DMARC
Internet-Draft
Obsoletes: 7489 (if approved)
Intended status: Standards Track
Expires: May 15, 2021

A. Brotman (ed)
Comcast, Inc.

November 11, 2020

DMARC Aggregate Reporting
draft-ietf-dmarc-aggregate-reporting-00

Abstract

DMARC allows for domain holders 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 to the domain holder's specified destination as supported by the receiver.

This document (along with others) obsoletes RFC7489.

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 May 15, 2021.

include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction

A key component of DMARC is the ability for domain holders to request that receivers provide various types of reports. These reports allow domain holders 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 SPF [RFC7208] & DKIM [RFC6376] validation.

1.1. Terminology

The keywords MUST, MUST NOT, REQUIRED, SHALL, SHALL NOT, SHOULD, SHOULD NOT, RECOMMENDED, MAY, and OPTIONAL, when they appear in this document, are to be interpreted as described in [RFC2119].

2. 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.

2.1. Verifying External Destinations

[<https://trac.ietf.org/trac/dmarc/ticket/76>]

2.2. 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 email streams processed by Mail Receivers.

Aggregate DMARC feedback provides visibility into real-world email streams that Domain Owners need to make informed decisions regarding the publication of 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) Mail Receiver-originated feedback reports 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.

The format for these reports is defined in Appendix C.

The report SHOULD 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 alignment

- Data for each Domain Owner's subdomain separately from mail from the sender's Organizational Domain, even if there is no explicit subdomain policy

- Sending and receiving domains

- The policy requested by the Domain Owner and the policy actually applied (if different)

- 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

Note that Domain Owners or their agents may change the published DMARC policy 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;

- generate a report whose end time occurs when the updated policy was detected, regardless of any requested report interval.

The report SHOULD 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 alignment

- Data for each Domain Owner's subdomain separately from mail from the sender's Organizational Domain, even if there is no explicit subdomain policy

- Sending and receiving domains

- The policy requested by the Domain Owner and the policy actually applied (if different)

- 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

Note that Domain Owners or their agents may change the published DMARC policy 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;

- generate a report whose end time occurs when the updated policy was detected, regardless of any requested report interval.

2.2.1. Transport

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.

The Mail Receiver, after preparing a report, MUST evaluate the provided reporting URIs in the order given. Any reporting URI that includes a size limitation exceeded by the generated report (after compression and after any encoding required by the particular transport mechanism) MUST NOT be used. An attempt MUST be made to deliver an aggregate report to every remaining URI, up to the Receiver's limits on supported URIs.

If transport 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, or all provided URIs specify size limits exceeded by the generated record), the Mail Receiver SHOULD send a short report (see Section 7.2.2) indicating that a report is available but could not be sent. The Mail Receiver MAY cache that data and try again later, or MAY discard data that could not be sent.

2.2.1.1. Email

The message generated by the Mail Receiver MUST be a [MAIL] message formatted per [MIME]. The aggregate report itself MUST be included in one of the parts of the message. A human-readable portion MAY be included as a MIME part (such as a text/plain part).

The aggregate data MUST be an XML file that SHOULD be subjected to GZIP compression. Declining to apply compression can cause the report to be too large for a receiver to process (a commonly observed receiver limit is ten megabytes); doing the compression increases the chances of acceptance of the report at some compute cost. The aggregate data SHOULD be present using the media type "application/gzip" if compressed (see [GZIP]), and "text/xml" otherwise. The filename is typically constructed using the following ABNF:

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

unique-id = 1*(ALPHA / DIGIT)

receiver = domain
           ; imported from [MAIL]

policy-domain   = domain

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

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

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.

For example, this is a possible 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.gz
```

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

Email streams carrying DMARC feedback data MUST conform to the DMARC mechanism, thereby resulting in an aligned "pass" (see Section 3.1). This practice minimizes the risk of report consumers processing fraudulent reports.

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

```
dmarc-subject = %x52.65.70.6f.72.74 1*FWS       ; "Report"
                %x44.6f.6d.61.69.6e.3a 1*FWS    ; "Domain:"
                domain-name 1*FWS               ; from RFC 6376
                %x53.75.62.6d.69.74.74.65.72.3a ; "Submitter:"
                1*FWS domain-name 1*FWS
                %x52.65.70.6f.72.74.2d.49.44.3a ; "Report-ID:"
                msg-id                          ; from RFC 5322
```

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 line-wrapped as allowed by [MAIL]:

```
Subject: Report Domain: example.com
    Submitter: mail.receiver.example
    Report-ID: <2002.02.15.1>
```

This transport mechanism potentially encounters a problem when feedback data size exceeds maximum allowable attachment sizes for

either the generator or the consumer. See Section 7.2.2 for further discussion.

2.2.1.2. Other Methods

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

3. Security Considerations

TBD

4. IANA Considerations

TBD

5. Appendix A. DMARC XML Schema

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://dmarc.org/dmarc-xml/0.1">

<!-- The time range in UTC covered by messages in this report,
     specified in seconds since epoch. -->
<xs:complexType name="DateRangeType">
  <xs:all>
    <xs:element name="begin" type="xs:integer"/>
    <xs:element name="end" type="xs:integer"/>
  </xs:all>
</xs:complexType>

<!-- Report generator metadata. -->
<xs:complexType name="ReportMetadataType">
  <xs:sequence>
    <xs:element name="org_name" type="xs:string"/>
    <xs:element name="email" type="xs:string"/>
    <xs:element name="extra_contact_info" type="xs:string"
                minOccurs="0"/>
    <xs:element name="report_id" type="xs:string"/>
    <xs:element name="date_range" type="DateRangeType"/>
    <xs:element name="error" type="xs:string" minOccurs="0"
                maxOccurs="unbounded"/>
  </xs:sequence>
</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 and sp in the
         DMARC 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 DMARC policy that applied to the messages in
         this report. -->
    <xs:complexType name="PolicyPublishedType">
      <xs:all>
        <!-- The domain at which the DMARC record was found. -->
        <xs:element name="domain" type="xs:string"/>
        <!-- The DKIM alignment mode. -->
        <xs:element name="adkim" type="AlignmentType"
                    minOccurs="0"/>
        <!-- The SPF alignment mode. -->
        <xs:element name="aspf" type="AlignmentType"
                    minOccurs="0"/>
        <!-- The policy to apply to messages from the domain. -->
        <xs:element name="p" type="DispositionType"/>
        <!-- The policy to apply to messages from subdomains. -->
        <xs:element name="sp" type="DispositionType"/>
        <!-- The percent of messages to which policy applies. -->
        <xs:element name="pct" type="xs:integer"/>
        <!-- Failure reporting options in effect. -->
        <xs:element name="fo" type="xs:string"/>
      </xs:all>
    </xs:complexType>

    <!-- 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
         thereof. -->
```

```
    <xs:simpleType name="PolicyOverrideType">
      <xs:restriction base="xs:string">
        <xs:enumeration value="forwarded"/>
        <xs:enumeration value="sampled_out"/>
        <xs:enumeration value="trusted_forwarder"/>
        <xs:enumeration value="mailing_list"/>
        <xs:enumeration value="local_policy"/>
        <xs:enumeration value="other"/>
      </xs:restriction>
    </xs:simpleType>

    <!-- How do we allow report generators to include new
         classes of override reasons if they want to be more
         specific than "other"? -->
    <xs:complexType name="PolicyOverrideReason">
      <xs:all>
        <xs:element name="type" type="PolicyOverrideType"/>
        <xs:element name="comment" type="xs:string"
                    minOccurs="0"/>
      </xs:all>
    </xs:complexType>

    <!-- Taking into account everything else in the record,
         the results of applying DMARC. -->
    <xs:complexType name="PolicyEvaluatedType">
      <xs:sequence>
        <xs:element name="disposition" type="DispositionType"/>
        <xs:element name="dkim" type="DMARCResultType"/>
        <xs:element name="spf" type="DMARCResultType"/>
        <xs:element name="reason" type="PolicyOverrideReason"
                    minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:complexType>

    <!-- Credit to Roger L. Costello for IPv4 regex
         http://mailman.ic.ac.uk/pipermail/xml-dev/1999-December/
              018018.html -->
    <!-- Credit to java2s.com for IPv6 regex
         http://www.java2s.com/Code/XML/XML-Schema/
              IPv6addressesareeasiertodescribeusingasimpleregex.htm -->
    <xs:simpleType name="IPAddress">
      <xs:restriction base="xs:string">
        <xs:pattern value="((1?[0-9]?[0-9]|2[0-4][0-9]|25[0-5]).){3}
                    (1?[0-9]?[0-9]|2[0-4][0-9]|25[0-5])|
                    ([A-Fa-f0-9]{1,4}:){7}[A-Fa-f0-9]{1,4}"/>
      </xs:restriction>
    </xs:simpleType>
```

```
    <xs:complexType name="RowType">
      <xs:all>
        <!-- The connecting IP. -->
        <xs:element name="source_ip" type="IPAddress"/>
        <!-- The number of matching messages. -->
        <xs:element name="count" type="xs:integer"/>
        <!-- The DMARC disposition applying to matching
             messages. -->
        <xs:element name="policy_evaluated"
                    type="PolicyEvaluatedType"
                    minOccurs="1"/>
      </xs:all>
    </xs:complexType>

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

    <!-- DKIM verification result, according to RFC 7001
         Section 2.6.1. -->
    <xs:simpleType name="DKIMResultType">
      <xs:restriction base="xs:string">
        <xs:enumeration value="none"/>
        <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=" parameter in the signature. -->
        <xs:element name="domain" type="xs:string"
                    minOccurs="1"/>
        <!-- The "s=" parameter in the signature. -->
```

```
        <xs:element name="selector" type="xs:string"
                    minOccurs="0"/>
        <!-- The DKIM verification result. -->
        <xs:element name="result" type="DKIMResultType"
                    minOccurs="1"/>
        <!-- Any extra information (e.g., from
             Authentication-Results). -->
        <xs:element name="human_result" type="xs:string"
                    minOccurs="0"/>
      </xs:all>
    </xs:complexType>

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

    <!-- SPF result. -->
    <xs:simpleType name="SPFResultType">
      <xs:restriction base="xs:string">
        <xs:enumeration value="none"/>
        <xs:enumeration value="neutral"/>
        <xs:enumeration value="pass"/>
        <xs:enumeration value="fail"/>
        <xs:enumeration value="softfail"/>
        <!-- "TempError" commonly implemented as "unknown". -->
        <xs:enumeration value="temperror"/>
        <!-- "PermError" commonly implemented as "error". -->
        <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"/>
        <!-- The scope of the checked domain. -->
        <xs:element name="scope" type="SPFDomainScope" minOccurs="1"/>
        <!-- The SPF verification result. -->
        <xs:element name="result" type="SPFResultType"
                    minOccurs="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 no DKIM signatures, or multiple DKIM
             signatures. -->
        <xs:element name="dkim" type="DKIMAuthResultType"
          minOccurs="0" maxOccurs="unbounded"/>
        <!-- There will always be at least one SPF result. -->
        <xs:element name="spf" type="SPFAuthResultType" minOccurs="1"
                    maxOccurs="unbounded"/>
      </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"/>
        <xs:element name="identifiers" type="IdentifierType"/>
        <xs:element name="auth_results" type="AuthResultType"/>
      </xs:sequence>
    </xs:complexType>

    <!-- Parent -->
    <xs:element name="feedback">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="version"
                      type="xs:decimal"/>
          <xs:element name="report_metadata"
                      type="ReportMetadataType"/>
          <xs:element name="policy_published"
                      type="PolicyPublishedType"/>
          <xs:element name="record" type="RecordType"
                      maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    </xs:schema>
```

6. References

6.1. Normative References

[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>.

[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>.

6.2. Informative References

[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>.

Author's Address

Alex Brotman
Comcast, Inc.

Email: alex_brotman@comcast.com