
Stream: Internet Engineering Task Force (IETF)
RFC: [9990](#)
Obsoletes: [7489](#)
Category: Standards Track
Published: May 2026
ISSN: 2070-1721
Author: A. Brotman, Ed.
Comcast, Inc.

RFC 9990

Domain-Based Message Authentication, Reporting, and Conformance (DMARC) Aggregate Reporting

Abstract

Domain-based Message Authentication, Reporting, and Conformance (DMARC) allows for Domain Owners to request aggregate reports from receivers. This report is an XML document and contains extensible elements that allow for other types of data to be specified later. The aggregate reports can be submitted by the receiver to the Domain Owner's specified destination as declared in the associated DNS record.

This document obsoletes RFC 7489.

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at <https://www.rfc-editor.org/info/rfc9990>.

Copyright Notice

Copyright (c) 2026 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	3
1.1. Terminology	4
1.1.1. Notation	4
1.1.2. DMARC Terminology	4
2. Document Status	4
3. DMARC Feedback	4
3.1. Aggregate Reports	4
3.1.1. Description of the Content of the XML File	5
3.1.2. Handling Domains in Reports	12
3.1.3. DKIM Signatures in Aggregate Reports	13
3.1.4. Unique Identifiers in Aggregate Reporting	13
3.1.5. Error Element	13
3.1.6. Policy Override Reason	13
3.2. Extensions	14
3.3. Changes in Policy During a Reporting Period	14
3.4. Report Request Discovery	14
3.5. Report Delivery	14
3.5.1. Definition of Report-ID	15
3.5.2. Email	15
3.5.3. Other Methods	17
3.5.4. Handling of Duplicates	17
4. Verifying External Destinations	18
5. Extensible Reporting	19
6. IANA Considerations	20
6.1. Registration for the DMARC Namespace	20
6.2. Registration for the DMARC XML Schema	21

7. Privacy Considerations	21
7.1. Report Recipients	21
7.2. Data Contained Within Reports	21
7.3. Feedback Leakage	22
8. Security Considerations	22
8.1. Report Content as an Attack	23
8.2. False Information	23
8.3. Disclosure of Filtering Information	23
9. Operational Considerations	23
9.1. Report Generation	23
9.2. Report Evaluation	23
9.3. Report Storage	24
10. References	24
10.1. Normative References	24
10.2. Informative References	25
Appendix A. DMARC XML Schema	26
Appendix B. Sample Report	33
Appendix C. Differences from RFC 7489	34
Acknowledgements	34
Author's Address	34

1. Introduction

A key component of Domain-based Message Authentication, Reporting, and Conformance (DMARC) [RFC9989] is the ability for Domain Owners to request that Mail Receivers provide various types of reports. These reports allow Domain Owners to have insight into which IP addresses are sending on their behalf and some insight into whether or not the volume may be legitimate.

These reports expose information relating to the DMARC policy, as well as the outcome of Sender Policy Framework (SPF) [RFC7208] and DomainKeys Identified Mail (DKIM) [RFC6376] validation.

1.1. Terminology

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] [RFC8174] when, and only when, they appear in all capitals, as shown here.

1.1.1. Notation

Certain properties of mail messages described in this document are referenced using notation found in [RFC5598] (e.g., "RFC5322.From").

This specification uses the Augmented Backus-Naur Form (ABNF) notation of [RFC5234] and [RFC7405].

1.1.2. DMARC Terminology

There are a number of terms defined in [RFC9989] that are used within this document. Understanding those definitions will aid in reading this document. The terms below are of noted interest:

- Author Domain
- DMARC Policy Record
- Domain Owner
- Mail Receiver
- Organizational Domain
- Report Consumer

2. Document Status

This document, in part, along with [RFC9989] and [RFC9991], obsoletes and replaces DMARC [RFC7489].

3. 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.

3.1. Aggregate Reports

The DMARC aggregate feedback report is designed to provide Domain Owners with precise insight into:

- authentication results,

- corrective action that needs to be taken by Domain Owners, and
- the effect of Domain Owner DMARC policy on mail streams processed by Mail Receivers.

Aggregate DMARC feedback provides visibility into real-world mail streams that Domain Owners need in order to make informed decisions regarding the publication of a DMARC policy. When Domain Owners know what legitimate mail they are sending, what the authentication results are on that mail, and what forged Mail Receivers are getting, they can make better decisions about the policies they need and the steps they need to take to enable those policies. When Domain Owners set policies appropriately and understand their effects, Mail Receivers can act on them confidently.

Visibility comes in the form of daily (or more frequent) feedback reports that are originated from Mail Receivers and that contain aggregate data on message streams relevant to the Domain Owner. This information includes data about messages that passed DMARC authentication as well as those that did not.

A separate report **MUST** be generated for each DMARC Policy Domain encountered during the reporting period. See below for further explanation in [Section 3.1.2](#) ("Handling Domains in Reports").

The report may include the following data:

- The DMARC policy discovered and applied, if any
- The selected message disposition
- The identifier evaluated by SPF and the SPF result, if any
- The identifier evaluated by DKIM and the DKIM result, if any
- For both DKIM and SPF, an indication of whether the identifier was in DMARC alignment (see [Section 3.2.10](#) of [\[RFC9989\]](#))
- Sending and receiving domains
- The number of successful authentications
- The counts of messages based on all messages received, even if their delivery is ultimately blocked by other filtering agents.

Each report **MUST** contain data for only one DMARC Policy Domain. A single report **MUST** contain data for one policy configuration. If multiple configurations were observed during a single reporting period, a reporting entity **MAY** choose to send multiple reports; otherwise, the reporting entity **SHOULD** note only the final configuration observed during the period. See below for further information.

3.1.1. Description of the Content of the XML File

The format for these reports is defined in the XML Schema Definition (XSD) in [Appendix A](#). The XSD includes the possible values for some of the elements below. Most of these values have a definition tied to [\[RFC9989\]](#).

The format is also described in the following sections. Each section describes a collection of sibling elements in the XML hierarchy. There are pointers to where in the hierarchy each table fits.

If a document does not match the specified format, the document evaluator **SHOULD** discard the report. The evaluator **MAY** choose to try to utilize some of the data; however, if the format is in question, the data may be as well. The report evaluator **MAY** choose to contact the report generator so that they may be alerted to an issue with the report format.

The column "#" specifies how many times an element may appear -- this is sometimes referred to as multiplicity. The possible values are:

- O: **OPTIONAL**, zero or one element
- R: **REQUIRED**, exactly one element
- *: **OPTIONAL**, zero or more elements
- +: **REQUIRED**, one or more elements

Some elements contain text meant for humans and support an optional "lang" attribute whose value indicates the language of its contents. The default value is "en". Elements supporting this optional attribute are marked with "[@lang]" at the start of their content description in the following tables.

3.1.1.1. XML Root Element

DMARC aggregate feedback reports have the root element "feedback" with its XML namespace set to the DMARC namespace.

Element name	#	Content
feedback	R	First level elements; see Section 3.1.1.2

Table 1: The XML Root Element

3.1.1.2. First Level Elements

The elements in this table **MUST** appear in the order listed.

Element name	#	Content
version	O	MUST have the value 1.0.
report_metadata	R	Report generator metadata; see Section 3.1.1.3 .
policy_published	R	The DMARC policy configuration observed by the receiving system; see Section 3.1.1.5 .

Element name	#	Content
extension	O	Allows for future extensibility; see Section 3.1.1.6 .
record	+	Record(s) of the feedback from the report generator; see Section 3.1.1.7 .

Table 2: First Level Elements of the Aggregate Feedback Report

There **MUST** be at least one "record" element; these elements contain data stating that IP addresses were seen to have delivered messages for the Author Domain to the receiving system. For each IP address that is being reported, there will be at least one "record" element.

3.1.1.3. Report Generator Metadata

Element name	#	Content
org_name	R	Name of the Reporting Organization.
email	R	Contact to use when contacting the Reporting Organization.
extra_contact_info	O	[@lang] Additional contact details.
report_id	R	Unique Report-ID; see Section 3.5.1 .
date_range	R	The reporting period; see Section 3.1.1.4 .
error	O	[@lang] Error messages encountered when processing the DMARC Policy Record; see Section 3.1.5 .
generator	O	The name and version of the report generator; this can help the Report Consumer find out where to report bugs.

Table 3: Report Generator Metadata

3.1.1.4. Contents of the "date_range" Element

This element describes the time range in UTC defining the reporting period of this report.

Element name	#	Content
begin	R	Start of the reporting period.
end	R	End of the reporting period.

Table 4: Contents of the "date_range" Element

- "begin" and "end" contain the number of seconds since the epoch.

The "begin" and "end" elements are meant to denote the reporting period and not the first/last observed message from the reporting period. When generating reports, these reporting periods **SHOULD NOT** overlap. Typically, the reporting period will encompass a single UTC day, beginning at 0000UTC.

3.1.1.5. Contents of the "policy_published" Element

This element provides information on the DMARC Policy Record published for the Author Domain. The elements from "p" and onwards contain the discovered or default value for the DMARC policy applied.

Unspecified tags have their default values.

Element name	#	Content
domain	R	The DMARC Policy Domain.
discovery_method	O	The method used to discover the DMARC Policy Record used during evaluation.
p	R	A Domain Owner Assessment Policy.
sp	O	A Domain Owner Assessment Policy.
np	O	A Domain Owner Assessment Policy.
fo	O	The value for the failure reporting options.
adkim	O	The DKIM Identifier Alignment mode.
aspf	O	The SPF Identifier Alignment mode.
testing	O	The value of the "t" tag.

Table 5: Contents of the "policy_published" Element

- "discovery_method" can have the value "psl" or "treewalk", where "psl" is the method from [RFC7489] and "treewalk" is described in [RFC9989].
- Many of the items above (p, sp, etc.) are defined in [RFC9989].

3.1.1.6. Contents of the "extension" Element

Use of extensions may cause elements to be added here. These elements **MUST** be namespaced.

Element name	#	Content
<any namespaced element>	*	File level elements defined by an extension.

Table 6: Contents of the "extension" Element

- "<any namespaced element>": Zero or more elements in the namespace of the related extension declared in the XML root element.

3.1.1.7. Contents of the "record" Element

The report **MUST** contain one or more records stating which IP addresses were seen to have delivered messages for the Author Domain to the receiving system. For each IP address that is being reported, there will be at least one "record" element.

This element contains all the authentication results that were evaluated by the receiving system for the given set of messages.

An unlimited number of "record" elements may be specified.

Use of extensions may cause other elements to be added to the end of the record; such elements **MUST** be namespaced.

One record per (IP, result, authentication identifiers) tuples.

The elements in this table **MUST** appear in the order listed.

Element name	#	Content
row	R	See Section 3.1.1.8 .
identifiers	R	The data that was used to apply policy for the given "row"; see Section 3.1.1.10 .
auth_results	R	The data related to authenticating the messages associated with this sending IP address; see Section 3.1.1.11 .
<any namespaced element>	*	Record level elements defined by an extension.

Table 7: Contents of the "record" Element

- "<any namespaced element>": Zero or more elements in the namespace of the related extension declared in the XML root element.

3.1.1.8. Contents of the "row" Element

A "row" element contains the details of the connecting system, and how many mail messages were received from it, for the particular combination of the policy evaluated.

Element name	#	Content
source_ip	R	The connecting IP address. IPv4address or IPv6address as defined in Section 3.2.2 of [RFC3986].
count	R	Number of messages for which the "policy_evaluated" was applied.
policy_evaluated	R	The DMARC disposition applied to matching messages; see Section 3.1.1.9 .

Table 8: Contents of the "row" Element

3.1.1.9. Contents of the "policy_evaluated" Element

This element describes the results of applying the DMARC policy. If alignment fails and the policy applied does not match the DMARC Policy Domain's configured policy, the "reason" element **MUST** be included.

The elements in this table **MUST** appear in the order listed.

Element name	#	Content
disposition	R	The result of applying the DMARC policy.
dkim	R	The result of the DKIM DMARC Identifier Alignment test.
spf	R	The result of the SPF DMARC Identifier Alignment test.
reason	*	Policy override reason; see Section 3.1.1.14 .

Table 9: Contents of the "policy_evaluated" Element

- "spf" and "dkim" **MUST** be the evaluated values as they relate to DMARC, not the values the receiver may have used when overriding the policy.
- "reason" elements are meant to contain any notes the reporter might want to include as to why the "disposition" policy does not match the "policy_published", such as a local policy override.

3.1.1.10. Contents of the "identifiers" Element

Element name	#	Content
header_from	R	The RFC5322.From domain from the message.
envelope_from	O	The RFC5321.MailFrom domain that the SPF check has been applied to.

Element name	#	Content
envelope_to	O	The RFC5321.RcptTo domain from the message.

Table 10: Contents of the "identifiers" Element

- "envelope_from" **MAY** exist but be empty if the message had a null reverse-path (see [Section 4.5.5](#) of [RFC5321]).

3.1.1.11. Contents of the "auth_results" Element

This element contains DKIM and SPF results, uninterpreted with respect to DMARC.

If validation is attempted for any DKIM signature, the results **MUST** be included in the report (within reason; see [Section 3.1.3](#) ("DKIM Signatures in Aggregate Reports") below for handling numerous signatures).

The elements in this table **MUST** appear in the order listed.

Element name	#	Content
dkim	*	DKIM authentication result; see Section 3.1.1.12 .
spf	O	SPF authentication result; see Section 3.1.1.13 .

Table 11: Contents of the "auth_results" Element

3.1.1.12. Contents of the "dkim" Element

Element name	#	Content
domain	R	The domain that was used during validation (the "d=" tag in the signature).
selector	R	The selector that was used during validation (the "s=" tag in the signature).
result	R	DKIM verification result; see below.
human_result	O	[@lang] More descriptive information to the Domain Owner relating to evaluation failures.

Table 12: Contents of the "dkim" Element

- "result" is a lowercase string where the value is one of the results defined in [Section 2.7.1](#) of [RFC8601].

3.1.1.13. Contents of the "spf" Element

Only the "MAIL FROM" identity (see [Section 2.4](#) of [\[RFC7208\]](#)) is used in DMARC.

Element name	#	Content
domain	R	The domain that was used during validation.
scope	O	The source of the domain used during validation.
result	R	SPF verification result; see below.
human_result	O	[@lang] More descriptive information to the Domain Owner relating to evaluation failures.

Table 13: Contents of the "spf" Element

- The only valid value for the "scope" element is "mfrom".
- "result" is a lowercase string where the value is one of the results defined in [Section 2.7.2](#) of [\[RFC8601\]](#).

3.1.1.14. Contents of the "reason" Element

The policy override reason consists of a pre-defined override type and free-text comment; see [Section 3.1.6](#).

Element name	#	Content
type	R	The reason the DMARC policy was overridden
comment	O	[@lang] Further details, if available.

Table 14: Contents of the "reason" Element

3.1.2. Handling Domains in Reports

In the same report, there **MUST** be a single DMARC Policy Domain, though there could be multiple RFC5322.From domains. Each RFC5322.From domain will create its own "record" within the report. Consider the case where there are three domains with traffic volume to report: example.com, foo.example.com, and bar.example.com. There will be explicit DMARC Policy Records for example.com and bar.example.com, with distinct policies. There is no explicit DMARC Policy Record for foo.example.com, so it will be reliant on the policy described for example.com. For a report period, there would now be two reports.

The first report will be for bar.example.com and contain only one "record", for bar.example.com. The second report will be for example.com and will contain multiple "record" elements, one for example.com and one for foo.example.com (and extensibly, other "record" elements for subdomains that likewise did not have an explicit DMARC Policy Record).

3.1.3. DKIM Signatures in Aggregate Reports

Within a single message, the possibility exists that there could be multiple DKIM signatures. When validation of the message occurs, some signatures may pass, while some may not. As these pertain to DMARC, and especially to aggregate reporting, reporters may not find it clear which DKIM signatures they should include in a report. Signatures, regardless of outcome, could help the report ingester determine the source of a message. However, there is a preference as to which signatures are included.

1. A signature that passes DKIM, in strict alignment with the RFC5322.From domain
2. A signature that passes DKIM, in relaxed alignment with the RFC5322.From domain
3. Any other DKIM signatures that pass
4. DKIM signatures that do not pass

A report **SHOULD** contain no more than 100 signatures for a given "row", in decreasing priority.

3.1.4. Unique Identifiers in Aggregate Reporting

There are a few places where a unique identifier is specified as part of the body of the report, the subject, and so on. These unique identifiers should be consistent per each report. Specified below, the reader will see a "Report-ID" and "unique-id". These are the fields that **MUST** be identical when used.

3.1.5. Error Element

Here are a few examples of information contained within the "error" element(s):

- DMARC Policy Record evaluation errors (invalid "rua", "sp", etc.)
- Multiple DMARC Policy Records at a given location

Be mindful that the "error" element is an unbounded string but should not contain an extremely large body. Provide enough information to assist the Domain Owner with understanding some issues with their authentication or DMARC Policy Record.

3.1.6. Policy Override Reason

The "reason" element, indicating an override of the DMARC policy, consists of a mandatory "type" element and an optional "comment" element. The "type" element **MUST** have one of the pre-defined values listed below. The "comment" element is an unbounded string for providing further details.

Possible values for the policy override type:

"local_policy": The Mail Receiver's local policy exempted the message from being subjected to the Domain Owner's requested policy action.

"mailing_list": Local heuristics determined that the message arrived via a mailing list, and thus authentication of the original message was not expected to succeed.

"other": Some policy exception not covered by the other entries in this list occurred. Additional details can be found in the "comment" element.

"policy_test_mode": The message was exempted from application of policy by the testing mode ("t" tag) in the DMARC Policy Record.

"trusted_forwarder": Message authentication failure was anticipated by other evidence linking the message to a locally maintained list of known and trusted forwarders.

3.2. Extensions

The document format supports optional elements for extensions. The absence or existence of this section **SHOULD NOT** create an error when processing reports. This will be covered in [Section 5](#) ("Extensible Reporting").

3.3. Changes in Policy During a Reporting Period

Note that Domain Owners or their agents may change the published DMARC Policy Record for a domain or subdomain at any time. From a Mail Receiver's perspective, this will occur during a reporting period and may be noticed during that period, at the end of that period when reports are generated, or during a subsequent reporting period, all depending on the Mail Receiver's implementation. Under these conditions, it is possible that a Mail Receiver could do any of the following:

- generate for such a reporting period a single aggregate report that includes message dispositions based on the old policy, or a mix of the two policies, even though the report only contains a single "policy_published" element
- generate multiple reports for the same period, one for each published policy occurring during the reporting period

Such policy changes are expected to be infrequent for any given domain, whereas more stringent policy monitoring requirements on the Mail Receiver would produce a very large burden at Internet scale. Therefore, it is the responsibility of Report Consumers (i.e., vendors) and Domain Owners to be aware of this situation and expect such mixed reports during the propagation of the new policy to Mail Receivers.

3.4. Report Request Discovery

A Mail Receiver discovers reporting requests when it looks up a DMARC Policy Record that corresponds to an RFC5322.From domain on received mail. The presence of the "rua" tag specifies where to send feedback.

3.5. Report Delivery

The Mail Receiver, after preparing a report, **MUST** evaluate the provided reporting URIs (see [\[RFC9989\]](#)) in the order given. If any of the URIs are malformed, they **SHOULD** be ignored. An attempt **MUST** be made to deliver an aggregate report to every remaining URI, up to the Receiver's limits on supported URIs.

If delivery is not possible because the services advertised by the published URIs are not able to accept reports (e.g., the URI refers to a service that is unreachable), the Mail Receiver **MAY** cache that data and try again later or **MAY** discard data that could not be sent.

Where the URI specified in a "rua" tag does not specify otherwise, a Mail Receiver generating a feedback report **SHOULD** employ a secure transport mechanism, meaning the report should be delivered over a channel employing TLS (SMTP+STARTTLS).

3.5.1. Definition of Report-ID

This identifier **MUST** be unique among reports to the same domain to aid receivers in identifying duplicate reports should they happen. The Report-ID value should be constructed using the following ABNF:

```
ridfmt = (dot-atom-text ["@" dot-atom-text]) ; from RFC 5322
ridtxt = ("<" ridfmt ">") / ridfmt
```

The format specified here is not very strict, as the key goal is uniqueness. In order to create this uniqueness, the Mail Receiver may wish to use elements such as the receiving domain, the sending domain, and a timestamp in combination. An example string might be "1721054318-example.com@example.org". An alternate could use a date string such as "2024-03-27_example.com@example.org".

3.5.2. Email

The message generated by the Mail Receiver **MUST** be a [RFC5322] message formatted per [RFC2045]. The aggregate report itself **MUST** be included in one of the parts of the message, as an attachment with a corresponding media type from below. A human-readable annotation **MAY** be included as a body part (with a human-friendly content-type, such as "text/plain" or "text/html").

The aggregate data **MUST** be an XML file that **SHOULD** be subjected to GZIP [RFC1952] compression. Declining to apply compression can cause the report to be too large for a receiver to process (the total message size could exceed the receiver SMTP size limit); doing the compression increases the chances of acceptance of the report at some compute cost. The aggregate data **MUST** be present using the media type "application/gzip" if compressed (see [RFC6713]) and "text/xml" otherwise. The attachment filename **MUST** be constructed using the following ABNF:

```
filename = receiver "!" policy-domain "!" begin-timestamp
          "!" end-timestamp [ "!" unique-id ] "." extension

receiver = domain-name
          ; imported from RFC 6376

policy-domain = domain-name

begin-timestamp = 1*DIGIT
                 ; seconds since 00:00:00 UTC January 1, 1970
                 ; indicating start of the time range contained
                 ; in the report

end-timestamp = 1*DIGIT
               ; seconds since 00:00:00 UTC January 1, 1970
               ; indicating end of the time range contained
               ; in the report

unique-id = 1*(ALPHA / DIGIT)

extension = "xml" / "xml.gz"
```

The following primitive tokens that are used but otherwise unspecified are taken from "Core Rules" ([Appendix B.1](#) of [\[RFC5234\]](#)): DIGIT, ALPHA.

The extension **MUST** be "xml" for a plain XML file or "xml.gz" for an XML file compressed using GZIP.

"unique-id" allows an optional unique ID generated by the Mail Receiver to distinguish among multiple reports generated simultaneously by different sources within the same Domain Owner. A viable option may be to explore Universally Unique Identifiers (UUIDs) [\[RFC9562\]](#).

If a report generator needs to re-send a report, the system **MUST** use the same filename as the original report. This would allow the receiver to overwrite the data from the original or discard the second instance of the report.

For example, this is a sample filename for the gzip file of a report to the Domain Owner "example.com" from the Mail Receiver "mail.receiver.example":

```
mail.receiver.example!example.com!1013662812!1013749130.xml.gz
```

No specific MIME message structure is required for the message body. It is presumed that the aggregate reporting address will be equipped to extract body parts with the prescribed media type and filename and ignore the rest.

Mail streams carrying DMARC feedback data **MUST** conform to the DMARC mechanism, thereby resulting in an aligned "pass" (see [Section 4.4](#) of [\[RFC9989\]](#)). This practice minimizes the risk of Report Consumers processing fraudulent reports.

The RFC5322.Subject field for individual report submissions **MUST** conform to the following ABNF:

```
; FWS is imported from RFC 5322
dmarc-subject = %s"Report" 1*FWS %s"Domain:"
                1*FWS domain-name 1*FWS           ; policy domain
                %s"Submitter:" 1*FWS
                domain-name 1*FWS                 ; report generator
                [ %s"Report-ID:" 1*FWS ridtxt ] ; defined above
```

The first domain-name indicates the DNS domain name about which the report was generated. The second domain-name indicates the DNS domain name representing the Mail Receiver generating the report. The purpose of the Report-ID: portion of the field is to enable the Domain Owner to identify and ignore duplicate reports that might be sent by a Mail Receiver.

For instance, this is a possible Subject field for a report to the Domain Owner "example.com" from the Mail Receiver "mail.receiver.example". It is folded as allowed by [\[RFC5322\]](#):

```
Subject: Report Domain: example.com
        Submitter: mail.receiver.example
        Report-ID: <sample-ridtxt@example.com>
```

This transport mechanism potentially encounters a problem when feedback data size exceeds maximum allowable attachment sizes for either the generator or the consumer.

Optionally, the report sender **MAY** choose to use the same "ridtxt" as a part or whole of the RFC5322.Message-Id header included with the report. Doing so may help receivers distinguish when a message is a re-transmission or duplicate report.

3.5.3. Other Methods

The specification as written allows for the addition of other registered URI schemes to be supported in later versions.

3.5.4. Handling of Duplicates

There may be a situation where the report generator attempts to deliver duplicate information to the receiver. This may manifest as an exact duplicate of the report or as duplicate information between two reports. In these situations, the decision of how to handle the duplicate data lies with the receiver. As noted above, the sender **MUST** use the same unique identifiers when sending the report. This allows the receiver to better understand when duplicates happen. Here are a few options on how to handle that duplicate information:

- Reject back to sender, ideally with a permfail error noting the duplicate receipt
- Discard upon receipt
- Inspect the contents to evaluate the timestamps and reported data, act as appropriate
- Accept the duplicate data

When accepting the data, that's likely in a situation where it's not yet noticed or a one-off experience. Long-term duplicate data is not ideal. In the situation of a partial time frame overlap, there is no clear way to distinguish the impact of the overlap. The receiver would need to accept or reject the duplicate data in whole.

4. Verifying External Destinations

It is possible to specify destinations for the different reports that are outside the authority of the Domain Owner making the request. This allows domains that do not operate mail servers to request reports and have them go someplace that is able to receive and process them.

Without checks, this would allow a bad actor to publish a DMARC Policy Record that requests that reports be sent to a victim address and then send a large volume of mail that will fail both DKIM and SPF checks to a wide variety of destinations; the victim will in turn be flooded with unwanted reports. Therefore, a verification mechanism is included.

When a Mail Receiver discovers a DMARC Policy Record in the DNS, and the Organizational Domain at which that record was discovered is not identical to the Organizational Domain of the host part of the authority component of a [RFC3986] specified in the "rua" tag, the following verification steps **MUST** be taken:

1. Extract the host portion of the authority component of the URI. Call this the "destination host", as it refers to a Report Receiver.
2. Prepend the string "_report._dmarc".
3. Prepend the domain name from which the policy was retrieved, after conversion to an A-label [RFC5890] if needed.
4. If the length of the constructed name exceed DNS limits, a positive determination of the external reporting relationship cannot be made; stop.
5. Query the DNS for a TXT record at the constructed name. If the result of this request is a temporary DNS error of some kind (e.g., a timeout), the Mail Receiver **MAY** elect to temporarily fail the delivery so the verification test can be repeated later.
6. For each record returned, parse the result as a series of "tag=value" pairs, i.e., the same overall format as the DMARC Policy Record (see Section 4.7 of [RFC9989]). In particular, the "v=DMARC1" tag is mandatory and **MUST** appear first in the list. Discard any that do not pass this test. A trailing ";" is optional.
7. If the result includes no TXT resource records that pass basic parsing, a positive determination of the external reporting relationship cannot be made; stop.
8. If at least one TXT resource record remains in the set after parsing, then the external reporting arrangement was authorized by the Report Consumer.
9. If a "rua" tag is thus discovered, replace the corresponding value extracted from the domain's DMARC Policy Record with the one found in this record. This permits the Report Consumer to override the report destination. However, to prevent loops or indirect abuse, the overriding URI **MUST** use the same destination host from the first step.

For example, if the DMARC Policy Record for "blue.example.com" contained "rua=mailto:reports@red.example.net", the Organizational Domain host extracted from the latter ("red.example.net") does not match "blue.example.com", so this procedure is enacted. A TXT query for "blue.example.com._report._dmarc.red.example.net" is issued. If a single reply comes back containing a tag of "v=DMARC1", then the relationship between the two is confirmed. Moreover, "red.example.net" has the opportunity to override the report destination requested by "blue.example.com" if needed.

Where the above algorithm fails to confirm that the external reporting was authorized by the Report Consumer, the URI **MUST** be ignored by the Mail Receiver generating the report. Further, if the confirming record includes a URI whose host is again different than the domain publishing that override, the Mail Receiver generating the report **MUST NOT** generate a report to either the original or the override URI. A Report Consumer publishes such a record in its DNS if it wishes to receive reports for other domains.

A Report Consumer that is willing to receive reports for any domain can use a wildcard DNS record. For example, a TXT resource record at "*._report._dmarc.example.com" containing at least "v=DMARC1" confirms that example.com is willing to receive DMARC reports for any domain.

If the Report Consumer is overcome by volume, it can simply remove the confirming DNS record. However, due to positive caching, the change could take as long as the Time to Live (TTL) on the record to go into effect.

If the length of the DNS query is excessively long (Step 4 above), the Domain Owner may need to reconsider the domain being used to be shorter or reach out to another party that may allow for a shorter DNS label.

5. Extensible Reporting

DMARC reports allow for some extensibility, as defined by future documents that utilize DMARC as a foundation. These extensions **MUST** be properly formatted XML and meant to exist within the structure of a DMARC report. Two positions of type "<any>" are provided in the existing DMARC structure: one at file level in an "<extension>" element after "<policy_published>" and one at record level after "<auth_results>". In either case, the extensions **MUST** contain a URI to the definition of the extension so that the receiver understands how to interpret the data.

At file level:

```

<feedback xmlns="urn:ietf:params:xml:ns:dmarc-2.0"
  xmlns:ext="URI for an extension-supplied name space">
  ...
  <policy_published>
    <domain>example.com</domain>
    <p>quarantine</p>
    <sp>none</sp>
    <testing>n</testing>
  </policy_published>
  <extension>
    <ext:arc-override>never</ext:arc-override>
  </extension>

```

Within the "record" element:

```

<feedback xmlns="urn:ietf:params:xml:ns:dmarc-2.0"
  xmlns:ext="URI for an extension-supplied name space">
  ...
  <record>
    <row>
      ...
    </row>
    <identifiers>
      ...
    </identifiers>
    <auth_results>
      ...
    </auth_results>
    <ext:arc-results>
      ...
    </ext:arc-results>
  </record>
</record>
  ...

```

Here "arc-override" and "arc-results" are hypothetical element names defined in the extension's namespace.

Extension elements are optional. Any number of extensions is allowed. If a processor is unable to handle an extension in a report, it **SHOULD** ignore the data and continue to the next extension.

6. IANA Considerations

This document uses URNs to describe XML namespaces and XML schemas conforming to a registry mechanism described in [RFC3688]. Two URI assignments have been registered by the IANA.

6.1. Registration for the DMARC Namespace

IANA has registered the following URI in the "ns" registry within the "IETF XML Registry" group:

URI: urn:ietf:params:xml:ns:dmarc-2.0
Registrant Contact: The IESG (iesg@ietf.org)
XML: N/A; the requested URI is an XML namespace.

6.2. Registration for the DMARC XML Schema

IANA has registered the following URI in the "schema" registry within the "IETF XML Registry" group:

URI: urn:ietf:params:xml:schema:dmarc-2.0
Registrant Contact: The IESG (iesg@ietf.org)
XML: See the DMARC XML schema [[W3C.REC-xmlschema-1](#)] [[W3C.REC-xmlschema-2](#)] in [Appendix A](#) of this document.

7. Privacy Considerations

This section discusses exposure related to DMARC aggregate reporting.

7.1. Report Recipients

A DMARC Policy Record can specify that reports should be sent to an intermediary operating on behalf of the Domain Owner. This is done when the Domain Owner contracts with an entity to monitor mail streams for abuse and performance issues. Receipt by third parties of such data may or may not be permitted by the Mail Receiver's privacy policy, terms of use, or other similar governing document. Domain Owners and Mail Receivers should both review and understand if their own internal policies constrain the use and transmission of DMARC reporting.

Some potential exists for report recipients to perform traffic analysis, making it possible to obtain metadata about the Receiver's traffic. In addition to verifying compliance with policies, Receivers need to consider that before sending reports to a third party.

7.2. Data Contained Within Reports

Aggregate feedback reports contain aggregated data relating to messages purportedly originating from the Domain Owner. The data does not contain any identifying characteristics about individual users. No personal information such as individual mail addresses, IP addresses of individuals, or the content of any messages is included in reports.

Mail Receivers should have no concerns in sending reports, as they do not contain personal information. In all cases, the data within the reports relates to the domain-level authentication information provided by mail servers sending messages on behalf of the Domain Owner. This information is necessary to assist Domain Owners in implementing and maintaining DMARC.

Domain Owners should have no concerns in receiving reports, as they do not contain personal information. The reports only contain aggregated data related to the domain-level authentication details of messages claiming to originate from their domain. This information is

essential for the proper implementation and operation of DMARC. Domain Owners who are unable to receive reports for organizational reasons can choose to exclusively direct the reports to an external processor.

7.3. Feedback Leakage

Providing feedback reporting to Public Suffix Operators (PSOs) for a Public Suffix Domain (PSD) [RFC9989] can, in some cases, cause information to leak out of an organization to the PSO. This leakage could potentially be utilized as part of a program of pervasive surveillance (see [RFC7624]). There are roughly three cases to consider:

Single Organization PSDs (e.g., ".mil"):

Aggregate reports based on PSD DMARC have the potential to contain information about mail related to entities managed by the organization. Since both the PSO and the Organizational Domain Owners are common, there is no additional privacy risk for either normal or non-existent domain reporting due to PSD DMARC.

Multi-organization PSDs requiring DMARC usage (e.g., ".bank"):

Aggregate reports based on PSD DMARC will only be generated for domains that do not publish a DMARC Policy Record at the Organizational Domain or host level. For domains that do publish the required DMARC Policy Records, the feedback reporting addresses of the Organizational Domain (or hosts) will be used. The only direct risk of feedback leakage for these PSDs are for Organizational Domains that are out of compliance with PSD policy. Data on non-existent domains would be sent to the PSO.

Multi-organization PSDs not requiring DMARC usage (e.g., ".com"):

Privacy risks for Organizational Domains that have not deployed DMARC within such PSDs can be significant. For non-DMARC Organizational Domains, all DMARC feedback will be directed to the PSO if that PSO itself has a DMARC Policy Record that specifies a "rua" tag. Any non-DMARC Organizational Domain would have its feedback reports redirected to the PSO. The content of such reports, particularly for existing domains, is privacy sensitive.

PSOs will receive feedback on non-existent domains, which may be similar to existing Organizational Domains. Feedback related to such domains have a small risk of carrying information related to an actual Organizational Domain. To minimize this potential concern, PSD DMARC feedback **MUST** be limited to aggregate reports. Failure reports carry more detailed information and present a greater risk.

8. Security Considerations

While reviewing this document and its security considerations, it is ideal that the reader also review the Privacy Considerations section above, as well as the privacy considerations and security considerations in Sections 10 and 11 of [RFC9989].

8.1. Report Content as an Attack

Aggregate reports are supposed to be processed automatically. An attacker might attempt to compromise the integrity or availability of the report processor by sending malformed reports. In particular, the archive decompressor and XML parser are at risk to resource exhaustion attacks (zip bomb or XML bomb).

8.2. False Information

The data contained within aggregate reports may be forged. An attacker might attempt to interfere with or influence policy decisions by submitting false reports in large volume. The attacker could also be attempting to influence platform architecture decisions. A volume-based attack may also impact the ability for a Report Receiver to accept reports from other entities.

8.3. Disclosure of Filtering Information

While not specified in this document itself, the availability of extensions could enable the report generator to disclose information about message placement (Inbox/Spam/etc.). This is very much discouraged as it could relay this information to a malicious party, allowing them to understand more about filtering methodologies at a receiving entity.

9. Operational Considerations

9.1. Report Generation

- The error fields should be reasonably terse and usable.
- If reports cannot be generated, the system should ideally log a useful error that helps troubleshoot the issue.

9.2. Report Evaluation

As noted above, if a report does not match the specified format, the evaluator will likely find the contents to be in question. Alternately, the evaluator may decide to sideline those reports so they can more easily collaborate with the report generator to identify where the issues are happening.

It's quite likely that the data contained within the reports will be extracted and stored in a system that allows for easy reporting, dashboarding, and/or monitoring. The XML reports themselves are not human readable in bulk, and a system such as the above may aid the Domain Owner with identifying issues.

9.3. Report Storage

Once a report is accepted and properly parsed by the report evaluator, it is entirely up to that evaluator as to what they wish to do with the XML documents. For some domains, the quantity of reports could be fairly high, or the size of the reports themselves could be large. Once the data from the reports has been extracted and indexed, the reports seemingly have little value in most situations.

10. References

10.1. Normative References

- [RFC1952] Deutsch, P., "GZIP file format specification version 4.3", RFC 1952, DOI 10.17487/RFC1952, May 1996, <<https://www.rfc-editor.org/info/rfc1952>>.
- [RFC2045] Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies", RFC 2045, DOI 10.17487/RFC2045, November 1996, <<https://www.rfc-editor.org/info/rfc2045>>.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <<https://www.rfc-editor.org/info/rfc3688>>.
- [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>>.
- [RFC5234] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, DOI 10.17487/RFC5234, 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", RFC 5322, DOI 10.17487/RFC5322, 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>>.

-
- [RFC6713] Levine, J., "The 'application/zlib' and 'application/gzip' Media Types", RFC 6713, DOI 10.17487/RFC6713, August 2012, <<https://www.rfc-editor.org/info/rfc6713>>.
- [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>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8601] Kucherawy, M., "Message Header Field for Indicating Message Authentication Status", RFC 8601, DOI 10.17487/RFC8601, May 2019, <<https://www.rfc-editor.org/info/rfc8601>>.
- [RFC9989] Herr, T., Ed. and J. R. Levine, Ed., "Domain-Based Message Authentication, Reporting, and Conformance (DMARC)", RFC 9989, DOI 10.17487/RFC9989, May 2026, <<https://www.rfc-editor.org/info/rfc9989>>.
- [W3C.REC-xmlschema-1] Thompson, H. S., Ed., Beech, D., Ed., Maloney, M., Ed., and N. Mendelsohn, Ed., "XML Schema Part 1: Structures Second Edition", W3C Recommendation, 28 October 2004, <<https://www.w3.org/TR/2004/REC-xmlschema-1-20041028/>>. Latest version available at <https://www.w3.org/TR/xmlschema-1/>.
- [W3C.REC-xmlschema-2] Biron, P. V., Ed. and A. Malhotra, Ed., "XML Schema Part 2: Datatypes Second Edition", W3C Recommendation, 28 October 2004, <<https://www.w3.org/TR/2004/REC-xmlschema-2-20041028/>>. Latest version available at <https://www.w3.org/TR/xmlschema-2/>.

10.2. Informative References

- [RFC5598] Crocker, D., "Internet Mail Architecture", RFC 5598, DOI 10.17487/RFC5598, July 2009, <<https://www.rfc-editor.org/info/rfc5598>>.
- [RFC7624] Barnes, R., Schneier, B., Jennings, C., Hardie, T., Trammell, B., Huitema, C., and D. Borkmann, "Confidentiality in the Face of Pervasive Surveillance: A Threat Model and Problem Statement", RFC 7624, DOI 10.17487/RFC7624, August 2015, <<https://www.rfc-editor.org/info/rfc7624>>.
- [RFC9562] Davis, K., Peabody, B., and P. Leach, "Universally Unique Identifiers (UUIDs)", RFC 9562, DOI 10.17487/RFC9562, May 2024, <<https://www.rfc-editor.org/info/rfc9562>>.

[RFC9991] Jones, S. M., Ed. and A. Vesely, Ed., "Domain-Based Message Authentication, Reporting, and Conformance (DMARC) Failure Reporting", RFC 9991, DOI 10.17487/RFC9991, May 2026, <<https://www.rfc-editor.org/info/rfc9991>>.

Appendix A. DMARC XML Schema

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  targetNamespace="urn:ietf:params:xml:ns:dmarc-2.0"
  xmlns="urn:ietf:params:xml:ns:dmarc-2.0"
  elementFormDefault="qualified">

  <!-- Elements with an optional "lang" attribute. -->
  <xs:complexType name="langAttrString">
    <xs:simpleContent>
      <xs:extension base="xs:string">
        <xs:attribute name="lang" type="xs:language"
          use="optional" default="en"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>

  <!-- The time range in UTC defining the reporting period of
    this report, specified in seconds since epoch. -->
  <xs:complexType name="DateRangeType">
    <xs:all>
      <xs:element name="begin" type="xs:integer"
        minOccurs="1" maxOccurs="1"/>
      <xs:element name="end" type="xs:integer"
        minOccurs="1" maxOccurs="1"/>
    </xs:all>
  </xs:complexType>

  <!-- Report generator metadata. -->
  <xs:complexType name="ReportMetadataType">
    <xs:all>
      <!-- Reporting Organization -->
      <xs:element name="org_name" type="xs:string"
        minOccurs="1" maxOccurs="1"/>
      <!-- Contact to use when contacting the Reporting Organization -->
      <xs:element name="email" type="xs:string"
        minOccurs="1" maxOccurs="1"/>
      <!-- Additional contact details -->
      <xs:element name="extra_contact_info" type="langAttrString"
        minOccurs="0" maxOccurs="1"/>
      <!-- Unique Report-ID -->
      <xs:element name="report_id" type="xs:string"
        minOccurs="1" maxOccurs="1"/>
      <!-- Timestamps used when forming report data -->
      <xs:element name="date_range" type="DateRangeType"
        minOccurs="1" maxOccurs="1"/>
      <!-- Optional error messages when processing DMARC policy -->
      <xs:element name="error" type="langAttrString"
        minOccurs="0" maxOccurs="1"/>
      <!-- Optional information about the generating software -->
```

```
<xs:element name="generator" type="xs:string"
  minOccurs="0" maxOccurs="1"/>
</xs:all>
</xs:complexType>

<!-- Alignment mode (relaxed or strict) for DKIM and SPF. -->
<xs:simpleType name="AlignmentType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="r"/>
    <xs:enumeration value="s"/>
  </xs:restriction>
</xs:simpleType>

<!-- The policy actions specified by p, sp, and np in the
  DMARC Policy Record. -->
<xs:simpleType name="DispositionType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="none"/>
    <xs:enumeration value="quarantine"/>
    <xs:enumeration value="reject"/>
  </xs:restriction>
</xs:simpleType>

<!-- The policy actions utilized on messages for this record. -->
<!--
  "none": No action taken
  "pass": No action, passing DMARC w/enforcing policy
  "quarantine": Failed DMARC, message marked for quarantine
  "reject": Failed DMARC, marked as reject
-->
<xs:simpleType name="ActionDispositionType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="none"/>
    <xs:enumeration value="pass"/>
    <xs:enumeration value="quarantine"/>
    <xs:enumeration value="reject"/>
  </xs:restriction>
</xs:simpleType>

<!-- The method used to discover the DMARC Policy Record used during
  evaluation. The available values are "psl" and "treewalk",
  where "psl" is the method from RFC 7489 and "treewalk" is
  described in RFC 9989. -->
<xs:simpleType name="DiscoveryType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="psl"/>
    <xs:enumeration value="treewalk"/>
  </xs:restriction>
</xs:simpleType>

<!-- The published DMARC policy. Unspecified tags have their
  default values. -->
<xs:complexType name="PolicyPublishedType">
  <xs:all>
    <!-- The domain at which the DMARC record was found. -->
    <xs:element name="domain" type="xs:string"
      minOccurs="1" maxOccurs="1"/>
    <!-- The policy published for messages from: -->
```

```

<!-- * the domain. -->
<xs:element name="p" type="DispositionType"
  minOccurs="1" maxOccurs="1"/>
<!-- * subdomains. -->
<xs:element name="sp" type="DispositionType"
  minOccurs="0" maxOccurs="1"/>
<!-- * non-existent subdomains. -->
<xs:element name="np" type="DispositionType"
  minOccurs="0" maxOccurs="1"/>
<!-- The DKIM alignment mode. -->
<xs:element name="adkim" type="AlignmentType"
  minOccurs="0" maxOccurs="1"/>
<!-- The SPF alignment mode. -->
<xs:element name="aspf" type="AlignmentType"
  minOccurs="0" maxOccurs="1"/>
<!-- Method used to find/obtain DMARC policy. -->
<xs:element name="discovery_method" type="DiscoveryType"
  minOccurs="0" maxOccurs="1"/>
<!-- Failure reporting options in effect. -->
<xs:element name="fo" type="xs:string"
  minOccurs="0" maxOccurs="1"/>
<!-- Whether testing mode was declared in the DMARC Record. -->
<xs:element name="testing" type="TestingType"
  minOccurs="0" maxOccurs="1"/>
</xs:all>
</xs:complexType>

<!-- Values for Testing mode attached to policy. -->
<xs:simpleType name="TestingType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="n"/>
    <xs:enumeration value="y"/>
  </xs:restriction>
</xs:simpleType>

<!-- The DMARC-aligned authentication result. -->
<xs:simpleType name="DMARCResultType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="pass"/>
    <xs:enumeration value="fail"/>
  </xs:restriction>
</xs:simpleType>

<!-- Reasons that may affect DMARC disposition or execution. -->
<xs:simpleType name="PolicyOverrideType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="local_policy"/>
    <xs:enumeration value="mailing_list"/>
    <xs:enumeration value="other"/>
    <xs:enumeration value="policy_test_mode"/>
    <xs:enumeration value="trusted_forwarder"/>
  </xs:restriction>
</xs:simpleType>

<!-- Override reason consists of pre-defined override type and
  free-text comment. -->
<xs:complexType name="PolicyOverrideReason">
  <xs:all>

```

```
<xs:element name="type" type="PolicyOverrideType"
  minOccurs="1" maxOccurs="1"/>
<xs:element name="comment" type="langAttrString"
  minOccurs="0" maxOccurs="1"/>
</xs:all>
</xs:complexType>

<!-- Taking into account everything else in the record,
  the results of applying DMARC. If alignment fails
  and the policy applied does not match the domain's
  configured policy, the reason element MUST be specified. -->
<xs:complexType name="PolicyEvaluatedType">
  <xs:sequence>
    <xs:element name="disposition" type="ActionDispositionType"
      minOccurs="1" maxOccurs="1"/>
    <xs:element name="dkim" type="DMARCResultType"
      minOccurs="1" maxOccurs="1"/>
    <xs:element name="spf" type="DMARCResultType"
      minOccurs="1" maxOccurs="1"/>
    <xs:element name="reason" type="PolicyOverrideReason"
      minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="RowType">
  <xs:all>
    <!-- The connecting IP. IPv4address or IPv6address
      as defined in RFC 3986, Section 3.2.2. -->
    <xs:element name="source_ip" type="xs:string"
      minOccurs="1" maxOccurs="1"/>
    <!-- The number of messages for which the
      PolicyEvaluatedType was applied. -->
    <xs:element name="count" type="xs:integer"
      minOccurs="1" maxOccurs="1"/>
    <!-- The DMARC disposition applied to matching messages. -->
    <xs:element name="policy_evaluated" type="PolicyEvaluatedType"
      minOccurs="1" maxOccurs="1"/>
  </xs:all>
</xs:complexType>

<xs:complexType name="IdentifierType">
  <xs:all>
    <!-- The RFC5322.From domain. -->
    <xs:element name="header_from" type="xs:string"
      minOccurs="1" maxOccurs="1"/>
    <!-- The RFC5321.MailFrom domain. -->
    <xs:element name="envelope_from" type="xs:string"
      minOccurs="0" maxOccurs="1"/>
    <!-- The envelope recipient domain. -->
    <xs:element name="envelope_to" type="xs:string"
      minOccurs="0" maxOccurs="1"/>
  </xs:all>
</xs:complexType>

<!-- DKIM verification result; see RFC 8601, Section 2.7.1. -->
<xs:simpleType name="DKIMResultType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="none"/>
  </xs:restriction>
</xs:simpleType>
```

```

<xs:enumeration value="pass"/>
<xs:enumeration value="fail"/>
<xs:enumeration value="policy"/>
<xs:enumeration value="neutral"/>
<xs:enumeration value="temperror"/>
<xs:enumeration value="permerror"/>
</xs:restriction>
</xs:simpleType>

<xs:complexType name="DKIMAuthResultType">
  <xs:all>
    <!-- The "d=" tag in the signature. -->
    <xs:element name="domain" type="xs:string"
      minOccurs="1" maxOccurs="1"/>
    <!-- The "s=" tag in the signature. -->
    <xs:element name="selector" type="xs:string"
      minOccurs="1" maxOccurs="1"/>
    <!-- The DKIM verification result. -->
    <xs:element name="result" type="DKIMResultType"
      minOccurs="1" maxOccurs="1"/>
    <!-- Any extra information (e.g., from Authentication-Results). -->
    <xs:element name="human_result" type="langAttrString"
      minOccurs="0" maxOccurs="1"/>
  </xs:all>
</xs:complexType>

<!-- SPF domain scope. -->
<xs:simpleType name="SPFDomainScope">
  <xs:restriction base="xs:string">
    <xs:enumeration value="mfrom"/>
  </xs:restriction>
</xs:simpleType>

<!-- SPF verification result; see RFC 8601, Section 2.7.2. -->
<xs:simpleType name="SPFResultType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="none"/>
    <xs:enumeration value="pass"/>
    <xs:enumeration value="fail"/>
    <xs:enumeration value="softfail"/>
    <xs:enumeration value="policy"/>
    <xs:enumeration value="neutral"/>
    <xs:enumeration value="temperror"/>
    <xs:enumeration value="permerror"/>
  </xs:restriction>
</xs:simpleType>

<xs:complexType name="SPFAuthResultType">
  <xs:all>
    <!-- The checked domain. -->
    <xs:element name="domain" type="xs:string"
      minOccurs="1" maxOccurs="1"/>
    <!-- The scope of the checked domain. -->
    <xs:element name="scope" type="SPFDomainScope"
      minOccurs="0" maxOccurs="1"/>
    <!-- The SPF verification result. -->
    <xs:element name="result" type="SPFResultType"
      minOccurs="1" maxOccurs="1"/>
  </xs:all>
</xs:complexType>

```

```

    <!-- Any extra information (e.g., from Authentication-Results).
         The information in the field below should be for a
         person to be provided with additional information
         that may be useful when debugging SPF authentication
         issues. This could include broken records, invalid
         DNS responses, etc. -->
    <xs:element name="human_result" type="langAttrString"
        minOccurs="0" maxOccurs="1"/>
</xs:all>
</xs:complexType>

<!-- This element contains DKIM and SPF results, uninterpreted
     with respect to DMARC. -->
<xs:complexType name="AuthResultType">
  <xs:sequence>
    <!-- There may be zero or more DKIM signatures. -->
    <xs:element name="dkim" type="DKIMAuthResultType"
        minOccurs="0" maxOccurs="unbounded"/>
    <!-- There may be zero or one SPF result. -->
    <xs:element name="spf" type="SPFAuthResultType"
        minOccurs="0" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>

<!-- This element contains all the authentication results that
     were evaluated by the receiving system for the given set of
     messages. -->
<xs:complexType name="RecordType">
  <xs:sequence>
    <xs:element name="row" type="RowType"
        minOccurs="1" maxOccurs="1"/>
    <xs:element name="identifiers" type="IdentifierType"
        minOccurs="1" maxOccurs="1"/>
    <xs:element name="auth_results" type="AuthResultType"
        minOccurs="1" maxOccurs="1"/>
    <!-- Extension at record level -->
    <xs:any processContents="lax"
        minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="ExtensionType">
  <xs:sequence>
    <xs:any processContents="lax"
        minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

<!-- Parent -->
<xs:element name="feedback">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="version" type="xs:decimal"
          minOccurs="0" maxOccurs="1"/>
      <xs:element name="report_metadata" type="ReportMetadataType"
          minOccurs="1" maxOccurs="1"/>
      <xs:element name="policy_published" type="PolicyPublishedType"
          minOccurs="1" maxOccurs="1"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

```
<!-- Extension at top level -->
<xs:element name="extension" type="ExtensionType"
  minOccurs="0" maxOccurs="1"/>
<!-- One record per (IP, result, IDs Auths) tuples -->
<xs:element name="record" type="RecordType"
  minOccurs="1" maxOccurs="unbounded"/>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>
```

Appendix B. Sample Report

```
<feedback xmlns="urn:ietf:params:xml:ns:dmarc-2.0">
  <version>1.0</version>
  <report_metadata>
    <org_name>Sample Reporter</org_name>
    <email>report_sender@example-reporter.com</email>
    <extra_contact_info>...</extra_contact_info>
    <report_id>3v98abbb8ya9n3va8yr8oa3ya</report_id>
    <date_range>
      <begin>302832000</begin>
      <end>302918399</end>
    </date_range>
    <generator>Example DMARC Aggregate Reporter v1.2</generator>
  </report_metadata>
  <policy_published>
    <domain>example.com</domain>
    <p>quarantine</p>
    <sp>none</sp>
    <np>none</np>
    <testing>n</testing>
    <discovery_method>treewalk</discovery_method>
  </policy_published>
  <record>
    <row>
      <source_ip>192.0.2.123</source_ip>
      <count>123</count>
      <policy_evaluated>
        <disposition>pass</disposition>
        <dkim>pass</dkim>
        <spf>fail</spf>
      </policy_evaluated>
    </row>
    <identifiers>
      <envelope_from>example.com</envelope_from>
      <header_from>example.com</header_from>
    </identifiers>
    <auth_results>
      <dkim>
        <domain>example.com</domain>
        <result>pass</result>
        <selector>abc123</selector>
      </dkim>
      <spf>
        <domain>example.com</domain>
        <result>fail</result>
      </spf>
    </auth_results>
  </record>
</feedback>
```

Appendix C. Differences from RFC 7489

Here is a bulleted list of some of the more noticeable/important differences between DMARC [RFC7489] and this document:

- Many elements of the defining XSD have been clarified, which means the structure of the report should be more consistent
- The report identifier has more structure
- Clarification provided about the number of domains to be addressed per report
- The addition of extensions as part of the report structure
- PSD is now included as part of the specification
- Selector is now required when reporting a DKIM signature

Furthermore, the original DMARC specification was contained within a single document: [RFC7489]. The original document has been split into three documents: [RFC9989], this document, and [RFC9991]. This allows these pieces to potentially be altered in the future without re-opening the entire document, as well as allowing them to move through the IETF process independently.

Acknowledgements

Many thanks are deserved to those that helped create this document. Much of the content was created from [RFC7489] and has now been updated to be more clear and correct some outstanding issues. The IETF DMARC Working Group has spent much time working to finalize this effort, and significant contributions were made by Seth Blank, Todd Herr, Steve Jones, Scott Kitterman, Murray S. Kucherawy, Daniel Kvål, Barry Leiba, John Levine, Martijn van der Lee, Alessandro Veseley, and Matthäus Wander.

Author's Address

Alex Brotman (EDITOR)

Comcast, Inc.

Email: alex_brotman@comcast.com