DNSWL Email Authentication Method Extension
draft-vesely-authmethod-dnswl-11

Abstract

This document describes an additional Email Authentication Method compliant with RFC 8601. The method consists in looking up the sender’s IP address in a DNS whitelist.

This document does not consider black lists.

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 http://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 April 17, 2020.

Copyright Notice

Copyright (c) 2019 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 (http://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 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.
1. Introduction

One of the many checks that mail servers carry out is to query DNS whitelists (DNSWL, [RFC5782]). The lookup is based on the connecting client’s IP address, so this check can occur very early in an SMTP transaction. The result can be used to counterweight policies that typically occur at early stages too, such as the Sender Policy Framework (SPF, the last paragraph of Appendix D.3 of [RFC7208] is illustrated in Appendix A). In addition, the result of a DNSWL lookup can also be used at later stages; for example, a delivery agent can use it to learn the trustworthiness of a mail relay in order to estimate the spamminess of an email message. The latter possibility needs a place to collect query results for downstream use, which is precisely what the Authentication-Results header field aims at providing.

Results often contain additional data, encoded according to DNSWL-specific criteria. The present method considers only whitelists --one of the major branches considered by [RFC5782]. In case of DNSxL, the boundary MTA (see [RFC5598]) which carries out the check and possibly stores the result, has to be able to discern at least the color of "x", which is required to make accept/reject decisions. The present method can then be used only if the DNSxL record to be reported turns out to be a whitelisting one.

Data conveyed in A and TXT records can be stored as method’s properties. In effect, they are tantamount to local policies, albeit outsourced. Downstream agents need to know DNSWL-specific encoding to understand the meaning of that data. In order to smooth operations, this document endorses a usage of TXT fields consistent with other authentication methods. Namely, to serve the domain name in the TXT record.
2. Method Details

The following ptype.property items define how the data provided by the whitelist lookup can be saved.

**dns.zone:** DNSWL query root domain, which defines the meaning of the policy.ip property below. Note that an MTA can use a local mirror with a different name. The name stored here has to be the best available reference for all foreseeable downstream consumers. If the message is handed outside the internal network, dns.zone had better be the global zone.

**policy.ip:** The bit mask value received in type A response, in dotted quad. Multiple entries can be arranged in a comma-separated list.

**policy.txt:** The TXT record, if any. Multiple records are concatenated in the usual way (explained, for example, in Section 3.3 of [RFC7208]). See Section 3 for the resulting content and query options.

**dns.sec:** This is a generic property stating whether the relevant data was validated using DNSSEC ([RFC4033]). For the present method, the relevant data consists of the reported policy properties above, or, if the method result is "none", their non-existence. This property has three possible values:

- **yes:** DNSSEC validation confirms the integrity of data. Section 5.1 considers how that is related to the DNS response.
- **no:** The data is not signed. See Section 5.1.
- **na:** Not applicable. No DNSSEC validation can be performed, possibly because the lookup is run through a different means than a security-aware DNS resolver. This does not necessarily imply less security. In particular, "na" is used if the data was downloaded in bulk and then loaded on a local nameserver --which is the case of an MTA querying a local zone different from the reported dns.zone.
  DNS errors, including validation errors, can also report "na". This is also the value assumed by default.

The result of the method states how the query did, up to the interpretation of the result. In particular, some DNSBLs are known to return special codes to signal over quota, for example 127.0.0.255. If the MTA cannot interpret that value, that case results in a false positive.
The method has four possible results:

**pass:** The query successfully returned applicable records. This result is usually accompanied by one or both the policy properties described above. Agents unable to interpret those properties can still derive a positive value from the fact that the sender is whitelisted.

**none:** The query worked but yielded no A record, or returned NXDOMAIN, so the sender is not whitelisted.

**temperror:** The DNS evaluation could not be completed due to some error that is likely transient in nature, such as a temporary DNS error, e.g., a DNS RCODE of 2, commonly known as SERVFAIL, or other error condition resulted. A later attempt may produce a final result.

**permerror:** The DNS evaluation cannot work because test entries don’t work, that is, DNSWL is broken, or because queries are overquota, e.g., a DNS RCODE of 5, commonly known as REFUSED, or a DNSWL-specific policy.ip was returned. A later attempt is unlikely to produce a final result. Human intervention is required.

Note that there is no fail result.

### 3. TXT Record Contents

According to [RFC5782], TXT records describe the reason why IP addresses are listed in a DNSWL. The TXT record is useful if it contains the domain name(s). The domain name would correspond to the DNS domain name used by or within the ADMD operating the relevant MTA, sometimes called the "organizational domain". In that case, the authentication provided by this method is equivalent to a DKIM signature ([RFC6376]) or an SPF check host ([RFC7208]).

According to a DNSWL’s policy, attributing responsibility of an IP address to an organization may require something more than a mere PTR record consistency. If no domain names can be responsibly associated to a given IP, for example because the IP was added without direct involvement of the organization concerned, DNSWLs can use a subdomain of .INVALID ([RFC2606]) where the leftmost label hints at why an address is whitelisted. For example, if the address 192.0.2.38 was added by the list managers solely based on their knowledge, the corresponding TXT record might be AUTOPROMOTED.INVALID, so as to avoid to explicitly identify an entity who didn’t opt-in.

Following the example of Multicast DNS (see the second paragraph of Section 16 of [RFC6762]) names containing non-ASCII characters can be encoded in UTF-8 ([RFC3629]) using the normalization form canonical composition (NFC) as described in Unicode Format for Network.
Interchange ([RFC5198]). Inclusion of unaltered UTF-8 TXT values in the header entails an environment compatible with EAI ([RFC6530]).

DNS queries with a QTYPE of ANY may lead to inconsistent replies, depending on the cache status. In addition, ANY is not "all", and the provisions for queries that have QTYPE=ANY ([RFC8482]) don’t cover DNSxLs. A mail server can issue two simultaneous queries, A and TXT. Otherwise, a downstream filter can issue a TXT query on its own, if it knows that an A query was successful and that the DNSWL serves useful TXT records. It is unlikely that TXT records exist if a query for QTYPE A brought a result of none.

4. IANA Considerations

There is a registry of Email Authentication Methods. The method described in this document is referred by Table 1, along with its ptype.property values.

<table>
<thead>
<tr>
<th>Method</th>
<th>ptype</th>
<th>property</th>
<th>Value</th>
<th>Status</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>dnswl</td>
<td>dns</td>
<td>zone</td>
<td>DNSWL publicly accessible query root domain</td>
<td>active</td>
<td>1</td>
</tr>
<tr>
<td>dnswl</td>
<td>policy</td>
<td>ip</td>
<td>type A response received (or comma-separated list thereof)</td>
<td>active</td>
<td>1</td>
</tr>
<tr>
<td>dnswl</td>
<td>policy</td>
<td>txt</td>
<td>type TXT query response</td>
<td>active</td>
<td>1</td>
</tr>
<tr>
<td>dnswl</td>
<td>dns</td>
<td>sec</td>
<td>one of &quot;yes&quot; for DNSSEC authenticated data, &quot;no&quot; for not signed, or &quot;na&quot; for not applicable</td>
<td>active</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 1: Email Authentication Method

A new ptype, "dns" is introduced in Table 2. It is meant to be used for properties related to the Domain Name System (DNS [RFC1034]).
<table>
<thead>
<tr>
<th>ptype</th>
<th>Definition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dns</td>
<td>[this doc]</td>
<td>The property being reported belongs to the Domain Name System</td>
</tr>
</tbody>
</table>

Table 2: Email Authentication Property Type

This method reuses four of the values already defined in the Email Authentication Result Names associated registry. They are listed in Table 3.

<table>
<thead>
<tr>
<th>Auth Method</th>
<th>Code</th>
<th>Specification</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>dnswl</td>
<td>pass</td>
<td>Sender is whitelisted, up to returned code interpretation</td>
<td>active</td>
</tr>
<tr>
<td>dnswl</td>
<td>none</td>
<td>NXDOMAIN or no record, sender is not whitelisted</td>
<td>active</td>
</tr>
<tr>
<td>dnswl</td>
<td>temperror</td>
<td>Transient DNS error during the query</td>
<td>active</td>
</tr>
<tr>
<td>dnswl</td>
<td>pererror</td>
<td>Query cannot work, human intervention needed</td>
<td>active</td>
</tr>
</tbody>
</table>

Table 3: Email Authentication Result Names

5. Security Considerations

5.1. Security of DNSSEC Validation

The dns.sec property is meant to be as secure as DNSSEC results. It makes sense to use it in an environment where the DNSSEC validation can succeed.

Section 7 of [RFC4033] examines various ways of setting up a stub resolver which either validates DNSSEC locally or trusts the validation provided through a secure channel. For a different class, it is possible to set up a dedicated, caching, dnssec-enabled resolver reachable by the mail server through interprocess communication on 127.0.0.1. In such cases, the property dns.sec=yes corresponds to the Authenticated Data (AD) bit in the DNS response header.

When the response contains no DNSSEC data, a security-aware resolver seeks a signed proof of the non-existence of a DS record, at some...
delegation point. If no error is returned, the zone is unsigned and
dns.sec=no can be set. Quoting the Security Considerations
Section of [RFC3225]: The absence of DNSSEC data in response to a
query with the DO bit set MUST NOT be taken to mean no security
information is available for that zone as the response may be forged
or a non-forged response of an altered (DO bit cleared) query.

If the application verifies the DNSSEC signatures on its own, it
effectively behaves like a validating stub resolver, and hence can
set dns.sec correspondingly.

When the data is downloaded in bulk and made available on a trusted
channel without using DNSSEC, set dns.sec=na or not at all. DNSWL
who publish bulk versions of their data can also sign that data, for
example using OpenPGP ([RFC4880]). It is the responsibility of
system administrators to authenticate the data by downloading and
validating the signature. The result of such validation is not
reported using dns.sec.

5.2. Inherited Security Considerations

For DNSSEC, the considerations of Section 12 of [RFC4033] apply.

All of the considerations described in Section 7 of [RFC8601] apply.
That includes securing against tampering all the channels after the
production of this Authentication-Results header field.

In addition, the usual caveats apply about importing text from
external online sources. Although queried DNSWLs are well known,
trusted entities, it is suggested that TXT records be reported only
if, upon inspection, their content is deemed actually actionable, and
their format compatible with the computing environment.

6. References

6.1. Normative References

[RFC2606] Eastlake 3rd, D. and A. Panitz, "Reserved Top Level DNS
Names", BCP 32, RFC 2606, DOI 10.17487/RFC2606, June 1999,

[RFC5782] Levine, J., "DNS Blacklists and Whitelists", RFC 5782,
editor.org/info/rfc5782>.
6.2. Informative References


Appendix A.  Example

Delivered-To: recipient@example.org
Return-Path: <sender@example.com>
Authentication-Results: mta.example.org; dkim=pass (whitelisted) header.i=@example.com
Authentication-Results: mta.example.org; dnswl=pass dns.zone=list.dnswl.example dns.sec=na
  policy.ip=127.0.10.1
  policy.txt="fwd.example https://dnswl.example/?d=fwd.example"
Received-SPF: fail (Address does not pass Sender Policy Framework)
  client-ip=192.0.2.1;
  envelope-from="sender@example.com";
  helo=mailout.fwd.example;
  receiver=mta.example.org;
Received: from mailout.fwd.example (mailout.fwd.example [192.0.2.1])
(TLS: TLSv1/SSLv3,128bits,ECDHE-RSA-AES128-GCM-SHA256)
  by mta.example.org with ESMTPS; Thu, 03 Ocy 2019 19:23:11 +0200
  id 00000000005DC044.00000005702D87C.000007FC

Trace fields added at the top of the header by multiple agents at various stages during processing at the final MTA

The message went through a third party, fwd.example, which forwarded it to the final MTA.  Such mail path was not arranged beforehand with the involved MTAs, it emerged spontaneously.  This message would not have made it to the target without whitelisting, because:

- the author domain published a strict SPF policy (-all),
- the forwarder did not alter the bounce address, and
- the target usually honors reject-on-fail, according to Section 8.4 of [RFC7208].
However, the target also implemented the last paragraph of Appendix D.3 of [RFC7208]. Rather than rejecting the message outright before DATA, the MTA received it, recorded the SPF fail result, and indicated the local policy mechanism which was applied in order to override that result. Subsequent filtering detected no malware and verified DKIM [RFC6376]. It would still have been possible to reject the message, based on its content. It is at these later stages, after receiving the body and also during delivery, that a deeper knowledge of the policy values obtained from dnswl.example can allow weighting that score against other factors.

Author’s Address

Alessandro Vesely
v. L. Anelli 13
Milano, MI  20122
IT

Email: vesely@tana.it