Using TLS in Applications
Internet-Draft
Intended status: Standards Track
Expires: October 5, 2017

D. Margolis
Google, Inc
A. Brotman
Comcast, Inc
B. Ramakrishnan
Yahoo!, Inc
J. Jones
Microsoft, Inc
M. Risher
Google, Inc
April 3, 2017

SMTP TLS Reporting
draft-ietf-uta-smtp-tlsrpt-04

Abstract

A number of protocols exist for establishing encrypted channels between SMTP Mail Transfer Agents, including STARTTLS [RFC3207], DANE [RFC6698], and MTA-STS (TODO: Add ref). These protocols can fail due to misconfiguration or active attack, leading to undelivered messages or delivery over unencrypted or unauthenticated channels. This document describes a reporting mechanism and format by which sending systems can share statistics and specific information about potential failures with recipient domains. Recipient domains can then use this information to both detect potential attackers and diagnose unintentional misconfigurations.

Table of Contents

1. Introduction 3
1.1. Terminology 3
2. Related Technologies 4
3. Reporting Policy 4
3.1. Example Reporting Policy 5
3.1.1. Report using MAILTO 5
3.1.2. Report using HTTPS 5
4. Reporting Schema 5
4.1. Report Time-frame 6
4.2. Delivery Summary 6
4.2.1. Success Count 6
4.2.2. Failure Count 7
4.3. Result Types 7
4.3.1. Negotiation Failures 7
4.3.2. Policy Failures 7
4.3.3. General Failures 8
4.3.4. Transient Failures 8
5. Report Delivery 8
5.1. Report Filename 8
5.2. Compression 9
5.3. Email Transport 9
5.4. HTTPS Transport 10
5.5. Delivery Retry 10
6. IANA Considerations 11
7. Security Considerations 11
8. Appendix 1: Example Reporting Policy 11
8.1. Report using MAILTO 11
8.2. Report using HTTPS 12
9. Appendix 2: JSON Report Schema 12
10. Appendix 3: Example JSON Report 14
11. Normative References 16

Authors' Addresses 17

1. Introduction

The STARTTLS extension to SMTP [RFC3207] allows SMTP clients and hosts to establish secure SMTP sessions over TLS. The protocol design is based on "Opportunistic Security" (OS) [RFC7435], which maintains interoperability with clients that do not support STARTTLS but means that any attacker who can delete parts of the SMTP session (such as the "250 STARTTLS" response) or redirect the entire SMTP session (perhaps by overwriting the resolved MX record of the delivery domain) can perform a downgrade or interception attack.

Because such "downgrade attacks" are not necessarily apparent to the receiving MTA, this document defines a mechanism for sending domains to report on failures at multiple stages of the MTA-to-MTA conversation.

Recipient domains may also use the mechanisms defined by MTA-STS (TODO: Add ref) or DANE [RFC6698] to publish additional encryption and authentication requirements; this document defines a mechanism for sending domains that are compatible with MTA-STS or DANE to share success and failure statistics with recipient domains.

Specifically, this document defines a reporting schema that covers failures in routing, STARTTLS negotiation, and both DANE [RFC6698] and MTA-STS (TODO: Add ref) policy validation errors, and a standard TXT record that recipient domains can use to indicate where reports in this format should be sent.

This document is intended as a companion to the specification for SMTP MTA Strict Transport Security (MTA-STS, TODO: Add ref).

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].

We also define the following terms for further use in this document:

- MTA-STS Policy: A definition of the expected TLS availability, behavior, and desired actions for a given domain when a sending MTA encounters problems in negotiating a secure channel. MTA-STS is defined in [TODO]

- DANE Policy: A mechanism by which administrators can supply a record that can be used to validate the certificate presented by an MTA. DANE is defined in [RFC6698].

- TLSRPT Policy: A policy specifying the endpoint to which sending MTAs should deliver reports.

- Policy Domain: The domain against which an MTA-STS or DANE Policy is defined.

- Sending MTA: The MTA initiating the delivery of an email message.

2. Related Technologies

- This document is intended as a companion to the specification for SMTP MTA Strict Transport Security (MTA-STS, TODO: Add ref).

- The Public Key Pinning Extension for HTTP [RFC7469] contains a JSON-based definition for reporting individual pin validation failures.

- The Domain-based Message Authentication, Reporting, and Conformance (DMARC) [RFC7489] contains an XML-based reporting format for aggregate and detailed email delivery errors.

3. Reporting Policy

A domain publishes a record to its DNS indicating that it wishes to receive reports. These SMTP TLSRPT policies are distributed via DNS from the Policy Domain's zone, as TXT records (similar to DMARC policies) under the name "_smtp-tlsrpt". For example, for the Policy Domain "example.com", the recipient's TLSRPT policy can be retrieved from "_smtp-tlsrpt.example.com".

Policies consist of the following directives:

- "v": This value MUST be equal to "TLSRPTv1".

- "rua": A URI specifying the endpoint to which aggregate information about policy failures should be sent (see the section _Reporting_ _Schema_ for more information). Two URI schemes are supported: "mailto" and "https".

  * In the case of "https", reports should be submitted via POST ([RFC2818]) to the specified URI.

  * In the case of "mailto", reports should be submitted to the specified email address. When sending failure reports via

SMTP, sending MTAs MUST deliver reports despite any TLS-related failures. This may mean that the reports are delivered in the clear.

The formal definition of the "_smtp-tlsrpt" TXT record, defined using [RFC5234], is as follows:

```
tlsrpt-record    = tlsrpt-version *WSP %x3B tlsrpt-rua

tlsrpt-version   = "v" *WSP "=" *WSP %x54 %x4C %x53
                   %x52 %x50 %x54 %x76 %x31

tlsrpt-rua       = "rua" *WSP "=" *WSP tlsrpt-uri

tlsrpt-uri       = URI
                 ; "URI" is imported from [@!RFC3986]; commas (ASCII
                 ; 0x2C) and exclamation points (ASCII 0x21)
                 ; MUST be encoded; the numeric portion MUST fit
                 ; within an unsigned 64-bit integer
```

If multiple TXT records for "_smtp-tlsrpt" are returned by the resolver, records which do not begin with "v=TLSRPTv1;" are discarded. If the number of resulting records is not one, senders MUST assume the recipient domain does not implement TLSRPT.

3.1. Example Reporting Policy

3.1.1. Report using MAILTO

```
_smtp-tlsrpt.example.com. IN TXT \
    "v=TLSRPTv1;rua=mailto:reports@example.com"
```

3.1.2. Report using HTTPS

```
_smtp-tlsrpt.example.com. IN TXT \
    "v=TLSRPTv1; \
    rua=https://reporting.example.com/v1/tlsrpt"
```

4. Reporting Schema

The report is composed as a plain text file encoded in the JSON format ([RFC7159]).

Aggregate reports contain the following fields:

- Report metadata:
  - The organization responsible for the report

    * Contact information for one or more responsible parties for the contents of the report

    * A unique identifier for the report

    * The reporting date range for the report

- Policy, consisting of:

    * One of the following policy types: (1) The MTA-STS policy applied (as a string) (2) The DANE TLSA record applied (as a string, with each RR entry of the RRset listed and separated by a semicolon) (3) The literal string "no-policy-found", if neither a TLSA nor MTA-STS policy could be found.

    * The domain for which the policy is applied

    * The MX host

    * An identifier for the policy (where applicable)

- Aggregate counts, comprising result type, sending MTA IP, receiving MTA hostname, session count, and an optional additional information field containing a URI for recipients to review further information on a failure type.

Note that the failure types are non-exclusive; an aggregate report may contain overlapping "counts" of failure types when a single send attempt encountered multiple errors.

4.1. Report Time-frame

The report SHOULD cover a full day, from 0000-2400 UTC. This should allow for easier correlation of failure events.

4.2. Delivery Summary

4.2.1. Success Count

- "success-count": This indicates that the sending MTA was able to successfully negotiate a policy-compliant TLS connection, and serves to provide a "heartbeat" to receiving domains that reporting is functional and tabulating correctly. This field contains an aggregate count of successful connections for the reporting system.

4.2.2. Failure Count

- "failure-count": This indicates that the sending MTA was unable to successfully establish a connection with the receiving platform. The "Result Types" section will elaborate on the failed negotiation attempts. This field contains an aggregate count of failed connections.

4.3. Result Types

The list of result types will start with the minimal set below, and is expected to grow over time based on real-world experience. The initial set is:

4.3.1. Negotiation Failures

- "starttls-not-supported": This indicates that the recipient MX did not support STARTTLS.

- "certificate-host-mismatch": This indicates that the certificate presented did not adhere to the constraints specified in the MTA-STS or DANE policy, e.g. if the MX does not match any identities listed in the Subject Alternate Name (SAN) [RFC5280].

- "certificate-expired": This indicates that the certificate has expired.

- "certificate-not-trusted": This a label that covers multiple certificate related failures that include, but not limited to errors such as untrusted/unknown CAs, certificate name constraints, certificate chain errors etc. When using this declaration, the reporting MTA SHOULD utilize the "failure-reason" to provide more information to the receiving entity.

- "validation-failure": This indicates a general failure for a reason not matching a category above. When using this declaration, the reporting MTA SHOULD utilize the "failure-reason" to provide more information to the receiving entity.

4.3.2. Policy Failures

4.3.2.1. DANE-specific Policy Failures

- "tlsa-invalid": This indicates a validation error in the TLSA record associated with a DANE policy. None of the records in the RRset were found to be valid.

- "dnssec-invalid": This would indicate that no valid records were returned from the recursive resolver. The request returned with SERVFAIL for the requested TLSA record.

4.3.2.2. MTA-STS-specific Policy Failures

- "sts-invalid": This indicates a validation error for the overall MTA-STS policy.

- "webpki-invalid": This indicates that the MTA-STS policy could not be authenticated using PKIX validation.

4.3.3. General Failures

When a negotiation failure can not be categorized into one of the "Negotiation Failures" stated above, the reporter SHOULD use the "validation-failure" category. As TLS grows and becomes more complex, new mechanisms may not be easily categorized. This allows for a generic feedback category. When this category is used, the reporter SHOULD also use the "failure-reason-code" to give some feedback to the receiving entity. This is intended to be a short text field, and the contents of the field should be an error code or error text, such as "X509_V_ERR_UNHANDLED_CRITICAL_CRL_EXTENSION".

4.3.4. Transient Failures

Transient errors due to too-busy network, TCP timeouts, etc. are not required to be reported.

5. Report Delivery

Reports can be delivered either as an email message via SMTP or via HTTP POST.

5.1. Report Filename

The filename is typically constructed using the following ABNF:

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

unique-id = 1*(ALPHA / DIGIT)

sender = domain        ; imported from [@!RFC5322]

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 = "json" / "json.gz"
```

The extension MUST be "json" for a plain JSON file, or "json.gz" for a JSON file compressed using GZIP.

"unique-id" allows an optional unique ID generated by the Sending MTA to distinguish among multiple reports generated simultaneously by different sources within the same Policy Domain. For example, this is a possible filename for the gzip file of a report to the Policy Domain "example.net" from the Sending MTA "mail.sender.example.com":

`mail.sender.example.com!example.net!1470013207!1470186007!001.json.gz`

5.2. Compression

The report SHOULD be subjected to GZIP compression for both email and HTTPS transport. 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); compressing the file increases the chances of acceptance of the report at some compute cost.

5.3. Email Transport

The report MAY be delivered by email. 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.

If compressed, the report should use the media type "application/gzip" if compressed (see [RFC6713]), and "text/json" otherwise.

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

```
tlsrpt-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 Sending MTA generating the report. The purpose of the Report-ID: portion of the field is to enable the Policy Domain to identify and ignore duplicate reports that might be sent by a Sending MTA.

For instance, this is a possible Subject field for a report to the Policy Domain "example.net" from the Sending MTA "mail.sender.example.com". It is line-wrapped as allowed by [RFC5322]:

```
Subject: Report Domain: example.net
    Submitter: mail.sender.example.com
    Report-ID: <735ff.e317+bf22029@mailexample.net>
```

Note that, when sending failure reports via SMTP, sending MTAs MUST NOT honor MTA-STS or DANE TLSA failures.

5.4. HTTPS Transport

The report MAY be delivered by POST to HTTPS. If compressed, the report should use the media type "application/gzip" (see [RFC6713]), and "text/json" otherwise.

5.5. Delivery Retry

In the event of a delivery failure, regardless of the delivery method, a sender SHOULD attempt redelivery for up to 24hrs after the initial attempt. As previously stated the reports are optional, so while it is ideal to attempt redelivery, it is not required. If multiple retries are attempted, they should be on a logarithmic scale.

6. IANA Considerations

There are no IANA considerations at this time.

7. Security Considerations

SMTP TLS Reporting provides transparency into misconfigurations or attempts to intercept or tamper with mail between hosts who support STARTTLS. There are several security risks presented by the existence of this reporting channel:

- Flooding of the Aggregate report URI (rua) endpoint: An attacker could flood the endpoint and prevent the receiving domain from accepting additional reports. This type of Denial-of-Service attack would limit visibility into STARTTLS failures, leaving the receiving domain blind to an ongoing attack.

- Untrusted content: An attacker could inject malicious code into the report, opening a vulnerability in the receiving domain. Implementers are advised to take precautions against evaluating the contents of the report.

- Report snooping: An attacker could create a bogus TLSRPT record to receive statistics about a domain the attacker does not own. Since an attacker able to poison DNS is already able to receive counts of SMTP connections (and, absent DANE or MTA-STS policies, actual SMTP message payloads), this does not present a significant new vulnerability.

- Reports as DDoS: TLSRPT allows specifying destinations for the reports that are outside the authority of the Policy Domain, which allows domains to delegate processing of reports to a partner organization. However, an attacker who controls the Policy Domain DNS could also use this mechanism to direct the reports to an unwitting victim, flooding that victim with excessive reports. DMARC [RFC7489] defines an elegant solution for verifying delegation; however, since the attacker had less ability to generate large reports than with DMARC failures, and since the reports are generated by the sending MTA, such a delegation mechanism is left for a future version of this specification.

8. Appendix 1: Example Reporting Policy

8.1. Report using MAILTO

```
_smtp-tlsrpt.mail.example.com. IN TXT \
        "v=TLSRPTv1;rua=mailto:reports@example.com"
```

8.2. Report using HTTPS

```
_smtp-tlsrpt.mail.example.com. IN TXT \
        "v=TLSRPTv1; \
        rua=https://reporting.example.com/v1/tlsrpt"
```

9. Appendix 2: JSON Report Schema

The JSON schema is derived from the HPKP JSON schema [RFC7469] (cf. Section 3)

```
{
  "organization-name": organization-name,
  "date-range": {
    "start-datetime": date-time,
    "end-datetime": date-time
  },
  "contact-info": email-address,
  "report-id": report-id,
  "policy": {
    "policy-type": policy-type,
    "policy-string": policy-string,
    "policy-domain": domain,
    "mx-host": mx-host-pattern
  },
  "summary": {
    "success-aggregate": total-successful-session-count,
    "failure-aggregate:" total-failure-session-count
  }
  "failure-details": [
    {
      "result-type": result-type,
      "sending-mta-ip": ip-address,
      "receiving-mx-hostname": receiving-mx-hostname,
      "receiving-mx-helo": receiving-mx-helo,
      "session-count": failed-session-count,
      "additional-information": additional-info-uri,
      "failure-reason-code": "Text body"
    }
  ]
}
```

Figure: JSON Report Format

- "organization-name": The name of the organization responsible for the report. It is provided as a string.

- "date-time": The date-time indicates the start- and end-times for the report range. It is provided as a string formatted according to Section 5.6, "Internet Date/Time Format", of [RFC3339]. The report should be for a full UTC day, 0000-2400.

- "email-address": The contact information for a responsible party of the report. It is provided as a string formatted according to Section 3.4.1, "Addr-Spec", of [RFC5322].

- "report-id": A unique identifier for the report. Report authors may use whatever scheme they prefer to generate a unique identifier. It is provided as a string.

- "policy-type": The type of policy that was applied by the sending domain. Presently, the only three valid choices are "tlsa", "sts", and the literal string "no-policy-found". It is provided as a string.

- "policy-string": The string serialization of the policy, whether TLSA record or MTA-STS policy. Any linefeeds from the original policy MUST be replaced with [SP]. TODO: Help with specifics.

- "domain": The Policy Domain upon which the policy was applied. For messages sent to "user@example.com" this field would contain "example.com". It is provided as a string.

- "mx-host-pattern": The pattern of MX hostnames from the applied policy. It is provided as a string, and is interpreted in the same manner as the "Checking of Wildcard Certificates" rules in Section 6.4.3 of [RFC6125].

- "result-type": A value from the _Result Types_ section above.

- "ip-address": The IP address of the sending MTA that attempted the STARTTLS connection. It is provided as a string representation of an IPv4 or IPv6 address in dot-decimal or colon-hexadecimal notation.

- "receiving-mx-hostname": The hostname of the receiving MTA MX record with which the sending MTA attempted to negotiate a STARTTLS connection.

- "receiving-mx-helo": (optional) The HELO or EHLO string from the banner announced during the reported session.

- "success-aggregate": The aggregate number (integer) of successfully negotiated SSL-enabled connections to the receiving site.

- "failure-aggregate": The aggregate number (integer) of failures to negotiate an SSL-enabled connection to the receiving site.

- "session-count": The number of (attempted) sessions that match the relevant "result-type" for this section.

- "additional-info-uri": An optional URI pointing to additional information around the relevant "result-type". For example, this URI might host the complete certificate chain presented during an attempted STARTTLS session.

- "failure-reason-code": A text field to include an SSL-related error code or error message.

10. Appendix 3: Example JSON Report

```
{
  "organization-name": "Company-X",
  "date-range": {
    "start-datetime": "2016-04-01T00:00:00Z",
    "end-datetime": "2016-04-01T23:59:59Z"
  },
  "contact-info": "sts-reporting@company-x.com",
  "report-id": "5065427c-23d3-47ca-b6e0-946ea0e8c4be",
  "policy": {
    "policy-type": "sts",
    "policy-string": "{ \"version\": \"STSv1\",\"mode\": \"report\", \"mx\":
[\"*.mail.company-y.com\"], \"max_age\": 86400 }",
    "policy-domain": "company-y.com",
    "mx-host": "*.mail.company-y.com"
  },
  "summary": {
    "success-aggregate": 5326,
    "failure-aggregate": 303
  }
  "failure-details": [{
    "result-type": "certificate-expired",
    "sending-mta-ip": "98.136.216.25",
    "receiving-mx-hostname": "mx1.mail.company-y.com",
    "session-count": 100
  }, {
    "result-type": "starttls-not-supported",
    "sending-mta-ip": "98.22.33.99",
    "receiving-mx-hostname": "mx2.mail.company-y.com",
    "session-count": 200,
    "additional-information": "hxxps://reports.company-x.com/
      report_info?id=5065427c-23d3#StarttlsNotSupported"
  }, {
    "result-type: "validation-failure",
    "sending-mta-ip": "47.97.15.2",
    "receiving-mx-hostname: "mx-backup.mail.company-y.com",
    "session-count": 3,
    "failure-error-code": "X509_V_ERR_PROXY_PATH_LENGTH_EXCEEDED"
  }]
}
```

Figure: Example JSON report for a messages from Company-X to Company-Y, where 100 sessions were attempted to Company Y servers with an expired certificate and 200 sessions were attempted to Company Y servers that did not successfully respond to the "STARTTLS" command. Additionally 3 sessions failed due to "X509_V_ERR_PROXY_PATH_LENGTH_EXCEEDED".

11. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

[RFC2818] Rescorla, E., "HTTP Over TLS", RFC 2818, DOI 10.17487/RFC2818, May 2000, <http://www.rfc-editor.org/info/rfc2818>.

[RFC3207] Hoffman, P., "SMTP Service Extension for Secure SMTP over Transport Layer Security", RFC 3207, DOI 10.17487/RFC3207, February 2002, <http://www.rfc-editor.org/info/rfc3207>.

[RFC3339] Klyne, G. and C. Newman, "Date and Time on the Internet: Timestamps", RFC 3339, DOI 10.17487/RFC3339, July 2002, <http://www.rfc-editor.org/info/rfc3339>.

[RFC5234] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, DOI 10.17487/RFC5234, January 2008, <http://www.rfc-editor.org/info/rfc5234>.

[RFC5322] Resnick, P., Ed., "Internet Message Format", RFC 5322, DOI 10.17487/RFC5322, October 2008, <http://www.rfc-editor.org/info/rfc5322>.

[RFC6125] Saint-Andre, P. and J. Hodges, "Representation and Verification of Domain-Based Application Service Identity within Internet Public Key Infrastructure Using X.509 (PKIX) Certificates in the Context of Transport Layer Security (TLS)", RFC 6125, DOI 10.17487/RFC6125, March 2011, <http://www.rfc-editor.org/info/rfc6125>.

[RFC6698] Hoffman, P. and J. Schlyter, "The DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS) Protocol: TLSA", RFC 6698, DOI 10.17487/RFC6698, August 2012, <http://www.rfc-editor.org/info/rfc6698>.

[RFC6713] Levine, J., "The 'application/zlib' and 'application/gzip' Media Types", RFC 6713, DOI 10.17487/RFC6713, August 2012, <http://www.rfc-editor.org/info/rfc6713>.

[RFC7159] Bray, T., Ed., "The JavaScript Object Notation (JSON) Data Interchange Format", RFC 7159, DOI 10.17487/RFC7159, March 2014, <http://www.rfc-editor.org/info/rfc7159>.

[RFC7435] Dukhovni, V., "Opportunistic Security: Some Protection Most of the Time", RFC 7435, DOI 10.17487/RFC7435, December 2014, <http://www.rfc-editor.org/info/rfc7435>.

[RFC7469] Evans, C., Palmer, C., and R. Sleevi, "Public Key Pinning Extension for HTTP", RFC 7469, DOI 10.17487/RFC7469, April 2015, <http://www.rfc-editor.org/info/rfc7469>.

[RFC7489] Kucherawy, M., Ed. and E. Zwicky, Ed., "Domain-based Message Authentication, Reporting, and Conformance (DMARC)", RFC 7489, DOI 10.17487/RFC7489, March 2015, <http://www.rfc-editor.org/info/rfc7489>.

Authors' Addresses

Daniel Margolis
Google, Inc

Email: dmargolis (at) google.com

Alexander Brotman
Comcast, Inc

Email: alex_brotman (at) comcast.com

Binu Ramakrishnan
Yahoo!, Inc

Email: rbinu (at) yahoo-inc (dot com)

Janet Jones
Microsoft, Inc

Email: janet.jones (at) microsoft (dot com)

Mark Risher
Google, Inc

Email: risher (at) google (dot com)