

Network Working Group  
Internet-Draft  
Updates: [1939](#), [3464](#), [3501](#), [5068](#), [6186](#)  
(if approved)  
Intended status: Standards Track  
Expires: August 17, 2015

K. Moore  
Network Heretics  
C. Newman  
Oracle  
February 13, 2015

**Deployable Enhanced Email Privacy (DEEP)**  
**draft-ietf-uta-email-deep-00.txt**

Abstract

This specification defines a set of requirements and facilities designed to improve email privacy. This provides mechanisms intended to increase use of already deployed Transport Layer Security (TLS) technology, provide a model for mail user agent's confidentiality assurance, and enable mail service providers to advertise improved TLS privacy facilities.

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 August 17, 2015.

Copyright Notice

Copyright (c) 2015 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.

## Table of Contents

<a href="#">1.</a>	<a href="#">Introduction . . . . .</a>	<a href="#">3</a>
<a href="#">2.</a>	<a href="#">Conventions and Terminology Used in This Document . . . . .</a>	<a href="#">4</a>
<a href="#">3.</a>	<a href="#">Mail Account Confidentiality Assurance Level . . . . .</a>	<a href="#">4</a>
<a href="#">3.1.</a>	<a href="#">High Confidentiality Assurance . . . . .</a>	<a href="#">5</a>
<a href="#">3.2.</a>	<a href="#">Certificate Pinning . . . . .</a>	<a href="#">6</a>
<a href="#">3.3.</a>	<a href="#">No Confidentiality Assurance . . . . .</a>	<a href="#">6</a>
<a href="#">3.4.</a>	<a href="#">Other Confidentiality Assurance Levels . . . . .</a>	<a href="#">6</a>
<a href="#">4.</a>	<a href="#">Implicit TLS . . . . .</a>	<a href="#">7</a>
<a href="#">4.1.</a>	<a href="#">Implicit TLS for POP . . . . .</a>	<a href="#">7</a>
<a href="#">4.2.</a>	<a href="#">Implicit TLS for IMAP . . . . .</a>	<a href="#">7</a>
<a href="#">4.3.</a>	<a href="#">Implicit TLS for SMTP Submission . . . . .</a>	<a href="#">8</a>
<a href="#">4.4.</a>	<a href="#">Implicit TLS Connection Closure for POP, IMAP and SMTP . . . . .</a>	<a href="#">8</a>
<a href="#">5.</a>	<a href="#">Email Security Upgrading Using Security Latches . . . . .</a>	<a href="#">8</a>
<a href="#">5.1.</a>	<a href="#">Email Security Tags . . . . .</a>	<a href="#">9</a>
<a href="#">5.2.</a>	<a href="#">Initial Set of Email Security Tags . . . . .</a>	<a href="#">10</a>
<a href="#">5.3.</a>	<a href="#">Server DEEP Status . . . . .</a>	<a href="#">10</a>
<a href="#">5.4.</a>	<a href="#">Email Security Tag Latch Failures . . . . .</a>	<a href="#">10</a>
<a href="#">6.</a>	<a href="#">Recording TLS Cipher Suite in Received Header . . . . .</a>	<a href="#">11</a>
<a href="#">7.</a>	<a href="#">Extensions for DEEP Status and Reporting . . . . .</a>	<a href="#">11</a>
<a href="#">7.1.</a>	<a href="#">IMAP DEEP Extension . . . . .</a>	<a href="#">12</a>
<a href="#">7.2.</a>	<a href="#">POP DEEP Extension . . . . .</a>	<a href="#">14</a>
<a href="#">7.3.</a>	<a href="#">SMTP DEEP Extension . . . . .</a>	<a href="#">15</a>
<a href="#">7.4.</a>	<a href="#">SMTP Error Extension . . . . .</a>	<a href="#">16</a>
<a href="#">8.</a>	<a href="#">Use of SRV records in Establishing Configuration . . . . .</a>	<a href="#">16</a>
<a href="#">9.</a>	<a href="#">Implementation Requirements . . . . .</a>	<a href="#">17</a>
<a href="#">9.1.</a>	<a href="#">All Implementations (Client and Server) . . . . .</a>	<a href="#">17</a>
<a href="#">9.1.1.</a>	<a href="#">Client Certificate Authentication . . . . .</a>	<a href="#">18</a>
<a href="#">9.2.</a>	<a href="#">Mail Server Implementation Requirements . . . . .</a>	<a href="#">18</a>
<a href="#">9.3.</a>	<a href="#">Mail User Agent Implementation Requirements . . . . .</a>	<a href="#">19</a>
<a href="#">9.4.</a>	<a href="#">Non-configurable MUAs and nonstandard access protocols . . . . .</a>	<a href="#">20</a>
<a href="#">9.5.</a>	<a href="#">DEEP Compliance for Anti-Virus/Anti-Spam Software and Services . . . . .</a>	<a href="#">20</a>
<a href="#">10.</a>	<a href="#">Mail Service Provider Requirements . . . . .</a>	<a href="#">20</a>
<a href="#">10.1.</a>	<a href="#">Server Requirements . . . . .</a>	<a href="#">20</a>
<a href="#">10.2.</a>	<a href="#">MSPs MUST provide Submission Servers . . . . .</a>	<a href="#">20</a>
<a href="#">10.3.</a>	<a href="#">TLS Server Certificate Requirements . . . . .</a>	<a href="#">21</a>
<a href="#">10.4.</a>	<a href="#">Recommended DNS records for mail protocol servers . . . . .</a>	<a href="#">21</a>
<a href="#">10.4.1.</a>	<a href="#">MX records . . . . .</a>	<a href="#">21</a>
<a href="#">10.4.2.</a>	<a href="#">SRV records . . . . .</a>	<a href="#">21</a>
<a href="#">10.4.3.</a>	<a href="#">TLSA records . . . . .</a>	<a href="#">22</a>
<a href="#">10.4.4.</a>	<a href="#">DNSSEC . . . . .</a>	<a href="#">22</a>
<a href="#">10.5.</a>	<a href="#">MSP Server Monitoring . . . . .</a>	<a href="#">22</a>



<a href="#">10.6.</a>	Advertisement of DEEP status . . . . .	<a href="#">22</a>
<a href="#">10.7.</a>	Require TLS . . . . .	<a href="#">22</a>
<a href="#">10.8.</a>	Changes to Internet Facing Servers . . . . .	<a href="#">22</a>
<a href="#">11.</a>	IANA Considerations . . . . .	<a href="#">23</a>
<a href="#">11.1.</a>	Security Tag Registry . . . . .	<a href="#">23</a>
<a href="#">11.2.</a>	Initial Set of Security Tags . . . . .	<a href="#">23</a>
<a href="#">11.3.</a>	POP3S Port Registration Update . . . . .	<a href="#">25</a>
<a href="#">11.4.</a>	IMAPS Port Registration Update . . . . .	<a href="#">25</a>
<a href="#">11.5.</a>	Submissions Port Registration . . . . .	<a href="#">26</a>
<a href="#">11.6.</a>	DEEP IMAP Capability . . . . .	<a href="#">26</a>
<a href="#">11.7.</a>	DEEP POP3 Capability . . . . .	<a href="#">26</a>
<a href="#">11.8.</a>	DEEP SMTP EHLO Keyword . . . . .	<a href="#">27</a>
<a href="#">11.9.</a>	SMTP Enhanced Status Code . . . . .	<a href="#">27</a>
<a href="#">11.10.</a>	MAIL Parameters Additional-registered-clauses Sub-Registry . . . . .	<a href="#">27</a>
<a href="#">12.</a>	Security Considerations . . . . .	<a href="#">28</a>
<a href="#">13.</a>	References . . . . .	<a href="#">28</a>
<a href="#">13.1.</a>	Normative References . . . . .	<a href="#">28</a>
<a href="#">13.2.</a>	Informative References . . . . .	<a href="#">30</a>
<a href="#">Appendix A.</a>	Design Considerations . . . . .	<a href="#">31</a>
<a href="#">Appendix B.</a>	Open Issues . . . . .	<a href="#">32</a>
<a href="#">Appendix C.</a>	Change Log . . . . .	<a href="#">33</a>
<a href="#">Appendix D.</a>	Acknowledgements . . . . .	<a href="#">35</a>
	Authors' Addresses . . . . .	<a href="#">35</a>

## **[1.](#) Introduction**

Software that provides email service via Internet Message Access Protocol (IMAP) [[RFC3501](#)], Post Office Protocol (POP) [[RFC1939](#)] and/or Simple Mail Transfer Protocol (SMTP) Submission [[RFC6409](#)] usually has Transport Layer Security (TLS) [[RFC5246](#)] support but often does not use it in a way that maximizes end-user confidentiality. This specification proposes changes to email software and deployments intended to increase the use of TLS and record when that use occurs.

In brief, this memo now recommends that:

- o MUAs associate a confidentiality assurance level with each mail account, and the default level requires use of TLS with certificate validation for all TCP connections;
- o TLS on a well-known port ("Implicit TLS") be supported for IMAP, POP, and SMTP Submission [[RFC6409](#)] for all electronic mail user agents (MUAs), servers, and service providers;
- o MUAs and mail protocol servers cooperate (via mechanisms defined in this specification) to upgrade security/privacy feature use and record/indicate that usage appropriately.



Improved use of TLS with SMTP for message relaying is described in a separate document [[I-D.ietf-dane-smtp-with-dane](#)].

The recommendations in this memo do not replace the functionality of, and are not intended as a substitute for, end-to-end encryption of electronic mail.

This draft is subject to change. Implementation of this proposal is not recommended at this time. Please discuss this proposal on the ietf-uta mailing list.

## **2. Conventions and Terminology Used in This Document**

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 [[RFC2119](#)].

This specification expresses syntax using the Augmented Backus-Naur Form (ABNF) as described in [[RFC5234](#)], including the core rules in [Appendix B](#) and rules from [[RFC5322](#)].

In examples, "C:" and "S:" indicate lines sent by the client and server respectively. If a single "C:" or "S:" label applies to multiple lines, then the line breaks between those lines are for editorial clarity only and are not part of the actual protocol exchange.

## **3. Mail Account Confidentiality Assurance Level**

A "mail account" refers to the network services an end user uses to read, submit and manage email communications on the Internet. This typically involves at least one mail access server (IMAP or POP) and at least one SMTP submission server. An end users uses a mail user agent (MUA) to access a mail account and most MUAs support one or more mail accounts. This document uses the term "confidentiality assurance level" to indicate the degree to which the network connections between an MUA and a mail account have confidentiality protection from both passive and active attackers on the network.

The configuration necessary for a mail account includes an email address, connection information and authentication credentials for network services. MUAs compliant with this specification **MUST** also associate a confidentiality assurance level with each mail account. MUAs **MUST** implement a high confidentiality assurance level as described in the next section.

MUAs **SHOULD** continuously indicate to the user the confidentiality assurance level of the account currently in use when reading,



submitting and managing mail (e.g., via a lock icon, background colors and indications similar to those commonly used in web browsers for a similar purpose) and SHOULD indicate the confidentiality assurance level for each account whenever displaying a list of mail accounts. Note that the displayed confidentiality assurance level could be higher than the level set at account configuration but never lower. If multiple active connections are associated with an account or view, the indication should match the level provided by the least confidential connection.

Account configuration occurs when an MUA is first used to access a particular service, when a user wishes to access or submit mail through servers in addition to those specified or found during first use, or when a user explicitly requests to change account configuration parameters such as server names, user names, passwords, client certificates, etc. Account configuration can be entirely manual (entering server names explicitly) or partially automated via a mechanism such as DNS SRV records [[RFC6186](#)]. MUAs SHOULD use the high confidentiality assurance level as the default for newly configured accounts.

### **3.1. High Confidentiality Assurance**

A mail account has a high confidentiality assurance when the following conditions are met on all TCP server connections associated with an account. This includes connections to POP, IMAP and SMTP submission servers as well as any other associated protocols defined now or in the future. Examples of protocols associated with a mail account include managesieve [[RFC5804](#)] and MTQP [[RFC3887](#)].

- o TCP connections MUST attempt to negotiate TLS via either Implicit TLS [Section 4](#) or STARTTLS.
- o MUAs MUST implement [[I-D.melnikov-email-tls-certs](#)] and PKIX [[RFC5280](#)].
- o MUAs MAY implement DANE [[RFC6698](#)].
- o User agents MUST abort a TLS session if the TLS negotiation fails or the server's certificate or identity fails to verify. A user may reconfigure the account to lower the expected level of confidentiality if he/she chooses. Reduction of expected account confidentiality MUST NOT be done on a click-through basis.

The end user is part of the system that protects the user's privacy and security. As a result, it's critical not to present the end user with a simple action that reduces their privacy in response to certificate validation failure. An MUA which offers a user actions





such as "connect anyway", "trust certificate for future connections" or "lower confidentiality assurance for this account" in response to certificate validation failure is not providing a high confidentiality assurance as defined in this section and thus does not comply with this document. Examples of acceptable actions to offer would be "work offline", "try again later", and "open service provider status web page".

### **3.2. Certificate Pinning**

MUAs MAY implement certificate pinning as part of account setup, but MUST NOT offer this as an option in response to a failed certificate validation for an existing account. Certificate pinning occurs when the user agent saves a server certificate with the account settings and trusts that certificate for subsequent connections to that server. An MUA that allows certificate pinning MUST NOT allow a certificate pinned for one account to validate connections for other accounts.

A pinned certificate is subject to a man-in-the-middle attack at account setup time, and lacks a mechanism to revoke or securely refresh the certificate. Therefore use of a pinned certificate does not provide a high confidentiality assurance and an MUA MUST NOT indicate a high level for an account or connection using a pinned certificate.

### **3.3. No Confidentiality Assurance**

MUAs MAY implement a no confidentiality assurance level for accounts. At this level, the MUA MUST attempt to negotiate TLS, but MAY ignore server certificate validation failures. MUAs MAY support use of connections without TLS, but if they do they SHOULD attempt TLS first if available and MUST implement code to reconnect without TLS if TLS negotiation fails for reasons other than server certificate validity.

Note that if the TLS certificate is not successfully validated as described in [Section 3.1](#) or a version of SSL/TLS prior to TLS 1.0 is used, the client MUST NOT present a high confidentiality indication for the account or connection.

### **3.4. Other Confidentiality Assurance Levels**

This specification is not intended to limit experimentation and innovation with respect to user privacy. As a result more confidentiality assurance levels are permitted. However, levels below "no confidentiality assurance" described in the previous section are discouraged and implementers are cautioned that end users may be confused by too many confidentiality assurance levels.



## **4. Implicit TLS**

Previous standards for use of email protocols with TLS used the STARTTLS mechanism: [RFC2595], [RFC3207], and [RFC3501]. With STARTTLS, the client establishes a clear text application session and determines whether to issue a STARTTLS command based on server capabilities and client configuration. If the client issues a STARTTLS command, a TLS handshake follows that can upgrade the connection. While this mechanism has been deployed, an alternate mechanism where TLS is negotiated immediately at connection start on a separate port (referred to in this document as "Implicit TLS") has been deployed more successfully. To increase use of TLS, this specification recommends use of implicit TLS by new POP, IMAP and SMTP Submission software.

### **4.1. Implicit TLS for POP**

When a TCP connection is established for the "pop3s" service (default port 995), a TLS handshake begins immediately. Clients MUST implement the certificate validation mechanism described in [I-D.melnikov-email-tls-certs]. Once the TLS session is established, POP3 [RFC1939] protocol messages are exchanged as TLS application data for the remainder of the TCP connection. After the server sends a +OK greeting, the server and client MUST enter AUTHORIZATION state, even if client credentials were supplied during the TLS handshake.

See [Section 9.1.1](#) for additional information on client certificate authentication. See [Section 11.3](#) for port registration information.

### **4.2. Implicit TLS for IMAP**

When a TCP connection is established for the "imaps" service (default port 993), a TLS handshake begins immediately. Clients MUST implement the certificate validation mechanism described in [RFC3501] and SHOULD implement the certificate validation mechanism described in [I-D.melnikov-email-tls-certs]. Once the TLS session is established, IMAP [RFC3501] protocol messages are exchanged as TLS application data for the remainder of the TCP connection. If client credentials were provided during the TLS handshake that the server finds acceptable, the server MAY issue a PREAUTH greeting in which case both the server and client enter AUTHENTICATED state. If the server issues an OK greeting then both server and client enter NOT AUTHENTICATED state.

See [Section 9.1.1](#) for additional information on client certificate authentication. See [Section 11.4](#) for port registration information.



#### **4.3. Implicit TLS for SMTP Submission**

When a TCP connection is established for the "submissions" service (default port 465), a TLS handshake begins immediately. Clients MUST implement the certificate validation mechanism described in [[I-D.melnikov-email-tls-certs](#)]. Once a TLS session is established, message submission protocol data [[RFC6409](#)] is exchanged as TLS application data for the remainder of the TCP connection. (Note: the "submissions" service name is defined in [section 10.3](#) of this document, and follows the usual convention that the name of a service layered on top of Implicit TLS consists of the name of the service as used without TLS, with an "s" appended.)

Note that the submissions port provides access to a Mail Submission Agent (MSA) as defined in [[RFC6409](#)] so requirements and recommendations for MSAs in that document apply to the submissions port, including the requirement to implement SMTP AUTH [[RFC4954](#)].

See [Section 9.1.1](#) for additional information on client certificate authentication. See [Section 11.5](#) for port registration information.

#### **4.4. Implicit TLS Connection Closure for POP, IMAP and SMTP**

When a client or server wishes to close the connection, it SHOULD initiate the exchange of TLS close alerts before TCP connection termination. The client MAY, after sending a TLS close alert, gracefully close the TCP connection without waiting for a TLS response from the server.

### **5. Email Security Upgrading Using Security Latches**

Once an improved email security or privacy mechanism is deployed and ready for general use, it is desirable to continue using it for all future email service. For example, TLS is widely deployed in email software, but use of TLS is often not required. At the time this is written, deployed mail user agents (MUAs) [[RFC5598](#)] usually make a determination if TLS is available when an account is first configured and may require use of TLS with that account if and only if it was initially available. If the service provider makes TLS available after initial client configuration, many MUAs will not notice the change.

Alternatively, a security feature may be purely opportunistic and thus subject to downgrade attacks. For example, at the time this was written, most TLS stacks that support TLS 1.2 will use an older TLS version if the peer does not support TLS 1.2 and some do so without alerting the client of the reduced security. Thus a variety of active attacks could cause the loss of TLS 1.2 benefits. Only if



client policy is upgraded to require TLS 1.2 can the client prevent all downgrade attacks. However, this sort of security policy upgrade will be ignored by most users unless it is automated.

This section describes a mechanism, called "security latches", which is designed to permit an MUA to recognize when a service provider has committed to provide certain server security features, and that it's safe for the client to change its configuration for that account to require that such features be present in future sessions with that server. When an MUA implements both confidentiality assurance levels and security latches, then both the end-user and the service provider independently have the ability to improve the end-user's privacy.

Note that security latches are a mechanism similar to HTTP Strict Transport Security (HSTS) [[RFC6797](#)] but are extensible.

### **5.1. Email Security Tags**

Each security latch is given a name known as an email security tag. An email security tag is a short alphanumeric token that represents a security facility that can be used by an IMAP, POP or SMTP Submission session. When a server advertises a security tag it is making a commitment to support that security facility indefinitely and recommending that the client save that security tag with the account configuration and require that security feature for future connections to that server. When a security tag is saved by the client in this way, it is then considered latched. For the "tls10" and/or "tls12" tags, the client SHOULD refuse to connect to the server unless the appropriate level of TLS is successfully negotiated. The client SHOULD NOT latch these two tags until TLS has been successfully negotiated as described in the tag definition. If the tags are advertised within an appropriate TLS-protected connection, the client SHOULD latch these tags. Other security tags are latched if they are advertised by the server, TLS is active and the client successfully authenticates the server with the TLS session. Once a security tag is latched, all subsequent connections to that host require that security feature. For this confidentiality protection to work as desired clients MUST NOT offer a click-through-to-connect action when unable to achieve connection security matching the latched security tags.

An identifier for a security tag has the following formal syntax:

```
security-tag = ALPHA *63(ALPHA / DIGIT / "-" / "_")
```





## 5.2. Initial Set of Email Security Tags

This section describes an initial set of email security tags. The IANA Considerations [Section 11](#) defines a registry so that more tags can be defined in the future. The initial set of tags are defined in [Section 11.2](#) and include `tls10`, `tls12`, `tls-cert` and `tls-dane-tlsa`.

## 5.3. Server DEEP Status

Servers supporting this extension MUST advertise a DEEP status. This status includes a list of security-tags the server administrator has explicitly configured as recommended for use by end-users (the list MAY be empty), an optional https Uniform Resource Locator (URL) [[RFC2818](#)] that the client can save and subsequently resolve for the user in the event of a security connection problem, and the DEEP status can be extended by future updates to this specification. DEEP status has the following formal syntax:

```
EXTCHAR      = 0x20-21 / 0x23-2E / 0x30-3B / 0x3D-40
               / 0x5B-60 / 0x7B-7E
               ; printable characters excluding " \ < and ALPHA

deep-extend   = EXTCHAR *(EXTCHAR / ALPHA / "<")
               ; clients MUST ignore, for future extensibility

deep-status   = [deep-tag *(SP deep-tag)]

deep-tag      = deep-https / security-tag / deep-extend

deep-https    = "<" <URI from RFC 3986 with https scheme> ">"
```

The syntax for a Uniform Resource Identifier (URI) is defined in [[RFC3986](#)]. Protocol extensions to advertise DEEP status are defined in [Section 7](#).

If the client successfully negotiates TLS and authenticates the server (e.g., via `tls-cert`, `tls-dane-tlsa` or SCRAM-SHA1-PLUS with channel bindings [[RFC5802](#)]), then the client SHOULD record the server's DEEP status information in the account configuration with the server's hostname. Otherwise, the client SHOULD ignore the server-provided DEEP status except for the `"tls10"` and `"tls12"` security tags.

## 5.4. Email Security Tag Latch Failures

When a security tag latch has been set for connections from a client to a server and the property identified by that tag is no longer available, this results in a connection failure. An MUA SHOULD



inform the user of a potential threat to their confidentiality and offer to resolve a previously-recorded DEEP status https URL if one is available. MUAs are discouraged from offering a lightweight option to reset or ignore latches as this defeats the benefit they provide to end users.

## **6. Recording TLS Cipher Suite in Received Header**

The ESMTPS transmission type [[RFC3848](#)] provides trace information that can indicate TLS was used when transferring mail. However, TLS usage by itself is not a guarantee of confidentiality or security. The TLS cipher suite provides additional information about the level of security made available for a connection. This defines a new SMTP "tls" Received header additional-registered-clause that is used to record the TLS cipher suite that was negotiated for the connection. The value included in this additional clause SHOULD be the registered cipher suite name (e.g., TLS\_DHE\_RSA\_WITH\_AES\_128\_CBC\_SHA) included in the TLS cipher suite registry. In the event the implementation does not know the name of the cipher suite (a situation that should be remedied promptly), a four-digit hexadecimal cipher suite identifier MAY be used. The ABNF for the field follows:

```
tls-cipher-clause = CFWS "tls" FWS tls-cipher

tls-cipher         = tls-cipher-suite-name / tls-cipher-suite-hex

tls-cipher-name    = ALPHA *(ALPHA / DIGIT / "_")
                   ; as registered in IANA cipher suite registry

tls-cipher-hex     = "0x" 4HEXDIG
```

## **7. Extensions for DEEP Status and Reporting**

This memo defines optional mechanisms for use by MUAs to communicate DEEP status to servers and for servers to advertise available latches. One purpose of such mechanisms is to permit servers to determine which and how many clients have latched security facilities, and thus, to permit operators to be aware of potential impact to their users should support for such facilities be changed. For IMAP, the existing ID command is extended to provide this capability. For SMTP Submission, a new CLIENT command is defined. No similar mechanism is defined for POP in this version of the memo to keep POP simpler, but one may be added in the future if deemed necessary.

In addition, for each of IMAP, POP, and SMTP, a new DEEP capability is defined so the client can access the server's DEEP status.



### **7.1. IMAP DEEP Extension**

When an IMAP server advertises the DEEP capability, that indicates the IMAP server implements IMAP4 ID [[RFC2971](#)] with additional field values defined here. This is grouped with the ID command because that is the existing IMAP mechanism for clients to report data for server logging, and provides a way for the server to report the DEEP status.

**deep** From server to client, the argument to this ID field is the server DEEP status. Servers **MUST** provide this information in response to an ID command.

**latch** From client to server, this is a space-separated list of security tags the client has latched for this server. Servers **MAY** record this information so administrators know the expected latch-related security properties of the client and can thus act to avoid security latch failures (e.g., by renewing server certificates on time, etc).

**latch-fail** From client to server, a space-separated list including one or more security tag the client has latched that the client was unable to achieve. This allows clients to report errors to the server prior to terminating the connection to the server in the event an acceptable security level is unavailable.

**security-tags** From client to server, this is a space-separated list of security tags the client supports that are not latched.

**tls** Server-side IMAP proxies that accept TLS connections from clients and connect in-the-clear over a fully private secure network to the server **SHOULD** use this field to report the tls-cipher (syntax as defined in [Section 6](#)) to the server.

IMAP clients **SHOULD** use the IMAP ID command to report latch failures and determine the server DEEP status. Clients **MAY** use the ID command to report other latch or security tag information. IMAP servers **MUST** implement the ID command at least to report DEEP status to clients.



```
<client connected to port 993 and negotiated TLS successfully>
S: * OK [CAPABILITY IMAP4rev1 DEEP ID AUTH=PLAIN
    AUTH=SCRAM-SHA-1] hello
C: a001 ID ("name" "Demo Mail" "version" "1.5" "latch"
    "tls10 tls-cert" "security-tags" "tls12")
S: * ID ("name" "Demo Server" "version" "1.7" "deep-status"
    "<https://www.example.com/privacy-support.html>")
S: a001 OK ID completed
```

#### Example 1

This example shows a client that successfully negotiated TLS version 1.0 or later and verified the server's certificate as required by IMAP. The client supports TLS 1.2. However, even if the client successfully negotiated TLS 1.2, it will not latch that security tag automatically because the server did not advertise that tag. If the client successfully validated the server certificate, it will latch the provided URL.

```
<client connected to port 993 and negotiated TLS successfully>
S: * OK [CAPABILITY IMAP4rev1 DEEP ID AUTH=PLAIN
    AUTH=SCRAM-SHA-1] hello
C: a001 ID ("name" "Demo Mail" "version" "1.5" "latch-failure"
    "tls-cert")
S: * ID ("name" "Demo Server" "version" "1.7" "deep-status"
    "tls10 <https://www.example.com/privacy-support.html>")
S: a001 OK ID completed
C: a002 LOGOUT
```

#### Example 2

This example shows a client that negotiated TLS, but was unable to verify the server's certificate. The latch-failure informs the server of this problem, at which point the client can disconnect. If the client had previously latched a URI for security problems from this server, it could offer to resolve that URI. However, the deep-status in this exchange is ignored due to the latch failure.





```
<IMAP Proxy connected over private network on port 143, there is
a client connected to the proxy on port 993 that negotiated TLS>
S: * OK [CAPABILITY IMAP4rev1 DEEP ID AUTH=PLAIN
    AUTH=SCRAM-SHA-1] hello
C: a001 ID ("name" "Demo Mail" "version" "1.5" "latch"
    "tls10 tls-cert" "security-tags" "tls12"
    "tls" "TLS_RSA_WITH_AES_128_CBC_SHA")
S: * ID ("name" "Demo Server" "version" "1.7" "deep-status"
    "tls10 tls-cert <https://www.example.com/support.html>")
S: a001 OK ID completed
```

### Example 3

This example shows the connection from an IMAP proxy to a back-end server. The client connected to the proxy and sent the ID command shown in example 1, and the proxy has added the "tls" item to the ID command so the back-end server can log the cipher suite that was used on the connection from the client.

## 7.2. POP DEEP Extension

POP servers supporting this specification MUST implement the POP3 extension mechanism [[RFC2449](#)]. POP servers MUST advertise the DEEP capability with an argument indicating the server's DEEP status.

```
<client connected to port 995 and negotiated TLS successfully>
S: +OK POP server ready
C: CAPA
S: +OK Capability list follows
S: TOP
S: SASL PLAIN SCRAM-SHA-1
S: RESP-CODES
S: PIPELINING
S: UIDL
S: DEEP tls10 tls12 <https://www.example.com/privacy-support.html>
S: .
```

### Example 4

After verifying the TLS server certificate and issuing CAPA, the client can latch any or all of the DEEP status. If the client connects to this same server later and has a security failure, the client can direct the user's browser to the previously-latched URI where the service provider may provide advice to the end user.



### 7.3. SMTP DEEP Extension

SMTP Submission servers supporting this specification MUST implement the DEEP SMTP extension. The name of this extension is DEEP. The EHLO keyword value is DEEP and the deep-status ABNF is the syntax of the EHLO keyword parameters. This does not add parameters to the MAIL FROM or RCPT TO commands. This also adds a CLIENT command to SMTP which is used to report client information to the server. The formal syntax for the command follows:

```

deep-cmd           = "CLIENT" 1*(SP deep-parameter)

deep-parameter     = name / version / latch / latch-fail
                    / security-tags / tls / future-extension

name               = "name=" esmtp-value

version            = "version=" esmtp-value

latch              = "latch=" security-tag *("," security-tag)

latch-fail         = "latch-fail=" security-tag
                    *("," security-tag)

security-tags      = "security-tags=" security-tag
                    *("," security-tag)

tls                = "tls=" tls-cipher

future-extension   = esmtp-param

esmtp-param        = <as defined in RFC 5321>

esmtp-value        = <as defined in RFC 5321>

```

The CLIENT command parameters listed here have the same meaning as the parameters used in the IMAP DEEP extension ([Section 7.1](#)). The server responds to the CLIENT command with a "250" if the command has correct syntax and a "501" if the command has incorrect syntax.



```
<client connected to port 465 and negotiated TLS successfully>
S: 220 example.com Demo SMTP Submission Server
C: EHLO client.example.com
S: 250-example.com
S: 250-8BITMIME
S: 250-PIPELINING
S: 250-DSN
S: 250-AUTH PLAIN LOGIN
S: 250-DEEP tls10 tls-cert <https://www.example.com/status.html>
S: 250-BURL imap
S: 250 SIZE 0
C: CLIENT name=demo_submit version=1.5 latch=tls10,tls-cert
    security-tags=tls12
S: 250 OK
```

#### Example 5

### 7.4. SMTP Error Extension

Although this document focuses on SMTP Submission, it is possible to use security latches for SMTP transport as well. When MTA transport fails due to a security latch, the MTA MUST use the SMTP enhanced status code X.7.TBD (RFC Editor note: update this TBD). The SMTP notary response [[RFC3464](#)] for a security latch failure MUST include an additional "SMTP-Security-Latch" recipient-specific header field that includes a space-delimited list including one or more security latch that failed. The ABNF for this new field follows:

```
CFWS                = <defined in RFC 5322>

FWS                 = <defined in RFC 5322>

smtp-security-latch = "SMTP-Security-Latch:" CFWS
                      security-tag *(FWS security-tag)
```

### 8. Use of SRV records in Establishing Configuration

This section updates [[RFC6186](#)] by changing the preference rules and adding a new SRV service label `_submissions._tcp` to refer to Message Submission with implicit TLS.

User-configurable MUAs SHOULD support use of [[RFC6186](#)] for account setup. However, when using configuration information obtained by this method, MUAs SHOULD default to a high confidentiality assurance level, unless the user has explicitly requested reduced confidentiality. This will have the effect of causing the MUA to ignore advertised configurations that do not support TLS, even when



those advertised configurations have a higher priority than other advertised configurations.

When using [[RFC6186](#)] configuration information, Mail User Agents SHOULD NOT automatically establish new configurations that do not require TLS for all servers, unless there are no advertised configurations using TLS. If such a configuration is chosen, prior to attempting to authenticate to the server or use the server for message submission, the MUA SHOULD warn the user that traffic to that server will not be encrypted and that it will therefore likely be intercepted by unauthorized parties. The specific wording is to be determined by the implementation, but it should adequately capture the sense of risk given the widespread incidence of mass surveillance of email traffic.

When establishing a new configuration for connecting to an IMAP, POP, or SMTP Submission server, an MUA SHOULD NOT blindly trust SRV records unless they are signed by DNSSEC and have a valid signature. Instead, the MUA SHOULD warn the user that the DNS-advertised mechanism for connecting to the server is not authenticated, and request the user to manually verify the connection details by reference to his or her mail service provider's documentation.

Similarly, an MUA MUST NOT consult SRV records to determine which servers to use on every connection attempt, unless those SRV records are signed by DNSSEC and have a valid signature. However, an MUA MAY consult SRV records from time to time to determine if an MSP's server configuration has changed, and alert the user if it appears that this has happened. This can also serve as a means to encourage users to upgrade their configurations to require TLS if and when their MSPs support it.

## **9. Implementation Requirements**

This section details requirements for implementations of electronic mail protocol clients and servers. A requirement for a client or server implementation to support a particular feature is not the same thing as a requirement that a client or server running a conforming implementation be configured to use that feature. Requirements for Mail Service Providers (MSPs) are distinct from requirements for protocol implementations, and are listed in a separate section.

### **9.1. All Implementations (Client and Server)**

These requirements apply to MUAs as well as POP, IMAP and SMTP Submission servers.





- o All implementations MUST be configurable to support implicit TLS using the TLS 1.2 protocol or later [[RFC5246](#)] including support for the mandatory-to-implement TLS 1.2 cipher suite TLS\_RSA\_WITH\_AES\_128\_CBC\_SHA.
- o IMAP implementations MUST support the IMAP4rev1 mandatory-to-implement cipher suite TLS\_RSA\_WITH\_RC4\_128\_MD5 for any connections made or received via IMAP although this MAY be disabled by default.
- o All implementations MUST be configurable to require TLS before performing any operation other than capability discovery and STARTTLS.

#### **9.1.1. Client Certificate Authentication**

MUAs and mail servers MAY implement client certificate authentication on the implicit TLS port. Servers MUST NOT request a client certificate during the TLS handshake unless the server is configured to accept some client certificates as sufficient for authentication and the server has the ability to determine a mail server authorization identity matching such certificates. How to make this determination is presently implementation specific. Clients MUST NOT provide a client certificate during the TLS handshake unless the server requests one and the client has determined the certificate can be safely used with that specific server, OR the client has been explicitly configured by the user to use that particular certificate with that server. How to make this determination is presently implementation specific. If the server accepts the client's certificate as sufficient for authorization, it MUST enable the SASL EXTERNAL [[RFC4422](#)] mechanism. An IMAPS server MAY issue a PREAUTH greeting instead of enabling SASL EXTERNAL. A client supporting client certificate authentication with implicit TLS MUST implement the SASL EXTERNAL [[RFC4422](#)] mechanism using the appropriate authentication command (AUTH for POP3 [[RFC5034](#)], AUTH for SMTP Submission [[RFC4954](#)], AUTHENTICATE for IMAP [[RFC3501](#)]).

#### **9.2. Mail Server Implementation Requirements**

These requirements apply to servers that implement POP, IMAP or SMTP Submission.

- o Servers MUST implement the DEEP extension described in [Section 7](#)
- o IMAP and SMTP submission servers SHOULD implement and be configurable to support STARTTLS. This enables discovery of new TLS availability, and can increase usage of TLS by legacy clients.



- o Servers MUST NOT advertise STARTTLS if it is unlikely to succeed based on server configuration (e.g., there is no server certificate installed).
- o SMTP message submission servers that have negotiated TLS SHOULD add a Received header field to the message including the tls clause described in [Section 6](#).
- o Servers MUST be configurable to include the TLS cipher information in any connection or user logging or auditing facility they provide.

### **9.3. Mail User Agent Implementation Requirements**

This section describes requirements on Mail User Agents (MUAs) using IMAP, POP, and/or Submission protocols. Note: Requirements pertaining to use of Submission servers are also applicable to use of SMTP servers (e.g., port 25) for mail submission.

- o User agents SHOULD indicate to users at configuration time, the expected level of confidentiality based on appropriate security inputs such as which security latches are pre-set, the number of trust anchors, certificate validity, use of an extended validation certificate, TLS version supported, and TLS cipher suites supported by both server and client. This indication SHOULD also be present when editing or viewing account configuration.
- o MUAs SHOULD detect when STARTTLS and/or implicit TLS becomes available for a protocol and set the tls10 latch if the server advertises the tls10 security tag after a successful TLS negotiation.
- o Whenever requested to establish any configuration that does not require both TLS and server certificate verification to talk to a server or account, an MUA SHOULD warn its user that his or her mail traffic (including password, if applicable) will be exposed to attackers, and give the user an opportunity to abort the connection prior to transmission of any such password or traffic.
- o MUAs SHOULD implement the "tls12" security latch (the TLS library has to provide an API that controls permissible TLS versions and communicates the negotiated TLS protocol version to the application for this to be possible).
- o See [Section 3](#) for additional requirements.



#### **9.4. Non-configurable MUAs and nonstandard access protocols**

MUAs which are not configurable to use user-specified servers **MUST** implement TLS or similarly other strong encryption mechanism when communicating with their mail servers. This generally applies to MUAs that are pre-configured to operate with one or more specific services, whether or not supplied by the vendor of those services.

MUAs using protocols other than IMAP, POP, and Submission to communicate with mail servers, **MUST** implement TLS or other similarly robust encryption mechanism in conjunction with those protocols.

#### **9.5. DEEP Compliance for Anti-Virus/Anti-Spam Software and Services**

There are multiple ways to connect an Anti-Virus and/or Anti-Spam (AVAS) service to a mail server. Some mechanisms, such as the de-facto milter protocol do not impact DEEP. However, some services use an SMTP relay proxy that intercepts mail at the application layer to perform a scan and proxy to the real MTA. Deploying AVAS services in this way can cause many problems [[RFC2979](#)] including direct interference with DEEP and confidentiality or security reduction. An AVAS product or service is considered DEEP compliant if all IMAP, POP and SMTP-related software it includes is DEEP compliant and it advertises and supports all security latches that the actual MTA advertises.

### **10. Mail Service Provider Requirements**

This section details requirements for providers of IMAP, POP, and/or SMTP submission services, for providers who claim to conform to this specification.

#### **10.1. Server Requirements**

Mail Service Providers **MUST** use server implementations that conform to this specification.

#### **10.2. MSPs **MUST** provide Submission Servers**

This document updates the advice in [[RFC5068](#)] by making Implicit TLS on port 465 the preferred submission port.

Mail Service Providers that accept mail submissions from end-users using the Internet Protocol **MUST** provide one or more SMTP Submission servers for this purpose, separate from the SMTP servers used to process incoming mail. Those submission servers **MUST** be configured to support Implicit TLS on port 465 and **SHOULD** support STARTTLS if port 587 is used.



MSPs MAY also support submission of messages via one or more designated SMTP servers to facilitate compatibility with legacy MUAs.

Discussion: SMTP servers used to accept incoming mail or to relay mail are expected to accept mail in cleartext. This is incompatible with the purpose of this memo which is to encourage encryption of traffic between mail servers. There is no such requirement for mail submission servers to accept mail in cleartext or without authentication. For other reasons, use of separate SMTP submission servers has been best practice for many years.

### **10.3. TLS Server Certificate Requirements**

MSPs MUST maintain valid server certificates for all servers. Those server certificates SHOULD present DNS-IDs and SRV-IDs conforming to [\[RFC6125\]](#) and which will be recognized by MUAs meeting the requirements of that specification. In addition, those server certificates MAY provide other DNS-IDs, SRV-IDs, or CN-IDs needed for compatibility with existing MUAs.

If a protocol server provides service for more than one mail domain, it MAY use a separate IP address for each domain and/or a server certificate that advertises multiple domains. This will generally be necessary unless and until it is acceptable to impose the constraint that the server and all clients support the Server Name Indication extension to TLS [\[RFC6066\]](#).

### **10.4. Recommended DNS records for mail protocol servers**

This section discusses not only the DNS records that are recommended, but also implications of DNS records for server configuration and TLS server certificates.

#### **10.4.1. MX records**

It is recommended that MSPs advertise MX records for handling of inbound mail (instead of relying entirely on A or AAAA records), and that those MX records be signed using DNSSEC. This is mentioned here only for completeness, as handling of inbound mail is out of scope for this document.

#### **10.4.2. SRV records**

MSPs SHOULD advertise SRV records to aid MUAs in determination of proper configuration of servers, per the instructions in [\[RFC6186\]](#).

MSPs SHOULD advertise servers that support Implicit TLS in preference to those which support cleartext and/or STARTTLS operation.





#### **10.4.3. TLSA records**

MSPs SHOULD advertise TLSA records to provide an additional trust anchor for public keys used in TLS server certificates. However, TLSA records MUST NOT be advertised unless they are signed using DNSSEC.

#### **10.4.4. DNSSEC**

All DNS records advertised by an MSP as a means of aiding clients in communicating with the MSP's servers, SHOULD be signed using DNSSEC.

#### **10.5. MSP Server Monitoring**

MSPs SHOULD regularly and frequently monitor their various servers to make sure that: TLS server certificates remain valid and are not about to expire, TLSA records match the public keys advertised in server certificates, are signed using DNSSEC, server configurations are consistent with SRV advertisements, and DNSSEC signatures are valid and verifiable. Failure to detect expired certificates and DNS configuration errors in a timely fashion can result in significant loss of service for an MSP's users and a significant support burden for the MSP.

#### **10.6. Advertisement of DEEP status**

MSPs SHOULD advertise a DEEP status that includes `tls10`, `tls-cert` and an HTTPS URL that can be used to inform clients of service outages or problems impacting client confidentiality. Note that advertising `tls-cert` is a commitment to maintain and renew server certificates.

#### **10.7. Require TLS**

New servers and services SHOULD be configured to require TLS unless it's necessary to support legacy clients or existing client configurations.

#### **10.8. Changes to Internet Facing Servers**

When an MSP changes the Internet Facing Servers providing mail access and mail submission services, including SMTP-based spam/virus filters, it is generally necessary to support the same and/or a newer version of TLS and the same security tags that were previously advertised.



## **11. IANA Considerations**

### **11.1. Security Tag Registry**

IANA shall create (has created) the registry "Email Security Tags". This registry is a single table and will use an expert review process [[RFC5226](#)]. Each registration will contain the following fields:

Name: The name of the security tag. This follows the security-tag ABNF.

Description: This describes the meaning of the security tag and the conditions under which the tag is latched.

Intended Usage: One of COMMON, LIMITED USE or OBSOLETE.

Reference: Optional reference to specification.

Submitter: The identify of the submitter or submitters.

Change Controller: The identity of the change controller for the registration. This will be "IESG" in case of registrations in IETF-produced documents.

The expert reviewer will verify the tag name follows the ABNF, and that the description field is clear, unambiguous, does not overlap existing deployed technology, does not create security or privacy problems and appropriately considers interoperability issues. Email security tags intended for LIMITED USE have a lower review bar (interoperability and overlap issues are less of a concern). The reviewer may approve a registration, reject for a stated reason or recommend the proposal have standards track review due to importance or difficult subtleties.

Standards-track registrations may be updated if the relevant standards are updated as a consequence of that action. Non-standards-track entries may be updated by the listed change controller. The entry's name and submitter may not be changed. In exceptional cases, any aspect of any registered entity may be updated at the direction of the IESG (for example, to correct a conflict).

### **11.2. Initial Set of Security Tags**

This document defines four initial security tags for the security tag registry as follows:

Name: tls10



Description: This indicates TLS version 1.0 [[RFC2246](#)] or later was negotiated successfully including negotiation of a strong encryption layer with a symmetric key of at least 128 bits. This tag does not indicate the server certificate was valid. This tag is latched if the client sees this tag in the advertised server DEEP status provided after successfully negotiating TLS version 1.0 or later.

Intended Usage: COMMON

Reference: RFC XXXX (this document once published)

Submitter: Authors of this document

Change Controller: IESG

Name: tls12

Description: This indicates TLS version 1.2 [[RFC5246](#)] or later was negotiated successfully including negotiation of a strong encryption layer with a symmetric key of at least 128 bits. This tag does not indicate the server certificate was valid. This tag is latched if the client sees this tag in the advertised server DEEP status provided after successfully negotiating TLS version 1.2 or later.

Intended Usage: COMMON

Reference: RFC XXXX (this document once published)

Submitter: Authors of this document

Change Controller: IESG

Name: tls-cert

Description: This tag indicates that TLS was successfully negotiated and the server certificate was successfully verified by the client using PKIX [[RFC5280](#)] and the server certificate identity was verified using the algorithm appropriate for the protocol (see [Section 4](#)). This tag is latched if the client sees this tag in the advertised server DEEP status after successfully negotiating TLS and verifying the certificate and server identity.

Intended Usage: COMMON

Reference: RFC XXXX (this document once published)



Submitter: Authors of this document

Change Controller: IESG

Name: tls-dane-tlsa

Description: This tag indicates that TLS was successfully negotiated and the server certificate was successfully verified by the client using the procedures described in [[RFC6698](#)] and the server certificate identity was verified using the algorithm appropriate for the protocol (see [Section 4](#)). This tag is latched if the client sees this tag in the advertised server DEEP status after successfully negotiating TLS and verifying the certificate and server identity.

Intended Usage: COMMON

Reference: RFC XXXX (this document once published)

Submitter: Authors of this document

Change Controller: IESG

### **[11.3.](#) POP3S Port Registration Update**

IANA is asked to update the registration of the TCP well-known port 995 using the following template ([[RFC6335](#)]):

Service Name: pop3s  
Transport Protocol: TCP  
Assignee: IETF <iesg@ietf.org>  
Contact: IESG <iesg@ietf.org>  
Description: POP3 over TLS protocol  
Reference: RFC XXXX (this document once published)  
Port Number: 995

### **[11.4.](#) IMAPS Port Registration Update**

IANA is asked to update the registration of the TCP well-known port 993 using the following template ([[RFC6335](#)]):

Service Name: imaps  
Transport Protocol: TCP  
Assignee: IETF <iesg@ietf.org>  
Contact: IESG <iesg@ietf.org>  
Description: IMAP over TLS protocol  
Reference: RFC XXXX (this document once published)  
Port Number: 993





### **11.5. Submissions Port Registration**

IANA is asked to assign an alternate usage of port 465 in addition to the current assignment using the following template ([[RFC6335](#)]):

```
Service Name: submissions
Transport Protocol: TCP
Assignee: IETF <iesg@ietf.org>
Contact: IESG <iesg@ietf.org>
Description: Message Submission over TLS protocol
Reference: RFC XXXX (this document once published)
Port Number: 465
```

This is a one time procedural exception to the rules in [RFC 6335](#). This requires explicit IESG approval and does not set a precedent. Historically, port 465 was briefly registered as the "smtps" port. This registration made no sense as the SMTP transport MX infrastructure has no way to specify a port so port 25 is always used. As a result, the registration was revoked and was subsequently reassigned to a different service. In hindsight, the "smtps" registration should have been renamed or reserved rather than revoked. Unfortunately, some widely deployed mail software interpreted "smtps" as "submissions" [[RFC6409](#)] and used that port for email submission by default when an end-user requests security during account setup. If a new port is assigned for the submissions service, email software will either continue with unregistered use of port 465 (leaving the port registry inaccurate relative to de-facto practice and wasting a well-known port), or confusion between the de-facto and registered ports will cause harmful interoperability problems that will deter use of TLS for message submission. The authors believe both of these outcomes are less desirable than a wart in the registry documenting real-world usage of a port for two purposes. Although STARTTLS-on-port-587 has deployed, it has not replaced deployed use of implicit TLS submission on port 465.

### **11.6. DEEP IMAP Capability**

This document adds the DEEP capability to the IMAP capabilities registry. This is described in [Section 7.1](#).

### **11.7. DEEP POP3 Capability**

This document adds the DEEP capability to the POP3 capabilities registry.

CAPA Tag: DEEP

Arguments: deep-status



Added Commands: none

Standard Commands affected: none

Announced status / possible differences: both / may change after STLS

Commands Valid in States: N/A

Specification Reference: This document

Discussion: See [Section 7.2](#).

#### **[11.8](#). DEEP SMTP EHLO Keyword**

This document adds the DEEP EHLO Keyword to the SMTP Service Extension registry. This is described in [Section 7.3](#).

#### **[11.9](#). SMTP Enhanced Status Code**

This document adds the following entry to the "SMTP Enhanced Status Codes" registry created by [[RFC5248](#)].

Code: X.7.TBD (IANA, please assign the next available number)

Sample Text: Message Transport Failed due to missing required security.

Associated Basic Status Code: 450, 454, 550, 554

Description This code indicates an SMTP server was unable to forward a message to the next host necessary for delivery because it required a higher level of transport security or confidentiality than was available. The temporary form of this error is preferred in case the problem is caused by a temporary administrative error such as an expired server certificate.

Reference This document

Submitter C. Newman

Change Controller IESG

#### **[11.10](#). MAIL Parameters Additional-registered-clauses Sub-Registry**

This document adds the following entry to the "Additional-registered-clauses" sub-registry of the "MAIL Parameters" registry, created by [[RFC5321](#)]:



Clause Name: `tls`

Description: Indicates the TLS cipher suite used for a transport connection.

Syntax Summary: See `tls-cipher` ABNF [Section 6](#)

Reference: This document.

## **[12.](#) Security Considerations**

This entire document is about security considerations. In general, this is targeted to improve mail privacy and to mitigate threats external to the email system such as network-level snooping or interception; this is not intended to mitigate active attackers who have compromised service provider systems.

It could be argued that sharing the name and version of the client software with the server has privacy implications. Although providing this information is not required, it is encouraged so that mail service providers can more effectively inform end-users running old clients that they need to upgrade to protect their security, or know which clients to use in a test deployment prior to upgrading a server to have higher security requirements.

## **[13.](#) References**

### **[13.1.](#) Normative References**

- [RFC1939] Myers, J. and M. Rose, "Post Office Protocol - Version 3", STD 53, [RFC 1939](#), May 1996.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [RFC2449] Gellens, R., Newman, C., and L. Lundblade, "POP3 Extension Mechanism", [RFC 2449](#), November 1998.
- [RFC2818] Rescorla, E., "HTTP Over TLS", [RFC 2818](#), May 2000.
- [RFC2971] Showalter, T., "IMAP4 ID extension", [RFC 2971](#), October 2000.
- [RFC3207] Hoffman, P., "SMTP Service Extension for Secure SMTP over Transport Layer Security", [RFC 3207](#), February 2002.
- [RFC3501] Crispin, M., "INTERNET MESSAGE ACCESS PROTOCOL - VERSION 4rev1", [RFC 3501](#), March 2003.



- [RFC3464] Moore, K. and G. Vaudreuil, "An Extensible Message Format for Delivery Status Notifications", [RFC 3464](#), January 2003.
- [RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, [RFC 3986](#), January 2005.
- [RFC5034] Siemborski, R. and A. Menon-Sen, "The Post Office Protocol (POP3) Simple Authentication and Security Layer (SASL) Authentication Mechanism", [RFC 5034](#), July 2007.
- [RFC5068] Hutzler, C., Crocker, D., Resnick, P., Allman, E., and T. Finch, "Email Submission Operations: Access and Accountability Requirements", [BCP 134](#), [RFC 5068](#), November 2007.
- [RFC5234] Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, [RFC 5234](#), January 2008.
- [RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", [RFC 5246](#), August 2008.
- [RFC5248] Hansen, T. and J. Klensin, "A Registry for SMTP Enhanced Mail System Status Codes", [BCP 138](#), [RFC 5248](#), June 2008.
- [RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", [BCP 26](#), [RFC 5226](#), May 2008.
- [RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", [RFC 5280](#), May 2008.
- [RFC5321] Klensin, J., "Simple Mail Transfer Protocol", [RFC 5321](#), October 2008.
- [RFC5322] Resnick, P., Ed., "Internet Message Format", [RFC 5322](#), October 2008.
- [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](#), March 2011.





- [RFC6186] Daboo, C., "Use of SRV Records for Locating Email Submission/Access Services", [RFC 6186](#), March 2011.
- [RFC6409] Gellens, R. and J. Klensin, "Message Submission for Mail", STD 72, [RFC 6409](#), November 2011.
- [I-D.melnikov-email-tls-certs]  
Melnikov, A., "Updated TLS Server Identity Check Procedure for Email Related Protocols", [draft-melnikov-email-tls-certs-01](#) (work in progress), October 2013.
- [I-D.ietf-dane-smtp-with-dane]  
Dukhovni, V. and W. Hardaker, "SMTP security via opportunistic DANE TLS", [draft-ietf-dane-smtp-with-dane-02](#) (work in progress), October 2013.

### **13.2. Informative References**

- [RFC2246] Dierks, T. and C. Allen, "The TLS Protocol Version 1.0", [RFC 2246](#), January 1999.
- [RFC2595] Newman, C., "Using TLS with IMAP, POP3 and ACAP", [RFC 2595](#), June 1999.
- [RFC2979] Freed, N., "Behavior of and Requirements for Internet Firewalls", [RFC 2979](#), October 2000.
- [RFC3848] Newman, C., "ESMTP and LMTP Transmission Types Registration", [RFC 3848](#), July 2004.
- [RFC3887] Hansen, T., "Message Tracking Query Protocol", [RFC 3887](#), September 2004.
- [RFC4422] Melnikov, A. and K. Zeilenga, "Simple Authentication and Security Layer (SASL)", [RFC 4422](#), June 2006.
- [RFC4954] Siemborski, R. and A. Melnikov, "SMTP Service Extension for Authentication", [RFC 4954](#), July 2007.
- [RFC5598] Crocker, D., "Internet Mail Architecture", [RFC 5598](#), July 2009.
- [RFC5802] Newman, C., Menon-Sen, A., Melnikov, A., and N. Williams, "Salted Challenge Response Authentication Mechanism (SCRAM) SASL and GSS-API Mechanisms", [RFC 5802](#), July 2010.
- [RFC5804] Melnikov, A. and T. Martin, "A Protocol for Remotely Managing Sieve Scripts", [RFC 5804](#), July 2010.



- [RFC6066] Eastlake, D., "Transport Layer Security (TLS) Extensions: Extension Definitions", [RFC 6066](#), January 2011.
- [RFC6335] Cotton, M., Eggert, L., Touch, J., Westerlund, M., and S. Cheshire, "Internet Assigned Numbers Authority (IANA) Procedures for the Management of the Service Name and Transport Protocol Port Number Registry", [BCP 165](#), [RFC 6335](#), August 2011.
- [RFC6698] Hoffman, P. and J. Schlyter, "The DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS) Protocol: TLSA", [RFC 6698](#), August 2012.
- [RFC6797] Hodges, J., Jackson, C., and A. Barth, "HTTP Strict Transport Security (HSTS)", [RFC 6797](#), November 2012.

## **[Appendix A](#). Design Considerations**

This section is not normative.

The first version of this was written independently from [draft-moore-email-tls-00.txt](#); subsequent versions merge ideas from both drafts.

One author of this document was also the author of [RFC 2595](#) that became the standard for TLS usage with POP and IMAP, and the other author was perhaps the first to propose that idea. In hindsight both authors now believe that that approach was a mistake. At this point the authors believe that while anything that makes it easier to deploy TLS is good, the desirable end state is that these protocols always use TLS, leaving no need for a separate port for cleartext operation except to support legacy clients while they continue to be used. The separate port model for TLS is inherently simpler to implement, debug and deploy. It also enables a "generic TLS load-balancer" that accepts secure client connections for arbitrary foo-over-TLS protocols and forwards them to a server that may or may not support TLS. Such load-balancers cause many problems because they violate the end-to-end principle and the server loses the ability to log security-relevant information about the client unless the protocol is designed to forward that information (as this specification does for the cipher suite). However, they can result in TLS deployment where it would not otherwise happen which is a sufficiently important goal that it overrides the problems.

Although STARTTLS appears only slightly more complex than separate-port TLS, we again learned the lesson that complexity is the enemy of security in the form of the STARTTLS command injection vulnerability (CERT vulnerability ID #555316). Although there's nothing inherently wrong with STARTTLS, the fact it resulted in a common implementation



error (made independently by multiple implementers) suggests it is a less secure architecture than Implicit TLS.

[Section 7 of RFC 2595](#) critiques the separate-port approach to TLS. The first bullet was a correct critique. There are proposals in the http community to address that, and use of SRV records as described in [RFC 6186](#) resolves that critique for email. The second bullet is correct as well, but not very important because useful deployment of security layers other than TLS in email is small enough to be effectively irrelevant. The third bullet is incorrect because it misses the desirable option of "use and latch-on TLS if available". The fourth bullet may be correct, but is not a problem yet with current port consumption rates. The fundamental error was prioritizing a perceived better design based on a mostly valid critique over real-world deployability. But getting security and confidentiality facilities actually deployed is so important it should trump design purity considerations.

## [Appendix B](#). Open Issues

There are many open issues with this document. Here is an attempt to enumerate some of them:

- o Port 465 is presently used for two purposes: for submissions by a large number of clients and service providers and for the "urd" protocol by one vendor. Actually documenting this current state is controversial as discussed in the IANA considerations section. However, there is no good alternative. Registering a new port for submissions when port 465 is widely used for that purpose already will just create interoperability problems. Registering a port that's only used if advertised by an SRV record ([RFC 6186](#)) would not create interoperability problems but would require all client and server deployments and software to change significantly which is contrary to the goal of promoting more TLS use. Encouraging use of STARTTLS on port 587 would not create interoperability problems, but is unlikely to have impact on current undocumented use of port 465 and makes the guidance in this document less consistent.
- o This document should reference [draft-ietf-uta-tls-bcp](#) and possibly other guidance documents. Suggested text on where/how to reference this and possibly other TLS guidance (e.g., must staple). would be welcome.
- o One author believes that the security latch model is complementary with [draft-ietf-dane-smtp-with-dane-02](#) but hasn't thought about the issues in depth. We welcome feedback on this point.



- o The two authors of this document and the author of [draft-melnikov-email-tls-certs](#) are willing to merge these two documents. However, it is undesirable to delay publication of either document so this will be done only if the latter document is not yet through IESG processing when this document is ready for the IESG.
- o It might make sense to split this in two or more documents if it's getting too long to evaluate in one IETF last call. In particular, it might make sense to put implementation requirements and service provider requirements in separate documents. The authors prefer to edit one document for now and defer discussion of splitting the document until all technical issues are resolved.
- o The use of SRV records [[RFC6186](#)] for account setup or refresh is presently not secure from DNS active attacks unless DNSSEC is used. As this document is now focusing on MUA security/privacy, discussing how to do SRV record account setup or account refresh securely, probably using DANE, would be in scope for this document. It has been suggested that we add this.
- o This document does not cover use of TLS with SMTP relay.

#### **[Appendix C](#). Change Log**

Changes since [draft-newman-email-deep-02](#):

- o Changed "privacy assurance" to "confidentiality assurance"
- o Changed "low privacy assurance" to "no confidentiality assurance"
- o Attempt to improve definition of confidentiality assurance level.
- o Add SHOULD indicate when MUA is showing list of mail accounts.
- o Add SHOULD NOT latch tls10, tls12 tags until TLS negotiated.
- o Removed sentence about deleting and re-creating the account in latch failure section.
- o Remove use of word "fallback" with respect to TLS version negotiation.
- o Added bullet about changes to Internet facing servers to MSP section.
- o minor wording improvements based on feedback

Changes since -01:





- o Updated abstract, introduction and document structure to focus more on mail user agent privacy assurance.
- o Added email account privacy section, also moving section on account setup using SRV records to that section.
- o Finished writing IANA considerations section
- o Remove provisional concept and instead have server explicitly list security tags clients should latch.
- o Added note that rules for the submissions port follow the same rules as those for the submit port.
- o Reference and update advice in [[RFC5068](#)].
- o Fixed typo in Client Certificate Authentication section.
- o Removed tls-pfs security latch and all mention of perfect forward secrecy as it was controversial.
- o Added reference to HSTS.

Changes since -00:

- o Rewrote introduction to merge ideas from [draft-moore-email-tls-00](#).
- o Added Implicit TLS section, Account configuration section and IANA port registration updates based on [draft-moore-email-tls-00](#).
- o Add protocol details necessary to standardize implicit TLS for POP/IMAP/submission, using ideas from [draft-melnikov-pop3-over-tls](#).
- o Reduce initial set of security tags based on feedback.
- o Add deep status concept to allow a window for software updates to be backed out before latches make that problematic, as well as to provide service providers with a mechanism they can use to assist customers in the event of a privacy failure.
- o Add DNS SRV section from [draft-moore-email-tls-00](#).
- o Write most of the missing IANA considerations section.
- o Rewrite most of implementation requirements section based more on [draft-moore-email-tls-00](#). Remove new cipher requirements for now because those may be dealt with elsewhere.



## **Appendix D. Acknowledgements**

Many thanks to Ned Freed for discussion of the initial latch concepts in this document. Thanks to Alexey Melnikov for [draft-melnikov-pop3-over-tls-02](#), which was the basis of the POP3 implicit TLS text. Thanks to Dan Newman and Alexey Melnikov for review feedback. Thanks to Paul Hoffman for interesting feedback in initial conversations about this idea.

### Authors' Addresses

Keith Moore  
Network Heretics  
PO Box 1934  
Knoxville, TN 37901  
US

Email: [moore@network-heretics.com](mailto:moore@network-heretics.com)

Chris Newman  
Oracle  
440 E. Huntington Dr., Suite 400  
Arcadia, CA 91006  
US

Email: [chris.newman@oracle.com](mailto:chris.newman@oracle.com)

