Abstract

This memo updates the registry of properties in Authentication-Results: message header fields to allow relaying of the results of an email sent using STARTTLS [RFC3207] or not.

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 June 22, 2015.

Copyright Notice

Copyright (c) 2014 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

STARTTLS [RFC3207] defines how to send an email over an SMTP [RFC5321] encrypted session between two mail servers.

This memo thus registers an additional reporting property allowing a TLS result to be relayed as an annotation in a message header.

1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

1.2. Discussion

STARTTLS [RFC3207] defines how to send an email over an encrypted session between two mail servers, Message Transfer Agent (MTA), using the TLS [RFC5246] protocol.

Most of these exchanges are opportunistic, meaning a best effort is done to establish an encrypted message exchange regardless of the strength of the cipher or the validity of the certificates used. However, the results of this negotiation should be recorded in the message via the Authentication-Results header [RFC7001] to indicate to other message processing algorithms, including Messaging User Agents (MUA), how securely this message was transmitted from the MTA client to the MTA server.

The concept of authentication here is related to the presentation of a certificate which is verified valid by a set of trusted Certificate
Authorities (CA), via DANE [RFC6698] or by local policy. This does not indicate that any string in the certificate is related to any string in the email.

The usage and usefulness of the Authentication-Results header is discussed in [RFC7001].

2. Definitions

This memo adds to the "Email Authentication Methods" registry, created by IANA upon publication of [RFC7001], the following:

- The method "tls";
- Associated with that method, the properties (reporting items) "cert.client", "cert.server", "cert.verif", "tls.v", "key.ciphersuite", "key.fingerprint", "key.length" and "key.strength".

2.1. results meaning

The "tls" method can have the following results:

none: the message was sent in clear.

pass: the message was sent encrypted and the client certificate was verified valid either using a trusted CA, via DANE [RFC6698] or via a local policy and host identity was verified.

selfsigned: the message was sent encrypted but the client certificate is self signed.

invalidhost: the message was sent encrypted and the client certificate is verified valid but the host identity is invalid.

fail: the message was sent encrypted but the client certificate is not valid. It is advised to use comments to indicate the nature of the problem like certificate expired, not linked to a trusted CA,...

temperror: the message was sent encrypted and the server was not able to verify the client certificate at this time. This may indicate for instance that the server could not fetch the CRL.

permerror: the message was sent encrypted and the client certificate was not verified by the MTA server. MTA should always attempt to verify the client certificate.
2.2. properties

- **cert.client**: the subject of the X.509 certificate used by the client to initiate TLS.
- **cert.server**: the subject of the X.509 certificate used by the server to initiate TLS (optional).
- **cert.clientalt**: the subject alternative name of the X.509 certificate used by the client to initiate TLS (optional).
- **cert.serveralt**: the subject alternative name of the X.509 certificate used by the server to initiate TLS (optional).
- **cert.clientissuer**: the issuer of the X.509 certificate used by the client to initiate TLS (optional).
- **cert.serverissuer**: the issuer of the X.509 certificate used by the server to initiate TLS (optional).
- **cert.verif**: the type of certification performed: CA, DANE [RFC6698], LOCAL (optional).
- **tls.v**: the protocol version used to encrypt SSL2.0, SSL3.0, TLS1.0, TLS1.1,... (optional)
- **key.ciphersuite**: the description of the TLS cipher suite used as defined in the IANA cipher suite registry.
- **key.fingerprint**: the fingerprint of the key used (optional).
- **key.length**: the length in bits of the key used (optional).
- **key.strength**: as many SMTP TLS are opportunistic in nature this property is an arbitrary value set by the MTA server to indicate the strength of the encryption (optional).

While ciphers strength vary over time, and key length in bits does not indicate a comparable strength between various ciphers, it may be difficult for all the processors of the authentication-results header to redo the analysis based on the cipher used and all to arrive to the same conclusion. It seems, therefore, best if the receiving MTA does that analysis and communicate it to the other layers. This is the purpose of the key.strength. For instance this value could be used by the MUA to indicate to the end user some quality of the encryption channel.
3. IANA Considerations

Per [IANA], the following items have been added to the "Email Authentication Methods" registry:

<table>
<thead>
<tr>
<th>Method</th>
<th>Defined</th>
<th>ptype</th>
<th>property</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>tls</td>
<td>RFC 3207</td>
<td>cert</td>
<td>client</td>
<td>subject of client certificate, section 4.1.2.6 of RFC 5280</td>
</tr>
<tr>
<td>tls</td>
<td>RFC 3207</td>
<td>cert</td>
<td>server</td>
<td>subject of server certificate, section 4.1.2.6 of RFC 5280</td>
</tr>
<tr>
<td>tls</td>
<td>RFC 3207</td>
<td>cert</td>
<td>clientalt</td>
<td>alternate subject of client certificate, section 4.2.1.6 of RFC 5280</td>
</tr>
<tr>
<td>tls</td>
<td>RFC 3207</td>
<td>cert</td>
<td>serveralt</td>
<td>alternate subject of server certificate, section 4.2.1.6 of RFC 5280</td>
</tr>
<tr>
<td>tls</td>
<td>RFC 3207</td>
<td>cert</td>
<td>clientissuer</td>
<td>issuer of client certificate, section 4.1.2.4 of RFC 5280</td>
</tr>
<tr>
<td>tls</td>
<td>RFC 3207</td>
<td>cert</td>
<td>serverissuer</td>
<td>issuer of server certificate, section 4.1.2.4 of RFC 5280</td>
</tr>
<tr>
<td>tls</td>
<td>RFC 3207</td>
<td>cert</td>
<td>verif</td>
<td>CA, DANE, LOCAL</td>
</tr>
<tr>
<td>tls</td>
<td>RFC 3207</td>
<td>tls</td>
<td>v</td>
<td>protocol version description from RFC 5246</td>
</tr>
<tr>
<td>-------</td>
<td>----------</td>
<td>-------</td>
<td>------------</td>
<td>---------------------------------------------</td>
</tr>
<tr>
<td>tls</td>
<td>RFC 3207</td>
<td>key</td>
<td>ciphersuite</td>
<td>IANA cipher suite registry description from RFC 5246</td>
</tr>
<tr>
<td>tls</td>
<td>RFC 3207</td>
<td>key</td>
<td>fingerprint</td>
<td>key fingerprint from RFC 5246</td>
</tr>
<tr>
<td>tls</td>
<td>RFC 3207</td>
<td>key</td>
<td>length</td>
<td>in bits</td>
</tr>
<tr>
<td>tls</td>
<td>RFC 3207</td>
<td>key</td>
<td>strength</td>
<td>low medium high</td>
</tr>
</tbody>
</table>

Also, the following items have been added to the "Email Authentication Result Names" registry:
<table>
<thead>
<tr>
<th>Code</th>
<th>Existing/New</th>
<th>Defined In</th>
<th>Method</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>none</td>
<td>existing</td>
<td>RFC 7001</td>
<td>tls</td>
<td>this memo</td>
</tr>
<tr>
<td>pass</td>
<td>existing</td>
<td>RFC 7001</td>
<td>tls</td>
<td>this memo</td>
</tr>
<tr>
<td>selfsigned</td>
<td>existing</td>
<td>RFC 7001</td>
<td>tls</td>
<td>this memo</td>
</tr>
<tr>
<td>invalidhost</td>
<td>existing</td>
<td>RFC 7001</td>
<td>tls</td>
<td>this memo</td>
</tr>
<tr>
<td>fail</td>
<td>existing</td>
<td>RFC 7001</td>
<td>tls</td>
<td>this memo</td>
</tr>
<tr>
<td>temperror</td>
<td>existing</td>
<td>RFC 7001</td>
<td>tls</td>
<td>this memo</td>
</tr>
<tr>
<td>permerror</td>
<td>existing</td>
<td>RFC 7001</td>
<td>tls</td>
<td>this memo</td>
</tr>
</tbody>
</table>

4. Security Considerations

This memo creates a mechanism for relaying STARTTLS [RFC3207] results using the structure already defined by [RFC7001]. The Security Considerations sections of those documents should be consulted.

By this mechanism, some identifiers of the client certificates get to live pass the receiving MTA. This is a change in the sender expectation on where the client certificate is used.

5. References

5.1. Normative References


5.2. Informative References


Appendix A. Authentication-Results Examples

This section presents an example of the use of this new header field to indicate TLS results.

A.1. TLS Results
A message that went over a successful TLS session:

Authentication-Results: mail-router.example.net;
dkim=pass (good signature) header.d=newyork.example.com
header.b=oINEO8hg;
tls=pass (verified, expires 20140505)
cert.verif=CA
cert.client="CN=smtp.example.com,O=ACME,L=ToonTown,
ST=CA,C=US"
cert.clientalt="DNS:smtp.example.com, DNS:newyork.example.com"
cert.issuer="C=US, O=AcmeCert Inc, CN=AcmeCert CA"
key.ciphersuite=TLS_RSA_WITH_RC4_128_SHA
tls.v=TLS1.0
AD:5C:B9:C9:40"
key.length=128
key.strength=MEDIUM;
Received: from newyork.example.com
  (newyork.example.com [192.0.2.250])
  by mail-router.example.net (8.11.6/8.11.6)
  for <recipient@example.net>
  with ESMTPS id i7PK0sH7021929;
  Fri, Feb 15 2002 17:19:22 -0800
DKIM-Signature: v=1; a=rsa-sha256; s=rashani;
d=newyork.example.com;
t=1188964191; c=relaxed/simple;
h=From:Date:To:VBR-Info:Message-Id:Subject;
bh=sEu28nfs9fu2GD/pSr7A9ysbY3jtdaQP7x9xPQtS0m7=;
  b=oINEO8hgn/gnunsg ... 9n9ODSNFSDij3=
From: sender@newyork.example.com
Date: Fri, Feb 15 2002 16:54:30 -0800
To: meetings@example.net
Message-Id: <12345.abc@newyork.example.com>
Subject: here’s a sample

Author’s Address

Franck Martin (editor)
LinkedIn
Mountain View, CA
US

Email: fmartin@linkedin.com