Workgroup: DMARC
Internet-Draft: draft-ietf-dmarc-dmarcbis-13
Obsoletes: 7489, 9091 (if approved)
Published: 25 July 2022
Intended Status: Standards Track
Expires: 26 January 2023
Authors: T. Herr (ed) J. Levine (ed)
Valimail Standcore LLC

Domain-based Message Authentication, Reporting, and Conformance (DMARC)

Abstract

This document describes the Domain-based Message Authentication, Reporting, and Conformance (DMARC) protocol.

DMARC permits the owner of an email author's domain name to enable verification of the domain's use, to indicate the Domain Owner's or Public Suffix Operator's message handling preference regarding failed verification, and to request reports about use of the domain name. Mail receiving organizations can use this information when evaluating handling choices for incoming mail.

This document obsoletes RFC 7489.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 26 January 2023.

Copyright Notice

Copyright (c) 2022 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (https://trustee.ietf.org/license-info) in effect on the date of

publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction
2. Requirements
  2.1. High-Level Goals
  2.2. Anti-Phishing
  2.3. Scalability
  2.4. Out of Scope
3. Terminology and Definitions
  3.1. Conventions Used in This Document
  3.2. Definitions
    3.2.1. Authenticated Identifiers
    3.2.2. Author Domain
    3.2.3. Domain Owner
    3.2.4. Identifier Alignment
    3.2.5. Mail Receiver
    3.2.6. Non-existent Domains
    3.2.7. Organizational Domain
    3.2.8. Public Suffix Domain (PSD)
    3.2.9. Public Suffix Operator (PSO)
    3.2.10. PSO Controlled Domain Names
    3.2.11. Report Consumer
4. Overview and Key Concepts
  4.1. DMARC Basics
  4.2. Use of RFC5322.From
  4.3. Authentication Mechanisms
  4.4. Identifier Alignment Explained
    4.4.1. DKIM-Authenticated Identifiers
    4.4.2. SPF-Authenticated Identifiers
    4.4.3. Alignment and Extension Technologies
  4.5. Flow Diagram
  4.6. DNS Tree Walk
  4.7. DMARC Policy Discovery
  4.8. Organizational Domain Discovery
5. Policy
  5.1. DMARC Policy Record
  5.2. DMARC URIs
  5.3. General Record Format
  5.4. Formal Definition
  5.5. Domain Owner Actions
    5.5.1. Publish an SPF Policy for an Aligned Domain

5.5.2. Configure Sending System for DKIM Signing Using an Aligned Domain
5.5.3. Setup a Mailbox to Receive Aggregate Reports
5.5.4. Publish a DMARC Policy for the Author Domain and Organizational Domain
5.5.5. Collect and Analyze Reports
5.5.6. Decide If and When to Update DMARC Policy
5.6. PSO Actions
5.7. Mail Receiver Actions
5.7.1. Extract Author Domain
5.7.2. Determine Handling Policy
5.7.3. Store Results of DMARC Processing
5.7.4. Send Aggregate Reports
5.8. Policy Enforcement Considerations
6. DMARC Feedback
7. Changes from RFC 7489
7.1. IETF Category
7.2. Changes to Terminology and Definitions
7.2.1. Terms Added
7.2.2. Definitions Updated
7.3. Policy Discovery and Organizational Domain Determination
7.4. Reporting
7.5. Tags
7.5.1. Tags Added:
7.5.2. Tags Removed:
7.6. Domain Owner Actions
7.7. Report Generator Recommendations
7.8. General Editing and Formatting
8. Other Topics
8.1. Issues Specific to SPF
8.2. DNS Load and Caching
8.3. Rejecting Messages
8.4. Identifier Alignment Considerations
8.5. Interoperability Issues
9. IANA Considerations
9.1. Authentication-Results Method Registry Update
9.2. Authentication-Results Result Registry Update
9.3. Feedback Report Header Fields Registry Update
9.4. DMARC Tag Registry
9.5. DMARC Report Format Registry
9.6. Underscored and Globally Scoped DNS Node Names Registry
10. Security Considerations
10.1. Authentication Methods
10.2. Attacks on Reporting URIs
10.3. DNS Security
10.4. Display Name Attacks
10.5. External Reporting Addresses
10.6. Secure Protocols

10.7. Determination of the Organizational Domain For Relaxed Alignment
11. Normative References
12. Informative References
Appendix A. Technology Considerations
A.1. S/MIME
A.2. Method Exclusion
A.3. Sender Header Field
A.4. Domain Existence Test
A.5. Issues with ADSP in Operation
A.6. Organizational Domain Discovery Issues
A.7. Removal of the "pct" Tag
Appendix B. Examples
B.1. Identifier Alignment Examples
B.1.1. SPF
B.1.2. DKIM
B.2. Domain Owner Example
B.2.1. Entire Domain, Monitoring Only
B.2.2. Entire Domain, Monitoring Only, Per-Message Reports
B.2.3. Per-Message Failure Reports Directed to Third Party
B.2.4. Subdomain, Testing, and Multiple Aggregate Report URIs
B.3. Mail Receiver Example
B.3.1. SMTP Session Example
B.4. Organizational and Policy Domain Tree Walk Examples
B.4.1. Simple Organizational and Policy Example
B.4.2. Deep Tree Walk Example
B.4.3. Example with a PSD DMARC Record
B.5. Utilization of Aggregate Feedback: Example
Acknowledgements
Authors' Addresses

1. Introduction

RFC EDITOR: PLEASE REMOVE THE FOLLOWING PARAGRAPH BEFORE PUBLISHING: The source for this draft is maintained in GitHub at: https://github.com/ietf-wg-dmarc/draft-ietf-dmarc-dmarcbis

Abusive email often includes unauthorized and deceptive use of a domain name in the RFC5322.From header field. The domain typically belongs to an organization expected to be known to - and presumably trusted by - the recipient. The Sender Policy Framework (SPF) [RFC7208] and DomainKeys Identified Mail (DKIM) [RFC6376] protocols provide domain-level authentication but are not directly associated with the RFC5322.From domain. DMARC leverages these two protocols, providing a method for Domain Owners to publish a DNS record describing the email authentication policies for the RFC5322.From domain and to request specific handling for messages using that domain that fail authentication checks.

As with SPF and DKIM, DMARC classes results as "pass" or "fail". In order to get a DMARC result of "pass", a pass from either SPF or DKIM is required. In addition, the passed domain can be "aligned" with the RFC5322.From domain in one of two modes - "relaxed" or "strict". The mode is expressed in the domain's DMARC policy record. Domains are said to be "in relaxed alignment" if they have the same "Organizational Domain", which is the domain at the top of the domain hierarchy for the RFC5322.From domain while having the same administrative authority as the RFC5322.From domain. Domains are "in strict alignment" if and only if they are identical.

A DMARC pass indicates only that the RFC5322.From domain has been authenticated for that message. Authentication does not carry an explicit or implicit value assertion about that message or about the Domain Owner. Furthermore, a mail-receiving organization that performs DMARC verification can choose to honor the Domain Owner's requested message handling for authentication failures, but it is under no obligation to do so; it might choose different actions entirely.

For a mail-receiving organization supporting DMARC, a message that passes verification is part of a message stream that is reliably associated with the RFC5322.From field Domain Owner. Therefore, reputation assessment of that stream by the mail-receiving organization is not encumbered by accounting for unauthorized use of that domain in the RFC5322.From field. A message that fails this verification is not necessarily associated with the Domain Owner's domain and its reputation.

DMARC policy records can also cover non-existent sub-domains, below the "Organizational Domain", as well as domains at the top of the name hierarchy, controlled by Public Suffix Operators (PSOs).

DMARC, in the associated [DMARC-Aggregate-Reporting] and [DMARC-Failure-Reporting] documents, also specifies a reporting framework. Using it, a mail-receiving domain can generate regular reports about messages that claim to be from a domain publishing DMARC policies, sending those reports to the address(es) specified by the Domain Owner in the latter's DMARC policy record. Domain Owners can use these reports, especially the aggregate reports, to identify not only sources of mail attempting to fraudulently use their domain, but also (and perhaps more importantly) gaps in their own authentication practices. However, as with honoring the Domain Owner's stated mail handling preference, a mail-receiving organization supporting DMARC is under no obligation to send requested reports, although it is recommended that they do send aggregate reports.

Use of DMARC creates some interoperability challenges that require due consideration before deployment, particularly with configurations that can cause mail to be rejected. These are discussed in Section 8.

2. Requirements

Specification of DMARC is guided by the following high-level goals, security dependencies, detailed requirements, and items that are documented as out of scope.

2.1. High-Level Goals

DMARC has the following high-level goals:

- Allow Domain Owners and PSOs to assert their desired message handling for authentication failures for messages purporting to have authorship within the domain.

- Allow Domain Owners and PSOs to verify their authentication deployment.

- Minimize implementation complexity for both senders and receivers, as well as the impact on handling and delivery of legitimate messages.

- Reduce the amount of successfully delivered spoofed email.

- Work at Internet scale.

2.2. Anti-Phishing

DMARC is designed to prevent bad actors from sending mail that claims to come from legitimate senders, particularly senders of transactional email (official mail that is about business transactions). One of the primary uses of this kind of spoofed mail is phishing (enticing users to provide information by pretending to be the legitimate service requesting the information). Thus, DMARC is significantly informed by ongoing efforts to enact large-scale, Internet-wide anti-phishing measures.

Although DMARC can only be used to combat specific forms of exact-domain spoofing directly, the DMARC mechanism has been found to be useful in the creation of reliable and defensible message streams.

DMARC does not attempt to solve all problems with spoofed or otherwise fraudulent email. In particular, it does not address the use of visually similar domain names ("cousin domains") or abuse of the RFC5322.From human-readable <display-name>.

2.3. Scalability

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.

Although DMARC does not introduce third-party senders (namely external agents authorized to send on behalf of an operator) to the email-handling flow, it also does not preclude them. Such third parties are free to provide services in conjunction with DMARC.

2.4. Out of Scope

Several topics and issues are specifically out of scope for this work. These include the following:

* Different treatment of messages that are not authenticated versus those that fail authentication;

* Evaluation of anything other than RFC5322.From header field;

* Multiple reporting formats;

* Publishing policy other than via the DNS;

* Reporting or otherwise evaluating other than the last-hop IP address;

* Attacks in the RFC5322.From header field, also known as "display name" attacks;

* Authentication of entities other than domains, since DMARC is built upon SPF and DKIM, which authenticate domains; and

* Content analysis.

3. Terminology and Definitions

This section defines terms used in the rest of the document.

3.1. Conventions Used in This Document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] and [RFC8174] when, and only when, they appear in all capitals, as shown here.

Readers are encouraged to be familiar with the contents of [RFC5598]. In particular, that document defines various roles in the messaging infrastructure that can appear the same or separate in various contexts. For example, a Domain Owner could, via the messaging security mechanisms on which DMARC is based, delegate the ability to send mail as the Domain Owner to a third party with another role. This document does not address the distinctions among such roles; the reader is encouraged to become familiar with that material before continuing.

3.2. Definitions

The following sections define terms used in this document.

3.2.1. Authenticated Identifiers

Domain-level identifiers that are verified using authentication technologies are referred to as "Authenticated Identifiers". See Section 4.3 for details about the supported mechanisms.

3.2.2. Author Domain

The domain name of the apparent author, as extracted from the RFC5322.From header field.

3.2.3. Domain Owner

An entity or organization that owns a DNS domain. The term "owns" here indicates that the entity or organization being referenced holds the registration of that DNS domain. Domain Owners range from complex, globally distributed organizations, to service providers working on behalf of non-technical clients, to individuals responsible for maintaining personal domains. This specification uses this term as analogous to an Administrative Management Domain as defined in [RFC5598]. It can also refer to delegates, such as Report Consumers, when those are outside of their immediate management domain.

3.2.4. Identifier Alignment

When the domain in the address in the RFC5322.From header field has the same Organizational Domain as a domain verified by an authenticated identifier, it has Identifier Alignment. (see Section 3.2.7)

3.2.5. Mail Receiver

The entity or organization that receives and processes email. Mail Receivers operate one or more Internet-facing Mail Transport Agents (MTAs).

3.2.6. Non-existent Domains

For DMARC purposes, a non-existent domain is consistent with the meaning of the term as described in [RFC8020]. That is, if the response code received for a query for a domain name is NXDOMAIN, then the domain name and all the names under it do not exist.

3.2.7. Organizational Domain

The Organizational Domain is typically a domain that was registered with a domain name registrar. The Organizational Domain for any domain is determined by applying the algorithm found in Section 4.8.

3.2.8. Public Suffix Domain (PSD)

Some domains allow registration of subdomains that are "owned" by independent organizations. Real-world examples of these points are ".com", ".org", ".us", and ".co.uk". These domains are called "Public Suffix Domains (PSDs)". For example, "ietf.org" is a registered domain name, and ".org" is its PSD.

3.2.9. Public Suffix Operator (PSO)

A Public Suffix Operator is an organization that manages operations within a PSD, particularly the DNS records published for names at and under that domain name.

3.2.10. PSO Controlled Domain Names

PSO-Controlled Domain Names are names in the DNS that are managed by a PSO. PSO-controlled Domain Names may have one label (e.g., ".com") or more (e.g., ".co.uk"), depending on the PSD's policy.

3.2.11. Report Consumer

An operator that receives reports from another operator implementing the reporting mechanisms described in this document and/or the documents [DMARC-Aggregate-Reporting] and [DMARC-Failure-Reporting]. Such an operator might be receiving reports about messages related to a domain for which it is the Domain Owner or PSO, or reports about messages related to another operator's domain. This term applies collectively to the system components that receive and process these reports and the organizations that operate them.

4. Overview and Key Concepts

This section provides a general overview of the design and operation of the DMARC environment.

4.1. DMARC Basics

DMARC permits a Domain Owner or PSO to enable verification of a domain's use in an email message, to indicate the Domain Owner's or PSO's message handling preference regarding failed verification, and to request reports about use of the domain name. All information about a Domain Owner's or PSO's DMARC policy is published and retrieved via the DNS.

DMARC's verification function is based on whether the RFC5322.From domain is aligned with a domain name used in a supported authentication mechanism, as described in Section 4.3. When a DMARC policy exists for the domain name found in the RFC5322.From header field, and that domain name is not verified through an aligned supported authentication mechanism, the handling of that message can be affected based on the DMARC policy when delivered to a participating Mail Receiver.

A message satisfies the DMARC checks if at least one of the supported authentication mechanisms:

1. produces a "pass" result, and

2. produces that result based on an identifier that is in alignment, as described in Section 4.4.

It is important to note that the authentication mechanisms employed by DMARC authenticate only a DNS domain and do not authenticate the local-part of any email address identifier found in a message, nor do they validate the legitimacy of message content.

DMARC's feedback component involves the collection of information about received messages claiming to be from the Author Domain for periodic aggregate reports to the Domain Owner or PSO. The parameters and format for such reports are discussed in [DMARC-Aggregate-Reporting]

A DMARC-enabled Mail Receiver might also generate per-message reports that contain information related to individual messages that fail authentication checks. Per-message failure reports are a useful source of information when debugging deployments (if messages can be determined to be legitimate even though failing authentication) or in analyzing attacks. The capability for such services is enabled by DMARC but defined in other referenced material such as [RFC6591] and [DMARC-Failure-Reporting]

4.2. Use of RFC5322.From

One of the most obvious points of security scrutiny for DMARC is the choice to focus on an identifier, namely the RFC5322.From address,

which is part of a body of data that has been trivially forged throughout the history of email. This field is the one used by end users to identify the source of the message, and so it has always been a prime target for abuse through such forgery and other means.

Several points suggest that it is the most correct and safest thing to do in this context:

- Of all the identifiers that are part of the message itself, this is the only one guaranteed to be present.

- It seems the best choice of an identifier on which to focus, as most MUAs display some or all of the contents of that field in a manner strongly suggesting those data as reflective of the true originator of the message.

- Many high-profile email sources, such as email service providers, require that the sending agent have authenticated before email can be generated. Thus, for these mailboxes, the mechanism described in this document provides recipient end users with strong evidence that the message was indeed originated by the agent they associate with that mailbox, if the end user knows that these various protections have been provided.

- The absence of a single, properly formed RFC5322.From header field renders the message invalid. Handling of such a message is outside of the scope of this specification.

Since the sorts of mail typically protected by DMARC participants tend to only have single Authors, DMARC participants generally operate under a slightly restricted profile of RFC5322 with respect to the expected syntax of this field. See Section 5.7 for details.

4.3. Authentication Mechanisms

The following mechanisms for determining Authenticated Identifiers are supported in this version of DMARC:

- DKIM, [RFC6376], which provides a domain-level identifier in the content of the "d=" tag of a verified DKIM-Signature header field.

- SPF, [RFC7208], which can authenticate both the domain found in an SMTP [RFC5321] HELO/EHLO command (the HELO identity) and the domain found in an SMTP MAIL command (the MAIL FROM identity). As noted earlier, however, DMARC relies solely on SPF authentication of the domain found in SMTP MAIL FROM command. Section 2.4 of [RFC7208] describes MAIL FROM processing for cases in which the MAIL command has a null path.

4.4. Identifier Alignment Explained

Email authentication technologies authenticate various (and disparate) aspects of an individual message. For example, DKIM [RFC6376] authenticates the domain that affixed a signature to the message, while SPF [RFC7208] can authenticate either the domain that appears in the RFC5321.MailFrom (MAIL FROM) portion of an SMTP [RFC5321] conversation or the RFC5321.EHLO/HELO domain, or both. These may be different domains, and they are typically not visible to the end user.

DMARC authenticates use of the RFC5322.From domain by requiring either that it have the same Organizational Domain as an Authenticated Identifier (a condition known as "relaxed alignment") or that it be identical to the domain of the Authenticated Identifier (a condition known as "strict alignment"). The choice of relaxed or strict alignment is left to the Domain Owner and is expressed in the domain's DMARC policy record. Domain names in this context are to be compared in a case-insensitive manner, per [RFC4343].

It is important to note that Identifier Alignment cannot occur with a message that is not valid per [RFC5322], particularly one with a malformed, absent, or repeated RFC5322.From header field, since in that case there is no reliable way to determine a DMARC policy that applies to the message. Accordingly, DMARC operation is predicated on the input being a valid RFC5322 message object, and handling of such non-compliant cases is outside of the scope of this specification. Further discussion of this can be found in Section 5.7.1.

Each of the underlying authentication technologies that DMARC takes as input yields authenticated domains as their outputs when they succeed.

4.4.1. DKIM-Authenticated Identifiers

DMARC requires Identifier Alignment based on the result of a DKIM authentication because a message can bear a valid signature from any domain, including domains used by a mailing list or even a bad actor. Therefore, merely bearing a valid signature is not enough to infer authenticity of the Author Domain.

DMARC permits Identifier Alignment based on the result of a DKIM authentication to be strict or relaxed. (Note that these terms are not related to DKIM's "simple" and "relaxed" canonicalization modes.)

In relaxed mode, the identifiers are considered to be aligned if the Organizational Domains of both the DKIM-authenticated signing domain

(taken from the value of the d= tag in the signature) and that of the RFC5322.From domain are equal. In strict mode, only an exact match between both Fully Qualified Domain Names (FQDNs) is considered to produce Identifier Alignment.

To illustrate, in relaxed mode, if a verified DKIM signature successfully verifies with a "d=" domain of "example.com", and the RFC5322.From address is "alerts@news.example.com", the DKIM "d=" domain and the RFC5322.From domain are considered to be "in alignment", because both domains have the same Organizational Domain of "example.com". In strict mode, this test would fail because the d= domain does not exactly match the RFC5322.From domain.

However, a DKIM signature bearing a value of "d=com" would never allow an "in alignment" result, as "com" does not have a DMARC record therefore cannot be an Organizational Domain.

Note that a single email can contain multiple DKIM signatures, and it is considered to produce a DMARC "pass" result if any DKIM signature is aligned and verifies.

4.4.2. SPF-Authenticated Identifiers

DMARC permits Identifier Alignment based on the result of an SPF authentication. As with DKIM, Identifier Alignement can be either strict or relaxed.

In relaxed mode, the identifiers are considered to be aligned if the Organizational Domains of the SPF-authenticated domain and RFC5322.From domain are equal. In strict mode, only an exact match between the two FQDNs is considered to produce Identifier Alignment.

For example, in relaxed mode, if a message passes an SPF check with an RFC5321.MailFrom domain of "cbg.bounces.example.com", and the address portion of the RFC5322.From header field contains "payments@example.com", the Authenticated RFC5321.MailFrom domain identifier and the RFC5322.From domain are considered to be "in alignment" because they have the same Organizational Domain ("example.com"). In strict mode, this test would fail because the two domains are not identical.

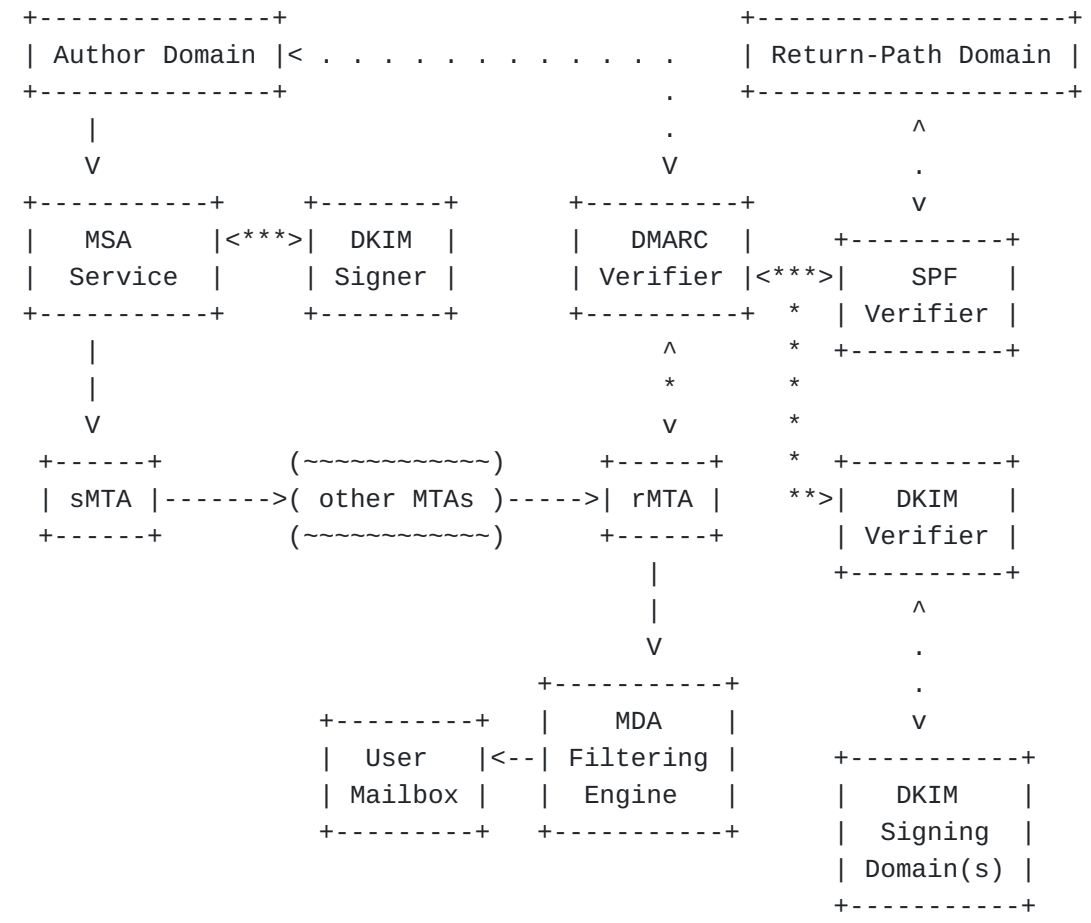
Note that SPF alignment checks in DMARC rely solely on the RFC5321.MailFrom domain. This differs from section 2.3 of [RFC7208], which recommends that SPF checks be done on not only the "MAIL FROM" but also on a separate check of the "HELO" identity.

4.4.3. Alignment and Extension Technologies

If in the future DMARC is extended to include the use of other authentication mechanisms, the extensions will need to allow for

domain identifier extraction so that alignment with the RFC5322.From domain can be verified.

4.5. Flow Diagram



MSA = Mail Submission Agent
MDA = Mail Delivery Agent

The above diagram shows a simple flow of messages through a DMARC-aware system. Solid lines denote the actual message flow, dotted lines involve DNS queries used to retrieve message policy related to the supported message authentication schemes, and asterisk lines indicate data exchange between message-handling modules and message authentication modules. "sMTA" is the sending MTA, and "rMTA" is the receiving MTA.

Put simply, when a message reaches a DMARC-aware rMTA, a DNS query will be initiated to determine if a DMARC policy exists that applies to the author domain. If a policy is found, the rMTA will use the results of SPF and DKIM verification checks to determine the ultimate DMARC authentication status. The DMARC status can then

factor into the message handling decision made by the recipient's mail system.

More details on specific actions for the parties involved can be found in Section 5.5 and Section 5.7.

4.6. DNS Tree Walk

The DMARC protocol defines a method for communicating information through the publishing of records in DNS. Both the content of the records and their location in the DNS hierarchy are used for two purposes: policy discovery (see Section 4.7) and Organizational Domain determination (see Section 4.8).

The relevant DMARC record for these purposes is not necessarily the DMARC policy record found in DNS at the same level as the name label for the domain in question. Instead, some domains will inherit their DMARC policy records from parent domains one level or more above them in the DNS hierarchy. Similarly, the Organizational Domain may be found at a higher level in the DNS hierarchy.

These records are discovered through the technique described here, known colloquially as the "DNS Tree Walk". The target of any DNS Tree Walk is a valid DMARC policy record, but the rules defining required content for that record depend on the reason for performing the Tree Walk.

To prevent possible abuse of the DNS, a shortcut is built into the process so that domains that have more than five labels do not result in more than five DNS queries.

The generic steps for a DNS Tree Walk are as follows:

1. Query the DNS for a DMARC TXT record at the DNS domain matching the one found in the domain(s) described above. A possibly empty set of records is returned.

2. Records that do not start with a "v=" tag that identifies the current version of DMARC are discarded. If multiple DMARC records are returned, they are all discarded. If a single record remains and it contains a "psd=n" tag, stop.

3. Determine the target for additional queries (if needed; see the note in Section 4.8), using steps 4 through 8 below.

4. Break the subject DNS domain name into a set of "n" ordered labels. Number these labels from right to left; e.g., for "a.mail.example.com", "com" would be label 1, "example" would be label 2, "mail" would be label 3, and so forth.

5. Count the number of labels found in the subject DNS domain. Let that number be "x". If x < 5, remove the left-most (highest-numbered) label from the subject domain. If x >= 5, remove the left-most (highest-numbered) labels from the subject domain until 4 labels remain. The resulting DNS domain name is the new target for subsequent lookups.

6. Query the DNS for a DMARC TXT record at the DNS domain matching this new target in place of the RFC5322.From domain in the message. A possibly empty set of records is returned.

7. Records that do not start with a "v=" tag that identifies the current version of DMARC are discarded. If multiple DMARC records are returned for a single target, they are all discarded. If a single record remains and it contains a "psd=n" or "psd=y" tag, stop.

8. Determine the target for additional queries by removing a single label from the target domain as described in step 5 and repeating steps 6 and 7 until the process stops or there are no more labels remaining.

To illustrate, for a message with the arbitrary RFC5322.From domain of "a.b.c.d.e.mail.example.com", a full DNS Tree Walk would require the following five queries, in order to locate the policy or Organizational Domain:

*_dmarc.a.b.c.d.e.mail.example.com

*_dmarc.e.mail.example.com

*_dmarc.mail.example.com

*_dmarc.example.com

*_dmarc.com

4.7. DMARC Policy Discovery

For policy discovery, a DNS Tree Walk starts at the point in the DNS hierarchy that matches the domain in the RFC5322.From header of the message. The DMARC policy to be applied to the message will be the record found at one of these three locations:

*The RFC5322.From domain

*The Organizational Domain (as determined by a separate DNS Tree Walk) of the RFC5322.From domain

*The Public Suffix Domain of the RFC5322.From domain

If the DMARC policy to be applied is that of the RFC5322.From domain, then the DMARC policy is taken from the p= tag of the record. If the DMARC policy is taken from either the Organizational Domain or the Public Suffix Domain and that domain is different than the RFC5322.From domain, then the DMARC policy is taken from the sp= tag (if any) if the RFC5322.From domain exists and the np= tag (if any) if the RFC5322.From domain does not exist. In the absence of applicable sp= or np= tags, the p= tag policy is used for subdomains.

If a retrieved policy record does not contain a valid "p" tag, or contains an "sp" or "np" tag that is not valid, then:

- If a "rua" tag is present and contains at least one syntactically valid reporting URI, the Mail Receiver MUST act as if a record containing "p=none" was retrieved, and continue processing;

- Otherwise, the Mail Receiver applies no DMARC processing to this message.

If the set produced by the DNS Tree Walk contains no DMARC policy record (i.e., any indication that there is no such record as opposed to a transient DNS error), Mail Receivers MUST NOT apply the DMARC mechanism to the message.

Handling of DNS errors when querying for the DMARC policy record is left to the discretion of the Mail Receiver. For example, to ensure minimal disruption of mail flow, transient errors could result in delivery of the message ("fail open"), or they could result in the message being temporarily rejected (i.e., an SMTP 4yx reply), which invites the sending MTA to try again after the condition has possibly cleared, allowing a definite DMARC conclusion to be reached ("fail closed").

Note: PSD policy is not used for Organizational Domains that have published a DMARC policy. Specifically, this is not a mechanism to provide feedback addresses (rua/ruf) when an Organizational Domain has declined to do so.

4.8. Organizational Domain Discovery

For Organizational Domain discovery, it may be necessary to perform multiple DNS Tree Walks in order to determine if any two domains are in alignment. This means that a DNS Tree Walk to discover an Organizational Domain might start at any of the following locations:

- The domain in the RFC5322.From header of the message.

- The RFC5321.MailFrom domain if there is an SPF pass result for the message.

- *Any DKIM d= domain if there is a DKIM pass result for the message for that domain.

Note: There is no need to perform Tree Walk searches for Organizational Domains under any of the following conditions:

- *The RFC5322.From domain and the RFC5321.MailFrom domain (if SPF authenticated), and/or the DKIM d= domain (if present and authenticated,) are all the same and that domain has a DMARC record. In this case, this common domain is treated as the Organizational Domain.

- *No applicable DMARC policy is discovered for the RFC5322.From domain during the tree walk for that domain. In this case, the DMARC mechanism does not apply to the message in question.

- *The record for the RFC5322.From domain indicates strict alignment. In this case, a simple string compare between the RFC5322.From domain and the RFC5321.MailFrom domain (if SPF authenticated), and/or the DKIM d= domain (if present and authenticated) is all that is required.

To discover the Organizational Domain for a domain, perform the DNS Tree Walk described in Section 4.6 as needed for any of the domains in question.

For each Tree Walk that retrieved valid DMARC records, select the Organizational Domain from the domains for which valid DMARC records were retrieved from the longest to the shortest:

1. If a valid DMARC record contains the psd= tag set to 'n' (psd=n), this is the Organizational Domain and the selection process is complete.

2. If a valid DMARC record, other than the one for the domain where the tree walk started, contains the psd= tag set to 'y' (psd=y), the Organizational Domain is the domain one label below this one in the DNS hierarchy, and the selection process is complete.

3. Otherwise select the record for the domain with the fewest number of labels. This is the Organizational Domain and the selection process is complete.

If this process does not determine the Organizational Domain, then the initial target domain is the Organizational Domain.

For example, given the starting domain "a.mail.example.com", a search for the Organizational Domain would require a series of DNS queries for DMARC records starting with "_dmarc.a.mail.example.com"

and finishing with "_dmarc.com". If there are DMARC records for "_dmarc.mail.example.com" and "_dmarc.example.com", but not for "_dmarc.a.mail.example.com" or "_dmarc.com", then the Organizational Domain for this domain would be "example.com".

As another example, given the starting domain "a.mail.example.com", if a search for the Organizational Domain only yields a DMARC record at "_dmarc.com" and that record contains the tag psd=y, then the Organizational Domain for this domain would be "example.com".

5. Policy

A Domain Owner or PSO advertises DMARC participation of one or more of its domains by adding a DNS TXT record (described in Section 5.1) to those domains. In doing so, Domain Owners and PSOs indicate their handling preference regarding failed authentication for email messages making use of their domain in the RFC5322.From header field as well as their desire for feedback about those messages. Mail Receivers in turn can take into account the Domain Owner's stated preference when making handling decisions about email messages that fail DMARC authentication checks.

A Domain Owner or PSO may choose not to participate in DMARC evaluation by Mail Receivers simply by not publishing an appropriate DNS TXT record for its domain(s). A Domain Owner can also choose to not have some underlying authentication technologies apply to DMARC evaluation of its domain(s). In this case, the Domain Owner simply declines to advertise participation in those schemes. For example, if the results of path authorization checks ought not be considered as part of the overall DMARC result for a given Author Domain, then the Domain Owner does not publish an SPF policy record that can produce an SPF pass result.

A Mail Receiver implementing the DMARC mechanism SHOULD make a best-effort attempt to adhere to the Domain Owner's or PSO's published DMARC Domain Owner Assessment Policy when a message fails the DMARC test. Since email streams can be complicated (due to forwarding, existing RFC5322.From domain-spoofing services, etc.), Mail Receivers MAY deviate from a published Domain Owner Assessment Policy during message processing and SHOULD make available the fact of and reason for the deviation to the Domain Owner via feedback reporting, specifically using the "PolicyOverride" feature of the aggregate report defined in [DMARC-Aggregate-Reporting]

5.1. DMARC Policy Record

Domain Owner and PSO DMARC preferences are stored as DNS TXT records in subdomains named "_dmarc". For example, the Domain Owner of "example.com" would post DMARC preferences in a TXT record at

"_dmarc.example.com". Similarly, a Mail Receiver wishing to query for DMARC preferences regarding mail with an RFC5322.From domain of "example.com" would issue a TXT query to the DNS for the subdomain of "_dmarc.example.com". The DNS-located DMARC preference data will hereafter be called the "DMARC record".

DMARC's use of the Domain Name Service is driven by DMARC's use of domain names and the nature of the query it performs. The query requirement matches with the DNS, for obtaining simple parametric information. It uses an established method of storing the information, associated with the target domain name, namely an isolated TXT record that is restricted to the DMARC context. Use of the DNS as the query service has the benefit of reusing an extremely well-established operations, administration, and management infrastructure, rather than creating a new one.

Per [RFC1035], a TXT record can comprise several "character-string" objects. Where this is the case, the module performing DMARC evaluation MUST concatenate these strings by joining together the objects in order and parsing the result as a single string.

5.2. DMARC URIs

[RFC3986] defines a generic syntax for identifying a resource. The DMARC mechanism uses this as the format by which a Domain Owner or PSO specifies the destination for the two report types that are supported.

The place such URIs are specified (see Section 5.3) allows a list of these to be provided. The list of URIs is separated by commas (ASCII 0x2c). A report SHOULD be sent to each listed URI provided in the DMARC record.

A formal definition is provided in Section 5.4.

5.3. General Record Format

DMARC records follow the extensible "tag-value" syntax for DNS-based key records defined in DKIM [RFC6376].

Section 9 creates a registry for known DMARC tags and registers the initial set defined in this document. Only tags defined in that registry are to be processed; unknown tags MUST be ignored.

The following tags are valid DMARC tags:

adkim: (plain-text; OPTIONAL; default is "r".) Indicates whether strict or relaxed DKIM Identifier Alignment mode is required by the Domain Owner. See Section 4.4.1 for details. Valid values are as follows:

r:
relaxed mode

s: strict mode

aspf: (plain-text; OPTIONAL; default is "r".) Indicates whether strict or relaxed SPF Identifier Alignment mode is required by the Domain Owner. See Section 4.4.2 for details. Valid values are as follows:

r: relaxed mode

s: strict mode

fo: Failure reporting options (plain-text; OPTIONAL; default is "0") Provides requested options for generation of failure reports. Report generators MAY choose to adhere to the requested options. This tag's content MUST be ignored if a "ruf" tag (below) is not also specified. Failure reporting options are shown below. The value of this tag is either "0", "1", or a colon-separated list of the options represented by alphabetic characters. The valid values and their meanings are:

0: Generate a DMARC failure report if all underlying authentication mechanisms fail to produce an aligned "pass" result.

1: Generate a DMARC failure report if any underlying authentication mechanism produced something other than an aligned "pass" result.

d: Generate a DKIM failure report if the message had a signature that failed evaluation, regardless of its alignment. DKIM-specific reporting is described in [RFC6651].

s: Generate an SPF failure report if the message failed SPF evaluation, regardless of its alignment. SPF-specific reporting is described in [RFC6652].

np: Domain Owner Assessment Policy for non-existent subdomains (plain-text; OPTIONAL). Indicates the message handling preference that the Domain Owner or PSO has for mail using non-existent subdomains of the domain queried. It applies only to non-existent subdomains of the domain queried and not to either existing subdomains or the domain itself. Its syntax is identical to that of the "p" tag defined below. If the "np" tag is absent, the policy specified by the "sp" tag (if the "sp" tag is present) or the policy specified by the "p" tag, if the "sp" tag is not present, MUST be applied for non-existent subdomains. Note that "np" will be ignored for DMARC records published on subdomains of

Organizational Domains and PSDs due to the effect of the DMARC policy discovery mechanism described in Section 4.7.

p: Domain Owner Assessment Policy (plain-text; RECOMMENDED for policy records). Indicates the message handling preference the Domain Owner or PSO has for mail using its domain but not passing DMARC verification. Policy applies to the domain queried and to subdomains, unless subdomain policy is explicitly described using the "sp" or "np" tags. If this tag is not present in an otherwise syntactically valid DMARC record, then the record is treated as if it included "p=none" (see Section 4.7). This tag is not applicable for third-party reporting records (see [DMARC-Aggregate-Reporting] and [DMARC-Failure-Reporting]) Possible values are as follows:

none: The Domain Owner offers no expression of preference.

quarantine: The Domain Owner considers such mail to be suspicious. It is possible the mail is valid, although the failure creates a significant concern.

reject: The Domain Owner considers all such failures to be a clear indication that the use of the domain name is not valid. See Section 8.3 for some discussion of SMTP rejection methods and their implications.

psd: A flag indicating whether the domain is a PSD. (plain-text; OPTIONAL; default is 'u'). Possible values are:

y: PSOs include this tag with a value of 'y' to indicate that the domain is a PSD. If a record containing this tag with a value of 'y' is found during policy discovery, this information will be used to determine the Organizational Domain and policy domain applicable to the message in question.

n: The DMARC policy record is published for a PSD, but it is the Organizational Domain for itself and its subdomain. There is no need to put psd=n in a DMARC record, except in the very unusual case of a parent PSD publishing a DMARC record without the requisite psd=y tag.

u: The default, indicating that the DMARC policy record is published for a domain that is not a PSD. Use the mechanism described in Section 4.8 for determining the Organizational Domain. There is no need to explicitly publish psd=u in a DMARC record.

rua: Addresses to which aggregate feedback is to be sent (comma-separated plain-text list of DMARC URIs; OPTIONAL). [DMARC-

Aggregate-Reporting] discusses considerations that apply when the domain name of a URI differs from that of the domain advertising the policy. See Section 10.5 for additional considerations. Any valid URI can be specified. A Mail Receiver MUST implement support for a "mailto:" URI, i.e., the ability to send a DMARC report via electronic mail. If the tag is not provided, Mail Receivers MUST NOT generate aggregate feedback reports for the domain. URIs not supported by Mail Receivers MUST be ignored. The aggregate feedback report format is described in [DMARC-Aggregate-Reporting]

ruf: Addresses to which message-specific failure information is to be reported (comma-separated plain-text list of DMARC URIs; OPTIONAL). If present, the Domain Owner or PSO is requesting Mail Receivers to send detailed failure reports about messages that fail the DMARC evaluation in specific ways (see the "fo" tag above). [DMARC-Aggregate-Reporting] discusses considerations that apply when the domain name of a URI differs from that of the domain advertising the policy. A Mail Receiver MUST implement support for a "mailto:" URI, i.e., the ability to send a DMARC report via electronic mail. If the tag is not provided, Mail Receivers MUST NOT generate failure reports for the domain. See Section 10.5 for additional considerations.

sp: Domain Owner Assessment Policy for all subdomains (plain-text; OPTIONAL). Indicates the message handling preference the Domain Owner or PSO has for mail using an existing subdomain of the domain queried but not passing DMARC verification. It applies only to subdomains of the domain queried and not to the domain itself. Its syntax is identical to that of the "p" tag defined above. If both the "sp" tag is absent and the "np" tag is either absent or not applicable, the policy specified by the "p" tag MUST be applied for subdomains. Note that "sp" will be ignored for DMARC records published on subdomains of Organizational Domains due to the effect of the DMARC policy discovery mechanism described in Section 4.7.

t: DMARC policy test mode (plain-text; OPTIONAL; default is 'n'). For the RFC5322.From domain to which the DMARC record applies, the "t" tag serves as a signal to the actor performing DMARC verification checks as to whether or not the domain owner wishes the assessment policy declared in the "p=", "sp=", and/or "np=" tags to actually be applied. This parameter does not affect the generation of DMARC reports. Possible values are as follows:

y: A request that the actor performing the DMARC verification check not apply the policy, but instead apply any special handling rules it might have in place, such as rewriting the

RFC5322.From header. The domain owner is currently testing its specified DMARC assessment policy.

n: The default, a request to apply the policy as specified to any message that produces a DMARC "fail" result.

v: Version (plain-text; REQUIRED). Identifies the record retrieved as a DMARC record. It MUST have the value of "DMARC1". The value of this tag MUST match precisely; if it does not or it is absent, the entire retrieved record MUST be ignored. It MUST be the first tag in the list.

A DMARC policy record MUST comply with the formal specification found in Section 5.4 in that the "v" tag MUST be present and MUST appear first. Unknown tags MUST be ignored. Syntax errors in the remainder of the record MUST be discarded in favor of default values (if any) or ignored outright.

Note that given the rules of the previous paragraph, addition of a new tag into the registered list of tags does not itself require a new version of DMARC to be generated (with a corresponding change to the "v" tag's value), but a change to any existing tags does require a new version of DMARC.

5.4. Formal Definition

The formal definition of the DMARC format, using [RFC5234] and [RFC7405], is as follows:

```
dmarc-uri       = URI
                  ; "URI" is imported from [RFC3986]; commas (ASCII
                  ; 0x2C) and exclamation points (ASCII 0x21)
                  ; MUST be encoded

dmarc-sep       = *WSP ";" *WSP

equals          = *WSP "=" *WSP

dmarc-record    = dmarc-version *(dmarc-sep dmarc-tag) [dmarc-sep]

dmarc-tag       = 1*ALPHA equals 1*dmarc-value

; any printing characters but semicolon
dmarc-value     = %x20-3A | %x3C-7E

dmarc-version   = "v" equals %s"DMARC1" ; case sensitive

; specialized syntax of DMARC values
dmarc-request   = "none" / "quarantine" / "reject"

dmarc-yorn      = "y" / "n"

dmarc-psd       = "y" / "n" / "u"

dmarc-rors      = "r" / "s"

dmarc-urilist   = dmarc-uri *(*WSP "," *WSP dmarc-uri)

dmarc-fo        = "0" / "1" / "d" / "s" / "d:s" / "s:d"
```

"Keyword" is imported from Section 4.1.2 of [RFC5321].

In each dmarc-tag, the dmarc-value has a syntax that depends on the tag name. The ABNF rule for each dmarc-value is specified in the following table:

Tag Name	Value Rule
p	dmarc-request
t	dmarc-yorn
psd	dmarc-psd
np	dmarc-request
sp	dmarc-request
adkim	dmarc-rors
aspf	dmarc-rors
rua	dmarc-urilist
ruf	dmarc-urilist
fo	dmarc-fo

Table 1: "Tag Names and Values"

5.5. Domain Owner Actions

This section describes Domain Owner actions to implement the DMARC mechanism.

5.5.1. Publish an SPF Policy for an Aligned Domain

Because DMARC relies on SPF [RFC7208] and DKIM [RFC6376], in order to take full advantage of DMARC, a Domain Owner SHOULD first ensure that SPF and DKIM authentication are properly configured. As a first step the Domain Owner SHOULD choose a domain to use as the RFC5321.MailFrom domain (i.e., the Return-Path domain) for its mail, one that aligns with the Author Domain, and then publish an SPF policy in DNS for that domain. The SPF record SHOULD be constructed at a minimum to ensure an SPF pass verdict for all known sources of mail for the RFC5321.MailFrom domain.

5.5.2. Configure Sending System for DKIM Signing Using an Aligned Domain

While it is possible to secure a DMARC pass verdict based on only SPF or DKIM, it is commonly accepted best practice to ensure that both authentication mechanisms are in place in order to guard against failure of just one of them. The Domain Owner SHOULD choose a DKIM-Signing domain (i.e., the d= domain in the DKIM-Signature header) that aligns with the Author Domain.

5.5.3. Setup a Mailbox to Receive Aggregate Reports

Proper consumption and analysis of DMARC aggregate reports is the key to any successful DMARC deployment for a Domain Owner. DMARC aggregate reports, which are XML documents and are defined in [DMARC-Aggregate-Reporting], contain valuable data for the Domain Owner, showing sources of mail using the Author Domain. Depending on how mature the Domain Owner's DMARC rollout is, some of these sources could be legitimate ones that were overlooked during the initial deployment of SPF and/or DKIM.

Because the aggregate reports are XML documents, it is recommended that they be machine-parsed, so setting up a mailbox involves more than just the physical creation of that mailbox. Many third-party services exist that will process DMARC aggregate reports, or the Domain Owner can create its own set of tools. No matter which method is chosen, the ability to parse these reports and consume the data contained in them will go a long way to ensuring a successful deployment.

5.5.4. Publish a DMARC Policy for the Author Domain and Organizational Domain

Once SPF, DKIM, and the aggregate reports mailbox are all in place, it's time to publish a DMARC record. For best results, Domain Owners SHOULD start with "p=none", with the rua tag containg a URI that references the mailbox created in the previous step. If the Organizational Domain is different than the Author Domain, a record also needs to be published for the Organizational Domain.

5.5.5. Collect and Analyze Reports

The reason for starting at "p=none" is to ensure that nothing's been missed in the initial SPF and DKIM deployments. In all but the most trivial setups, it is possible for a Domain Owner to overlook a server here or be unaware of a third party sending agreeement there. Starting at "p=none", therefore, takes advantage of DMARC's aggregate reporting function, with the Domain Owner using the reports to audit its own mail streams' authentication configurations.

5.5.6. Decide If and When to Update DMARC Policy

Once the Domain Owner is satisfied that it is properly authenticating all of its mail, then it is time to decide if it is appropriate to change the p= value in its DMARC record to p=quarantine or p=reject. Depending on its cadence for sending mail, it may take many months of consuming DMARC aggregate reports before a Domain Owner reaches the point where it is sure that it is properly authenticating all of its mail, and the decision on which p= value to use will depend on its needs.

5.6. PSO Actions

In addition to the DMARC Domain Owner actions, if a PSO publishes a DMARC record it MUST include the psd tag (see Section 5.3) with a value of 'y' ("psd=y").

5.7. Mail Receiver Actions

This section describes receiver actions in the DMARC environment.

5.7.1. Extract Author Domain

The domain in the RFC5322.From header field is extracted as the domain to be evaluated by DMARC. If the domain is a U-label, the domain name MUST be converted to an A-label, as described in Section 2.3 of [RFC5890], for further processing.

In order to be processed by DMARC, a message typically needs to contain exactly one RFC5322.From domain (a single From: field with a single domain in it). Not all messages meet this requirement, and the handling of those that are forbidden under [RFC5322] or that contain no meaningful domains is outside the scope of this document.

The case of a syntactically valid multi-valued RFC5322.From header field presents a particular challenge. When a single RFC5322.From header field contains multiple addresses, it is possible that there may be multiple domains used in those addresses. The process in this case is to only proceed with DMARC checking if the domain is identical for all of the addresses in a multi-valued RFC5322.From header field. Multi-valued RFC5322.From header fields with multiple domains MUST be exempt from DMARC checking.

Note that Public Suffix Domains are not exempt from DMARC policy application and reporting.

5.7.2. Determine Handling Policy

To arrive at a policy for an individual message, Mail Receivers MUST perform the following actions or their semantic equivalents. Steps 2-4 MAY be done in parallel, whereas steps 5 and 6 require input from previous steps.

The steps are as follows:

1. Extract the RFC5322.From domain from the message (as above).

2. Query the DNS for a DMARC policy record. Continue if one is found, or terminate DMARC evaluation otherwise. See Section 4.7 for details.

3. Perform DKIM signature verification checks. A single email could contain multiple DKIM signatures. The results of this step are passed to the remainder of the algorithm, MUST include "pass" or "fail", and if "fail", SHOULD include information about the reasons for failure. The results MUST further include the value of the "d=" and "s=" tags from each checked DKIM signature.

4. Perform SPF verification checks. The results of this step are passed to the remainder of the algorithm, MUST include "pass" or "fail", and if "fail", SHOULD include information about the reasons for failure. The results MUST further include the domain name used to complete the SPF check.

5. Conduct Identifier Alignment checks. With authentication checks and policy discovery performed, the Mail Receiver checks to see if Authenticated Identifiers fall into alignment as described

in Section 4.4. If one or more of the Authenticated Identifiers align with the RFC5322.From domain, the message is considered to pass the DMARC mechanism check. All other conditions (authentication failures, identifier mismatches) are considered to be DMARC mechanism check failures.

6. Apply policy, if appropriate. Emails that fail the DMARC mechanism check are handled in accordance with the discovered DMARC policy of the Domain Owner and any local policy rules enforced by the Mail Receiver. See Section 5.3 for details.

DMARC evaluation can only yield a "pass" result after one of the underlying authentication mechanisms passes for an aligned identifier. If neither passes and one or both of them fail due to a temporary error, the Mail Receiver evaluating the message is unable to conclude that the DMARC mechanism had a permanent failure; they therefore cannot apply the advertised DMARC policy. When otherwise appropriate, Mail Receivers MAY send feedback reports regarding temporary errors.

Handling of messages for which SPF and/or DKIM evaluation encounter a permanent DNS error is left to the discretion of the Mail Receiver.

5.7.3. Store Results of DMARC Processing

The results of Mail Receiver-based DMARC processing should be stored for eventual presentation back to the Domain Owner in the form of aggregate feedback reports. Section 5.3 and [DMARC-Aggregate-Reporting] discuss aggregate feedback.

5.7.4. Send Aggregate Reports

For a Domain Owner, DMARC aggregate reports provide data about all mailstreams making use of its domain in email, to include not only illegitimate uses but also, and perhaps more importantly, all legitimate uses. Domain Owners can use aggregate reports to ensure that all legitimate uses of their domain for sending email are properly authenticated, and once they are, express a stricter message handling preference in the p= tag in their DMARC policy records from none to quarantine to reject, if appropriate. In turn, DMARC policy records with p= tag values of 'quarantine' or 'reject' are higher value signals to Mail Receivers, ones that can assist Mail Receivers with handling decisions for a message in ways that p= tag values of 'none' cannot.

Given the above, in order to ensure maximum usefulness for DMARC across the email ecosystem, Mail Receivers SHOULD generate and send aggregate reports with a frequency of at least once every 24 hours.

5.8. Policy Enforcement Considerations

Mail Receivers MAY choose to reject or quarantine email even if email passes the DMARC mechanism check. The DMARC mechanism does not inform Mail Receivers whether an email stream is "good"; a DMARC result of "pass" only means that the domain in the RFC5322.From header has been verified by the DMARC mechanism. Mail Receivers are encouraged to maintain anti-abuse technologies to combat the possibility of DMARC-enabled criminal campaigns.

Mail Receivers MAY choose to accept email that fails the DMARC mechanism check even if the published Domain Owner Assessment Policy is "reject". Mail Receivers need to make a best effort not to increase the likelihood of accepting abusive mail if they choose not to honor the published Domain Owner Assessment Policy. At a minimum, addition of the Authentication-Results header field (see [RFC8601]) is RECOMMENDED when delivery of failing mail is done.

Mail Receivers are only obligated to report reject or quarantine policy actions in aggregate feedback reports that are due to published DMARC Domain Owner Assessment Policy. They are not required to report reject or quarantine actions that are the result of local policy. If local policy information is exposed, abusers can gain insight into the effectiveness and delivery rates of spam campaigns.

Final handling of a message is always a matter of local policy. An operator that wishes to favor DMARC policy over SPF policy, for example, will disregard the SPF policy, since enacting an SPF-determined rejection prevents evaluation of DKIM; DKIM might otherwise pass, satisfying the DMARC evaluation. There is a trade-off to doing so, namely acceptance and processing of the entire message body in exchange for the enhanced protection DMARC provides.

DMARC-compliant Mail Receivers typically disregard any mail-handling directive discovered as part of an authentication mechanism (e.g., Author Domain Signing Practices (ADSP) [RFC5617], SPF) where a DMARC record is also discovered that specifies a policy other than "none". Deviating from this practice introduces inconsistency among DMARC operators in terms of handling of the message. However, such deviation is not proscribed.

To enable Domain Owners to receive DMARC feedback without impacting existing mail processing, discovered policies of "p=none" MUST NOT modify existing mail handling processes.

Mail Receivers MUST also implement reporting instructions of DMARC, even in the absence of a request for DKIM reporting [RFC6651] or SPF

reporting [RFC6652]. Furthermore, the presence of such requests MUST NOT affect DMARC reporting.

6. DMARC Feedback

Providing Domain Owners with visibility into how Mail Receivers implement and enforce the DMARC mechanism in the form of feedback is critical to establishing and maintaining accurate authentication deployments. When Domain Owners can see what effect their policies and practices are having, they are better willing and able to use quarantine and reject policies.

The details of this feedback are described in [DMARC-Aggregate-Reporting]

Operational note for PSD DMARC: For PSOs, feedback for non-existent domains is desirable and useful, just as it is for org-level DMARC operators. See Section 4 of [RFC9091] for discussion of Privacy Considerations for PSD DMARC

7. Changes from RFC 7489

This document is intended to render obsolete [RFC7489]. As one might guess, that means that there are significant differences between RFC 7489 and this document. This section will summarize thos changes.

7.1. IETF Category

RFC 7489 was not an Internet Standards Track specification; it was instead published in the Informational Category. This document, by contrast, is intended to be Internet Standards Track.

7.2. Changes to Terminology and Definitions

The following changes were made to the Terminology and Defintions section.

7.2.1. Terms Added

These terms were added:

*Non-existent Domains

*Public Suffix Domain (PSD)

*Public Suffix Operator (PSO)

*PSO Controlled Domain Names

7.2.2. Definitions Updated

These definitions were updated:

- *Organizational Domain
- *Report Receiver (renamed to Report Consumer)

7.3. Policy Discovery and Organizational Domain Determination

The algorithms for DMARC policy discovery and for determining the Organizational Domain have been changed. Specifically, reliance on the Public Suffix List (PSL) has been replaced by a technique called a "DNS Tree Walk", and the methodology for the DNS Tree Walk is explained in detail in this document.

The DNS Tree Walk also incorporates PSD policy discovery, which was introduced in [RFC9091].

7.4. Reporting

Discussion of both aggregate and failure reporting have been moved to separate documents dedicated to the topics.

7.5. Tags

Several tags have been added to the "General Record Format" section of this document since RFC 7489 was published, and at the same time several others were removed.

7.5.1. Tags Added:

- *np - Policy for non-existent domains
- *psd - Flag indicating whether a domain is a Public Suffix Domain
- *t - Replacement for some pct tag functionality. See Appendix A.7 for further discussion

7.5.2. Tags Removed:

- *pct - Tag requesting application of DMARC policy to only a percentage of messages
- *rf - Tag specifying requested format of failure reports
- *ri - Tag specifying requested interval between aggregate reports

7.6. Domain Owner Actions

This section has been expanded upon from RFC 7489.

7.7. Report Generator Recommendations

In the cases where a DMARC policy record specifies multiple destinations for either aggregate reports or failure reports, RFC 7489 stated:

> Receivers MAY impose a limit on the number of URIs to which they will send reports but MUST support the ability to send to at least two.

This document in Section 5.2 says:

> A report SHOULD be sent to each listed URI provided in the DMARC record.

7.8. General Editing and Formatting

A great deal of the content from RFC 7489 was preserved in this document, but much of it was subject to either minor editing, re-ordering of sections, and/or both.

8. Other Topics

This section discusses some topics regarding choices made in the development of DMARC, largely to commit the history to record.

8.1. Issues Specific to SPF

Though DMARC does not inherently change the semantics of an SPF policy record, historically lax enforcement of such policies has led many to publish extremely broad records containing many large network ranges. Domain Owners are strongly encouraged to carefully review their SPF records to understand which networks are authorized to send on behalf of the Domain Owner before publishing a DMARC record.

Some Mail Receiver architectures might implement SPF in advance of any DMARC operations. This means that a "-" prefix on a sender's SPF mechanism, such as "-all", could cause that rejection to go into effect early in handling, causing message rejection before any DMARC processing takes place. Operators choosing to use "-all" should be aware of this.

8.2. DNS Load and Caching

DMARC policies are communicated using the DNS and therefore inherit a number of considerations related to DNS caching. The inherent conflict between freshness and the impact of caching on the reduction of DNS-lookup overhead should be considered from the Mail Receiver's point of view. If Domain Owners or PSOs publish a DNS

record with a very short TTL, the injection of large volumes of messages could cause Receivers to overwhelm the publisher's DNS. Although this is not a concern specific to DMARC, the implications of a very short TTL should be considered when publishing DMARC policies.

Conversely, long TTLs will cause records to be cached for long periods of time. This can cause a critical change to DMARC parameters advertised by a Domain Owner or PSO to go unnoticed for the length of the TTL (while waiting for DNS caches to expire). Avoiding this problem can mean shorter TTLs, with the potential problems described above. A balance should be sought to maintain responsiveness of DMARC preference changes while preserving the benefits of DNS caching.

8.3. Rejecting Messages

This protocol calls for rejection of a message during the SMTP session under certain circumstances. This is preferable to generation of a Delivery Status Notification [RFC3464], since fraudulent messages caught and rejected using DMARC would then result in annoying generation of such failure reports that go back to the RFC5321.MailFrom address.

This synchronous rejection is typically done in one of two ways:

* Full rejection, wherein the SMTP server issues a 5xy reply code as an indication to the SMTP client that the transaction failed; the SMTP client is then responsible for generating notification that delivery failed (see Section 4.2.5 of [RFC5321]).

* A "silent discard", wherein the SMTP server returns a 2xy reply code implying to the client that delivery (or, at least, relay) was successfully completed, but then simply discarding the message with no further action.

Each of these has a cost. For instance, a silent discard can help to prevent backscatter, but it also effectively means that the SMTP server has to be programmed to give a false result, which can confound external debugging efforts.

Similarly, the text portion of the SMTP reply may be important to consider. For example, when rejecting a message, revealing the reason for the rejection might give an attacker enough information to bypass those efforts on a later attempt, though it might also assist a legitimate client to determine the source of some local issue that caused the rejection.

In the latter case, when doing an SMTP rejection, providing a clear hint can be useful in resolving issues. A Mail Receiver might

indicate in plain text the reason for the rejection by using the word "DMARC" somewhere in the reply text. For example:

```
550 5.7.1 Email rejected per DMARC policy for example.com
```

Many systems are able to scan the SMTP reply text to determine the nature of the rejection. Thus, providing a machine-detectable reason for rejection allows the problems causing rejections to be properly addressed by automated systems.

If a Mail Receiver elects to defer delivery due to inability to retrieve or apply DMARC policy, this is best done with a 4xy SMTP reply code.

8.4. Identifier Alignment Considerations

The DMARC mechanism allows both DKIM and SPF-authenticated identifiers to authenticate email on behalf of a Domain Owner and, possibly, on behalf of different subdomains. If malicious or unaware users can gain control of the SPF record or DKIM selector records for a subdomain, the subdomain can be used to generate DMARC-passing email on behalf of the Organizational Domain.

For example, an attacker who controls the SPF record for "evil.example.com" can send mail with an RFC5322.From header field containing "foo@example.com" that can pass both authentication and the DMARC check against "example.com".

The Organizational Domain administrator should be careful not to delegate control of subdomains if this is an issue, and to consider using the "strict" Identifier Alignment option if appropriate.

8.5. Interoperability Issues

DMARC limits which end-to-end scenarios can achieve a "pass" result.

Because DMARC relies on SPF [RFC7208] and/or DKIM [RFC6376] to achieve a "pass", their limitations also apply.

Additional DMARC constraints occur when a message is processed by some Mediators, such as mailing lists. Transiting a Mediator often causes either the authentication to fail or Identifier Alignment to be lost. These transformations may conform to standards but will still prevent a DMARC "pass".

In addition to Mediators, mail that is sent by authorized, independent third parties might not be sent with Identifier Alignment, also preventing a "pass" result.

Issues specific to the use of policy mechanisms alongside DKIM are further discussed in [RFC6377], particularly Section 5.2.

9. IANA Considerations

This section describes actions completed by IANA.

9.1. Authentication-Results Method Registry Update

IANA has added the following to the "Email Authentication Methods" registry:

Method	Defined	ptype	Property	Value	Status	Version
dmarc	[RFC7489]	header	from	the domain portion of the RFC5322.From header field	active	1
dmarc	[RFC7489]	polrec	p	the p= value read from the discovered policy record	active	1
dmarc	[RFC7489]	polrec	domain	the domain at which the policy record was discovered, if different from the RFC5322.From domain	active	1

Table 2: "Authentication-Results Method Registry Update"

9.2. Authentication-Results Result Registry Update

IANA has added the following in the "Email Authentication Result Names" registry:

Code	Existing/ New Code	Defined	Auth Method	Meaning	Status
none	existing	[RFC8601]	dmarc (added)	No DMARC policy record was published for the aligned identifier, or	active

Code	Existing/ New Code	Defined	Auth Method	Meaning	Status
				no aligned identifier could be extracted.	
pass	existing	[RFC8601]	dmarc (added)	A DMARC policy record was published for the aligned identifier, and at least one of the authentication mechanisms passed.	active
fail	existing	[RFC8601]	dmarc (added)	A DMARC policy record was published for the aligned identifier, and none of the authentication mechanisms passed.	active
temperror	existing	[RFC8601]	dmarc (added)	A temporary error occurred during DMARC evaluation. A later attempt might produce a final result.	active
permerror	existing	[RFC8601]	dmarc (added)	A permanent error occurred during DMARC evaluation, such as encountering a syntactically incorrect DMARC record. A later attempt is unlikely to produce a final result.	active

Table 3: "Authentication-Results Result Registry Update"

9.3. Feedback Report Header Fields Registry Update

The following has been added to the "Feedback Report Header Fields" registry:

Field Name: Identity-Alignment

Description: indicates whether the message about which a report is being generated had any identifiers in alignment as defined in RFC 7489

Multiple Appearances: No

Related "Feedback-Type": auth-failure

Reference: RFC 7489

Status: current

9.4. DMARC Tag Registry

A new registry tree called "Domain-based Message Authentication, Reporting, and Conformance (DMARC) Parameters" has been created. Within it, a new sub-registry called the "DMARC Tag Registry" has been created.

Names of DMARC tags are registered with IANA in this new sub-registry. New entries are assigned only for values that have been documented in a manner that satisfies the terms of Specification Required, per [RFC8126]. Each registration includes the tag name; the specification that defines it; a brief description; and its status, which is one of "current", "experimental", or "historic". The Designated Expert needs to confirm that the provided specification adequately describes the new tag and clearly presents how it would be used within the DMARC context by Domain Owners and Mail Receivers.

To avoid version compatibility issues, tags added to the DMARC specification are to avoid changing the semantics of existing records when processed by implementations conforming to prior specifications.

The initial set of entries in this registry is as follows:

Tag Name	Reference	Status	Description
adkim	RFC 7489	current	DKIM alignment mode
aspf	RFC 7489	current	SPF alignment mode
fo	RFC 7489	current	Failure reporting options

Tag Name	Reference	Status	Description
np	RFC 7489	current	Requested handling policy for non-existent subdomains
p	RFC 7489	current	Requested handling policy
pct	RFC 7489	historic	Sampling rate
psd	RFC 7489	current	Indicates whether policy record is published by a Public Suffix Domain
rf	RFC 7489	historic	Failure reporting format(s)
ri	RFC 7489	historic	Aggregate Reporting interval
rua	RFC 7489	current	Reporting URI(s) for aggregate data
ruf	RFC 7489	current	Reporting URI(s) for failure data
sp	RFC 7489	current	Requested handling policy for subdomains
t	RFC 7489	current	Test mode for the specified policy
v	RFC 7489	current	Specification version

Table 4: "DMARC Tag Registry"

9.5. DMARC Report Format Registry

Also within "Domain-based Message Authentication, Reporting, and Conformance (DMARC) Parameters", a new sub-registry called "DMARC Report Format Registry" has been created.

Names of DMARC failure reporting formats are registered with IANA in this registry. New entries are assigned only for values that satisfy the definition of Specification Required, per [RFC8126]. In addition to a reference to a permanent specification, each registration includes the format name; a brief description; and its status, which must be one of "current", "experimental", or "historic". The Designated Expert needs to confirm that the provided specification adequately describes the report format and clearly presents how it would be used within the DMARC context by Domain Owners and Mail Receivers.

The initial entry in this registry is as follows:

Format Name	Reference	Status	Description
afrf	RFC 7489	current	Authentication Failure Reporting Format (see [RFC6591])

Table 5: "DMARC Report Format Registry"

9.6. Underscored and Globally Scoped DNS Node Names Registry

Per [RFC8552], please add the following entry to the "Underscored and Globally Scoped DNS Node Names" registry:

RR Type	_NODE NAME	Reference
TXT	_dmarc	RFC 7489

Table 6: "Underscored and Globally Scoped DNS Node Names" registry

10. Security Considerations

This section discusses security issues and possible remediations (where available) for DMARC.

10.1. Authentication Methods

Security considerations from the authentication methods used by DMARC are incorporated here by reference.

10.2. Attacks on Reporting URIs

URIs published in DNS TXT records are well-understood possible targets for attack. Specifications such as [RFC1035] and [RFC2142] either expose or cause the exposure of email addresses that could be flooded by an attacker, for example; MX, NS, and other records found in the DNS advertise potential attack destinations; common DNS names such as "www" plainly identify the locations at which particular services can be found, providing destinations for targeted denial-of-service or penetration attacks.

Thus, Domain Owners will need to harden these addresses against various attacks, including but not limited to:

- high-volume denial-of-service attacks;
- deliberate construction of malformed reports intended to identify or exploit parsing or processing vulnerabilities;
- deliberate construction of reports containing false claims for the Submitter or Reported-Domain fields, including the possibility of false data from compromised but known Mail Receivers.

10.3. DNS Security

The DMARC mechanism and its underlying technologies (SPF, DKIM) depend on the security of the DNS. Examples of how hostile parties can have an adverse impact on DNS traffic include:

- If they can snoop on DNS traffic, they can get an idea of who is sending mail.

*If they can block outgoing or reply DNS messages, they can prevent systems from discovering senders' DMARC policies, causing recipients to assume p=none by default.

*If they can send forged response packets, they can make aligned mail appear unaligned or vice-versa.

None of these threats are unique to DMARC, and they can be addressed using a variety of techniques, including, but not limited to:

*Signing DNS records with DNSSEC [RFC4033], which enables recipients to detect and discard forged responses.

*DNS over TLS [RFC7858] or DNS over HTTPS [RFC8484] can mitigate snooping and forged responses.

10.4. Display Name Attacks

A common attack in messaging abuse is the presentation of false information in the display-name portion of the RFC5322.From header field. For example, it is possible for the email address in that field to be an arbitrary address or domain name, while containing a well-known name (a person, brand, role, etc.) in the display name, intending to fool the end user into believing that the name is used legitimately. The attack is predicated on the notion that most common MUAs will show the display name and not the email address when both are available.

Generally, display name attacks are out of scope for DMARC, as further exploration of possible defenses against these attacks needs to be undertaken.

There are a few possible mechanisms that attempt mitigation of these attacks, such as the following:

*If the display name is found to include an email address (as specified in [RFC5322]), execute the DMARC mechanism on the domain name found there rather than the domain name discovered originally. However, this addresses only a very specific attack space, and spoofers can easily circumvent it by simply not using an email address in the display name. There are also known cases of legitimate uses of an email address in the display name with a domain different from the one in the address portion, e.g.,

From: "user@example.org via Bug Tracker" support@example.com

*In the MUA, only show the display name if the DMARC mechanism succeeds. This too is easily defeated, as an attacker could arrange to pass the DMARC tests while fraudulently using another domain name in the display name.

*In the MUA, only show the display name if the DMARC mechanism passes and the email address thus verified matches one found in the receiving user's list of known addresses.

10.5. External Reporting Addresses

To avoid abuse by bad actors, reporting addresses generally have to be inside the domains about which reports are requested. In order to accommodate special cases such as a need to get reports about domains that cannot actually receive mail, Section 3 of [DMARC-Aggregate-Reporting] describes a DNS-based mechanism for verifying approved external reporting.

The obvious consideration here is an increased DNS load against domains that are claimed as external recipients. Negative caching will mitigate this problem, but only to a limited extent, mostly dependent on the default TTL in the domain's SOA record.

Where possible, external reporting is best achieved by having the report be directed to domains that can receive mail and simply having it automatically forwarded to the desired external destination.

Note that the addresses shown in the "ruf" tag receive more information that might be considered private data, since it is possible for actual email content to appear in the failure reports. The URIs identified there are thus more attractive targets for intrusion attempts than those found in the "rua" tag. Moreover, attacking the DNS of the subject domain to cause failure data to be routed fraudulently to an attacker's systems may be an attractive prospect. Deployment of [RFC4033] is advisable if this is a concern.

10.6. Secure Protocols

This document encourages use of secure transport mechanisms to prevent loss of private data to third parties that may be able to monitor such transmissions. Unencrypted mechanisms should be avoided.

In particular, a message that was originally encrypted or otherwise secured might appear in a report that is not sent securely, which could reveal private information.

10.7. Determination of the Organizational Domain For Relaxed Alignment

DMARC evaluation for relaxed alignment is highly sensitive to errors in the determination of the organizational domain if the RFC5322.From domain does not have a published policy. If an incorrectly selected organizational domain is a parent of the correct organizational domain, then relaxed alignment could

potentially allow a malicious sender to obtain DMARC PASS. This potential exists for both the legacy [RFC7489] and current methods for determining the organizational domain, the latter described in Section 4.8.

This issue is completely avoided by use of strict alignment and publishing DMARC records for all domains/sub-domains used as RFC5322.From domain in an organization's email.

For cases where strict alignment is not appropriate, this issue can be mitigated by periodically checking the DMARC records, if any, of PSDs above the organization's domains in the DNS tree and (for legacy [RFC7489] checking that appropriate PSL entries remain present). If a PSD domain publishes a DMARC record without the appropriate psd=y tag, organizational domain owners can add psd=n to their organizational domain's DMARC record so that the PSD record will not be incorrectly evaluated to be the organizational domain

11. Normative References

[DMARC-Aggregate-Reporting] Brotman, A., Ed., "DMARC Aggregate Reporting", February 2021, <https://datatracker.ietf.org/doc/draft-ietf-dmarc-aggregate-reporting/>.

[DMARC-Failure-Reporting] Jones, S.M., Ed. and A. Vesely, Ed., "DMARC Failure Reporting", February 2021, <https://datatracker.ietf.org/doc/draft-ietf-dmarc-failure-reporting/>.

[RFC1035] Mockapetris, P V., "Domain names - implementation and specification", DOI 10.17487/RFC1035, RFC 1035, STD 13, November 1987, <https://www.rfc-editor.org/info/rfc1035>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", DOI 10.17487/RFC2119, BCP 14, RFC 2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, DOI 10.17487/RFC3986, January 2005, <https://www.rfc-editor.org/info/rfc3986>.

[RFC4343] Eastlake 3rd, D., "Domain Name System (DNS) Case Insensitivity Clarification", RFC 4343, DOI 10.17487/RFC4343, January 2006, <https://www.rfc-editor.org/info/rfc4343>.

[RFC5234] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", DOI 10.17487/RFC5234, RFC

5234, STD 68, January 2008, <https://www.rfc-editor.org/info/rfc5234>.

[RFC5321] Klensin, J., "Simple Mail Transfer Protocol", RFC 5321, DOI 10.17487/RFC5321, October 2008, <https://www.rfc-editor.org/info/rfc5321>.

[RFC5322] Resnick, P., Ed., "Internet Message Format", DOI 10.17487/RFC5322, RFC 5322, October 2008, <https://www.rfc-editor.org/info/rfc5322>.

[RFC5890] Klensin, J., "Internationalized Domain Names for Applications (IDNA): Definitions and Document Framework", RFC 5890, DOI 10.17487/RFC5890, August 2010, <https://www.rfc-editor.org/info/rfc5890>.

[RFC6376] Crocker, D., Ed., Hansen, T., Ed., and M. Kucherawy, Ed., "DomainKeys Identified Mail (DKIM) Signatures", STD 76, RFC 6376, DOI 10.17487/RFC6376, September 2011, <https://www.rfc-editor.org/info/rfc6376>.

[RFC6591] Fontana, H., "Authentication Failure Reporting Using the Abuse Reporting Format", RFC 6591, DOI 10.17487/RFC6591, April 2012, <https://www.rfc-editor.org/info/rfc6591>.

[RFC6651] Kucherawy, M., "Extensions to DomainKeys Identified Mail (DKIM) for Failure Reporting", DOI 10.17487/RFC6651, RFC 6651, June 2012, <https://www.rfc-editor.org/info/rfc6651>.

[RFC6652] Kitterman, S., "Sender Policy Framework (SPF) Authentication Failure Reporting Using the Abuse Reporting Format", DOI 10.17487/RFC6652, RFC 6652, June 2012, <https://www.rfc-editor.org/info/rfc6652>.

[RFC7208] Kitterman, S., "Sender Policy Framework (SPF) for Authorizing Use of Domains in Email, Version 1", RFC 7208, DOI 10.17487/RFC7208, April 2014, <https://www.rfc-editor.org/info/rfc7208>.

[RFC7405] Kyzivat, P., "Case-Sensitive String Support in ABNF", RFC 7405, DOI 10.17487/RFC7405, December 2014, <https://www.rfc-editor.org/info/rfc7405>.

[RFC7489] Kucherawy, M., Ed. and E. Zwicky, Ed., "Domain-based Message Authentication, Reporting, and Conformance

(DMARC)", RFC 7489, DOI 10.17487/RFC7489, March 2015, <https://www.rfc-editor.org/info/rfc7489>.

[RFC8552] Crocker, D., "Scoped Interpretation of DNS Resource Records through "Underscored" Naming of Attribute Leaves", BCP 222, RFC 8552, DOI 10.17487/RFC8552, March 2019, <https://www.rfc-editor.org/info/rfc8552>.

[RFC9091] Kitterman, S. and T. Wicinski, Ed., "Experimental Domain-Based Message Authentication, Reporting, and Conformance (DMARC) Extension for Public Suffix Domains", RFC 9091, DOI 10.17487/RFC9091, July 2021, <https://www.rfc-editor.org/info/rfc9091>.

12. Informative References

[RFC2142] Crocker, D., "Mailbox Names for Common Services, Roles and Functions", RFC 2142, DOI 10.17487/RFC2142, May 1997, <https://www.rfc-editor.org/info/rfc2142>.

[RFC2308] Andrews, M., "Negative Caching of DNS Queries (DNS NCACHE)", RFC 2308, DOI 10.17487/RFC2308, March 1998, <https://www.rfc-editor.org/info/rfc2308>.

[RFC3464] Moore, K. and G. Vaudreuil, "An Extensible Message Format for Delivery Status Notifications", DOI 10.17487/RFC3464, RFC 3464, January 2003, <https://www.rfc-editor.org/info/rfc3464>.

[RFC4033] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "DNS Security Introduction and Requirements", DOI 10.17487/RFC4033, RFC 4033, March 2005, <https://www.rfc-editor.org/info/rfc4033>.

[RFC5598] Crocker, D., "Internet Mail Architecture", DOI 10.17487/RFC5598, RFC 5598, July 2009, <https://www.rfc-editor.org/info/rfc5598>.

[RFC5617] Allman, E., Fenton, J., Delany, M., and J. Levine, "DomainKeys Identified Mail (DKIM) Author Domain Signing Practices (ADSP)", RFC 5617, DOI 10.17487/RFC5617, August 2009, <https://www.rfc-editor.org/info/rfc5617>.

[RFC6377] Kucherawy, M., "DomainKeys Identified Mail (DKIM) and Mailing Lists", RFC 6377, BCP 167, DOI 10.17487/RFC6377, September 2011, <https://www.rfc-editor.org/info/rfc6377>.

[RFC7858] Hu, Z., Zhu, L., Heidemann, J., Mankin, A., Wessels, D., and P. Hoffman, "Specification for DNS over Transport

Layer Security (TLS)", DOI 10.17487/RFC7858, RFC 7858, May 2016, <https://www.rfc-editor.org/info/rfc7858>.

[RFC8020] Bortzmeyer, S. and S. Huque, "NXDOMAIN: There Really Is Nothing Underneath", DOI 10.17487/RFC8020, RFC 8020, November 2016, <https://www.rfc-editor.org/info/rfc8020>.

[RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", RFC 8126, DOI 10.17487/RFC8126, BCP 26, June 2017, <https://www.rfc-editor.org/info/rfc8126>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", DOI 10.17487/RFC8174, RFC 8174, BCP 14, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8484] Hoffman, P. and P. McManus, "DNS Queries over HTTPS (DoH)", DOI 10.17487/RFC8484, RFC 8484, October 2018, <https://www.rfc-editor.org/info/rfc8484>.

[RFC8601] Kucherawy, M., "Message Header Field for Indicating Message Authentication Status", DOI 10.17487/RFC8601, RFC 8601, May 2019, <https://www.rfc-editor.org/info/rfc8601>.

Appendix A. Technology Considerations

This section documents some design decisions that were made in the development of DMARC. Specifically addressed here are some suggestions that were considered but not included in the design, with explanatory text regarding the decision.

A.1. S/MIME

S/MIME, or Secure Multipurpose Internet Mail Extensions, is a standard for encryption and signing of MIME data in a message. This was suggested and considered as a third security protocol for authenticating the source of a message.

DMARC is focused on authentication at the domain level (i.e., the Domain Owner taking responsibility for the message), while S/MIME is really intended for user-to-user authentication and encryption. This alone appears to make it a bad fit for DMARC's goals.

S/MIME also suffers from the heavyweight problem of Public Key Infrastructure, which means that distribution of keys used to verify signatures needs to be incorporated. In many instances, this alone is a showstopper. There have been consistent promises that PKI usability and deployment will improve, but these have yet to

materialize. DMARC can revisit this choice after those barriers are addressed.

S/MIME has extensive deployment in specific market segments (government, for example) but does not enjoy similar widespread deployment over the general Internet, and this shows no signs of changing. DKIM and SPF both are deployed widely over the general Internet, and their adoption rates continue to be positive.

Finally, experiments have shown that including S/MIME support in the initial version of DMARC would neither cause nor enable a substantial increase in the accuracy of the overall mechanism.

A.2. Method Exclusion

It was suggested that DMARC include a mechanism by which a Domain Owner could tell Mail Receivers not to attempt verification by one of the supported methods (e.g., "check DKIM, but not SPF").

Specifically, consider a Domain Owner that has deployed one of the technologies, and that technology fails for some messages, but such failures don't cause enforcement action. Deploying DMARC would cause enforcement action for policies other than "none", which would appear to exclude participation by that Domain Owner.

The DMARC development team evaluated the idea of policy exception mechanisms on several occasions and invariably concluded that there was not a strong enough use case to include them. The specific target audience for DMARC does not appear to have concerns about the failure modes of one or the other being a barrier to DMARC's adoption.

In the scenario described above, the Domain Owner has a few options:

1. Tighten up its infrastructure to minimize the failure modes of the single deployed technology.

2. Deploy the other supported authentication mechanism, to offset the failure modes of the first.

3. Deploy DMARC in a reporting-only mode.

A.3. Sender Header Field

It has been suggested in several message authentication efforts that the Sender header field be checked for an identifier of interest, as the standards indicate this as the proper way to indicate a re-mailing of content such as through a mailing list. Most recently, it was a protocol-level option for DomainKeys, but on evolution to DKIM, this property was removed.

The DMARC development team considered this and decided not to include support for doing so, for the following reasons:

1. The main user protection approach is to be concerned with what the user sees when a message is rendered. There is no consistent behavior among MUAs regarding what to do with the content of the Sender field, if present. Accordingly, supporting checking of the Sender identifier would mean applying policy to an identifier the end user might never actually see, which can create a vector for attack against end users by simply forging a Sender field containing some identifier that DMARC will like.

2. Although it is certainly true that this is what the Sender field is for, its use in this way is also unreliable, making it a poor candidate for inclusion in the DMARC evaluation algorithm.

3. Allowing multiple ways to discover policy introduces unacceptable ambiguity into the DMARC evaluation algorithm in terms of which policy is to be applied and when.

A.4. Domain Existence Test

The presence of the "np" tag in this specification seemingly implies that there would be an agreed-upon standard for determining a domain's existence.

Since the DMARC protocol is one focused on email, one might think that the definition of resolvable in [RFC5321] applies. That is, only names that resolve to MX Resource Records (RRs), A RRs, or AAAA RRs are deemed resolvable and therefore exist in the DNS. This is also consistent with the process documented in [RFC5617] (ADSP), and is a common practice among MTA operators to determine whether or not to accept a mail message before performing other more expensive processing.

The DMARC protocol makes no such requirement for the existence of specific DNS RRs in order for a domain to exist; instead, if any RR exists for a domain, then the domain exists. To use the terminology from [RFC2308], an "NXDOMAIN" response (rcode "Name Error") to a DNS query means that the domain name does not exist, while a "NODATA" response (rcode "NOERROR") means that the given resource record type queried for does not exist, but the domain name does.

Furthermore, in keeping with [RFC8020], if a query for a name returns NXDOMAIN, then not only does the name not exist, every name below it in the DNS hierarchy also does not exist.

A.5. Issues with ADSP in Operation

DMARC has been characterized as a "super-ADSP" of sorts.

Contributors to DMARC have compiled a list of issues associated with ADSP, gained from operational experience, that have influenced the direction of DMARC:

1. ADSP has no support for subdomains, i.e., the ADSP record for example.com does not explicitly or implicitly apply to subdomain.example.com. If wildcarding is not applied, then spammers can trivially bypass ADSP by sending from a subdomain with no ADSP record.

2. Nonexistent subdomains are explicitly out of scope in ADSP. There is nothing in ADSP that states Mail Receivers should simply reject mail from NXDOMAINs regardless of ADSP policy (which of course allows spammers to trivially bypass ADSP by sending email from nonexistent subdomains).

3. ADSP has no operational advice on when to look up the ADSP record.

4. ADSP has no support for using SPF as an auxiliary mechanism to DKIM.

5. ADSP has no support for a slow rollout, i.e., no way to configure a percentage of email on which the Mail Receiver should apply the policy. This is important for large-volume senders.

6. ADSP has no explicit support for an intermediate phase where the Mail Receiver quarantines (e.g., sends to the recipient's "spam" folder) rather than rejects the email.

7. The binding between the "From" header domain and DKIM is too tight for ADSP; they must match exactly.

A.6. Organizational Domain Discovery Issues

An earlier informational version of the DMARC protocol [RFC7489] noted that the DNS does not provide a method by which the "domain of record", or the domain that was actually registered with a domain registrar, can be determined given an arbitrary domain name. That version further mentioned suggestions that have been made that attempt to glean such information from SOA or NS resource records, but these too are not fully reliable, as the partitioning of the DNS is not always done at administrative boundaries.

That previous version posited that one could "climb the tree" to find the Organizational Domain, but expressed concern that an attacker could exploit this for a denial-of-service attack through sending a high number of messages each with a relatively large number of nonsense labels, causing a Mail Receiver to perform a large number of DNS queries in search of a policy record. This version defines a method for performing a DNS Tree Walk, described in Section 4.6, and further mitigates the risk of the denial-of-service attack by expressly limiting the number of DNS queries to execute regardless of the number of labels in the domain name.

As a matter of historical record, the method for finding the Organizational Domain described in [RFC7489] is preserved here:

1. Acquire a "public suffix" list (PSL), i.e., a list of DNS domain names reserved for registrations. Some country Top-Level Domains (TLDs) make specific registration requirements, e.g., the United Kingdom places company registrations under ".co.uk"; other TLDs such as ".com" appear in the IANA registry of top-level DNS domains. A PSL is the union of all of these.

   A PSL can be obtained from various sources. The most common one is maintained by the Mozilla Foundation and made public at http://publicsuffix.org. License terms governing the use of that list are available at that URI.

   Note that if operators use a variety of public suffix lists, interoperability will be difficult or impossible to guarantee.

2. Break the subject DNS domain name into a set of "n" ordered labels. Number these labels from right to left; e.g., for "example.com", "com" would be label 1 and "example" would be label 2.

3. Search the public suffix list for the name that matches the largest number of labels found in the subject DNS domain. Let that number be "x".

4. Construct a new DNS domain name using the name that matched from the public suffix list and prefixing to it the "x+1"th label from the subject domain. This new name is the Organizational Domain.

Thus, since "com" is an IANA-registered TLD, a subject domain of "a.b.c.d.example.com" would have an Organizational Domain of "example.com".

The process of determining a suffix is currently a heuristic one. No list is guaranteed to be accurate or current.

A.7. Removal of the "pct" Tag

An earlier informational version of the DMARC protocol [RFC7489] included a "pct" tag and specified all integers from 0 to 100 inclusive as valid values for the tag. The intent of the tag was to provide domain owners with a method to gradually change their preferred assessment policy (the p= tag) from 'none' to 'quarantine' or from 'quarantine' to 'reject' by requesting the stricter treatment for just a percentage of messages that produced DMARC results of "fail".

Operational experience showed that the pct tag was usually not accurately applied, unless the value specified was either "0" or "100" (the default), and the inaccuracies with other values varied widely from implementation to implementation. The default value was easily implemented, as it required no special processing on the part of the Mail Receiver, while the value of "0" took on unintended significance as a value used by some intermediaries and mailbox providers as an indicator to deviate from standard handling of the message, usually by rewriting the RFC5322.From header in an effort to avoid DMARC failures downstream.

These custom actions when the pct= tag was set to "0" proved valuable to the email community. In particular, header rewriting by an intermediary meant that a Domain Owner's aggregate reports could reveal to the Domain Owner how much of its traffic was routing through intermediaries that don't rewrite the RFC5322.From header. It required work on the part of the Domain Owner to compare aggregate reports from before and after the p= value was changed and pct= was included in the DMARC policy record with a value of "0", but the data was there. Consequently, knowing how much mail was subject to possible DMARC failure due to lack of RFC5322.From header rewriting by intermediaries could assist the Domain Owner in choosing whether or not to proceed from an applied policy of p=none to p=quarantine or p=reject. Armed with this knowledge, the Domain Owner could make an informed decision regarding subjecting its mail traffic to possible DMARC failures based on the Domain Owner's tolerance for such things.

Because of the value provided by "pct=0" to Domain Owners, it was logical to keep this functionality in the protocol; at the same time it didn't make sense to support a tag named "pct" that had only two valid values. This version of the DMARC protocol therefore introduces the "t" tag as shorthand for "testing", with the valid values of "y" and "n", which are meant to be analogous in their application by mailbox providers and intermediaries to the "pct" tag values "0" and "100", respectively.

Appendix B. Examples

This section illustrates both the Domain Owner side and the Mail Receiver side of a DMARC exchange.

B.1. Identifier Alignment Examples

The following examples illustrate the DMARC mechanism's use of Identifier Alignment. For brevity's sake, only message headers are shown, as message bodies are not considered when conducting DMARC checks.

B.1.1. SPF

The following SPF examples assume that SPF produces a passing result. Alignment cannot exist if SPF does not produce a passing result.

Example 1: SPF in alignment:

```
MAIL FROM: <sender@example.com>

From: sender@example.com
Date: Fri, Feb 15 2002 16:54:30 -0800
To: receiver@example.org
Subject: here's a sample
```

In this case, the RFC5321.MailFrom parameter and the RFC5322.From header field have identical DNS domains. Thus, the identifiers are in strict alignment.

Example 2: SPF in alignment (parent):

```
MAIL FROM: <sender@child.example.com>

From: sender@example.com
Date: Fri, Feb 15 2002 16:54:30 -0800
To: receiver@example.org
Subject: here's a sample
```

In this case, the RFC5322.From header parameter includes a DNS domain that is a parent of the RFC5321.MailFrom domain. Thus, the identifiers are in relaxed alignment, because they both have the same Organizational Domain (example.com).

Example 3: SPF not in alignment:

```
  MAIL FROM: <sender@example.net>

  From: sender@child.example.com
  Date: Fri, Feb 15 2002 16:54:30 -0800
  To: receiver@example.org
  Subject: here's a sample
```

In this case, the RFC5321.MailFrom parameter includes a DNS domain that is neither the same as, a parent of, nor a child of the RFC5322.From domain. Thus, the identifiers are not in alignment.

B.1.2. DKIM

The examples below assume that the DKIM signatures pass verification. Alignment cannot exist with a DKIM signature that does not verify.

Example 1: DKIM in alignment:

```
  DKIM-Signature: v=1; ...; d=example.com; ...
  From: sender@example.com
  Date: Fri, Feb 15 2002 16:54:30 -0800
  To: receiver@example.org
  Subject: here's a sample
```

In this case, the DKIM "d=" parameter and the RFC5322.From header field have identical DNS domains. Thus, the identifiers are in strict alignment.

Example 2: DKIM in alignment (parent):

```
  DKIM-Signature: v=1; ...; d=example.com; ...
  From: sender@child.example.com
  Date: Fri, Feb 15 2002 16:54:30 -0800
  To: receiver@example.org
  Subject: here's a sample
```

In this case, the DKIM signature's "d=" parameter includes a DNS domain that is a parent of the RFC5322.From domain. Thus, the identifiers are in relaxed alignment, as they have the same Organizational Domain (example.com).

Example 3: DKIM not in alignment:

```
  DKIM-Signature: v=1; ...; d=sample.net; ...
  From: sender@child.example.com
  Date: Fri, Feb 15 2002 16:54:30 -0800
  To: receiver@example.org
  Subject: here's a sample
```

In this case, the DKIM signature's "d=" parameter includes a DNS domain that is neither the same as, a parent of, nor a child of the RFC5322.From domain. Thus, the identifiers are not in alignment.

B.2. Domain Owner Example

A Domain Owner that wants to use DMARC should have already deployed and tested SPF and DKIM. The next step is to publish a DNS record that advertises a DMARC policy for the Domain Owner's Organizational Domain.

B.2.1. Entire Domain, Monitoring Only

The owner of the domain "example.com" has deployed SPF and DKIM on its messaging infrastructure. The owner wishes to begin using DMARC with a policy that will solicit aggregate feedback from Mail Receivers without affecting how the messages are processed, in order to:

- Confirm that its legitimate messages are authenticating correctly
- Verify that all authorized message sources have implemented authentication measures
- Determine how many messages from other sources would be affected by a blocking policy

The Domain Owner accomplishes this by constructing a policy record indicating that:

- The version of DMARC being used is "DMARC1" ("v=DMARC1;")
- Mail Receivers should not alter how they treat these messages because of this DMARC policy record ("p=none")
- Aggregate feedback reports are sent via email to the address "dmarc-feedback@example.com" ("rua=mailto:dmarc-feedback@example.com")
- All messages from this Organizational Domain are subject to this policy (no "t" tag present, so the default of "n" applies).

The DMARC policy record might look like this when retrieved using a common command-line tool:

```
% dig +short TXT _dmarc.example.com.
"v=DMARC1; p=none; rua=mailto:dmarc-feedback@example.com"
```

To publish such a record, the DNS administrator for the Domain Owner creates an entry like the following in the appropriate zone file (following the conventional zone file format):

```
; DMARC record for the domain example.com

_dmarc  IN TXT ( "v=DMARC1; p=none; "
                 "rua=mailto:dmarc-feedback@example.com" )
```

B.2.2. Entire Domain, Monitoring Only, Per-Message Reports

The Domain Owner from the previous example has used the aggregate reporting to discover some messaging systems that had not yet implemented DKIM correctly, but they are still seeing periodic authentication failures. In order to diagnose these intermittent problems, they wish to request per-message failure reports when authentication failures occur.

Not all Mail Receivers will honor such a request, but the Domain Owner feels that any reports it does receive will be helpful enough to justify publishing this record. The default per-message report format ([RFC6591]) meets the Domain Owner's needs in this scenario.

The Domain Owner accomplishes this by adding the following to its policy record from Appendix B.2.1:

> *Per-message failure reports are sent via email to the address "auth-reports@example.com" ("ruf=mailto:auth-reports@example.com")

The DMARC policy record might look like this when retrieved using a common command-line tool (the output shown would appear on a single line but is wrapped here for publication):

```
% dig +short TXT _dmarc.example.com.
"v=DMARC1; p=none; rua=mailto:dmarc-feedback@example.com;
 ruf=mailto:auth-reports@example.com"
```

To publish such a record, the DNS administrator for the Domain Owner might create an entry like the following in the appropriate zone file (following the conventional zone file format):

```
; DMARC record for the domain example.com

_dmarc  IN TXT ( "v=DMARC1; p=none; "
                 "rua=mailto:dmarc-feedback@example.com; "
                 "ruf=mailto:auth-reports@example.com" )
```

B.2.3. Per-Message Failure Reports Directed to Third Party

The Domain Owner from the previous example is maintaining the same policy but now wishes to have a third party serve as a Report Consumer. Again, not all Mail Receivers will honor this request, but those that do may implement additional checks to verify that the third party wishes to receive the failure reports for this domain.

The Domain Owner needs to alter its policy record from Appendix B.2.2 as follows:

* Per-message failure reports are sent via email to the address "auth-reports@thirdparty.example.net" ("ruf=mailto:auth-reports@thirdparty.example.net")

The DMARC policy record might look like this when retrieved using a common command-line tool (the output shown would appear on a single line but is wrapped here for publication):

```
% dig +short TXT _dmarc.example.com.
"v=DMARC1; p=none; rua=mailto:dmarc-feedback@example.com;
 ruf=mailto:auth-reports@thirdparty.example.net"
```

To publish such a record, the DNS administrator for the Domain Owner might create an entry like the following in the appropriate zone file (following the conventional zone file format):

```
; DMARC record for the domain example.com

_dmarc IN TXT ( "v=DMARC1; p=none; "
                "rua=mailto:dmarc-feedback@example.com; "
                "ruf=mailto:auth-reports@thirdparty.example.net" )
```

Because the address used in the "ruf" tag is outside the Organizational Domain in which this record is published, conforming Mail Receivers will implement additional checks as described in Section 3 of [DMARC-Aggregate-Reporting]. In order to pass these additional checks, the Report Consumer's Domain Owner will need to publish an additional DNS record as follows:

* Given the DMARC record published by the Domain Owner at "_dmarc.example.com", the DNS administrator for the Report Consumer will need to publish a TXT resource record at "example.com._report._dmarc.thirdparty.example.net" with the value "v=DMARC1;".

The resulting DNS record might look like this when retrieved using a common command-line tool (the output shown would appear on a single line but is wrapped here for publication):

```
% dig +short TXT example.com._report._dmarc.thirdparty.example.net
"v=DMARC1;"
```

To publish such a record, the DNS administrator for example.net might create an entry like the following in the appropriate zone file (following the conventional zone file format):

```
; zone file for thirdparty.example.net
; Accept DMARC failure reports on behalf of example.com

example.com._report._dmarc   IN   TXT    "v=DMARC1;"
```

Mediators and other third parties should refer to Section 3 of [DMARC-Aggregate-Reporting] for the full details of this mechanism.

B.2.4. Subdomain, Testing, and Multiple Aggregate Report URIs

The Domain Owner has implemented SPF and DKIM in a subdomain used for pre-production testing of messaging services. It now wishes to express a handling preference for messages from this subdomain that fail to authenticate to indicate to participating Mail Receivers that use of this domain is not valid.

As a first step, it will express that it considers to be suspicious messages using this subdomain that fail authentication. The goal here will be to enable examination of messages sent to mailboxes hosted by participating Mail Receivers as method for troubleshooting any existing authentication issues. Aggregate feedback reports will be sent to a mailbox within the Organizational Domain, and to a mailbox at a Report Consumer selected and authorized to receive same by the Domain Owner.

The Domain Owner will accomplish this by constructing a policy record indicating that:

- The version of DMARC being used is "DMARC1" ("v=DMARC1;")
- It is applied only to this subdomain (record is published at "_dmarc.test.example.com" and not "_dmarc.example.com")
- Mail Receivers are advised that the Domain Owner considers messages that fail to authenticate to be suspicious ("p=quarantine")
- Aggregate feedback reports are sent via email to the addresses "dmarc-feedback@example.com" and "example-tld-test@thirdparty.example.net" ("rua=mailto:dmarc-feedback@example.com, mailto:tld-test@thirdparty.example.net")

*The Domain Owner desires only that an actor performing a DMARC verification check apply any special handling rules it might have in place, such as rewriting the RFC53322.From header; the Domain Owner is testing its setup at this point, and so does not want the handling policy to be applied. ("t=y")

The DMARC policy record might look like this when retrieved using a common command-line tool (the output shown would appear on a single line but is wrapped here for publication):

```
% dig +short TXT _dmarc.test.example.com
"v=DMARC1; p=quarantine; rua=mailto:dmarc-feedback@example.com,
 mailto:tld-test@thirdparty.example.net; t=y"
```

To publish such a record, the DNS administrator for the Domain Owner might create an entry like the following in the appropriate zone file:

```
; DMARC record for the domain test.example.com

_dmarc IN  TXT  ( "v=DMARC1; p=quarantine; "
                  "rua=mailto:dmarc-feedback@example.com,"
                  "mailto:tld-test@thirdparty.example.net;"
                  "t=y" )
```

Once enough time has passed to allow for collecting enough reports to give the Domain Owner confidence that all legitimate email sent using the subdomain is properly authenticating and passing DMARC checks, then the Domain Owner can update the policy record to indicate that it considers authentication failures to be a clear indication that use of the subdomain is not valid. It would do this by altering the DNS record to advise Mail Receivers of its position on such messages ("p=reject") and removing the testing flag ("t=y").

After alteration, the DMARC policy record might look like this when retrieved using a common command-line tool (the output shown would appear on a single line but is wrapped here for publication):

```
% dig +short TXT _dmarc.test.example.com
"v=DMARC1; p=reject; rua=mailto:dmarc-feedback@example.com,
 mailto:tld-test@thirdparty.example.net"
```

To publish such a record, the DNS administrator for the Domain Owner might create an entry like the following in the appropriate zone file:

```
; DMARC record for the domain test.example.com

_dmarc IN  TXT  ( "v=DMARC1; p=reject; "
                  "rua=mailto:dmarc-feedback@example.com,"
                  "mailto:tld-test@thirdparty.example.net" )
```

B.3. Mail Receiver Example

A Mail Receiver that wants to use DMARC should already be checking SPF and DKIM, and possess the ability to collect relevant information from various email-processing stages to provide feedback to Domain Owners (possibly via Report Consumers).

B.3.1. SMTP Session Example

An optimal DMARC-enabled Mail Receiver performs authentication and Identifier Alignment checking during the SMTP [RFC5321] conversation.

Prior to returning a final reply to the DATA command, the Mail Receiver's MTA has performed:

1. An SPF check to determine an SPF-authenticated Identifier.

2. DKIM checks that yield one or more DKIM-authenticated Identifiers.

3. A DMARC policy lookup.

The presence of an Author Domain DMARC record indicates that the Mail Receiver should continue with DMARC-specific processing before returning a reply to the DATA command.

Given a DMARC record and the set of Authenticated Identifiers, the Mail Receiver checks to see if the Authenticated Identifiers align with the Author Domain (taking into consideration any strict versus relaxed options found in the DMARC record).

For example, the following sample data is considered to be from a piece of email originating from the Domain Owner of "example.com":

```
Author Domain: example.com
SPF-authenticated Identifier: mail.example.com
DKIM-authenticated Identifier: example.com
DMARC record:
  "v=DMARC1; p=reject; aspf=r;
   rua=mailto:dmarc-feedback@example.com"
```

In the above sample, both the SPF-authenticated Identifier and the DKIM-authenticated Identifier align with the Author Domain. The Mail

Receiver considers the above email to pass the DMARC check, avoiding the "reject" policy that is requested to be applied to email that fails to pass the DMARC check.

If no Authenticated Identifiers align with the Author Domain, then the Mail Receiver applies the DMARC-record-specified policy. However, before this action is taken, the Mail Receiver can consult external information to override the Domain Owner's Assessment Policy. For example, if the Mail Receiver knows that this particular email came from a known and trusted forwarder (that happens to break both SPF and DKIM), then the Mail Receiver may choose to ignore the Domain Owner's policy.

The Mail Receiver is now ready to reply to the DATA command. If the DMARC check yields that the message is to be rejected, then the Mail Receiver replies with a 5xy code to inform the sender of failure. If the DMARC check cannot be resolved due to transient network errors, then the Mail Receiver replies with a 4xy code to inform the sender as to the need to reattempt delivery later. If the DMARC check yields a passing message, then the Mail Receiver continues on with email processing, perhaps using the result of the DMARC check as an input to additional processing modules such as a domain reputation query.

Before exiting DMARC-specific processing, the Mail Receiver checks to see if the Author Domain DMARC record requests AFRF-based reporting. If so, then the Mail Receiver can emit an AFRF to the reporting address supplied in the DMARC record.

At the exit of DMARC-specific processing, the Mail Receiver captures (through logging or direct insertion into a data store) the result of DMARC processing. Captured information is used to build feedback for Domain Owner consumption. This is not necessary if the Domain Owner has not requested aggregate reports, i.e., no "rua" tag was found in the policy record.

B.4. Organizational and Policy Domain Tree Walk Examples

If a 5322.From domain has no DMARC record, a receiver uses a tree walk to find the policy domain. If the policy in a policy domain allows relaxed alignment and the SPF or DKIM domains are different from 5322.From domain, a receiver uses a tree walk to discover the respective Organizational domains.

B.4.1. Simple Organizational and Policy Example

A mail receiver receives an email with:

5322.From domain example.com

5321.MailFrom domain
example.com

DKIM signature d= signing.example.com

In this example, _dmarc.example.com and _dmarc.signing.example.com both have DMARC records while _dmarc.com does not. If SPF or DKIM yield pass results, they still have to be aligned to support a DMARC pass. Since not all domains are the same, if the alignment is relaxed then the tree walk is performed to determine the organizational domain for each:

For the 5322.From domain, query _dmarc.example.com and _dmarc.com; example.com is the last element of the DNS tree with a DMARC record, so it is the organizational domain for example.com.

For the 5321.MailFrom, the Organizational domain already found for example.com is example.com, so SPF is aligned.

For the DKIM d= domain, query _dmarc.signing.example.com, _dmarc.example.com, and _dmarc.com. Both signing.example.com and example.com have DMARC records, but example.com is the highest element in the tree with a DMARC record, so example.com is the organizational domain. Since this is also the organizational domain for 5322.From, DKIM is aligned for relaxed alignment.

Since both SPF and DKIM are aligned, they can be used to determine if the message has a DMARC pass result. If the result is not pass, then the policy domain's DMARC record is used to determine the appropriate policy. In this case, since the 5322.From domain has a DMARC record, that is the policy domain.

B.4.2. Deep Tree Walk Example

A mail receiver receives an email with:

5322.From domain a.b.c.d.e.f.g.h.i.j.k.example.com

5322.MailFrom domain example.com

DKIM signature d= signing.example.com

Both _dmarc.example.com and *dmarc.signing.example.com have DMARC records, while*dmarc.com does not. If SPF or DKIM yield pass results, they still have to be aligned to support a DMARC pass. Since not all domains are the same, if the alignment is relaxed then the tree walk is performed to determine the organizational domain for each:

For the 5322.From domain, query _dmarc.a.b.c.d.e.f.g.h.i.j.k.example.com, skip to

_dmarc.j.k.example.com, then query _dmarc.k.example.com, _dmarc.example.com, and _dmarc.com. None of a.b.c.d.e.f.g.h.i.j.k.example.com, j.k.example.com, or k.example.com have a DMARC record. Since example.com is the last element of the DNS tree with a DMARC record, it is the organizational domain for example.com.

For the 5322.MailFrom domain, the Organizational domain already found for example.com is example. com. SPF is aligned.

For the DKIM d= domain, query _dmarc.signing.example.com, _dmarc.example.com, and _dmarc.com. Both signing.example.com and example.com have DMARC records, but example.com is the highest element in the tree with a DMARC record, so example.com is the organizational domain. Since this is also the organizational domain for 5322.From, DKIM is aligned for relaxed alignment.

Since both SPF and DKIM are aligned, they can be used to determine if the message has a DMARC pass result. If the results for both are not pass, then the policy domain's DMARC record is used to determine the appropriate policy. In this case, the 5322.From domain does not have a DMARC record, so the policy domain is the highest element in the DNS tree with a DMARC record, example.com.

B.4.3. Example with a PSD DMARC Record

In rare cases, a PSD publishes a DMARC record with a psd tag, which the tree walk must take into account.

A mail receiver receives an email with:

5322.From domain notyourbank.example

5322.MailFrom domain notyourbank.example

DKIM signature d= notyourbank.example

In this example, _dmarc.notyourbank.example does not have a DMARC record, and _dmarc.example has a DMARC record which includes the psd=y tag. Since all three domains are the same, they are aligned (strict or relaxed).

Since both SPF and DKIM are aligned, they can be used to determine if the message has a DMARC pass result. If the result is not pass, then the policy domain's DMARC record is used to determine the appropriate policy. In this case, the 5322.From domain does not have a DMARC record, so the policy domain is the highest element in the DNS tree with a DMARC record, example.

The three identities are identical, so they are aligned automatically, and no tree walk is needed to determine alignment.

B.5. Utilization of Aggregate Feedback: Example

Aggregate feedback is consumed by Domain Owners to verify their understanding of how a given domain is being processed by the Mail Receiver. Aggregate reporting data on emails that pass all DMARC-supporting authentication checks is used by Domain Owners to verify that their authentication practices remain accurate. For example, if a third party is sending on behalf of a Domain Owner, the Domain Owner can use aggregate report data to verify ongoing authentication practices of the third party.

Data on email that only partially passes underlying authentication checks provides visibility into problems that need to be addressed by the Domain Owner. For example, if either SPF or DKIM fails to pass, the Domain Owner is provided with enough information to either directly correct the problem or understand where authentication-breaking changes are being introduced in the email transmission path. If authentication-breaking changes due to email transmission path cannot be directly corrected, then the Domain Owner at least maintains an understanding of the effect of DMARC-based policies upon the Domain Owner's email.

Data on email that fails all underlying authentication checks provides baseline visibility on how the Domain Owner's domain is being received at the Mail Receiver. Based on this visibility, the Domain Owner can begin deployment of authentication technologies across uncovered email sources, if the mail that is failing the checks was generated by or on behalf of the Domain Owner. Data regarding failing authentication checks can also allow the Domain Owner to come to an understanding of how its domain is being misused.

Acknowledgements

DMARC and the draft version of this document submitted to the Independent Submission Editor were the result of lengthy efforts by an informal industry consortium: DMARC.org (see http://dmarc.org). Participating companies included Agari, American Greetings, AOL, Bank of America, Cloudmark, Comcast, Facebook, Fidelity Investments, Google, JPMorgan Chase & Company, LinkedIn, Microsoft, Netease, PayPal, ReturnPath, The Trusted Domain Project, and Yahoo!. Although the contributors and supporters are too numerous to mention, notable individual contributions were made by J. Trent Adams, Michael Adkins, Monica Chew, Dave Crocker, Tim Draegen, Steve Jones, Franck Martin, Brett McDowell, and Paul Midgen. The contributors would also

like to recognize the invaluable input and guidance that was provided early on by J.D. Falk.

Additional contributions within the IETF context were made by Kurt Anderson, Michael Jack Assels, Les Barstow, Anne Bennett, Jim Fenton, J. Gomez, Mike Jones, Scott Kitterman, Eliot Lear, John Levine, S. Moonesamy, Rolf Sonneveld, Henry Timmes, and Stephen J. Turnbull.

Authors' Addresses

Todd M. Herr
Valimail

Email: todd.herr@valimail.com

John Levine
Standcore LLC

Email: standards@standore.com