* der jeweiligen Subdomain-Ebene stehen.

> Wer seine Domain über einen *Managed DNS Provider* (wie beispielsweise *Cloudflare*, *Route 53*, *Google Cloud DNS* oder *DigitalOcean*) administriert, kommt mit einem einzelnen Eintrag (**`*.hohlhupe.de`**) aus. Hier scheint man die Vorgabe der RFC4592 zwar etwas anders zu interpretieren, doch dies ganz im Sinne einer einfacheren Realisierung.

Mit wenigen zusätzlichen Einträgen hat man so die (hoffentlich schon) bestehende *SPF*-Konfiguration derart erweitert, dass auch für sämtliche[19] ungewollten Subdomains der eigenen Domain eine *SPF*-Regelung bzw. -Empfehlung ausgesprochen wird, diese als Absender von E-Mails zu ignorieren.

Bei Bedarf ließe sich sicherlich nachweisen, dass diese Empfehlung bereits zum Zeitpunkt der Verarbeitung durch den empfangenden E-Mail-Server abrufbar gewesen ist. Inwieweit dieser Empfehlung tatsächlich gefolgt wurde, liegt dann nicht mehr in der Verantwortung des Domaininhabers der mutmaßlichen Absenderadresse.

DMARC Policy setzen für Subdomains

Wie bereits anfangs erwähnt, baut *DMARC* auf *SPF* sowie *DKIM* auf und ermöglicht, über Policies festzulegen zu können, wie E-Mail-Server mit E-Mails der entsprechenden Domain umgehen sollen, bei denen die *DKIM*- und/oder *SPF*-Prüfung fehlschlägt.

```
_dmarc.hohlhupe.de.  IN  TXT  "v=DMARC1; p=quarantine;
↪  rua=mailto:postmaster@hohlhupe.de"
```

Dieser exemplarische *DMARC*-Eintrag ist als Mindesteinstellung zu verstehen. Im Internet verfügbare Beispiele mit einer deaktivierten Policy (**p=none**) und ohne einen hinterlegten Empfänger für Berichte, bilden keinerlei Mehrwert im Rahmen von *DMARC*. Je nach Umsetzung der Policy **`quarantine`** auf Empfängerseite, landen E-Mails mit fehlgeschlagenen Überprüfungen ohne jegliche weitere, für den Empfänger sichtbare Kennzeichnung direkt neben einwandfrei überprüften E-Mails im selben Eingangspostfach. Hinsichtlich eines sicheren E-Mail-Versands ist daher die Policy **`reject`** zu bevorzugen. Die über ihre Infrastruktur versendeten E-Mails werden ja über alle erforderlichen

[19] Anmerkung: *zumindest bis zu einer gewissen Anzahl von Subdomain-Ebenen*

Eigenschaften verfügen, sodass sich diese Policy nur auf E-Mails bezieht, die über andere Wege zum Empfänger gelangt ist.

> Jeglicher Raum, den sie sich für eigene Fehler lassen, kommt auch möglichen Angreifern zu Gute. Reduzieren sie eigene Fehler daher lieber durch standardisierte Tests nach Veränderungen und ein anschließendes, kontinuierliches Monitoring während des produktiven Einsatzes.

Abhängig von der Intensität der E-Mail-Nutzung empfiehlt es sich für die eingehenden Berichte, ein separates, exklusives E-Mail-Konto anzulegen, welches dann auch als Quelle für automatisierte Analysen dienen kann.

> Mittels eines zusätzlichen **`fo=1`** durch ein Semikolon von anderen Tags getrennt, lässt sich die Berichtsfunktion so einstellen, dass ein Bericht angefordert wird, sobald bereits eine Überprüfung durch *SPF* **oder** *DKIM* fehlschlägt. Standardmäßig geschieht dies erst, wenn **beide** Überprüfungen fehlschlagen.

Da bei *DMARC* das Präfix **`_dmarc`** am Anfang des *DNS-Name* erwartet wird und "*wildcards*" nur am Anfang verwendet werden können, lässt sich der oben für *SPF* beschriebene Ansatz leider nicht auch auf *DMARC*-Einträge anwenden.

> Grundsätzlich können durch die Verwendung eines Präfixes *DNS*-Abfragen gezielter erfolgen und so der Bereich der TXT-Einträge der Hauptdomain entlastet werden. *DNS*-Antworten mit mehr als **`4096 Bytes`** werden nicht unterstützt und durch die zufällige Anordnung einzelner, passender Antworten kann sich ein sporadisches Fehlerbild und eine sehr zeitaufwendige Fehlersuche ergeben.

Für den in diesem Dokument thematisierten Umgang mit Subdomains, existiert bereits seit der ersten Spezifikation von *DMARC* die Möglichkeit, eine "*separate policy*" (**`sp=`**) für Subdomains festzulegen.

Anders als bei *SPF*, sieht die *DMARC*-Spezifikation vor, den Eintrag der Hauptdomain zu verwenden, wenn für die Subdomain selbst kein eigener Eintrag existiert. In diesem Fall würde die "*separate policy*" greifen. Ist diese nicht gesetzt, wird die Policy der Hauptdomain (**`p=`**) verwendet.

> Das heißt, ohne eine gesetzte "*separate policy*" gilt für alle möglichen Subdomains, für die es keinen gesonderten *DMARC*-Eintrag gibt, die Policy der Hauptdomain. Mit **`p=none`** oder **`p=quarantine`** wäre somit die Annahme von E-Mails irgendwelcher Subdomains grundsätzlich empfohlen.

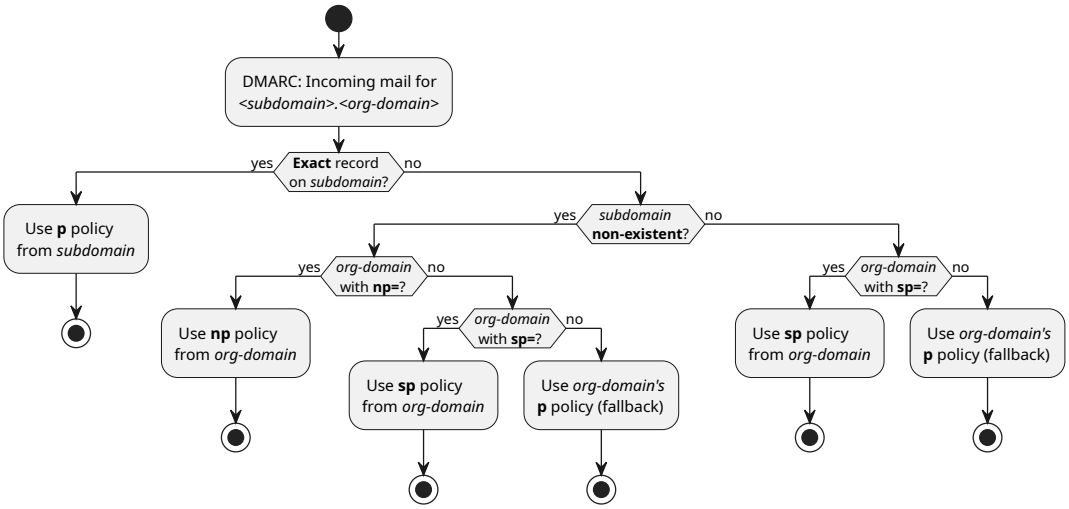


Abbildung 2: DMARC Umgang mit Subdomains

```
_dmarc.hohlhupe.de.  IN  TXT  "v=DMARC1; p=quarantine; sp=reject;
↪  rua=mailto:postmaster@hohlhupe.de"
```

Mit diesem *DMARC*-Eintrag wird den Empfängern empfohlen, alle problembehafteten E-Mails der Hauptdomain (hier: **`@hohlhupe.de`**) in Quarantäne zu stellen und sämtliche E-Mails von Subdomains dieser Domain direkt abzuweisen. Möchte man analog zu dem Beispiel im *SPF*-Kapitel E-Mails für **`@hr.hohlhupe.de`** zulassen, ist ein separater *DMARC*-Eintrag wie folgender erforderlich:

```
_dmarc.hr.hohlhupe.de.  IN  TXT  "v=DMARC1; p=quarantine; sp=reject;
↪  rua=mailto:postmaster@hohlhupe.de"
```

Ein **`sp=reject`** auf der Ebene der Subdomain **`hr`** sorgt dafür, dass die hier gesetzte Policy-Einstellung **`quarantine`** nicht für irgendwelche Subdomains von **`hr.hohlhupe.de`** angewendet wird.

> Aufgrund der *Fallback*-Funktion könnte man mit einem **`p=reject`** rein technisch auf ein **`sp=reject`** verzichten. Um möglichst wenig Spielraum für technische oder juristische Interpretationen zu lassen, empfehle ich die Entscheidung hinsichtlich des Umgangs mit Subdomains explizit im *DMARC*-Eintrag zu hinterlegen.

Im Rahmen der Erweiterung und Weiterentwicklung von *DMARC* gibt es seit Juli 2021 die Möglichkeit, mittels "*non-existent policy*" (**np=**)[20] für nicht existierende Subdomains gesondert eine Einstellung

[20] `RFC9091` - "*Experimental Domain-Based Message Authentication, Reporting, and Conformance (DMARC) Extension for Public Suffix Domains*"

festzulegen. Wie bereits bei *SPF* schon weiter oben erwähnt, gibt es auch hier nur eine sinnvolle Einstellung, nämlich **`np=reject`**. E-Mails von vom Domaininhaber nicht vorgesehener Subdomains können keinerlei Existenzberechtigung haben und sollten daher auch nicht weiter verarbeitet werden.

```
_dmarc.hohlhupe.de.  IN  TXT  "v=DMARC1; p=reject; sp=reject; np=reject;
↪  fo=1; rua=mailto:postmaster@hohlhupe.de"
```

Die "*non-existent policy*" ist Bestandteil der sich noch in Bearbeitung befindlichen Spezifikation `DMARCbis` [21]. Bis diese jedoch flächendeckend umgesetzt wurde, dürfte noch etwas Zeit verstreichen. Aus diesem Grund empfiehlt es sich, bis auf Weiteres sämtliche Subdomains mittels **`sp=reject`** abzuweisen und solche, bei denen ein E-Mail-Verkehr gewollt ist, mit einem eigenen *DMARC*-Eintrag explizit zu erlauben.

```
_dmarc.hr.hohlhupe.de.  IN  TXT  "v=DMARC1; p=quarantine; sp=reject;
↪  rua=mailto:postmaster@hohlhupe.de"
```

Anders als die "*separate policy*" macht die Definition einer "*non- existent policy*" auf der Ebene einer Subdomain praktisch keinen Sinn, weil dieser Eintrag primär für alle zufällige, nicht existierende Subdomains gilt. Der obenstehende *DMARC*-Eintrag für die Subdomain **`hr`** ist daher unverändert zu dem vorangegangenen Beispiel, soll aber zeigen, dass hier bei Bedarf auch eine abweichende Einstellung gesetzt werden kann.

Überprüfen der eigenen Einträge

Unabhängig davon, ob man eine der obenstehenden Anpassungen vorgenommen hat, ein *Best-Practice*-Beispiel übernommen hat oder bisher nur im Glauben war, die Themen *SPF* sowie *DMARC* bereits erfolgreich abgehandelt zu haben, sollten die eigenen Einträge überprüft werden. Neben der Erfüllung der gewünschten Funktion sollte auch eine Überprüfung der Syntax der Einträge sowie ihre technische Verfügbarkeit überprüft werden. Syntaktisch korrekte Einträge, die jedoch für die Empfänger nicht oder noch nicht über ihre jeweiligen *DNS*-Server abrufbar sind, können ihre Wirkung nicht entfalten. Aus diesem Grund sollte auch die weltweite Verbreitung innerhalb der *DNS*-Infrastruktur überprüft werden.

[21] `DMARCbis` - *Version 41* (04.04.2025)

Technische Verfügbarkeit

Bevor man jedoch mit der Überprüfung der gewünschten Funktion beginnt, indem man E-Mails von Absenderadressen mit verschiedensten Subdomains versendet, sollte sichergestellt werden, ob die erforderlichen Einträge einerseits von den für die Domain zuständigen *DNS*-Servern bereitgestellt werden und andererseits schon ausreichend über die *DNS*-Infrastruktur propagiert wurden. Ansonsten läuft man Gefahr, dass sich Tests noch vorangegangener Einstellungen bedienen und mögliche Probleme mit neuen Einstellungen übersehen werden.

> In der Praxis kann es maximal bis zu **`3x`** (lokaler *Caching-Nameserver*, *Rekursiver DNS-Server* des Providers, *sekundärer Nameserver*) der **gesetzten TTL-Zeit** ("*time to live*", Gültigkeitsdauer) dauern, bis die Veränderungen auch bei den *DNS*-Servern potentieller E-Mail-Empfänger ankommen. Es gilt die `TTL` des **vorangegangenen Eintrags**, weshalb es sich grundsätzlich empfiehlt im Vorfeld inhaltlicher Veränderungen die Zeit deutlich herabzusetzen und erst nach erfolgreichen Tests wieder hochzusetzen.

Auf einer handelsüblichen *BASH*-Kommandozeile lässt sich die technische Bereitstellung von *DNS*-Einträgen recht einfach überprüfen, indem man ihren entsprechenden Inhalt abfragt.

Für den *SPF*-Eintrag der Hauptdomain (hier: `hohlhupe.de`)[22] erfolgt dies beispielsweise durch das folgende Kommando:

```
dig hohlhupe.de TXT @46.38.225.225 +short
```

Je nachdem welche sonstigen Einstellungen für die Domain über den `TXT`-Eintrag gesetzt wurden, werden mehrere Zeilen als Ergebnis der Abfrage geliefert. Allerdings darf *nur eine* Zeile mit **`v=spf1`** beginnen und idealerweise mit dem indem vorangegangenen Kapitel gesetztem, weiteren Inhalt **`mx -all`** enden.

> Durch das **`@`** und der dahinter folgenden Angabe eines *DNS*-Servers lässt sich das jeweilige (zwischengespeicherte) Ergebnis zielgerichtet erfragen. Auf diese Weise kann man schon vor der vollständigen Synchronisation der Änderung gezielte Überprüfungen vornehmen. Eine Abfrage ohne diese Option würde voraussetzen, dass bereits alle dazwischengeschalteten *DNS*-Server entsprechend aktualisiert haben.

Da die Änderungen auf dem *Primary DNS*-Server der Domain **`hohlhupe.de`** (hier: `46.38.225.225`) vorgenommen werden, sollte dieser auch als erstes eine entsprechende Antwort liefern können.

[22] Anmerkung: *Die hier verwendete Beispieldomain `hohlhupe.de` existiert wirklich und kann somit für entsprechende DNS-Abfragen testweise verwendet werden.*

Das angehängte **`+short`** unterdrückt zusätzliche Informationen und erleichtert für Ungeübte die Lesbarkeit der Antworten.

Dies wird nun für jede explizit gesetzte Subdomain wiederholt:

```
dig hr.hohlhupe.de TXT @46.38.225.225 +short
```

Analog zu dem hier verwendetem Beispiel sollte die Antwort **`v=spf1 include:hohlhupe.de -all`** lauten.

Um die *Wildcard Records* zu testen, fragt man eine ausgedachte, nicht existierenden Subdomain der jeweiligen Subdomain-Ebene, wie beispielsweise **`un.schlagbar.naive.hohlhupe.de`** für die 3. Subdomain-Ebene, ab:

```
dig un.schlagbar.naive.hohlhupe.de TXT @46.38.225.225 +short
```

Hier sollte das Ergebnis stets **`v=spf1 -all`** lauten.

Die *DMARC*-Einträge lassen sich analog dazu prüfen, wobei das Präfix **`_dmarc`** vor der jeweils gewollten, explizit gesetzten Subdomain nicht vergessen werden darf:

```
dig _dmarc.hohlhupe.de TXT @46.38.225.225 +short
```

```
dig _dmarc.hr.hohlhupe.de TXT @46.38.225.225 +short
```

Im Rahmen des hier verwendeten Beispiels sollten sich die Antworten nur hinsichtlich **`p=reject`** und dem exemplarisch gesetztem **`p=quarantine`** für **`hr`** unterscheiden.

Ungewollte, beliebige Subdomains können aber nicht analog zu dem obenstehenden *SPF*-Abfragen getestet werden, da hier ja auf die Logik von *DMARC* zurückgegriffen wird, welche von dem jeweiligen empfangenden E-Mail-Server abgehandelt wird. Die Überprüfung hinsichtlich des Vorhandenseins entsprechender Regelungen endet daher mit der Nennung von **`sp=reject`** und dem in der Hauptdomain zusätzlichen **`np=reject`**[23].

Für eine vollständige Überprüfung und/oder in einem Fehlerfall, richtet man die aufgeführten *DNS*-Abfragen ebenfalls an die zuständigen *Secondary DNS*-Server der Domain. So lässt sich sicherzustellen, dass keine fehlerhaften oder veralteten Einträge an andere *DNS*-Server als quasi noch

[23] Anmerkung: *Aufgrund der noch ausstehenden Verbreitung von DMARCbis ist hiervon allerdings noch eher keinerlei Mehrwert zu erwarten.*

aktuelle Information ausgegeben wird. Im Zweifel lieber zu viele Abfragen überprüfen, als den einen Eintrag, der später Probleme bereitet, zu übersehen.

> Die für eine Domain zuständigen *DNS*-Server lassen sich mittels beispielsweise `dig hohlhupe.de NS +short` erfragen. Neben der *IP-Adresse* lässt sich auch der *Hostname* des jeweiligen *DNS* -Servers hinter das setzen - wobei die *IP-Adresse* zu bevorzugen ist.

Optional oder alternativ zur Kommandozeile kann die weltweite Verteilung der Änderungen auch mit einem *DNS Propagation Checker* (wie *whatismydns.net*[24]) überprüft werden.

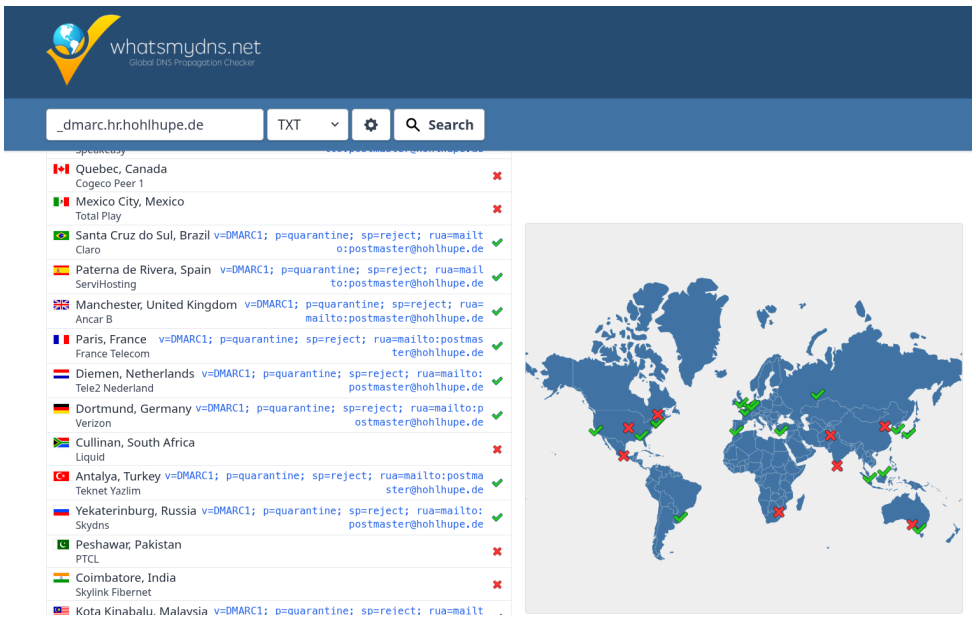


Abbildung 3: DNS Propagation Check

Sollte hier kein *DNS*-Server noch die veralten Inhalte wiedergeben, so kann man die Veränderung als "*erfolgreich propagiert*" betrachten. Dass ein paar *DNS*-Server temporär nicht erreichbar sind und kein Ergebnis liefern, ist durchaus üblich.

Syntaktische Überprüfung

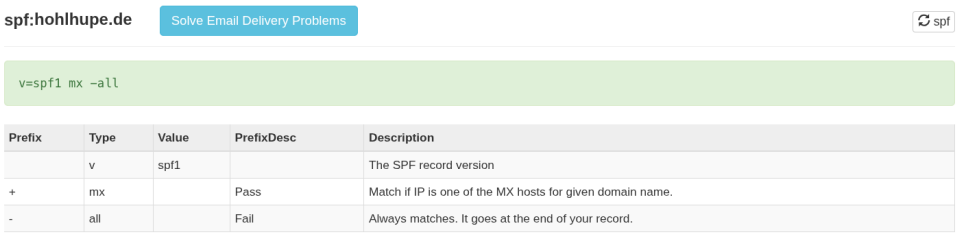
Die Überprüfung der Syntax eines *zu machenden* Eintrags sollte idealerweise *vor* einer Publikation erfolgen, während die Überprüfung *gemachter* Einträge gegebenenfalls erst *nach* der oben

[24] `whatismydns.net`

erwähnten, vollständigen *DNS*-Synchronisation möglich ist. Hierfür stehen zahlreiche Online-Tools (wie *SPF-Record*[25], *MXToolbox*[26], *EasyDMARC*[27], *Valimail*[28], *DMARCian*[29] *oder* PowerDMARC*[30]) zur Verfügung.

Gute Tools beziehen ihre Informationen direkt von den für die Domain zuständigen *DNS*-Servern und geben die zur Auswertung herangezogenen Einträge auch an. Mit etwas Übung lassen sich somit erforderliche Korrekturen noch vornehmen, bevor die zu korrigierende Version weitflächig verbreitet wurde.

Einige Tools erlauben die manuelle Eingabe eines Eintrages, wodurch eine Überprüfung ermöglicht wird, ohne den produktiven Betrieb zu beeinflussen.



Prefix	Type	Value	PrefixDesc	Description
	v	spf1		The SPF record version
+	mx		Pass	Match if IP is one of the MX hosts for given domain name.
-	all		Fail	Always matches. It goes at the end of your record.

Abbildung 4: SPF-Check mit MXToolbox

Die in der Abbildung gezeigte Überprüfung des *SPF*-Eintrags der Hauptdomain[31] signalisiert durch die grüne Hinterlegung des Eintrags, dass er insgesamt als syntaktisch korrekt angesehen wird. Die darunter stehenden Erklärungen hinsichtlich der einzelnen Optionen dienen nicht nur dem besseren Verständnis, sondern geben auch Auskunft darüber, ob die jeweilige Option von dem Tool erkannt und somit richtig interpretiert werden konnte.

[25] `SPF-Record.de`
[26] `MXToolbox.com`
[27] `EasyDMARC.com`
[28] `Valimail.com`
[29] `dmarcian.com`
[30] `PowerDMARC.com`
[31] `mxtoolbox.com` - "*spf:hohlhupe.de*"

dmarc:hohlhupe.de Find Problems Solve Email Delivery Problems dmarc

```
v=DMARC1; p=reject; sp=reject; np=reject; fo=1; rua=mailto:postmaster@hohlhupe.de
```

Tag	TagValue	Name	Description
v	DMARC1	Version	Identifies the record retrieved as a DMARC record. It must be the first tag in the list.
p	reject	Policy	Policy to apply to email that fails the DMARC test. Valid values can be 'none', 'quarantine', or 'reject'.
sp	reject	Sub-domain Policy	Requested Mail Receiver policy for all subdomains. Valid values can be 'none', 'quarantine', or 'reject'.
np	reject		Unknown
fo	1	Forensic Reporting	Provides requested options for generation of failure reports. Valid values are any combination of characters '01ds' seperated by ':'.
rua	mailto:postmaster@hohlhupe.de	Receivers	Addresses to which aggregate feedback is to be sent. Comma separated plain-text list of DMARC URIs.

	Test	Result
✔	DMARC Record Published	DMARC Record found
✔	DMARC Syntax Check	The record is valid
✔	DMARC Multiple Records	Multiple DMARC records corrected to a single record.
✔	DMARC Policy Not Enabled	DMARC Quarantine/Reject policy enabled
✔	DMARC External Validation	All external domains in your DMARC record are giving permission to send them DMARC reports.

Abbildung 5: DMARC-Check mit MXToolbox

Hier zeigt die Abbildung die Überprüfung des *DMARC*-Eintrags der Hauptdomain[32] nach demselben Prinzip.

Da im Kapitel *"General Record Format"*[33] der offiziellen *DMARC*-Spezifikation vorgegeben wird, dass alle unbekannten Tags ignoriert werden müssen. Es schadet daher nicht, ein **`np=reject`** bereits schon heute zu setzen. Die Abbildung zeigt, dass das **`np`** als "*unknown*" markiert wurde, während der gesamte Eintrag weiterhin grün hinterlegt ist.

Man im Hinterkopf haben, dass das ein oder andere Tool vielleicht noch nicht oder noch nicht umfänglich *DMARCBis* unterstützt und daher neue Tags gegebenenfalls fälschlicherweise auch als *"fehlerhaft"* markiert.

> Das könnte künftig auch mit veralteten Tag **`pct=100`** passieren, welcher durch **`t=n`** ersetzt wird. Mit der Bezeichnung "*test mode*", sollte es dann unmissverständlich sein, dass es sich hierbei um kein Tag für den produktiven Betrieb handelt.

Überprüfung der Funktion

Nachdem durch die vorangegangenen Überprüfungen sichergestellt wurde, dass möglichen Empfängern die syntaktisch korrekten Einträge weltweit technisch zur Verfügung stehen, bleibt noch zu prüfen, ob diese auch die erwünschte Funktion erreichen.

[32] `mxtoolbox.com` -"*dmarc:hohlhupe.de*"
[33] `RFC7489` - Kapitel 6.3 - "*General Record Format*"

Dabei macht es wenig Sinn, E-Mails **an** die eigene, gerade angepasste Domain zu senden, da man mit *SPF* oder "*DMARC*" keine Veränderung an der Logik des eingehenden E-Mail-Servers der Domain vorgenommen hat. Je nach Konfiguration, beinhaltet die domain-interne E-Mail-Zustellung, also das Senden einer E-Mail von einem Account zu einem anderen derselben Domain, gar keine *SPF*- oder *DMARC*-Prüfung. Um sicherzugehen, sollte man daher E-Mails **von** der angepassten Domain an eine andere Domain versenden, von der man sicher weiß, dass diese auch *Aggregate DMARC*-Reports versendet. Hierzu verwende ich gerne mal einen *Google*-, *Apple*- oder auch *Yahoo*-Mail-Account.

Zusätzlich lasse ich mir von dem Online-Tool `dkimvalidator`[34] eine E-Mail-Adresse generieren und schicke an diese eine Test-E-Mail. Nach kurzer Zeit kann ich mir nicht nur diese inklusive aller Kopfzeilen im "*Original*" ansehen, sondern erhalte auch noch eine exemplarische Auswertung von *SPF*, *DKIM* und *SpamAssassin*.

> Einige E-Mail-Server verwenden zur *Spam*-Abwehr das sogenannte *Greylisting*-Verfahren, bei dem E-Mails zunächst **temporär abgewiesen** werden und erst bei einer **erneuten Zustellung angenommen** werden. Es kann also durchaus normal sein, wenn die Zustellung mal einige Minuten dauert anstatt "*instant*" zu erfolgen.

Grundsätzlich sollte von jeder gewünschten Subdomain eine Test-E-Mail versendet werden, um die Nutzung als Absenderadresse nach einer Anpassung gewährleisten zu können. Das Auftauchen der E-Mails im jeweiligen Posteingang gilt als *bestanden*.

Für nicht legitimierte Absender-Domains sollte sich innerhalb der nächsten 48 Stunden im festgelegtem Posteingang (**`rua=`**) der *Aggregate DMARC*-Report ein entsprechender Eintrag befinden. Es spielt hierbei keine Rolle, ob der Versand über die sonst für die Hauptdomain übliche E-Mail-Infrastruktur versendet worden ist oder nicht. Möglicherweise lehnt die für die Domain zuständige Infrastruktur den Versand mit einer entsprechenden Fehlermeldung bereits ab oder im Report fehlen die entsprechenden Einträge. In diesem Fall sollte man prüfen, ob man nicht noch andere Testmöglichkeiten nutzen kann.[35] Für eine eindeutige Abnahme klingt das recht wage, aber der Test gilt eindeutig als *nicht bestanden*, wenn die E-Mail im Posteingang des verwendeten Empfängers auftaucht.

Allgemein sei noch zu erwähnen, dass die erstellten *SPF*- und *DMARC*-Einträgen stets der jeweiligen Interpretation sowie der technischen Umsetzung aufseiten des Empfängers unterliegt.

> Da man keinen Einfluss auf die technische Infrastruktur der Empfänger hat, sollten idealerweise **alle** *technisch-organisatorischen Maßnahmen* umgesetzt werden, sofern sie sich gegenseitig nicht beeinflussen.

[34] DKIM, SPF, SpamAssassin Email Validator
[35] Anmerkung: *Ich möchte an dieser Stelle keine Tipps geben, welche dem Versand von Spam ermöglichen könnten.*

Lieber zusätzlich noch eine weitere Spezifikation unterstützen, die kaum (noch) im Einsatz ist, als den Umgang von gegebenenfalls nicht eindeutig definierten Situationen gänzlich anderen zu überlassen…

Erklärung

Dieses Dokument wurde von mir, Maurice-Jörg Nießen, nach bestem Wissen und Gewissen erstellt. Es erhebt weder einen Anspruch auf Vollständigkeit, noch stellt es eine rechtliche Beratung dar. Die Nutzung der enthaltenen Informationen erfolgt auf eigenes Risiko. Für etwaige Schäden, die aus der Verwendung dieses Dokumentes resultieren, wird keine Haftung übernommen.

Jegliche Weitergabe oder Veröffentlichung dieses Dokuments – auch auszugsweise – bedarf der vorherigen schriftlichen Zustimmung des Autors.

Maurice-Jörg Nießen