[MS-SMTP]:
NT LAN Manager (NTLM) Authentication: Simple Mail Transfer Protocol (SMTP) Extension

---

Revision Summary

Date	Revision History	Revision Class	Comments
07/20/2007	0.1	Major	MCPP Milestone 5 Initial Availability
09/28/2007	0.1.1	Editorial	Revised and edited the technical content.
10/23/2007	0.2	Minor	Updated to use data types in MS-DTYP.
11/30/2007	0.2.1	Editorial	Revised and edited the technical content.
01/25/2008	0.2.2	Editorial	Revised and edited the technical content.
03/14/2008	0.2.3	Editorial	Revised and edited the technical content.
05/16/2008	1.0	Major	Updated and revised the technical content.
06/20/2008	2.0	Major	Updated and revised the technical content.
07/25/2008	2.1	Minor	Updated the technical content.
08/29/2008	3.0	Major	Updated and revised the technical content.
10/24/2008	4.0	Major	Updated and revised the technical content.
12/05/2008	5.0	Major	Updated and revised the technical content.
01/16/2009	5.1	Minor	Updated the technical content.
02/27/2009	5.1.1	Editorial	Revised and edited the technical content.
04/10/2009	5.1.2	Editorial	Revised and edited the technical content.
05/22/2009	5.2	Minor	Updated the technical content.
07/02/2009	5.3	Minor	Updated the technical content.
08/14/2009	5.3.1	Editorial	Revised and edited the technical content.
09/25/2009	6.0	Major	Updated and revised the technical content.
11/06/2009	7.0	Major	Updated and revised the technical content.
12/18/2009	8.0	Major	Updated and revised the technical content.
01/29/2010	8.1	Minor	Updated the technical content.
03/12/2010	8.1.1	Editorial	Revised and edited the technical content.
04/23/2010	8.1.2	Editorial	Revised and edited the technical content.
06/04/2010	9.0	Major	Updated and revised the technical content.
07/16/2010	10.0	Major	Significantly changed the technical content.

Date	Revision History	Revision Class	Comments
08/27/2010	10.0	No change	No changes to the meaning, language, or formatting of the technical content.
10/08/2010	11.0	Major	Significantly changed the technical content.
11/19/2010	11.0	No change	No changes to the meaning, language, or formatting of the technical content.
01/07/2011	11.1	Minor	Clarified the meaning of the technical content.
02/11/2011	11.1	No change	No changes to the meaning, language, or formatting of the technical content.

Contents

1 Introduction

The NT LAN Manager (NTLM) Authentication: Simple Mail Transfer Protocol (SMTP) Extension specifies the use of **NTLM** authentication (as specified in [MS-NLMP]) by the **Simple Mail Transfer Protocol (SMTP)** to facilitate client authentication to a Microsoft Windows® SMTP server. SMTP specifies a protocol for the distribution, inquiry, retrieval, and posting of news articles using a reliable, stream-based transmission of news. A detailed definition of SMTP is specified in [RFC2821].

The NT LAN Manager (NTLM) Authentication: Simple Mail Transfer Protocol (SMTP) Extension uses the SMTP-AUTH command (as specified in [RFC2554] section 4) and **SMTP** response codes to negotiate NTLM authentication and send authentication data.

1.1 Glossary

The following terms are defined in [MS-GLOS]:

base64
challenge/response authentication
NT LAN Manager (NTLM) Authentication Protocol
Simple Authentication and Security Layer (SASL)
Simple Mail Transfer Protocol (SMTP)

The following terms are specific to this document:

AUTH command: A **Simple Mail Transfer Protocol (SMTP)** command that is used to send authentication information, as specified in [RFC2554]. The structure of the **AUTH command** (as used in the NT LAN Manager (NTLM) Authentication: Simple Mail Transfer Protocol (SMTP) Extension) is as follows.

```
AUTH NTLM<CR><LF>
```

Or, optionally, it is as follows.

```
AUTH NTLM [initial-response]
```

Both command forms are accepted, as required by the RFC.

connection-oriented NTLM: One of the two variants of the NT LAN Manager (NTLM) Authentication Protocol.

NTLM AUTHENTICATE_MESSAGE: The **NTLM AUTHENTICATE_MESSAGE** packet defines an **NTLM** authenticate message that is sent from the client to the server after the **NTLM CHALLENGE_MESSAGE** is processed by the client. Message structure and other details of this packet are specified in [MS-NLMP].

NTLM CHALLENGE_MESSAGE: The **NTLM CHALLENGE_MESSAGE** packet defines an NTLM challenge message that is sent from the server to the client. **NTLM CHALLENGE_MESSAGE** is generated by the local **NTLM software** and passed to the application that supports embedded **NTLM** authentication. This message is used by the server to challenge the client to prove its identity. Message structure and other details of this packet are specified in [MS-NLMP].

NTLM message: An **NTLM message** carries authentication information. Its payload data is passed to the application that supports embedded **NTLM** authentication by the **NTLM software** installed on the local computer. **NTLM messages** are transmitted between the client and server embedded within the application protocol that is using **NTLM** authentication. There are three types of **NTLM messages**:

- **NTLM NEGOTIATE_MESSAGE**
- **NTLM CHALLENGE_MESSAGE**
- **NTLM AUTHENTICATE_MESSAGE**

NTLM NEGOTIATE_MESSAGE: The NEGOTIATE_MESSAGE packet defines an NTLM negotiate message that is sent from the client to the server. The **NTLM NEGOTIATE_MESSAGE** is generated by the local **NTLM software** and passed to the application that supports embedded **NTLM** authentication. This message allows the client to specify its supported **NTLM** options to the server. Message structure and other details are specified in [MS-NLMP].

NTLM software: Software that implements the NT LAN Manager (NTLM) Authentication Protocol.

MAY, SHOULD, MUST, SHOULD NOT, MUST NOT: These terms (in all caps) are used as specified in [RFC2119]. All statements of optional behavior use either MAY, SHOULD, or SHOULD NOT.

1.2 References

1.2.1 Normative References

We conduct frequent surveys of the normative references to assure their continued availability. If you have any issue with finding a normative reference, please contact dochelp@microsoft.com. We will assist you in finding the relevant information. Please check the archive site, http://msdn2.microsoft.com/en-us/library/E4BD6494-06AD-4aed-9823-445E921C9624, as an additional source.

[MS-NLMP] Microsoft Corporation, "NT LAN Manager (NTLM) Authentication Protocol Specification", June 2007.

[MSKB-163846] Microsoft Corporation, "SID Values For Default Windows NT Installations", Version 2.1, November 2006, http://support.microsoft.com/kb/163846

[RFC1521] Borenstein, N., and Freed, N., "MIME (Multipurpose Internet Mail Extensions) Part One: Mechanisms for Specifying and Describing the Format of Internet Message Bodies", RFC 1521, September, 1993, http://www.ietf.org/rfc/rfc1521.txt

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997, http://www.ietf.org/rfc/rfc2119.txt

[RFC2554] Myers, J., "SMTP Service Extension for Authentication", RFC 2554, March, 1999, http://www.ietf.org/rfc/rfc2554.txt

[RFC2821] Klensin, J., "Simple Mail Transfer Protocol", STD 10, RFC 2821, April 2001, http://www.ietf.org/rfc/rfc2821.txt

[RFC4234] Crocker, D., Ed., and Overell, P., "Augmented BNF for Syntax Specifications: ABNF", RFC 4234, October 2005, http://www.ietf.org/rfc/rfc4234.txt

1.2.2 Informative References

[MS-GLOS] Microsoft Corporation, "Windows Protocols Master Glossary", March 2007.

[SSPI] Microsoft Corporation, "SSPI", http://msdn.microsoft.com/en-us/library/aa380493.aspx

1.3 Overview

Client applications that connect to the Simple Mail Transfer Protocol (SMTP) service included in Microsoft Windows® 2000 Server operating system and Windows Server® 2003 operating system can use Microsoft Windows® NT LAN Manager Protocol (NTLM) authentication.

The NT LAN Manager (NTLM) Authentication: Simple Mail Transfer Protocol (SMTP) Extension specifies how an SMTP client and SMTP server can use the NTLM Authentication Protocol, as specified in [MS-NLMP], so that the SMTP server can authenticate the SMTP client. The NTLM Authentication Protocol, as specified in [MS-NLMP], is a **challenge/response authentication** protocol that depends on the application layer protocols to transport NTLM packets from client to server and from server to client.

The NT LAN Manager (NTLM) Authentication: Simple Mail Transfer Protocol (SMTP) Extension defines how the SMTP is extended to perform authentication using the NTLM Authentication Protocol, as specified in [MS-NLMP]. The SMTP standard defines an extensibility mechanism for arbitrary authentication protocols to be plugged in to the core protocol. This mechanism is the SMTP-AUTH mechanism.

The NT LAN Manager (NTLM) Authentication: Simple Mail Transfer Protocol (SMTP) Extension is an embedded protocol in which NTLM authentication data is first transformed into a **base64** representation (as specified in [RFC1521]) and then formatted by padding with SMTP status codes and SMTP keywords, as defined by the AUTH mechanism. The base64 encoding and the formatting are very rudimentary and solely intended to make the NTLM data look like other SMTP commands and responses. The following diagram illustrates the sequence of transformations performed on an **NTLM message** to produce a message that can be sent over SMTP.

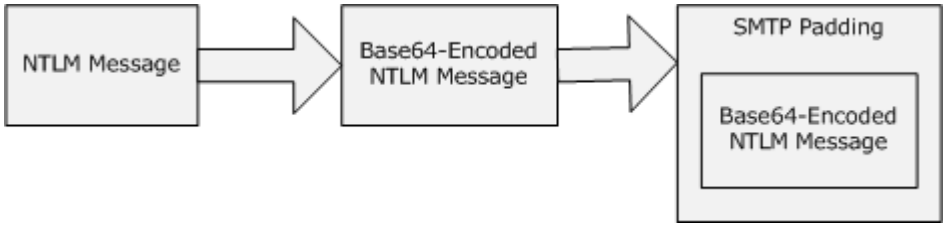

Figure 1: Relationship between NTLM message and SMTP (NTLM Authentication Protocol message)

The NT LAN Manager (NTLM) Authentication: Simple Mail Transfer Protocol (SMTP) Extension is a pass-through protocol that does not specify the structure of NTLM information. Instead, the protocol relies on the software that implements the NTLM Authentication Protocol (as specified in [MS-NLMP]) to process each NTLM message to be sent or received.

The NT LAN Manager (NTLM) Authentication: Simple Mail Transfer Protocol (SMTP) Extension defines both server and client roles.

When SMTP requests NTLM authentication, it interacts with the **NTLM software** appropriately. An overview of this interaction follows:

If acting as an SMTP client:

1. The NTLM software returns the first NTLM message to the client to be sent to the server.

2. The client should apply both the base64 encoding and SMTP padding transformations mentioned earlier (and described in detail later in this document) to produce an SMTP message, and then send this message to the server.

3. The client should wait for a response from the server. When the response is received, the client determines whether the response indicates either the end of authentication (success or failure) or the continuation of authentication.

4. If the authentication is continuing, the response message is stripped of the SMTP padding, is base64 decoded, and is passed into the NTLM software, on which the NTLM software may return another NTLM message that needs to be sent to the server. Steps 2 through 4 are repeated until authentication succeeds or fails.

If acting as an SMTP server:

1. The server waits to receive the first SMTP authentication message from the client.

2. When an SMTP message is received from the client, the SMTP padding is removed, the message is base64-decoded, and the resulting NTLM message is passed into the NTLM software.

3. The NTLM software will return a status indicating whether authentication completed successfully, failed, or more NTLM messages need to be exchanged to complete the authentication.

4. If the authentication is continuing, the NTLM software will return an NTLM message that needs to be sent to the server. This message is base64-encoded, and the SMTP padding is applied and sent to the client. Steps 2 through 4 are repeated until authentication succeeds or fails.

The sequence that follows shows the typical flow of packets between a client and server once NTLM authentication has been selected:

1. The SMTP client sends an **NTLM NEGOTIATE_MESSAGE** embedded in an SMTP_AUTH_NTLM_BLOB_Command packet to the server.

2. On receiving the SMTP packet with NTLM NEGOTIATE_MESSAGE, the server sends an **NTLM CHALLENGE_MESSAGE** embedded in an SMTP packet to the client.

3. In response, the SMTP client sends an **NTLM AUTHENTICATE_MESSAGE** embedded in an SMTP packet.

4. The server then sends an SMTP response to the client to successfully complete the authentication process.

The NTLM NEGOTIATE_MESSAGE, NTLM CHALLENGE_MESSAGE, and NTLM AUTHENTICATE_MESSAGE packets contain NTLM authentication data that must be processed by the NTLM software installed on the local computer. How to retrieve and process NTLM messages is specified in [MS-NLMP].

Implementers of the NT LAN Manager (NTLM) Authentication: Simple Mail Transfer Protocol (SMTP) Extension must possess a working knowledge of the following:

- Simple Mail Transfer Protocol (SMTP), as specified in [RFC2821] and [RFC2554]
- Multipurpose Internet Mail Extensions (MIME) base64 encoding method, as specified in [RFC1521]
- NTLM Authentication Protocol, as specified in [MS-NLMP]

1.4 Relationship to Other Protocols

The NT LAN Manager (NTLM) Authentication: Simple Mail Transfer Protocol (SMTP) Extension uses the SMTP-AUTH extension mechanism, as specified in [RFC2554], and is an embedded protocol. Unlike stand-alone application protocols, such as Telnet or Hypertext Transfer Protocol (HTTP), NTLM Authentication: SMTP Extension packets are embedded in Simple Mail Transfer Protocol (SMTP) commands and server responses.

The SMTP specifies only the sequence in which an SMTP server and an SMTP client must exchange NTLM messages to successfully authenticate the client to the server. It does not specify how the client obtains NTLM messages from the local NTLM software or how the SMTP server should process NTLM messages. The SMTP client and SMTP server implementations depend on the availability of an implementation of the NTLM Authentication Protocol (as specified in [MS-NLMP]) to obtain and process NTLM messages and on the availability of the base64 encoding and decoding mechanisms (as specified in [RFC1521]) to encode and decode the NTLM messages embedded in SMTP packets.

1.5 Prerequisites/Preconditions

Because the NT LAN Manager (NTLM) Authentication: Simple Mail Transfer Protocol (SMTP) Extension depends on NTLM to authenticate the client to the server, both server and client must have access to an implementation of the NTLM Authentication Protocol (as specified in [MS-NLMP]) that is capable of supporting **connection-oriented NTLM**.<1>

1.6 Applicability Statement

The NT LAN Manager (NTLM) Authentication: Simple Mail Transfer Protocol (SMTP) Extension must be used only when implementing an SMTP client that authenticates to an SMTP server by using NTLM authentication.

1.7 Versioning and Capability Negotiation

This document covers versioning issues in the following areas:

- Security and Authentication Methods: The NT LAN Manager (NTLM) Authentication: Simple Mail Transfer Protocol (SMTP) Extension supports the NTLM version 1 and NTLM version 2 authentication methods, as specified in [MS-NLMP].
- Capability Negotiation: The NTLM Authentication: SMTP Extension does not support negotiation of the NTLM Authentication Protocol (as specified in [MS-NLMP]) version to use. Instead, the NTLM Authentication Protocol (as specified in [MS-NLMP]) version must be configured on both the client and the server prior to authentication. NTLM Authentication Protocol (as specified in [MS-NLMP]) version mismatches are handled by the NTLM Authentication Protocol (as specified in [MS-NLMP]) implementation, and not by SMTP.

  The SMTP Service Extension for Authentication (as specified in [RFC2554]) does document the framework within which SMTP clients may discover (and SMTP servers may advertise) the capability to perform any given authentication mechanism, including (in particular) NTLM.

  The client discovers if the server supports NTLM AUTH through the SMTP-EHLO, at which time the server responds with a standard EHLO response, as specified in [RFC2821]. The EHLO keyword that is advertised if NTLM authentication is supported is "NTLM". Although "NTLM" is not an **SASL** mechanism (as defined in [RFC2554] section 3 bullet 3), the SMTP server and client behave exactly as if NTLM was an SASL mechanism with the name "NTLM". The messages involved are formally specified in other sections of this document.

1.8 Vendor-Extensible Fields

None.

1.9 Standards Assignments

None.

2 Messages

2.1 Transport

The NT LAN Manager (NTLM) Authentication: Simple Mail Transfer Protocol (SMTP) Extension does not establish transport connections. Instead, its messages are encapsulated in SMTP commands and responses. How NTLM Authentication: SMTP Extension messages must be encapsulated in SMTP commands is specified in section 2.2.

2.2 Message Syntax

NT LAN Manager (NTLM) Authentication: Simple Mail Transfer Protocol (SMTP) Extension messages are divided into three categories, depending on whether the message is sent by the server or the client.

The formal syntax of messages is provided in Augmented Backus-Naur Form (ABNF), as specified in [RFC4234].

2.2.1 SMTP AUTH Extensions

The first category of SMTP messages is within the SMTP-AUTH extensibility framework. These messages are defined in [RFC2554]. Some of the messages have parameters that MUST be customized by the extensibility mechanism (such as NTLM). The NT LAN Manager (NTLM) Authentication: Simple Mail Transfer Protocol (SMTP) Extension introduces the following customizations, as specified in sections 2.2.1.1 through 2.2.1.9.

At every point of time during the authentication exchange, the client MUST parse the status codes on the messages sent by the server and interpret them as specified in [RFC2554]. The status codes define various states such as success in authenticating, failure to authenticate, and any other arbitrary failures that the software may encounter.

The client may receive any one of the following responses during authentication:

- SMTP_AUTH_NTLM_BLOB_Response
- SMTP_AUTH_Fail_Response
- SMTP_AUTH_Other_Failure_Response
- SMTP_AUTH_NTLM_Succeeded_Response

Note that the syntax and meaning of these messages are completely defined by [RFC2554] except for the SMTP_AUTH_NTLM_BLOB_Response message, for which [RFC2554] does not define the data encapsulated within the SMTP message and leaves the definition and processing of that data to the extension mechanism. This specification will focus on precisely defining that data.

2.2.1.1 SMTP_AUTH_NTLM_Initiation_Command Message

The SMTP_AUTH_NTLM_Initiation_Command message initiates the NTLM authentication process for SMTP.

[RFC2554] section 4 defines the syntax of the SMTP **AUTH command** and related commands (for example, EHLO) to initiate authentication. The mechanism name for NTLM authentication is defined to be the string "NTLM" for the NTLM Authentication: SMTP Extension. The command to initiate an NTLM conversation by a client in ABNF form is shown as follows.

```
AUTH NTLM <CR><LF>
```

Or, optionally

```
AUTH NTLM [initial-response]<CR><LF>
```

Note There are two forms of this command. The client may optionally base64 encode (as specified in [RFC1521]) and send the initial NTLM NEGOTIATE_MESSAGE message as part of this command as the [initial-response].

If an [initial-response] string is supplied, the server will invoke the NTLM software, obtain the NTLM CHALLENGE_MESSAGE, base64 encode it, pad it (as defined elsewhere in this specification), and return it to the client as an SMTP message.

If NTLM is not supported, the SMTP server returns a failure status code, as defined in [RFC2554] section 6. This message is a standard message defined by the SMTP standard, and is not discussed here.

2.2.1.2 SMTP_NTLM_Supported_Response Message

The SMTP_NTLM_Supported_Response message indicates that the server supports NTLM authentication for SMTP.

If the [initial-response] string is not supplied in the client SMTP_AUTH_NTLM_Initiation_Command message, and NTLM is supported, the SMTP server will respond with an SMTP message prefixed with a status code of 334 to indicate that NTLM is supported. The only data in this message that is useful is the status code 334. The remaining data is a human-readable ASCII string whose contents are constrained by the specifications in section 4.5.3 in [RFC2821]. This data has no bearing on the authentication. The syntax of this command in ABNF form is shown as follows.

```
334 <human-readable-string><CR><LF>
```

A human-readable-string is formally defined in ABNF as follows.

```
human-readable-string = *CHAR
```

Note CHAR is the US-ASCII character set, excluding NULL.

The command sent by the client determines whether the server response MUST be interpreted as an SMTP_NTLM_Supported_Response or an SMTP_AUTH_NTLM_BLOB_Response. Based on ABNF syntax alone, SMTP_NTLM_Supported_Response and SMTP_AUTH_NTLM_BLOB_Response messages appear identical, making a successful distinction between the two impossible. Therefore, the parser MUST distinguish between these messages as follows:

- If the client previously sent an SMTP_AUTH_NTLM_Initiation_Command without an [initial-response], the server response MUST be parsed as an SMTP_NTLM_Supported_Response message with a human-readable-string. The human-readable-string SHOULD be ignored by the client, except to facilitate troubleshooting and debugging by a human operator. This string has no consequence on the operation of the protocol.

- If the client previously sent an SMTP_AUTH_NTLM_Initiation_Command with an [initial-response], the server response MUST be parsed as an SMTP_AUTH_NTLM_BLOB_Response message with a base64-encoded string. The client MUST NOT ignore the base64-encoded string and it MUST be processed by the NTLM subsystem, as described in this document.

Note Status code 334 is also returned by the SMTP_AUTH_NTLM_BLOB_Response message.

2.2.1.3 SMTP_AUTH_NTLM_BLOB_Response Message

The SMTP_AUTH_NTLM_BLOB_Response message is defined as follows. This message is partially defined in [RFC2554] section 4 as a "server challenge response". The 334 status code indicates ongoing authentication and indicates that the <base64-encoded-NTLM-message> is to be processed by the authentication subsystem.

```
334 <base64-encoded-NTLM-message><CR><LF>
```

Note that status code 334 is also returned by the SMTP_NTLM_Supported_Response message.

2.2.1.4 SMTP_AUTH_Fail_Response Message

SMTP_AUTH_Fail_Response is defined as follows. This message, identified by the 535 status code, is defined in [RFC2554] section 4, and indicates that the authentication has terminated unsuccessfully because the user name or password is incorrect.

```
535 5.7.3 <human-readable-string><CR><LF>
```

2.2.1.5 SMTP_AUTH_Other_Failure_Response Message

The SMTP_AUTH_Other_Failure_Response message is defined as follows. This is actually a class of messages whose syntax and interpretation are defined in [RFC2821] section 4.2 and [RFC2554] sections 4 and 6. They indicate an abnormal termination of the NTLM authentication negotiation, which may occur for various reasons such as software errors, lack of system resources, and so on. For the purposes of this document, SMTP_AUTH_Other_Failure_Response is defined as any SMTP message other than SMTP_AUTH_NTLM_Succeeded_Response, SMTP_AUTH_Fail_Response, and SMTP_AUTH_NTLM_BLOB_Response. The interpretation of SMTP_AUTH_Other_Failure_Response, and the suggested client action when receiving such a message, is defined in [RFC2821] section 4.3. This message represents an exit from AUTH and, as such, is not really a part of AUTH negotiation.

2.2.1.6 SMTP_AUTH_NTLM_Succeeded_Response Message

The SMTP_AUTH_NTLM_Succeeded_Response message is defined as follows. This message is defined in [RFC2554] section 4 and indicates that the authentication negotiation has completed with the client successfully authenticating to the server.

```
235 <human-readable-string><CR><LF>
```

2.2.1.7 SMTP_AUTH_NTLM_BLOB_Command Message

NTLM messages encapsulated by the client and sent to the server are referred to as SMTP_AUTH_NTLM_BLOB_Command messages in this document. They have the following syntax defined in ABNF and conform to the prescription of [RFC2554] section 4.

```
<base64-encoded-NTLM-message><CR><LF>
```

2.2.1.8 SMTP_NTLM_Not_Supported_Response Message

The SMTP_NTLM_Not_Supported_Response_Message is defined as follows. This message is defined in [RFC2554] section 4 and indicates that the authentication mechanism is not supported by the server. The server rejects the AUTH command with the following message.

```
504 <human-readable-string><CR><LF>
```

2.2.1.9 EHLO Discovery Message

The NT LAN Manager (NTLM) Authentication: Simple Mail Transfer Protocol (SMTP) Extension also supports the discovery of supported authentication procedures.

When the EHLO command is sent to the SMTP server, the SMTP server will list available authentication mechanisms using the syntax defined in [RFC2821] section 4.1.1.1. The NTLM mechanism is indicated by using the "NTLM" EHLO advertisement if NTLM authentication is enabled for the SMTP server. An example of such an advertisement of supported authentication procedures by the server can be found in [RFC2554] section 4. The line "S: 250 AUTH CRAM-MD5 DIGEST-MD5" in the conversation indicates that the server advertises the supported authentication procedures as CRAM-MD5, DIGEST-MD5.

The server responds with an EHLO-Response (including the EHLO-keyword AUTH) when the client sends the EHLO command with or without an argument.

[RFC2821] section 4.1.1.1 states that clients SHOULD send EHLO with an argument. The definition of SHOULD in [RFC2119] allows the client to exclude the EHLO argument in exceptional circumstances. The SMTP server MUST support such clients.

2.2.2 SMTP Server Messages

This section defines the creation of SMTP_AUTH_NTLM_BLOB_Response messages. These are NTLM messages that are sent by the server, and MUST be encapsulated as follows to conform to syntax specified by the SMTP-AUTH mechanism:

1. Encode the NTLM message data as base64 (as specified in [RFC1521]). This is required because NTLM messages contain data outside the ASCII character range whereas SMTP only supports the sending of ASCII characters within the context of SMTP-AUTH.
2. To the base64-encoded string, prefix the SMTP response code "334 " (that is, the numerals 334 followed by the ASCII space character 0x20).
3. Suffix the <CR> and <LF> characters (ASCII values 0x0D and 0x0A), as required by SMTP.

The ABNF definition of a server message is as follows.

```
334 <base64-encoded-NTLM-message><CR><LF>
```

De-encapsulation of these messages by the client follows the reverse logic:

1. Remove the <CR> and <LF> characters (ASCII values 0x0D and 0x0A).

2. Remove the SMTP response code "334" (that is, the numerals 334 followed by the ASCII space character 0x20).

3. base64 decode the SMTP data to produce the original NTLM message data.

2.2.3 SMTP Client Messages

This section defines the creation of SMTP_AUTH_NTLM_BLOB_Command messages. These NTLM messages sent by the client are encapsulated as follows to conform to the SMTP-AUTH mechanism:

1. base64-encode (as specified in [RFC1521]) the NTLM message data. This is required because NTLM messages contain data outside the ASCII character range whereas SMTP only supports ASCII characters to be sent within the context of SMTP-AUTH.

2. Suffix the <CR> and <LF> characters (ASCII values 0x0D and 0x0A), as required by SMTP.

The ABNF definition of a client message is as follows.

```
<base64-encoded-NTLM-message><CR><LF>
```

De-encapsulation of these messages by the server follows the reverse logic:

1. Remove the <CR> and <LF> characters (ASCII values 0x0D and 0x0A).

2. base64 decode the SMTP data to produce the original NTLM message data.

3 Protocol Details

3.1 Client Details

3.1.1 Abstract Data Model

3.1.1.1 SMTP State Model

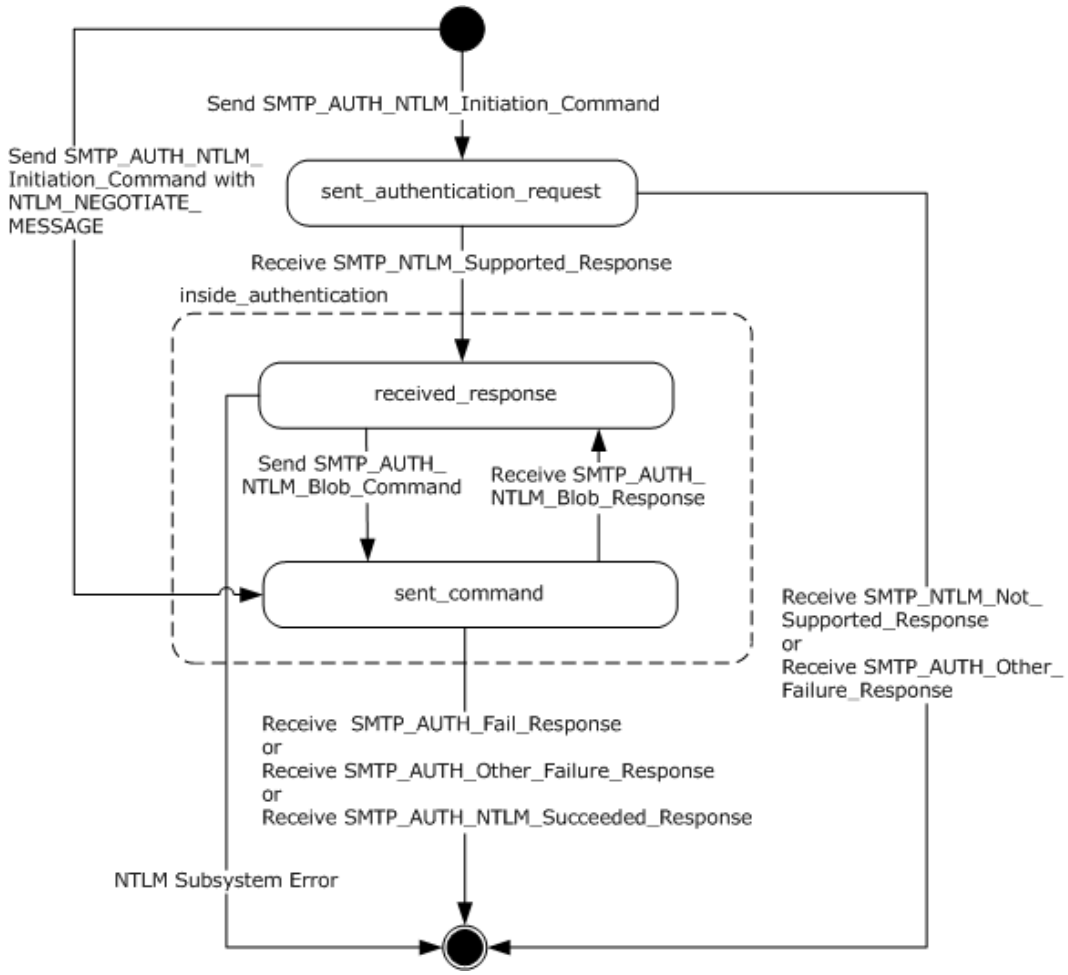

Figure 2: SMTP NTLM authentication client state model

The abstract data model for the NT LAN Manager (NTLM) Authentication: Simple Mail Transfer Protocol (SMTP) Extension has the following states:

1. start

   This is the state of the client before the SMTP_AUTH_NTLM_Initiation_Command message has been sent.

2. sent_authentication_request

This is the state of the client after the SMTP_AUTH_NTLM_Initiation_Command message has been sent.

3. inside_authentication

   This is a composite state that contains two states.

   - received_response

     This is the state entered by the client after it has received an SMTP_NTLM_Supported_Response message, or when the client receives a SMTP_AUTH_NTLM_Blob_Response message.

     When the client enters this state after receiving a SMTP_NTLM_Supported_Response message, the client invokes the NTLM software to get the NTLM_NEGOTIATE_MESSAGE and sends it to the server embedded inside the first SMTP_AUTH_NTLM Blob_Command. The client MUST transition the state to sent_command after it sends the SMTP_AUTH_NTLM Blob_Command.

     The client comes back to this state from the sent_command state after it receives SMTP_AUTH_NTLM_Blob_Response from the server.

     The client MUST transition the state to completed_authentication if it encounters an NTLM subsystem error.

   - sent_command

     This is the state entered by the client after it has sent SMTP_AUTH_NTLM_Initiation_Command with NTLM_NEGOTIATE_MESSAGE. During this state the client MUST wait for a response from the server. When SMTP_AUTH_NTLM_Blob_Response is received the client MUST transition the state to received_response.

     The client comes back to this state from the received_response state after it sends the SMTP_AUTH_NTLM Blob_Command to the server.

     The client MUST transition to completed_authentication if it receives SMTP_AUTH_FAIL_Response or SMTP_AUTH_Other_Failure_Response or SMTP_AUTH_NTLM_Succeeded_Response.

   The inside_authentication state is also described in section 3.1.5.4.

4. completed_authentication

   This is the state of the client on exiting the inside_authentication state. Section 3.1.5 defines the rules for how the inside_authentication state is exited. The completed_authentication represents the end state of the authentication protocol.

   This document does not address the behavior of SMTP in this state.

3.1.1.2 NTLM Software Interaction

During the inside_authentication phase, the SMTP client invokes the NTLM software, as specified in [MS-NLMP] section 3.1. The NTLM Authentication Protocol (as specified in [MS-NLMP]) is used with these options:

1. The negotiation is a connection-oriented NTLM negotiation.

2. None of the flags specified in [MS-NLMP] section 3.1.1 are passed to NTLM.

Following is a description of how SMTP uses NTLM. Remember that all NTLM messages are encapsulated as specified in section 2.1. The NTLM Authentication Protocol (as specified in [MS-NLMP] section 3.1.1) specifies the data-model, internal states, and sequencing of NTLM messages in greater detail.

1. The client initiates the authentication by invoking NTLM, at which time NTLM will return the NTLM NEGOTIATE_MESSAGE to be sent to the server. NTLM NEGOTIATE_MESSAGE may either be embedded in the initial SMTP_AUTH_NTLM_Initiation_Command message as the [initial-response] field (as specified in [RFC2554]), or it may be sent to the server embedded inside the first SMTP_AUTH_NTLM_BLOB_Command message.

2. Subsequently, the exchange of NTLM messages goes on as defined by the NTLM Authentication Protocol (as specified in [MS-NLMP]) with the SMTP client encapsulating the NTLM messages before sending them to the server, and de-encapsulating SMTP messages to obtain the NTLM message before giving it to the NTLM Authentication Protocol (as specified in [MS-NLMP]).

3. The NTLM Authentication Protocol (as specified in [MS-NLMP]) completes authentication, either successfully or unsuccessfully, as follows:

   - The server sends the SMTP_AUTH_NTLM_Succeeded_Response message to the client. On receiving this message, the client transitions to the completed_authentication state and SHOULD treat the authentication attempt as successful.

   - The server sends the SMTP_AUTH_Fail_Response message to the client. On receiving this message, the client transitions to the completed_authentication state and SHOULD treat the authentication attempt as failed.

   - The server sends the SMTP_AUTH_Other_Failure_Response message to the client. On receiving this message, the client transitions to the completed_authentication state and SHOULD treat the authentication attempt as failed.

   - Failures reported from the NTLM software (which can occur for any reason, including incorrect data being passed in, or implementation-specific errors) may be reported to the client by the NTLM Authentication Protocol (as specified in [MS-NLMP]) and cause the client to transition to the completed_authentication state.

3.1.2 Timers

None.

3.1.3 Initialization

None.

3.1.4 Higher-Layer Triggered Events

None.

3.1.5 Message Processing Events and Sequencing Rules

The NT LAN Manager (NTLM) Authentication: Simple Mail Transfer Protocol (SMTP) Extension is driven by a series of message exchanges between an SMTP server and an SMTP client. The rules governing the sequencing of commands and the internal states of the client and server are defined

by a combination of [RFC2554] and [MS-NLMP]. Section 3.1.1 completely defines how the rules specified in [RFC2554] and [MS-NLMP] govern SMTP authentication.

3.1.5.1 Sending an SMTP_AUTH_NTLM_BLOB_Command Message

Expected state is received_response.

This section defines the creation of SMTP_AUTH_NTLM_BLOB_Command messages. These NTLM messages sent by the client are encapsulated as follows to conform to the SMTP-AUTH mechanism:

1. base64-encode (as specified in [RFC1521]) the NTLM message data. This is required because NTLM messages contain data outside the ASCII character range whereas SMTP only supports ASCII characters to be sent within the context of SMTP-AUTH.
2. Suffix the <CR> and <LF> characters (ASCII values 0x0D and 0x0A), as required by SMTP.

The ABNF definition of a client message is as follows.

```
<base64-encoded-NTLM-message><CR><LF>
```

De-encapsulation of these messages by the server follows the reverse logic:

1. Remove the <CR> and <LF> characters (ASCII values 0x0D and 0x0A).
2. base64 decode the SMTP data to produce the original NTLM message data.

3.1.5.2 Receiving an SMTP_NTLM_Supported_Response Message

Expected state is sent_authentication_request.

On receiving this message, a client MUST generate the first NTLM message by calling the NTLM software. The NTLM software then generates NTLM NEGOTIATE_MESSAGE, as specified in [MS-NLMP]. The client MUST then encapsulate the NTLM message, as defined in section 2.2.3, and send it to the server.

Note The server will send the SMTP_NTLM_Supported_Response message only if the client did not embed an NTLM NEGOTIATE_MESSAGE in the SMTP_AUTH_NTLM_Initiation_Command [initial-response] optional parameter.

The state of the client is changed to inside_authentication.

3.1.5.3 Receiving an SMTP_NTLM_Not_Supported_Response Message

Expected state is sent_authentication_request.

On receiving this message, a client MUST de-encapsulate it to obtain the embedded NTLM message, and then pass it to the NTLM software for processing. The NTLM software MUST do the following:

This state terminates when the client receives an SMTP_AUTH_NTLM_Succeeded_Response, SMTP_AUTH_Fail_Response, or SMTP_AUTH_Other_Failure_Response message.

In this state, the client initializes the NTLM software and repeats the following steps:

- Encapsulates the NTLM message returned by the NTLM software into an SMTP message.
- Waits for a response from the server.

- De-encapsulates received SMTP message-data (if any) from the other party and converts it to NTLM message data.
- Passes it to the NTLM software.
- Sends the SMTP message to the other party.

For an overview of SMTP client authentication, see the SMTP client state model specified in section 3.1.1.1.

3.1.5.4 Receiving an SMTP_AUTH_NTLM_BLOB_Response Message

Expected state is sent_command.

On receiving this message, a client MUST change its internal state to received_response, de-encapsulate it to obtain the embedded NTLM message, and then pass it to the NTLM software for processing. The NTLM software then extracts the NTLM_CHALLENGE_MESSAGE and produces an NTLM_AUTHENTICATE_MESSAGE response. The client MUST then encapsulate the NTLM message, as defined in section 3.1.5.1, and send it to the server.

In this state, the client initializes the NTLM software and then performs the following steps.

1. Encapsulates the NTLM message returned by the NTLM software into an SMTP message
2. Sends the SMTP message to the server
3. Transition the state to sent_command

For an overview of SMTP client authentication, see the SMTP client state model specified in section 3.1.1.1.

3.1.5.4.1 Error from NTLM

If the NTLM software reports an error, the client MUST change its internal state to completed_authentication and consider that the authentication has failed. The client may then take any action it considers appropriate. This document does not mandate any specific course of action.

Typical actions are to attempt other (non-authentication related) SMTP commands or to disconnect the connection.

3.1.5.4.2 NTLM Reports Success and Returns an NTLM Message

The NTLM message should be encapsulated and sent to the server. No change occurs in the state of the client.

3.1.5.5 Receiving an SMTP_AUTH_NTLM_Succeeded_Response Message

Expected state: sent_command.

The SMTP client MUST change its internal state to completed_authentication and consider that the authentication has succeeded. The client then takes any action it considers appropriate. This document does not mandate any specific course of action.

3.1.5.6 Receiving an SMTP_AUTH_Fail_Response Message

Expected state: sent_command.

The SMTP client MUST change its internal state to completed_authentication and consider that the authentication has failed. The client then takes any action it considers appropriate. This document does not mandate any specific course of action.

3.1.5.7 Receiving an SMTP_AUTH_Other_Failure_Response Message

Expected state: sent_command.

The SMTP client MUST change its internal state to completed_authentication and consider that the authentication has failed. The client then takes any action it considers appropriate. This document does not mandate any specific course of action.

Note This response MUST be interpreted using the rules specified in [RFC2821]. These rules include using the three-digit status code to infer whether the failure is permanent or temporary, whether or not to generate non-delivery notifications for messages queued on the client, and so on.

3.1.6 Timer Events

None.

3.1.7 Other Local Events

None.

3.2 Server Details

3.2.1 Abstract Data Model

3.2.1.1 SMTP State Model

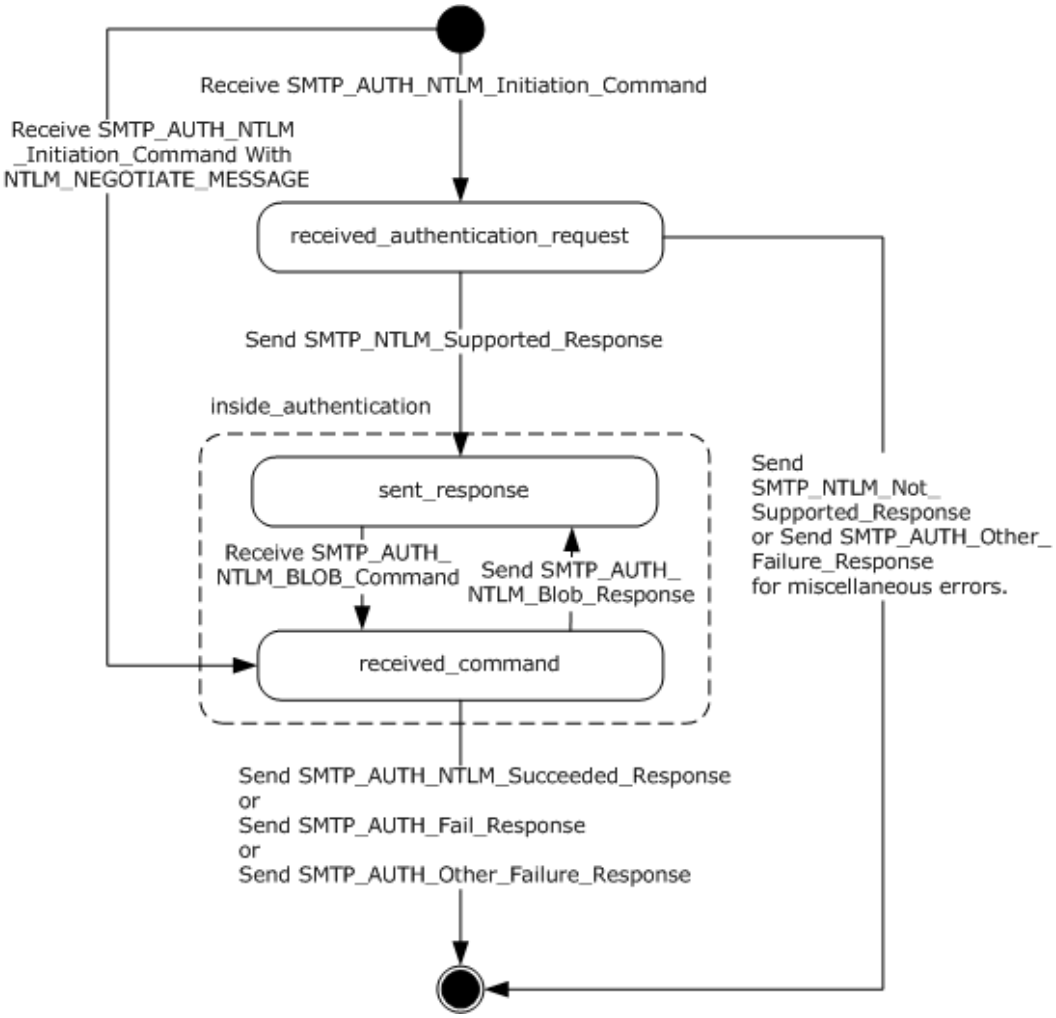

Figure 3: SMTP NTLM authentication server state model

The abstract data model for the NT LAN Manager (NTLM) Authentication: Simple Mail Transfer Protocol (SMTP) Extension has the following states:

1. start

   This is the state of the server before the SMTP_AUTH_NTLM_Initiation_Command (section 2.2.1.1) message has been received.

2. received_authentication_request

   This is the state of the server after the SMTP_AUTH_NTLM_Initiation_Command message has been received.

3. inside_authentication

   This is a composite state which contains two states.

   - sent_response

     This is the state entered by the server after it has sent an SMTP_NTLM_Supported_Response (section 2.2.1.2) or SMTP_AUTH_NTLM_Blob_Response (section 2.2.1.3) message.

     During this state the server SHOULD wait for SMTP_AUTH_NTLM_BLOB_Command (section 2.2.1.7) from the client and transition the state to received_response after receiving the SMTP_AUTH_NTLM_BLOB_Command.

     The server comes back to this state after it has sent SMTP_AUTH_NTLM_Blob_Response to the client.

   - received_command

     This is the state entered by the server after it has received the SMTP_AUTH_NTLM_Initiation_Command with NTLM_NEGOTIATE_MESSAGE or SMTP_AUTH_NTLM_BLOB_Command.

     During this state the server passes the SMTP_AUTH_NTLM_Initiation_Command with NTLM_NEGOTIATE_MESSAGE or SMTP_AUTH_NTLM_BLOB_Command to the NTLM software. If the NTLM software returns SMTP_AUTH_NTLM_Blob_Response message the server sends it back to the client.

     The server MUST transition the state to sent_response after it sends the SMTP_AUTH_NTLM_Blob_Response.

     The server comes back to this state after receiving SMTP_AUTH_NTLM_BLOB_Command.

     The server MUST transition the state to completed_authentication when it sends SMTP_AUTH_NTLM_Succeeded_Response (section 2.2.1.6) or SMTP_AUTH_Fail_Response (section 2.2.1.4) or SMTP_AUTH_Other_Failure_Response (section 2.2.1.5) to the client.

   The inside_authentication state is also described in section 3.2.5.2.

4. completed_authentication

   This is the state of the server on exiting the inside_authentication state. Section 3.1.5 defines the rules for how the inside_authentication state is exited. The completed_authentication represents the end state of the authentication protocol.

   This document does not address the behavior of SMTP in this state.

3.2.1.2 NTLM Software Interaction

During inside_authentication state, the SMTP server invokes the NTLM software, as specified in [MS-NLMP] section 3.1. The NTLM Authentication Protocol (as specified in [MS-NLMP]) is used with these options:

- The negotiation is a connection-oriented NTLM negotiation.
- None of the flags specified in [MS-NLMP] section 3.1.1 are passed to the NTLM Authentication Protocol (as specified in [MS-NLMP]).

Following is a description of how SMTP uses NTLM (see [MS-NLMP] section 3.1.1, which describes the data model and sequencing of NTLM packets in greater detail).

1. The server, on receiving the NTLM NEGOTIATE_MESSAGE, passes it to the NTLM software. If the NTLM NEGOTIATE_MESSAGE message is valid, an NTLM CHALLENGE_MESSAGE is returned.

2. Subsequently, the exchange of NTLM messages goes on as defined by the NTLM Authentication Protocol (as specified in [MS-NLMP]) with the SMTP server encapsulating the NTLM messages returned by the NTLM Authentication Protocol (as specified in [MS-NLMP]) before sending them to the client.

3. When the NTLM Authentication Protocol (as specified in [MS-NLMP]) completes authentication, either successfully or unsuccessfully, the NTLM software notifies SMTP.

    - On successful completion, the server MUST exit the inside_authentication state and enter the completed_authentication state, and then send the SMTP_AUTH_NTLM_Succeeded_Response message to the client. On receiving this message, the client MUST also transition to the completed_authentication state.

    - If a failure occurs due to an incorrect password error, as specified in [MS-NLMP] sections 3.3.1 and 3.3.2, the server SHOULD enter the completed_authentication state and send the client an SMTP_AUTH_Fail_Response message.

    - If a failure occurs on the server due to any reason other than the incorrect password error, the server enters the completed_authentication state and sends the client an SMTP_AUTH_Other_Failure_Response message. On receiving this message, the client MUST enter the completed_authentication state.

3.2.2 Timers

None.

3.2.3 Initialization

None.

3.2.4 Higher-Layer Triggered Events

None.

3.2.5 Message Processing Events and Sequencing Rules

The NT LAN Manager (NTLM) Authentication: Simple Mail Transfer Protocol (SMTP) Extension is driven by a series of message exchanges between an SMTP server and an SMTP client. The rules governing the sequencing of commands and the internal states of the client and server are defined by a combination of [RFC2554] and [MS-NLMP]. Section 3.2.1 completely defines how the rules specified in [RFC2554] and [MS-NLMP] govern SMTP authentication.

3.2.5.1 Receiving an SMTP_AUTH_NTLM_Initiation_Command Message

Expected state is start.

The server examines the received message to determine if the [initial-response] parameter is present in the message.

There are two actions possible, depending on whether or not the client has included the [initial-response] parameter in this message:

1. If the client has included the [initial-response] parameter, the server MUST change its internal state to received_command and de-encapsulate the NTLM NEGOTIATE_MESSAGE embedded within the [initial-response] and pass it to the NTLM software. The NTLM software MUST do one of the following:

   - Report success in processing the message. The server MUST send SMTP_AUTH_NTLM_BLOB_Response message to the client and change its internal state to sent_response.

   - Report that the authentication failed, which could be due to some other software error or message corruption. The server MUST change its state to completed_authentication and return an SMTP_AUTH_Other_Failure_Response message.

2. If the client has not included the [initial-response] parameter, the server MUST change its state to received_authenticaton_request and reply with the SMTP_NTLM_Supported_Response message if it supports NTLM and change its state to the sent_response state. If the server does not support NTLM, it MUST respond with the SMTP_NTLM_Not_Supported_Response message, and change the internal state to completed_authentication.

3.2.5.2 Receiving an SMTP_AUTH_NTLM_BLOB_Command Message

Expected state is sent_response.

On receiving this message, a server MUST change its internal state to received_command, de-encapsulate the message, obtain the embedded NTLM message, and pass it to the NTLM software. The NTLM software MUST do one of the following:

- Report success in processing the message and return an NTLM message to continue the authentication.

- Report that authentication completed successfully.

- Report that the authentication failed due to a bad user name or password, as specified in [MS-NLMP].

- Report that the authentication failed, which could be due to some other software error or message corruption.

For an overview of SMTP server authentication, see the SMTP server state model specified in section 3.2.1.1.

3.2.5.2.1 NTLM Returns Success, Returning an NTLM Message

Expected state is received_command.

The server MUST encapsulate the NTLM message, send it to the client, and change its internal state to sent_response.

3.2.5.2.2 NTLM Returns Success, Indicating that the Authentication Completed Successfully

Expected state is received_command.

The server MUST return the SMTP_AUTH_NTLM_Succeeded_Response message and change its internal state to completed_authentication.<2>

3.2.5.2.3 NTLM Returns Status, Indicating that the User Name or Password Is Incorrect

Expected state is received_command.

The server MUST return the SMTP_AUTH_Fail_Response message and change its internal state to completed_authentication.

3.2.5.2.4 NTLM Returns a Failure Status, Indicating Any Other Error

Expected state is received_command.

The server MUST return the SMTP_AUTH_Other_Failure_Response message and change its internal state to completed_authentication.

3.2.6 Timer Events

None.

3.2.7 Other Local Events

None.

4 Protocol Examples

4.1 SMTP Client Successfully Authenticating to an SMTP Server

This section illustrates the NT LAN Manager (NTLM) Authentication: Simple Mail Transfer Protocol (SMTP) Extension with an example scenario in which an SMTP client successfully authenticates to an SMTP server using NTLM.

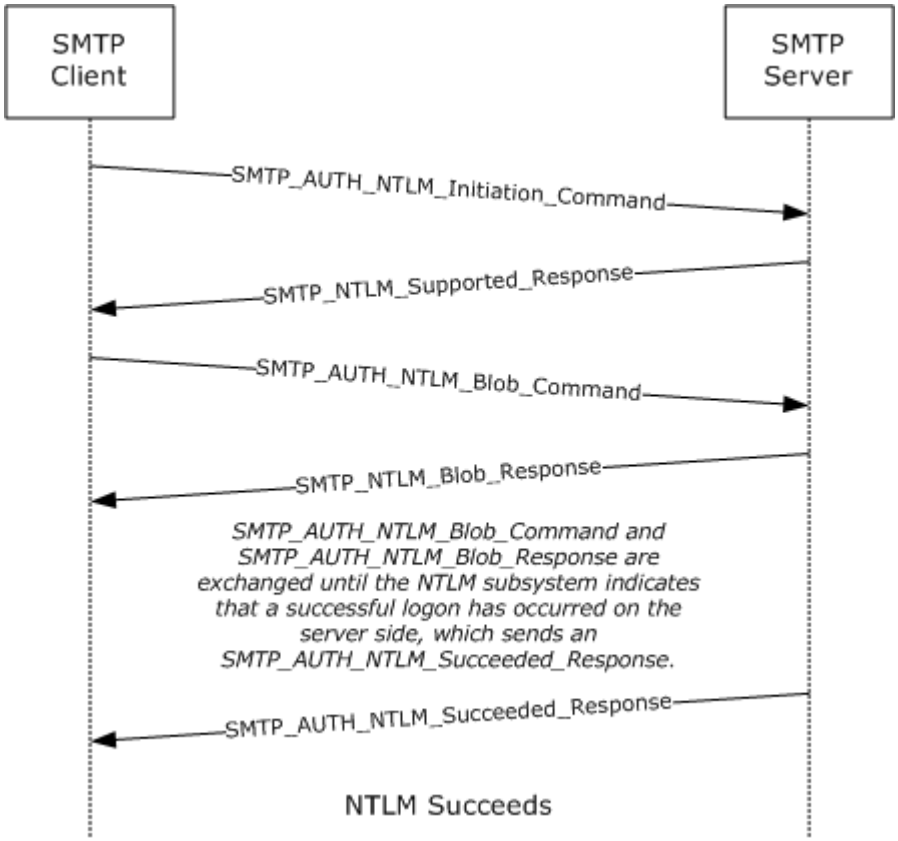

Figure 4: SMTP client successfully authenticating to SMTP server

1. The client sends an EHLO to the server. This command is specified in [RFC2821].

```
EHLO test.com
```

2. The server responds with an EHLO-Response (including the EHLO-keyword AUTH) to indicate that the authentication is supported. Among the parameters to the AUTH EHLO-response keyword is the keyword "NTLM", indicating that NTLM authentication is available.

```
250-exch-cli-66 Hello [127.0.0.1]
250-AUTH GSSAPI NTLM
250-TURN
250-SIZE 2097152
250-ETRN
250-PIPELINING
```

```
250-DSN
250-ENHANCEDSTATUSCODES
250-8bitmime
250-BINARYMIME
250-CHUNKING
250-VRFY
250 OK
```

3. The client then sends the SMTP AUTH command, SMTP_AUTH_NTLM_Initiation_Command, initiating auth. In this example, the AUTH command being sent is without the optional [initial-response] data.

```
AUTH NTLM
```

4. The server sends the SMTP_NTLM_Supported_Response message, indicating that it can perform NTLM authentication.

```
334 ntlm supported
```

5. The client sends an SMTP_AUTH_NTLM_BLOB_Command message containing a base64-encoded NTLM NEGOTIATE_MESSAGE.

```
TlRMTVNTUAABAAAAt4II4gAAAAAAAAAAAAAAAAAAAAAFAs4OAAAADw==
```

The content of the NTLM message after base64 decoding is as follows.

```
0x00000000  4E 54 4C 4D 53 53 50 00 01 00 00 00 B7 82 08 E2   NTLMSSP.....7_.b
0x00000010  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0x00000020  05 02 CE 0E 00 00 00 0F                           ..N.....
```

6. The server sends an SMTP_AUTH_NTLM_BLOB_Response message containing a base64-encoded NTLM CHALLENGE_MESSAGE.

```
334 TlRMTVNTUAACAAAAFgAWADgAAAA1goriZt7rI6Uq/ccAAAAAAAAAAGwAbABOAAA
ABQLODgAAAA9FAFgAQwBIAC0AQwBMAEkALQA2ADYAAgAWAEUAWABDAEgALQBDAEwASQ
AtADYANgABABYARQBYAEMASAAtAEMATABJAC0ANgA2AAQAFgBlAHgAYwBoAC0AYwBsA
GkALQA2ADYAAwAWAGUAeABjAGgALQBjAGwAaQAtADYANgAAAAAA
```

The content of the NTLM message after base64 decoding is as follows.

```
0x00000000  4E 54 4C 4D 53 53 50 00 02 00 00 00 16 00 16 00   NTLMSSP.........
0x00000010  38 00 00 00 35 82 8A E2 66 DE EB 23 A5 2A FD C7   8...5_bf^k#%*}G
0x00000020  00 00 00 00 00 00 00 00 6C 00 6C 00 4E 00 00 00   ........l.l.N...
0x00000030  05 02 CE 0E 00 00 00 0F 45 00 58 00 43 00 48 00   ..N.....E.X.C.H.
0x00000040  2D 00 43 00 4C 00 49 00 2D 00 36 00 36 00 02 00   -.C.L.I.-.6.6...
0x00000050  16 00 45 00 58 00 43 00 48 00 2D 00 43 00 4C 00   ..E.X.C.H.-.C.L.
```

```
0x00000060  49 00 2D 00 36 00 36 00 01 00 16 00 45 00 58 00   I.-.6.6.....E.X.
0x00000070  43 00 48 00 2D 00 43 00 4C 00 49 00 2D 00 36 00   C.H.-.C.L.I.-.6.
0x00000080  36 00 04 00 16 00 65 00 78 00 63 00 68 00 2D 00   6.....e.x.c.h.-.
0x00000090  63 00 6C 00 69 00 2D 00 36 00 36 00 03 00 16 00   c.l.i.-.6.6.....
0x000000A0  65 00 78 00 63 00 68 00 2D 00 63 00 6C 00 69 00   e.x.c.h.-.c.l.i.
0x000000B0  2D 00 36 00 36 00 00 00 00 00   -.6.6.....
```

7. The client sends an SMTP_AUTH_NTLM_BLOB_Command message containing a base64-encoded NTLM AUTHENTICATE_MESSAGE.

```
TlRMTVNTUAADAAAAGAAYAHwAAAAYABgAlAAAABYAFgBIAAAACAAIAF4AAAAWABYAZgA
AABAAEACsAAAANYKI4gUCzg4AAAAPZQB4AGMAaAAtAGMAbABpAC0ANgA2AHQAZQBzAH
QARQBYAEMASAAtAEMATABJAC0ANgA2AAZKkK42dvN2AAAAAAAAAAAAAAAAAAAAAABvqC
ZdJZ0NxuuMaNT5PPn5aZ6imuk9cPZkPUjEYNIRezkCGmTwS5G0=
```

The content of the NTLM message after base64 decoding is as follows.

```
0x00000000  4E 54 4C 4D 53 53 50 00 03 00 00 00 18 00 18 00   NTLMSSP.........
0x00000010  7C 00 00 00 18 00 18 00 94 00 00 00 16 00 16 00   |......._.......
0x00000020  48 00 00 00 08 00 08 00 5E 00 00 00 16 00 16 00   H.......^.......
0x00000030  66 00 00 00 10 00 10 00 AC 00 00 00 35 82 88 E2   f.......,...5__b
0x00000040  05 02 CE 0E 00 00 00 0F 65 00 78 00 63 00 68 00   ..N.....e.x.c.h.
0x00000050  2D 00 63 00 6C 00 69 00 2D 00 36 00 36 00 74 00   -.c.l.i.-.6.6.t.
0x00000060  65 00 73 00 74 00 45 00 58 00 43 00 48 00 2D 00   e.s.t.E.X.C.H.-.
0x00000070  43 00 4C 00 49 00 2D 00 36 00 36 00 06 4A 90 AE   C.L.I.-.6.6..J_.
0x00000080  36 76 F3 76 00 00 00 00 00 00 00 00 00 00 00 00   6vsv............
0x00000090  00 00 00 00 1B EA 09 97 49 67 43 71 BA E3 1A 35   .....j._IgCq:c.5
0x000000A0  3E 4F 3E 7E 5A 67 A8 A6 BA 4F 5C 3D 99 0F 52 31   >O>~Zg(&:O\=_.R1
0x000000B0  18 34 84 5E CE 40 86 99 3C 12 E4 6D               .4_^N@__<.dm
```

8. The server sends an SMTP_AUTH_NTLM_Succeeded_Response message.

```
235 2.7.0 Authentication successful
```

4.2 SMTP Client Not Successfully Authenticating to an SMTP Server

This section illustrates the NT LAN Manager (NTLM) Authentication: Simple Mail Transfer Protocol (SMTP) Extension with an example scenario in which an SMTP client attempts NTLM authentication to an SMTP server, and the authentication fails.

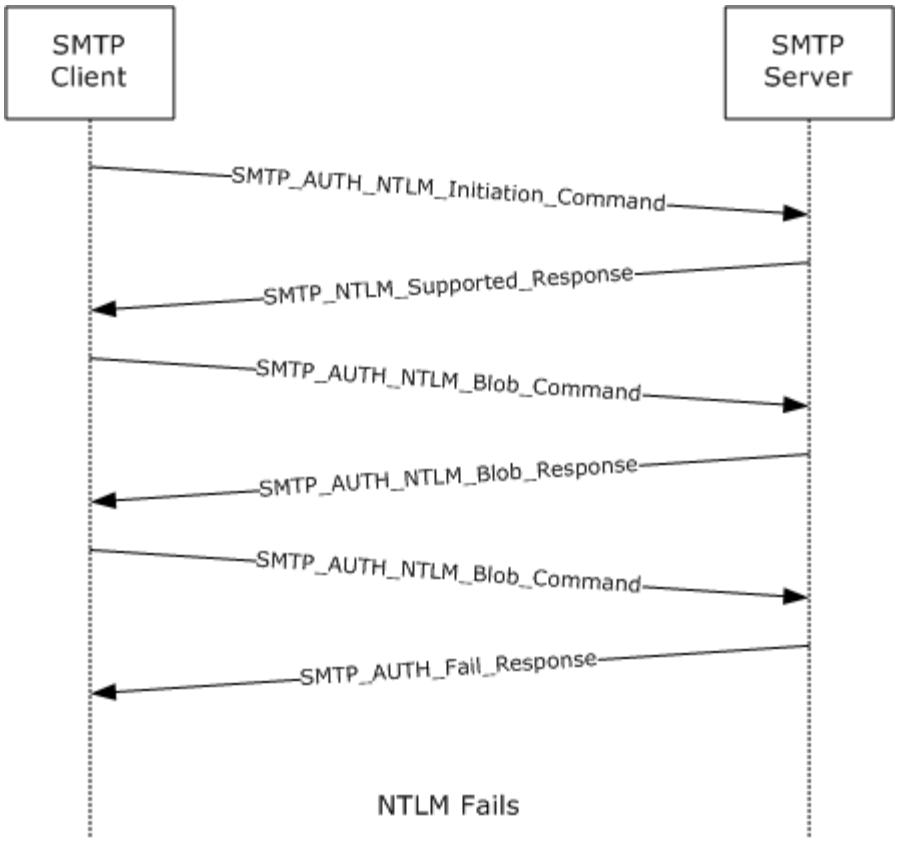

Figure 5: SMTP client unsuccessfully attempts authentication to SMTP server

1. As described in the previous example for unsuccessful AUTH, the SMTP client determines if the server supports NTLM authentication by sending the EHLO command and parsing the EHLO response.

2. The client sends an SMTP_AUTH_NTLM_Initiation_Command to the server.

```
AUTH NTLM
```

3. The server sends the SMTP_NTLM_Supported_Response message, indicating that it can perform NTLM authentication.

```
334 ntlm supported
```

4. The client sends an SMTP_AUTH_NTLM_BLOB_Command message.

```
TlRMTVNTUAABAAAAt4II4gAAAAAAAAAAAAAAAAAAAAAFAs4OAAAADw==
```

   The content of the NTLM message after base64 decoding is as follows.

```
0x00000000   4E 54 4C 4D 53 53 50 00 01 00 00 00 B7 82 08 E2   NTLMSSP.....7_.b
```

```
0x00000010  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0x00000020  05 02 CE 0E 00 00 00 0F   ..N.....
```

5. The server responds with an SMTP_AUTH_NTLM_BLOB_Response message.

```
334 TlRMTVNTUAACAAAAFgAWADgAAAA1goriYo7ENUsXagIAAAAAAAAAAGwAbABOAAA
ABQLODgAAAA9FAFgAQwBIAC0AQwBMAEkALQA2ADYAAgAWAEUAWABDAEgALQBDAEwASQ
AtADYANgABABYARQBYAEMASAAtAEMATABJAC0ANgA2AAQAFgBlAHgAYwBoAC0AYwBsA
GkALQA2ADYAAwAWAGUAeABjAGgALQBjAGwAaQAtADYANgAAAAAA
```

The content of the NTLM message after base64 decoding is as follows.

```
0x00000000  4E 54 4C 4D 53 53 50 00 02 00 00 00 16 00 16 00   NTLMSSP.........
0x00000010  38 00 00 00 35 82 8A E2 62 8E C4 35 4B 17 6A 02   8...5_bb_D5K.j.
0x00000020  00 00 00 00 00 00 00 00 6C 00 6C 00 4E 00 00 00   ........l.l.N...
0x00000030  05 02 CE 0E 00 00 00 0F 45 00 58 00 43 00 48 00   ..N.....E.X.C.H.
0x00000040  2D 00 43 00 4C 00 49 00 2D 00 36 00 36 00 02 00   -.C.L.I.-.6.6...
0x00000050  16 00 45 00 58 00 43 00 48 00 2D 00 43 00 4C 00   ..E.X.C.H.-.C.L.
0x00000060  49 00 2D 00 36 00 36 00 01 00 16 00 45 00 58 00   I.-.6.6.....E.X.
0x00000070  43 00 48 00 2D 00 43 00 4C 00 49 00 2D 00 36 00   C.H.-.C.L.I.-.6.
0x00000080  36 00 04 00 16 00 65 00 78 00 63 00 68 00 2D 00   6.....e.x.c.h.-.
0x00000090  63 00 6C 00 69 00 2D 00 36 00 36 00 03 00 16 00   c.l.i.-.6.6.....
0x000000A0  65 00 78 00 63 00 68 00 2D 00 63 00 6C 00 69 00   e.x.c.h.-.c.l.i.
0x000000B0  2D 00 36 00 36 00 00 00 00 00                     -.6.6.....
```

6. The client then sends an SMTP_AUTH_NTLM_BLOB_Command message.

```
TlRMTVNTUAADAAAAGAAYAHwAAAAYABgAlAAAABYAFgBIAAAACAAIAF4AAAAWABYAZgAAABAAEACsAAAANYKI4g
UCzg4AAAAPZQB4AGMAaAAtAGMAbABpAC0ANgA2AHQAZQBzAHQARQBYAEMASAAtAEMATABJAC0ANgA2AIqeV65h
hASwAAAAAAAAAAAAAAAAAAAAAAAAHZHDVfwTU5ci0RY04eRmWy0/VWZfIfjsqdUu2WmxYUKy83PyyxzbA8=
```

The content of the NTLM message after base64 decoding is as follows.

```
0x00000000  4E 54 4C 4D 53 53 50 00 03 00 00 00 18 00 18 00   NTLMSSP.........
0x00000010  7C 00 00 00 18 00 18 00 94 00 00 00 16 00 16 00   |........_.......
0x00000020  48 00 00 00 08 00 08 00 5E 00 00 00 16 00 16 00   H.......^.......
0x00000030  66 00 00 00 10 00 10 00 AC 00 00 00 35 82 88 E2   f.......,...5__b
0x00000040  05 02 CE 0E 00 00 00 0F 65 00 78 00 63 00 68 00   ..N.....e.x.c.h.
0x00000050  2D 00 63 00 6C 00 69 00 2D 00 36 00 36 00 74 00   -.c.l.i.-.6.6.t.
0x00000060  65 00 73 00 74 00 45 00 58 00 43 00 48 00 2D 00   e.s.t.E.X.C.H.-.
0x00000070  43 00 4C 00 49 00 2D 00 36 00 36 00 8A 9E 57 AE   C.L.I.-.6.6._W.
0x00000080  61 84 04 B0 00 00 00 00 00 00 00 00 00 00 00 00   a_.0............
0x00000090  00 00 00 00 76 47 0D 57 F0 4D 4E 5C 8B 44 58 D3   ....vG.WpMN\DXS
0x000000A0  87 91 99 6C B4 FD 55 99 7C 87 E3 B2 A7 54 BB 65   ___l4}U_|_c2'T;e
0x000000B0  A6 C5 85 0A CB CD CF CB 2C 73 6C 0F               &E_.KMOK,sl.
```

7. The server sends an SMTP_AUTH_Fail_Response message.

```
535 5.7.3 Authentication unsuccessful
```

5 Security

5.1 Security Considerations for Implementers

Implementers of the NT LAN Manager (NTLM) Authentication: Simple Mail Transfer Protocol (SMTP) Extension should be aware of the security considerations of using NTLM authentication (see [MS-NLMP] section 5).

5.2 Index of Security Parameters

Security parameter	Section
NTLM	2 and 3

6 Appendix A: Product Behavior

The information in this specification is applicable to the following Microsoft products or supplemental software. References to product versions include released service packs:

- Microsoft Windows® 2000 operating system
- Windows® XP operating system
- Windows Server® 2003 operating system
- Windows Vista® operating system
- Windows Server® 2008 operating system
- Windows® 7 operating system
- Windows Server® 2008 R2 operating system

Exceptions, if any, are noted below. If a service pack or Quick Fix Engineering (QFE) number appears with the product version, behavior changed in that service pack or QFE. The new behavior also applies to subsequent service packs of the product unless otherwise specified. If a product edition appears with the product version, behavior is different in that product edition.

Unless otherwise specified, any statement of optional behavior in this specification that is prescribed using the terms SHOULD or SHOULD NOT implies product behavior in accordance with the SHOULD or SHOULD NOT prescription. Unless otherwise specified, the term MAY implies that the product does not follow the prescription.

<1> Section 1.5: A Windows SMTP server and SMTP client use Security Support Provider Interface (SSPI) to obtain and process NTLM messages. For more information on SSPI, see [SSPI].

<2> Section 3.2.5.2.2: A Windows SMTP server does not permit a client to authenticate using credentials for the user identified as the "BUILTIN\Administrator" account, for security reasons. Internally, the NTLM software reports to the SMTP server that the authentication succeeded, but Windows SMTP then checks the user credentials and fails the authentication, sending the SMTP_AUTH_Fail_Response message even though NTLM actually succeeded the authentication.

For additional information on built-in accounts and groups, see "SID Values For Default Windows NT Installations", [MSKB-163846].

7 Change Tracking

No table of changes is available. The document is either new or has had no changes since its last release.

8 Index

T

V