                        DMARC Aggregate Reporting
                    draft-ietf-dmarc-aggregate-reporting-01

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 25 August 2021.

Copyright Notice

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

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 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 [URI] specified in the "rua" or "ruf" tag, the following verification steps are to 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 if needed.

4. 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.

5. For each record returned, parse the result as a series of
   "tag=value" pairs, i.e., the same overall format as the policy
   record (see Section 6.4). In particular, the "v=DMARC1" tag is
   mandatory and MUST appear first in the list. Discard any that do
   not pass this test.

6. If the result includes no TXT resource records that pass basic
   parsing, a positive determination of the external reporting
   relationship cannot be made; stop.

7. If at least one TXT resource record remains in the set after
   parsing, then the external reporting arrangement was authorized
   by the Report Receiver.

8. If a "rua" or "ruf" 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 Receiver 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 a DMARC policy query for "blue.example.com" contained
"rua=mailto:reports@red.example.net", the 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 Receiver, 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 Receiver publishes such a record in its DNS
if it wishes to receive reports for other domains.

A Report Receiver 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 Receiver 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.

   A Mail Receiver might decide not to enact this procedure if, for
   example, it relies on a local list of domains for which external
   reporting addresses are permitted.

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 A.

   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

   * A separate report should be generated for each 5322.From
      subdomain, regardless of which policy domain was used during
      receipt of messages

   * 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

   ProposedAddition: [[

   DMARC Aggregate Reports MUST contain two primary sections; one
   consisting of descriptive information and the other a set of IP-
   focused row-based data. Each report MUST contain data for only one
   Author Domain. A single report SHOULD 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.

   The informative section MUST contain two sub-sections. One will be
   the metadata section which MUST contain the fields related to
   "org_name", "email", "report_id", and "date_range". Optional fields
   MAY include "extra_contact_info" and an "error" field. The
   "date_range" section which will note "begin" and "end" values as
   epoch timestamps. The other sub-section will be the
   "policy_published", and record the policy configuration observed by
   the receiving system. Mandatory fields are "domain", "p", "sp",
   "pct". Optional fields are "fo", "adkim", "aspf".

   Within the data section, the report will contain row(s) of data
   stating which IPs were seen to have delivered messages for the Author
   Domain to the receiving system. For each IP that is being reported,
   there will be a "record" element, which will then have each of a
   "row", "identifiers", and "auth_results" sub-element. Within the
   "row" element, there MUST be "source_ip" and "count". There MUST
   also exist a "policy_evaluated", with subelements of "disposition",
   "dkim", and "spf". There MAY be an element for "reason", meant to
   include 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 (possible values listed in Appendex A). The "dkim" and "spf" elements MUST be the evaluated values as they relate to DMARC, not the values the receiver may have used when overriding the policy. Within the "identifiers" element, there MUST exist the data that was used to apply policy for the given IP. In most cases, this will be a "header_from" element, which will contain the 5322.From domain from the message.

There MUST be an "auth_results" element within the 'record' element. This will contain the data related to authenticating the messages associated with this sending IP. The "dkim" subelement is optional as not all messages are signed, while there MUST be at least one "spf" subelement. These elements MUST have a "domain" that was used during validation, as well as "result". Optionally, the "dkim" element MAY include a "selector" element that was observed during validation. For the "spf" element, the "result" element MUST contain a lower-case string where the value is one of none/neutral/pass/fail/softfail/temperror/permerror. The "dkim" result MUST contain a lower-case string where the value is one of none/pass/fail/policy/neutral/temperror/permerror.

2.3. Extensions

There MAY be an optional section for extensions within the "feedback" element. The absence or existence of this section SHOULD NOT create an error when processing reports. This will be covered in a separate section.

]]

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.

   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 and Domain
   Owners to be aware of this situation and allow for such mixed reports
   during the propagation of the new policy to Mail Receivers.

   Aggregate reports are most useful when they all cover a common time
   period. By contrast, correlation of these reports from multiple
   generators when they cover incongruent time periods is difficult or
   impossible. Report generators SHOULD, wherever possible, adhere to
   hour boundaries for the reporting period they are using. For
   example, starting a per-day report at 00:00; starting per-hour
   reports at 00:00, 01:00, 02:00; etc. Report generators using a
   24-hour report period are strongly encouraged to begin that period at
   00:00 UTC, regardless of local timezone or time of report production,
   in order to facilitate correlation.

   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.

2.3.1. 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.

[2.3.1.1](#). **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.

2.3.1.2. Other Methods

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

3. Extensible Reporting

   A DMARC report should 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. They MUST NOT alter the existing
   DMARC structure, but instead exist self-contained within an
   "<extensions>" element. This element MUST be a child of the
   "<feedback>" element.

```
<feedback>
  ...
  <extensions>
    <extension1 definition="https://path/to/spec">
      <data>...</data>
    </extension1>
  </extensions>
</feedback>
```

   A DMARC report receiver SHOULD NOT generate a processing error when
   this "<extensions>" element is absent or empty. Furthermore, if a
   processor is unable to handle an extension in a report, it SHOULD
   ignore the data, and continue to the next extension.

4. IANA Considerations

   TBD

5. Privacy Considerations

   This section will discuss exposure related to DMARC aggregate
   reporting.

5.1. Data Exposure Considerations

   Aggregate reports are limited in scope to DMARC policy and
   disposition results, to information pertaining to the underlying
   authentication mechanisms, and to the identifiers involved in DMARC
   validation.

   Aggregate report may expose sender and recipient identifiers,
   specifically the RFC5322.From addresses.

   Domain Owners requesting reports will receive information about mail
   claiming to be from them, which includes mail that was not, in fact,
   from them. Information about the final destination of mail where it
   might otherwise be obscured by intermediate systems will therefore be
   exposed.

When message-forwarding arrangements exist, Domain Owners requesting reports will also receive information about mail forwarded to domains that were not originally part of their messages' recipient lists. This means that destination domains previously unknown to the Domain Owner may now become visible.

Disclosure of information about the messages is being requested by the entity generating the email in the first place, i.e., the Domain Owner and not the Mail Receiver, so this may not fit squarely within existing privacy policy provisions. For some providers, aggregate reporting is viewed as a function similar to complaint reporting about spamming or phishing and are treated similarly under the privacy policy. Report generators (i.e., Mail Receivers) are encouraged to review their reporting limitations under such policies before enabling DMARC reporting.

5.2. Report Recipients

A DMARC 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.

5.3. Data Contained Within Reports (Tkt64)

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 email 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.

6. Security Considerations

TBD

7. 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>
```

8. **Appendix B**. Sample Report

```
<feedback>
  <report_metadata>
    <org_name>Sample Reporter</org_name>
    <email>report_sender@example-reporter.com</email>
    <extra_contact_info>...</export_contact_info>
    <report_id>3v98abbp8ya9n3va8yr8oa3ya</report_id>
    <date_range>
      <begin>161212415</begin>
      <end>161221511</end>
    </date_range>
  </report_metadata>
  <policy_published>
    <domain>example.com</domain>
    <p>quarantine</p>
    <sp>none</sp>
    <pct>100</pct>
  </policy_published>
  <record>
    <row>
      <source_ip>192.168.4.4</source_ip>
      <count>123</count>
      <policy_evaluated>
        <disposition>quarantine</disposition>
        <dkim>pass</dkim>
        <spf>fail</spf>
      </policy_evaluated>
    </row>
    <identifiers>
      <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>
```

9. **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>.

10. 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