[MS-OXPOP3]:
Post Office Protocol Version 3 (POP3) Extensions

Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation for protocols, file formats, languages, standards as well as overviews of the interaction among each of these technologies.
- **Copyrights.** This documentation is covered by Microsoft copyrights. Regardless of any other terms that are contained in the terms of use for the Microsoft website that hosts this documentation, you may make copies of it in order to develop implementations of the technologies described in the Open Specifications and may distribute portions of it in your implementations using these technologies or your documentation as necessary to properly document the implementation. You may also distribute in your implementation, with or without modification, any schema, IDL's, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications.
- **No Trade Secrets.** Microsoft does not claim any trade secret rights in this documentation.
- **Patents.** Microsoft has patents that may cover your implementations of the technologies described in the Open Specifications. Neither this notice nor Microsoft's delivery of the documentation grants any licenses under those or any other Microsoft patents. However, a given Open Specification may be covered by Microsoft's Open Specification Promise (available here: http://www.microsoft.com/interop/osp) or the Community Promise (available here: http://www.microsoft.com/interop/cp/default.mspx). If you would prefer a written license, or if the technologies described in the Open Specifications are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting iplg@microsoft.com.
- **Trademarks.** The names of companies and products contained in this documentation may be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights.

Reservation of Rights. All other rights are reserved, and this notice does not grant any rights other than specifically described above, whether by implication, estoppel, or otherwise.

Tools. The Open Specifications do not require the use of Microsoft programming tools or programming environments in order for you to develop an implementation. If you have access to Microsoft programming tools and environments you are free to take advantage of them. Certain Open Specifications are intended for use in conjunction with publicly available standard specifications and network programming art, and assumes that the reader either is familiar with the aforementioned material or has immediate access to it.

Revision Summary

Date	Revision History	Revision Class	Comments
04/04/2008	0.1		Initial Availability.
06/27/2008	1.0		Initial Release.
08/06/2008	1.01		Edited technical content.
09/03/2008	1.02		Revised and edited technical content.
12/03/2008	1.03		Minor editorial fixes.
03/04/2009	1.04		Revised and edited technical content.
04/10/2009	2.0		Updated technical content and applicable product releases.
07/15/2009	3.0	Major	Revised and edited for technical content.
11/04/2009	3.1.0	Minor	Updated the technical content.

Table of Contents

1 Introduction

This document specifies the implementation of extensions to the **POP3** protocol. The following extension is specified:

- **NTLM** authentication mechanism for the POP3 protocol. This is a proprietary extension that is used with the POP3 AUTH command, as documented in [RFC1734].
- Delegate access mechanism for the POP3 protocol.

For the purpose of this document, the Exchange Server NTLM POP3 extension will be referred to in subsequent sections as "NTLM POP3 Extension."

1.1 Glossary

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

ASCII
Augmented Backus-Naur Form (ABNF)
best body
Connection-Oriented NTLM
message
MIME
NTLM AUTHENTICATE_MESSAGE
NTLM CHALLENGE_MESSAGE
NTLM NEGOTIATE_MESSAGE
NTLM software
plain text

The following terms are specific to this document:

POP3 response: A message sent by a POP3 server in response to a message from a POP3 client. The structure of this message, as specified in [RFC1939], is as follows:

MAY, SHOULD, MUST, SHOULD NOT, MUST NOT: These terms (in all caps) are used as described 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", July 2006, http://go.microsoft.com/fwlink/?LinkId=111472.

[MS-OXBBODY] Microsoft Corporation, "Best Body Retrieval Protocol Specification", June 2008.

[MS-OXGLOS] Microsoft Corporation, "Exchange Server Protocols Master Glossary", June 2008.

[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.

[RFC1734] Myers, J., "POP3 AUTHentication command", RFC 1734, December 1994, http://www.ietf.org/rfc/rfc1734.txt.

[RFC1939] Myers, J. and Rose, M., "Post Office Protocol – Version 3", RFC 1939, May 1996, http://www.ietf.org/rfc/rfc1939.txt.

[RFC2045] Freed, N., et al., "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies", RFC 2045, November 1996, http://www.ietf.org/rfc/rfc2045.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.

[RFC2449] Gellens, R., Newman, C., and Lundblade, L., "POP3 Extension Mechanism", RFC 2449, November 1998, http://www.ietf.org/rfc/rfc2449.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.

[RFC822] Crocker, D.H., "Standard for ARPA Internet Text Messages", RFC 822, August 1982, http://www.ietf.org/rfc/rfc0822.txt.

1.2.2 Informative References

None.

1.3 Protocol Overview

Client applications that connect to the Post Office Protocol - Version 3 (POP3) service can use either standard **plain text** password authentication, as specified in [RFC1939], or NTLM authentication. <1>

The NTLM POP3 Extension specifies how a POP3 client and POP3 server can use the NT LAN Manager (NTLM) Authentication protocol, as specified in [MS-NLMP], so that the POP3 server can authenticate the POP3 client. NTLM 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.

This specification defines how the POP3 Authentication command, as specified in [RFC1734], is used to perform authentication by using the NTLM authentication protocol. The POP3 Authentication command standard defines an extensibility mechanism for arbitrary authentication protocols to be plugged in to the core protocol.

This specification defines an embedded protocol in which NTLM authentication data is first transformed into a base64 representation, and then formatted by padding with POP3 keywords as defined by the AUTH mechanism. The base64 encoding and the formatting are very rudimentary, and solely intended to make the NTLM data fit the framework specified in [RFC1734]. Figure 1 shows the sequence of transformations that are performed on an **NTLM Message** to produce a **Message** that can be sent over POP3.

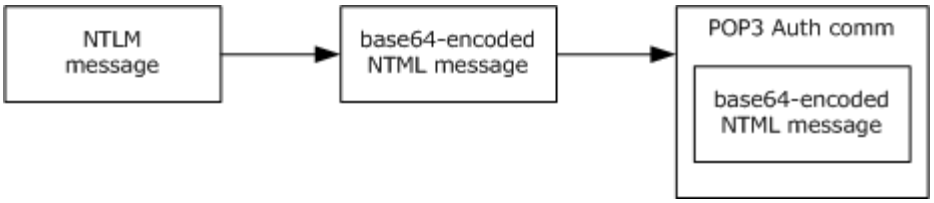

Figure 1: Relationship between NTLM Message **and** POP3 **:** NTLM **Authentication Protocol** Message

This document specifies 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 that is to be sent or received.

This specification defines a server and a client role.

When POP3 performs an NTLM authentication, it needs to interact with the NTLM subsystem appropriately. The following is an overview of this interaction.

If acting as a POP3 client:

1. The NTLM subsystem returns the first NTLM Message to the client, to be sent to the server.
2. The client applies the base64-encoding and POP3-padding transformations mentioned earlier and described in detail later in this **document** to produce a POP3 Message and send this Message to the server.
3. The client waits for a response from the server. When the response is received, the client checks to determine whether the response indicates the end of authentication (success or failure), or that authentication is continuing.
4. If the authentication is continuing, the response Message is stripped of the POP3 padding, base64 decoded, and passed into the NTLM subsystem, upon which the NTLM subsystem might return another NTLM Message that has to be sent to the server. Steps 2 through 4 are repeated until authentication succeeds or fails.

If acting as a POP3 server:

1. The server waits to receive the first POP3 authentication Message from the client.
2. When a POP3 Message is received from the client, the POP3 padding is removed, the Message is base64 decoded, and the resulting NTLM Message is passed into the NTLM subsystem.
3. The NTLM subsystem returns a status that indicates whether authentication completed successfully, failed, or whether more NTLM messages have to be exchanged to complete the authentication.
4. If the authentication is continuing, the NTLM subsystem will return an NTLM Message that has to be sent to the server. This Message is base64-encoded, and the POP3 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 client and server after NTLM authentication has been selected.

1. The POP3 client sends an **NTLM NEGOTIATE_MESSAGE** embedded in a POP3_AUTH_NTLM_Blob_Command packet to the server.

2. On receiving the POP3 packet with an NTLM NEGOTIATE_MESSAGE, the POP3 server sends an **NTLM CHALLENGE_MESSAGE** embedded in a POP3 packet to the client.

3. In response, the POP3 client sends an **NTLM AUTHENTICATE_MESSAGE** embedded in a POP3 packet.

4. The server then sends a **POP3 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 has to be processed by the **NTLM software** that is installed on the local computer. The way in which to retrieve and process NTLM messages is specified in [MS-NLMP].

This specification defines the delegate access mechanism that is used by a POP3 client.

Implementers of this specification have to conform to POP3, as specified [RFC1734] and [RFC1939], the **MIME** base64 encoding method, as specified in [RFC1521], and the NTLM Authentication protocol, as specified in [MS-NLMP].

1.4 Relationship to Other Protocols

The NTLM POP3 Extension uses the POP3 AUTH extension mechanism, as specified in [RFC1734], and is an embedded protocol. Unlike stand-alone application protocols, such as Telnet or HTTP, packets for this specification are embedded in POP3 commands <2> and server responses.

POP3 specifies only the sequence in which a POP3 server and POP3 client are required to 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 POP3 server processes NTLM Messages. The POP3 client and POP3 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 POP3 packets.

1.5 Prerequisites/Preconditions

Because POP3 depends on NTLM to authenticate the client to the server, both server and client will have access to an implementation of the NTLM Authentication protocol (as specified in [MS-NLMP]) that is capable of supporting **Connection-Oriented NTLM**.

1.6 Applicability Statement

The NTLM POP3 Extension is used only when implementing a POP3 client that has to authenticate to a POP3 server by using NTLM authentication.

1.7 Versioning and Capability Negotiation

This specification covers versioning issues in the following areas:

- Security and Authentication methods. The NTLM POP3 Extension supports the NTLMv1 and NTLMv2 authentication methods, as specified in [MS-NLMP].
- Capability Negotiation. POP3 does not support negotiation of which version of the NTLM Authentication protocol to use. Instead, the NTLM Authentication protocol version is configured on both the client and the server prior to authentication. NTLM Authentication protocol version mismatches are handled by the NTLM Authentication protocol implementation, not by POP3.

The client discovers whether the server supports NTLM AUTH through the AUTH command, issued without any arguments, upon which the server responds with a list of supported authentication mechanisms followed by a line that contains only a period (.). If NTLM is supported, the server will include the word "NTLM" in the list. The messages involved are formally described in section 2.2 of this document. <3><4>

1.8 Vendor-Extensible Fields

None.

1.9 Standards Assignments

None.

2 Messages

The following sections specify how the NTLM POP3 Extension messages are transported and NTLM POP3 Extension Message syntax.

2.1 Transport

The NTLM POP3 Extension does not establish transport connections. Instead, NTLM POP3 Extension messages are encapsulated in POP3 commands and responses. The way in which NTLM POP3 Extension messages are encapsulated in POP3 commands is specified in section 2.2.

2.2 Message Syntax

The NTLMPOP3 Extension messages are divided into the three categories, depending on whether the Message was sent by the server or the client:

- AUTH extensions
- POP3 server messages
- POP3 client messages

The POP3 USER command extension enables optional delegate access. The USER command extension adds an additional optional parameter that identifies the principal in a delegate access scenario. The USER command has four extended formats, as specified in [RFC1939].

2.2.1 AUTH Extensions

The first category of POP3 messages is messages that fall within the AUTH extensibility framework. These messages are specified in [RFC1734]. Some messages have parameters that have to be customized by the extensibility mechanism (such as NTLM). The following customizations are introduced in this specification:

- A client can query the server to see if NTLM is supported. This is accomplished by issuing the AUTH command without any parameters. This format is shown in **ABNF** in the following example (for more information about ABNF, see [RFC4234]).<5>

AUTH<CR><LF>

- The server responds to this Message with a Message followed by a list of supported authentication mechanisms, followed by a list termination Message. This sequence is shown in ABNF format in the following example.

+OK<CR><LF>

NTLM GSSAPI PLAIN<CR><LF>

.<CR><LF>

- [RFC1734] section 2 defines the syntax of the AUTH command to initiate authentication. The parameter "mechanism" is defined to be the string "NTLM" for the NTLM POP3 Extension. The command to initiate an NTLM conversation by a client in ABNF is shown in the following example. This is referred to as the POP3_AUTH_NTLM_Initiation_Command in this specification.

AUTH NTLM<CR><LF>

- If NTLM is supported, the POP3 server will respond with a POP3 Message to indicate that NTLM is supported. The syntax of this command in ABNF form is shown in the following example. This is referred to as POP3_NTLM_Supported_Response in this specification.

+<CR><LF>

- If NTLM is not supported, the POP3 server returns a failure status code as defined by [RFC1734]. The only data in this Message that is useful is the -ERR. The remaining data is human-readable data and has no bearing on the authentication. The syntax of this command in ABNF form is shown in the following example. This is referred to as POP3_AUTH_NTLM_Fail_Response in this specification.

-ERR <human_readable_string> <CR><LF>

- At every point of time during the authentication exchange, the client MUST parse the responses in the messages sent by the server and interpret them as defined by [RFC1734]. The responses define various states such as success in authenticating, failure to authenticate, and any other arbitrary failures that the software MAY encounter.

The client can receive any one of the following responses during authentication (note that the syntax and meaning of all these messages are specified in [RFC1734]):

- POP3_AUTH_NTLM_Blob_Response. This Message is partially defined in [RFC1734]. The '+' status code indicates ongoing authentication, and indicates that the <base64-encoded-NTLM-Message> is to be processed by the authentication subsystem. In this case, the client MUST de-encapsulate the data, and pass it to the NTLM subsystem.

+ < base64-encoded-NTLM-Message><CR><LF>

- POP3_AUTH_NTLM_Fail_Response. This Message is defined in [RFC1734] and indicates that the authentication has terminated unsuccessfully, either because the username or password was incorrect, or due to some other arbitrary error, such as a software or data corruption error.

-ERR <human-readable-string><CR><LF>

- POP3_AUTH_NTLM_Succeeded_Response. This Message is defined in [RFC1734] and indicates that the authentication negotiation has completed with the client successfully authenticating to the server.

+OK <human-readable-string><CR><LF>

- POP3_AUTH_NTLM_Cancelled_Response. This Message is defined in [RFC1734] and indicates that the authentication negotiation has been canceled with the client.

-ERR <human-readable-string><CR><LF>

- NTLM messages encapsulated by the client and sent to the server are referred to as POP3_AUTH_NTLM_Blob_Command in this specification. They have the following syntax defined in ABNF, and conform to the prescription as specified in [RFC1734].

< base64-encoded-NTLM-Message><CR><LF>

- The client is able to cancel the authentication request by issuing a POP3_AUTH_Cancellation_Command. This has the following syntax defined in ABNF:

*<CR><LF>

2.2.2 POP3 Server Messages

This section defines the creation of POP3_AUTH_NTLM_Blob_Response messages. These are NTLM Messages sent by the server and MUST be encapsulated as follows to conform to syntax specified by the AUTH mechanism:

1. Base64-encode the NTLM message data. <6>
2. To the base64-encoded string, prefix the POP3 response code with a plus sign (+).
3. Suffix the <CR> and <LF> character (**ASCII** values 0x0D and 0x0A) as required by POP3.

The ABNF definition of a server Message is as follows:

+ <base64-encoded-NTLM-Message><CR><LF>

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

1. Remove the <CR> and <LF> character (ASCII values 0x0D and 0x0A).
2. Remove the POP3 response code (+).
3. Decode the base64-encoded POP3 data to produce the original NTLMMessage data. <7>

2.2.3 POP3 Client Messages

This section defines the processing of POP3_AUTH_NTLM_Blob_Command messages. These NTLM messages sent by the client are encapsulated as follows to conform to the AUTH mechanism:

1. Base64-encode the NTLM message data. This is needed because NTLM messages contain data outside the ASCII character range, whereas POP3 only supports ASCII characters.
2. Send the base64-encoded string.
3. Suffix the <CR> and <LF> character (ASCII values 0x0D and 0x0A), as required by POP3.

The ABNF definition of a client Message is as follows:

<base64-encoded-NTLM-Message><CR><LF>

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

1. Remove the <CR> and <LF> character (ASCII values 0x0D and 0x0A).
2. Base64-decode the POP3 data to produce the original NTLM message data.

2.2.4 POP3 Delegate Access

There are four formats for using delegate access with POP3. In every case, the part after the last "/" of the user string is the mailbox identity in either alias or user principal name (UPN) format. The four formats are as follows:

- USER domain/delegateuseralias/principalalias
- USER domain/delegateuseralias/principalupn
- USER delegateuserupn/principalalias

- USER delegateuserupn/principalupn

3 Protocol Details

3.1 POP3 Client Details

3.1.1 Abstract Data Model

3.1.1.1 POP3 State Model

Figure 2 shows the client POP3 state model.

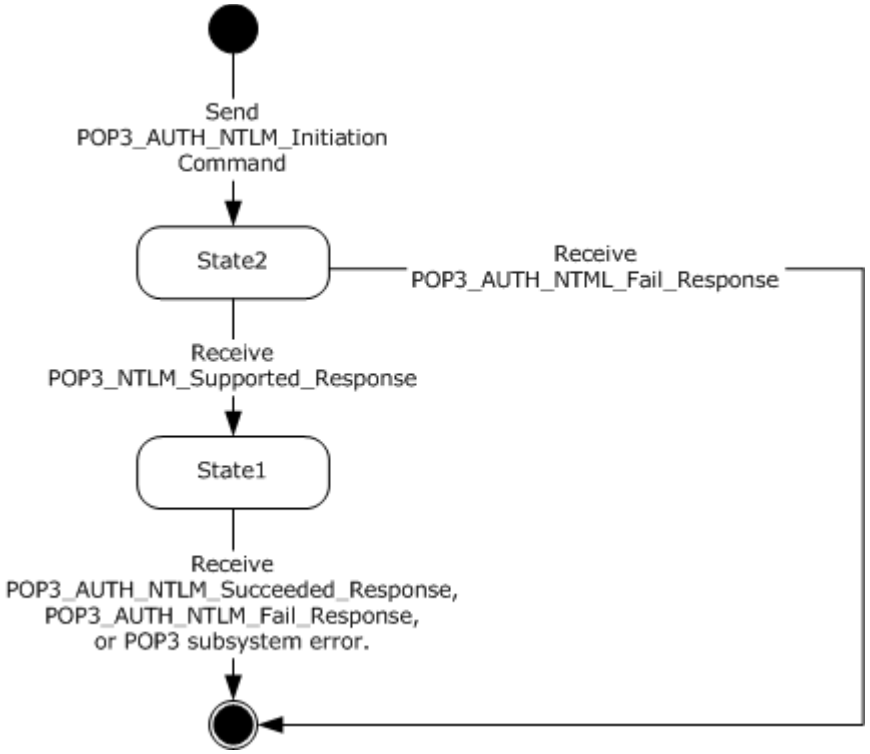

Figure 2: Client POP3 state model

The abstract data model for the NTLM POP3 Extension has the following states:

1. Start.

This is the state of the client before the POP3_AUTH_Initiation_Command has been sent.

2. State 2: sent_authentication_request.

This is the state of the client after the POP3_AUTH_Initiation_Command has been sent.

3. State 1: inside_authentication.

This is the state entered by a client after it has received a POP3_NTLM_Supported_Command. In this state, the client initializes the NTLM subsystem and repeats the following steps:

- Encapsulates the NTLM message, returned by the NTLM subsystem, into a POP3 Message. Waits for a response from the server.

- De-encapsulates received POP3 Message data (if any) from the other party and converts it to NTLM messages data.
- Passes it to the NTLM subsystem.
- Sends the POP3 Message to the other party.

This state terminates when:

- For the server: The NTLM subsystem reports completion with either a success or failed authentication status, upon which it sends the client a POP3_AUTH_NTLM_Suceeded_Response or POP3_AUTH_NTLM_Fail_Response, as specified in [RFC1734].
- For the client: a POP3_AUTH_NTLM_Suceeded_Response or POP3_AUTH_NTLM_Fail_Response is received.
- For either client or server: when any failure is reported by the NTLM subsystem.

4. Stop: completed_authentication.

This is the state of the client on exiting the inside_authentication state. The rules for how the inside_authentication state is exited are defined in section 3.1.5. The behavior of POP3 in this state is not in the scope of this specification. It represents the end state of the authentication protocol.

3.1.1.2 NTLM Subsystem Interaction

During the inside_authentication phase, the POP3 client invokes the NTLM subsystem, as specified in [MS-NLMP] section 3.1. The NTLM protocol 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 specific to NTLM.

The following is a description of how POP3 uses NTLM. All NTLM messages are encapsulated as specified in section 2.1. [MS-NLMP] section 3.1.1 describes the data model, internal states, and sequencing of NTLM messages in greater detail.

1. The client initiates the authentication by invoking NTLM, after which NTLM will return the NTLM NEGOTIATE_MESSAGE to be sent to the server.
2. Subsequently, the exchange of NTLM messages goes on as defined by the NTLM protocol, with the POP3 client encapsulating the NTLM messages before sending them to the server, and de-encapsulating POP3 messages to obtain the NTLM message before giving it to NTLM.
3. The NTLM protocol completes authentication, either successfully or unsuccessfully, as follows:

- The server sends the POP3_AUTH_NTLM_Succeeded_Response 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 POP3_AUTH_NTLM_Fail_Response 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 package (which can occur for any reason, including incorrect data being passed in, or implementation-specific errors) MAY be reported to the client by NTLM 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 NTLMPOP3 Extension is driven by a series of Message exchanges between a POP3 server and a POP3 client. The rules that govern the sequencing of commands and the internal states of the client and server are defined by a combination of [RFC1734] and [MS-NLMP]. Section 3.1.1 defines how the rules specified in [RFC1734] and [MS-NLMP] govern POP3 authentication.

3.1.5.1 Receiving a POP3_NTLM_Supported_Response Message

The expected state is sent_authentication_request.

On receiving this Message, a client MUST generate the first NTLM message by calling the NTLM subsystem. The NTLM subsystem then generates NTLM NEGOTIATE_MESSAGE, as specified in [MS-NLMP]. The NTLM message is then encapsulated as defined previously and sent to the server.

The state of the client is changed to "inside_authentication".

3.1.5.2 Receiving a POP3_AUTH_NTLM_Fail_Response Message

The expected state is sent_authentication_request.

On receiving this Message, a client MUST abort the NTLM authentication attempt.

3.1.5.3 Receiving a POP3_NTLM_Blob_Response

The expected state is inside_authentication.

On receiving this Message, a client MUST de-encapsulate it to obtain the embedded NTLM message, and pass it to the NTLM subsystem for processing. The NTLM subsystem can then either report an error, or report success and return an NTLM message to be sent to the server.

3.1.5.3.1 Error from NTLM

If the NTLM subsystem reports an error, the client MUST change its internal state to "completed_authentication" and consider that the authentication has failed. The client can then take any action it considers appropriate; this specification does not mandate any specific course of action.

Typical actions are to try other (non-authentication-related) POP3 commands, or to disconnect the connection.

3.1.5.3.2 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.

3.1.5.4 Receiving a POP3_AUTH_Succeeded_Response Message

Expected state: inside_authentication.

The POP3 client MUST change its internal state to completed_authentication and consider that the authentication has succeeded. The client can then take any action it considers appropriate. This specification does not mandate any specific course of action.

3.1.5.5 Receiving a POP3_AUTH_NTLM_Fail_Response

Expected state: inside_authentication.

The POP3 client MUST change its internal state to completed_authentication and consider that the authentication has failed. The client can then take any action it considers appropriate; this specification does not mandate any specific course of action.

3.1.6 Timer Events

None.

3.1.7 Other Local Events

None.

3.2 POP3 Server Details

3.2.1 Abstract Data Model

3.2.1.1 POP3 State Model

Figure 3 shows the server POP3 state model.

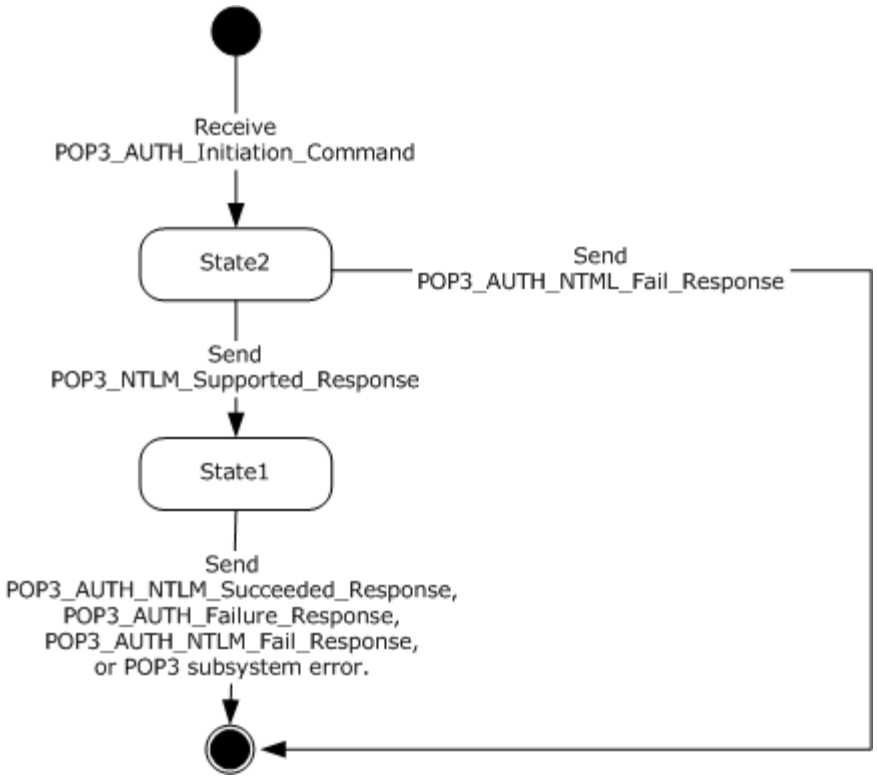

Figure 3: Server POP3 state model

The abstract data model for NTLM POP3 Extension has the following states:

1. Start.

This is the state of the server before the POP3_AUTH_NTLM_Initiation_Command has been received.

2. State 2: received_authentication_request.

This is the state of the client after the POP3_AUTH_NTLM_Initiation_Command has been received.

3. State 1: inside_authentication.

This is the state entered by a server after it has sent a POP3_NTLM_Supported_Response. In this **state**, the server initializes the NTLM subsystem and repeats the following steps:

- Waits for a Message from the client.
- De-encapsulates received POP3 Message data from the other party and obtains the embedded NTLM Message data.
- Passes it to the NTLM subsystem.
- Encapsulates the NTLM message returned by the NTLM subsystem into a POP3 Message.
- Sends the POP3 Message to the other party.

This state terminates when:

- The NTLM subsystem reports completion with either a success or failed authentication status, on which it sends the client and POP3_AUTH_NTLM_Succeeded_Response or POP3_AUTH_NTLM_Fail_Response, as specified in [RFC1734].
- Any failure is reported by the NTLM subsystem.

4. Stop: completed_authentication.

This is the state of the server on exiting the inside_authentication state. The rules for how the inside_authentication state is exited are defined in section 3.2.5. The behavior of POP3 in this state is specified in [RFC1734]—it represents the end_state of the authentication protocol.

3.2.1.2 NTLM Subsystem Interaction

During the inside_authentication state, the POP3 server invokes the NTLM subsystem as specified in [MS-NLMP] section 3.1.1. The NTLM protocol is used with the following options:

1. The negotiation is a Connection-Oriented NTLM negotiation.
2. None of the flags specified in [MS-NLMP] section 3.1.1 are specific to NTLM.

The following is a description of how POP3 uses NTLM. For more details, see [MS-NLMP] section 3.1.1, which describes the data model and sequencing of NTLM packets in greater detail.

1. On receiving the NTLM NEGOTIATE_MESSAGE, the server passes it to the NTLM subsystem and is returned the NTLM CHALLENGE_MESSAGE, if the NTLM NEGOTIATE_MESSAGE was valid.
2. Subsequently, the exchange of NTLMmessages goes on as defined by the NTLM protocol, with the POP3 server encapsulating the NTLMmessages that are returned by NTLM before sending them to the client.
3. When the NTLM protocol completes authentication, either successfully or unsuccessfully, the NTLM subsystem notifies POP3, and the following occurs:

- On successful completion, the server MUST exit the inside_authentication state and enter the completed_authentication state and send the POP3_AUTH_Success_Response 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 described in [MS-NLMP] section 3.3.1 and 3.3.2, the server SHOULD enter the completed_authentication state and send the client a POP3_AUTH_Failure_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 a POP3_AUTH_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 NTLM POP3 Extension is driven by a series of Message exchanges between a POP3 server and a POP3 client. The rules that govern the sequencing of commands and the internal states of the client and server are defined by a combination of [RFC1734] and [MS-NLMP]. Section 3.1.1 defines how the rules specified in [RFC1734] and [MS-NLMP] govern POP3 authentication.

3.2.5.1 Receiving a POP3_AUTH_NTLM_Initiation_Command Message

The expected state is start.

On receiving this Message, the server MUST reply with the POP3_NTLM_Supported_Response, if it supports NTLM, and change its state to the inside_authentication state.

If the server does not support NTLM, it MUST respond with the POP3_NTLM_AUTH_Fail_Response, and the internal state remains unchanged.

3.2.5.2 Receiving a POP3_AUTH_NTLM_Blob_Command Message

The expected state is inside_authentication.

On receiving this message, a server MUST de-encapsulate the message, obtain the embedded NTLM message, and pass it to the NTLM subsystem. The NTLM subsystem MAY:

1. Report success in processing the message and return an NTLM message to continue authentication.
2. Report that authentication completed successfully.
3. Report that authentication failed due to a bad user name or password, as specified in [MS-NLMP].
4. Report that the authentication failed due to some other software error or message corruption.

3.2.5.2.1 NTLM Returns Success, Returning an NTLM Message

The NTLM message MUST be encapsulated and sent to the client. The internal state of the POP3 server remains unchanged.

3.2.5.2.2 NTLM Returns Success, Indicating Authentication Completed Successfully

The server MUST return the POP3_NTLM_AUTH_Succeeded_Response and change its internal state to completed_authentication.

3.2.5.2.3 NTLM Returns Status, Indicating User Name or Password Was Incorrect

The server MUST return the POP3_AUTH_NTLM_Failed_Response and change its internal state to completed_authentication.

3.2.5.2.4 NTLM Returns a Failure Status, Indicating Any Other Error

The server MUST return the POP3_AUTH_NTLM_Failed_Response and change its internal state to completed_authentication.

3.2.5.2.5 NTLM Reports Success, Returning an NTLM Message

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

3.2.6 Timer Events

None.

3.2.7 Other Local Events

None.

4 Protocol Examples

The following section describes operations used in a common scenario to illustrate the function of the Post Office Protocol - Version3 (POP3).

4.1 POP3 Client Successfully Authenticating to a POP3 Server

This section illustrates the NTLM POP3 Extension with a scenario in which a POP3 client successfully authenticates to a POP3 server by using NTLM. Figure 4 shows a POP3 client authenticating to a POP3 server.

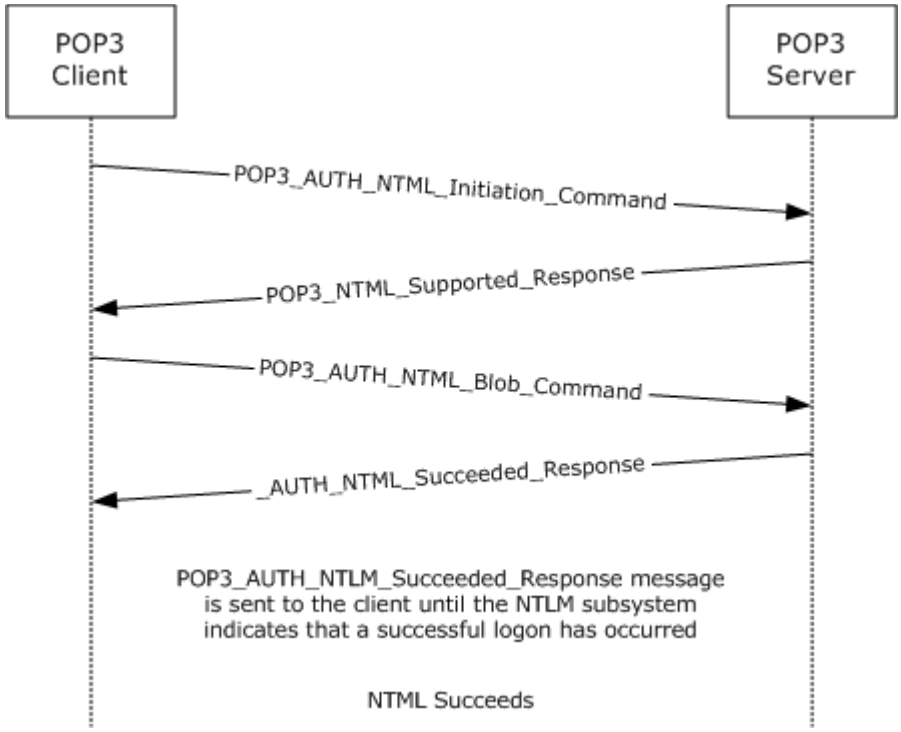

Figure 4: POP3 client successfully authenticating to POP3 server

1. The client sends a POP3_AUTH_NTLM_Initiation_Command to the server. This command is specified in [RFC1734] and does not carry any POP3-specific data. It is included in this example to provide a better understanding of the POP3 NTLM initiation command.

AUTH NTLM

2. The server sends the POP3_NTLM_Supported_Response Message, which indicates that it can perform NTLM authentication.

+OK

3. The client sends a POP3_AUTH_NTLM_Blob_Command Message that contains a base64-encoded NTLM NEGOTIATE_MESSAGE.

```
TlRMTVNTUAABAAAAB4IIogAAAAAAAAAAAAAAAAAAAAAFASgKAAAADw==
```

```
00000000:4e 54 4c 4d 53 53 50 00 01 00 00 00 07 82 08 a2 NTLMSSP......,.¢
00000010:00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00000020:05 01 28 0a 00 00 00 0f ..(.....
```

4. The server sends a POP3_AUTH_NTLM_Blob_Response Message that contains a base64-encoded NTLM CHALLENGE_MESSAGE.

```
+ TlRMTVNTUAACAAAAFAAUADgAAAAFgoqinziKqGYjdlEAAAAAAAAAAAGQAZABMAAAABQ
LODgAAAA9UAEUAUwBUAFMARQBSAFYARQBSAAIAFABUAEUAUwBUAFMARQBSAFYARQBSAA
EAFABUAEUAUwBUAFMARQBSAFYARQBSAAQAFABUAGUAcwB0AFMAZQByAHYAZQByAAMAFA
BUAGUAcwB0AFMAZQByAHYAZQByAAAAAAA=
00000000:4e 54 4c 4d 53 53 50 00 02 00 00 00 14 00 14 00 NTLMSSP.........
00000010:38 00 00 00 05 82 8a a2 9f 38 8a a8 66 23 76 51 8....,Š¢Ÿ8Š¨f#vQ
00000020:00 00 00 00 00 00 00 00 64 00 64 00 4c 00 00 00 ........d.d.L...
00000030:05 02 ce 0e 00 00 00 0f 54 00 45 00 53 00 54 00 ..Î.....T.E.S.T.
00000040:53 00 45 00 52 00 56 00 45 00 52 00 02 00 14 00 S.E.R.V.E.R.....
00000050:54 00 45 00 53 00 54 00 53 00 45 00 52 00 56 00 T.E.S.T.S.E.R.V.
00000060:45 00 52 00 01 00 14 00 54 00 45 00 53 00 54 00 E.R.....T.E.S.T.
00000070:53 00 45 00 52 00 56 00 45 00 52 00 04 00 14 00 S.E.R.V.E.R.....
00000080:54 00 65 00 73 00 74 00 53 00 65 00 72 00 76 00 T.e.s.t.S.e.r.v.
00000090:65 00 72 00 03 00 14 00 54 00 65 00 73 00 74 00 e.r.....T.e.s.t.
000000a0:53 00 65 00 72 00 76 00 65 00 72 00 00 00 00 00 S.e.r.v.e.r....
```

5. The client sends a POP3_AUTH_NTLM_Blob_Command Message that contains a base64-encoded NTLM AUTHENTICATE_MESSAGE.

```
TlRMTVNTUAADAAAAGAAYAGIAAAAYABgAegAAAAAAAABIAAAACAAIAEgAAAASABIAUAAA
AAAAAACSAAAABYKIogUBKAoAAAAPdQBzAGUAcgBOAEYALQBDAEwASQBFAE4AVABKMiQ4
djhcSgAAAAAAAAAAAAAAAAAAAAC7zUSgB0Auy98bRi6h3mwHMJfbKNtxmmo=
00000000:4e 54 4c 4d 53 53 50 00 03 00 00 00 18 00 18 00 NTLMSSP.........
00000010:62 00 00 00 18 00 18 00 7a 00 00 00 00 00 00 00 b.......z.......
00000020:48 00 00 00 08 00 08 00 48 00 00 00 12 00 12 00 H.......H.......
00000030:50 00 00 00 00 00 00 00 92 00 00 00 05 82 88 a2 P.......'....,ˆ¢
00000040:05 01 28 0a 00 00 00 0f 75 00 73 00 65 00 72 00 ..(.....u.s.e.r.
00000050:4e 00 46 00 2d 00 43 00 4c 00 49 00 45 00 4e 00 N.F.-.C.L.I.E.N.
00000060:54 00 4a 32 24 38 76 38 5c 4a 00 00 00 00 00 00 T.J2$8v8\J......
00000070:00 00 00 00 00 00 00 00 00 00 bb cd 44 a0 07 40 ..........»ÍD .@
00000080:2e cb df 1b 46 2e a1 de 6c 07 30 97 db 28 db 71 .Ëß.F.¡Þl.0—Û(Ûq
00000090:9a 6a šj
```

6. The server sends a POP3_Authentication_Succeeded_Response Message.

+OK User successfully logged on

4.2 POP3 Client Unsuccessfully Authenticating to a POP3 Server

This section illustrates the NTLM POP3 Extension with a scenario in which a POP3 client tries NTLM authentication to a POP3 server and the authentication fails. Figure 5 shows the unsuccessful attempt to authenticate to the POP3 server.

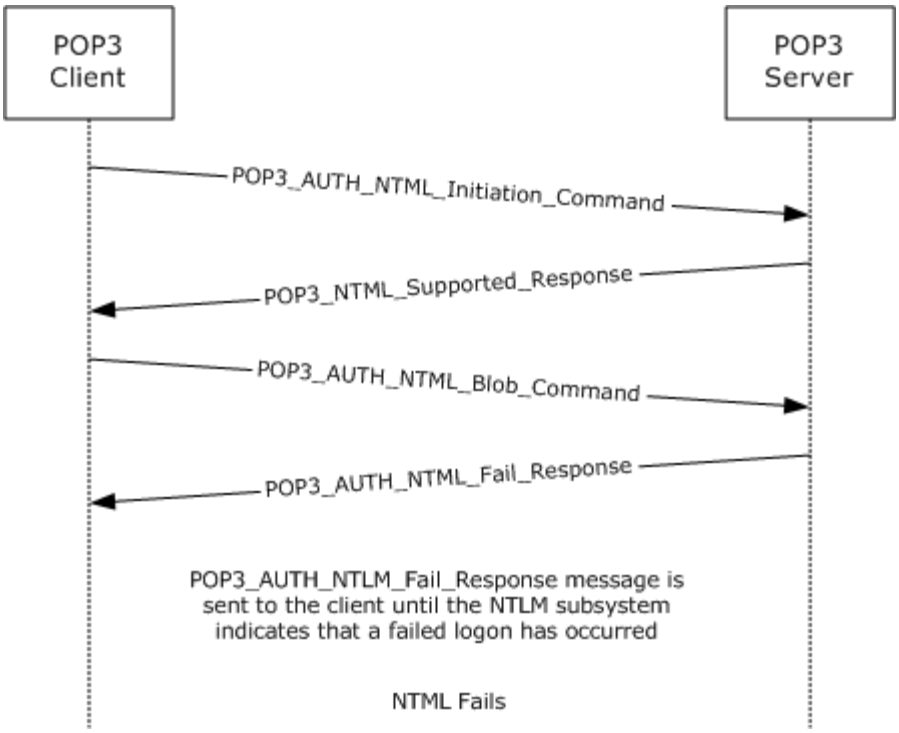

Figure 5: Client unsuccessfully authenticating to POP3 server

The client sends a POP3_AUTH_NTLM_Initiation_Command to the server. This command is defined in [RFC1734] and does not carry any POP3-specific data. It is included in this example to provide a better understanding.

The server sends the POP3_NTLM_Supported_Response Message, which indicates that it can perform NTLM authentication.

The client sends a POP3_AUTH_NTLM_Blob_Command Message.

The client sends a POP3_AUTH_NTLM_Blob_Command Message that contains a base-64-encoded NTLM NEGOTIATE_MESSAGE.

```
TlRMTVNTUAABAAAAB4IIogAAAAAAAAAAAAAAAAAAAAAFASgKAAAADw==

00000000:4e 54 4c 4d 53 53 50 00 01 00 00 00 07 82 08 a2 NTLMSSP.....,.¢
00000010:00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00000020:05 01 28 0a 00 00 00 0f ..(.....
```

The server sends a POP3_AUTH_NTLM_Blob_Response Message that contains a base64-encoded NTLM CHALLENGE_MESSAGE.

```
+ TlRMTVNTUAACAAAAFAAUADgAAAAFgoqieUWd5ES4Bi0AAAAAAAAAAGQAZABMAA

AABQLODgAAAA9UAEUAUwBUAFMARQBSAFYARQBSAAIAFABUAEUAUwBUAFMARQBSAF
YARQBSAAEAFABUAEUAUwBUAFMARQBSAFYARQBSAAQAFABUAGUAcwB0AFMAZQByAH
YAZQByAAMAFABUAGUAcwB0AFMAZQByAHYAZQByAAAAAAA=
```

```
00000000:4e 54 4c 4d 53 53 50 00 02 00 00 00 14 00 14 00 NTLMSSP.........
00000010:38 00 00 00 05 82 8a a2 79 45 9d e4 44 b8 06 2d 8.....,Š¢yE•äD¸.-
00000020:00 00 00 00 00 00 00 00 64 00 64 00 4c 00 00 00 ........d.d.L...
00000030:05 02 ce 0e 00 00 00 0f 54 00 45 00 53 00 54 00 ..Î.....T.E.S.T.
00000040:53 00 45 00 52 00 56 00 45 00 52 00 02 00 14 00 S.E.R.V.E.R.....
00000050:54 00 45 00 53 00 54 00 53 00 45 00 52 00 56 00 T.E.S.T.S.E.R.V.
00000060:45 00 52 00 01 00 14 00 54 00 45 00 53 00 54 00 E.R.....T.E.S.T.
00000070:53 00 45 00 52 00 56 00 45 00 52 00 04 00 14 00 S.E.R.V.E.R.....
00000080:54 00 65 00 73 00 74 00 53 00 65 00 72 00 76 00 T.e.s.t.S.e.r.v.
00000090:65 00 72 00 03 00 14 00 54 00 65 00 73 00 74 00 e.r.....T.e.s.t.
000000a0:53 00 65 00 72 00 76 00 65 00 72 00 00 00 00 00 S.e.r.v.e.r.....
```

The client sends a POP3_AUTH_NTLM_Blob_Command Message that contains a base-64-encoded NTLM AUTHENTICATE_MESSAGE.

```
TlRMTVNTUAADAAAAGAAYAGIAAAAYABgAegAAAAAAAABIAAAACAAIAEgAAAASABIA
UAAAAAAAAACSAAAABYKIogUBKAoAAAAPdQBzAGUAcgBOAEYALQBDAEwASQBFAE4A
VAAOarJ6lZ5ZNwAAAAAAAAAAAAAAAAAAAACD9mD8jmWs4FkZe59/nNb1cF2HkL0C
GZw=

00000000:4e 54 4c 4d 53 53 50 00 03 00 00 00 18 00 18 00 NTLMSSP.........
00000010:62 00 00 00 18 00 18 00 7a 00 00 00 00 00 00 00 b.......z.......
00000020:48 00 00 00 08 00 08 00 48 00 00 00 12 00 12 00 H.......H.......
00000030:50 00 00 00 00 00 00 00 92 00 00 00 05 82 88 a2 P.......'....,ˆ¢
00000040:05 01 28 0a 00 00 00 0f 75 00 73 00 65 00 72 00 ..(.....u.s.e.r.
00000050:4e 00 46 00 2d 00 43 00 4c 00 49 00 45 00 4e 00 N.F.-.C.L.I.E.N.
00000060:54 00 0e 6a b2 7a 95 9e 59 37 00 00 00 00 00 00 T..j²z•ţY7......
00000070:00 00 00 00 00 00 00 00 00 00 83 f6 60 fc 8e 65 ..........fö`üŢe
00000080:ac e0 59 19 7b 9f 7f 9c d6 f5 70 5d 87 90 bd 02 ¬àY.{ŸœeÖõp]‡•½.
00000090:19 9c .oe
```

The server sends a POP3_AUTH_Failed_Response Message.

```
-ERR, Error: Command not valid
```

5 Security

The following sections specify security considerations for implementers of the NTLMPOP3 Extension.

5.1 Security Considerations for Implementers

Implementers ought to be aware of the security considerations of using NTLM authentication. Information about the security considerations for using NTLM authentication is specified in [MS-NLMP].

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 product versions. References to product versions include released service packs.

- Microsoft Office Outlook 2003
- Microsoft Exchange Server 2003
- Microsoft Office Outlook 2007
- Microsoft Exchange Server 2007
- Microsoft Outlook 2010
- Microsoft Exchange Server 2010

Exceptions, if any, are noted below. If a service pack number appears with the product version, behavior changed in that service pack. The new behavior also applies to subsequent service packs of the product unless otherwise specified.

Unless otherwise specified, any statement of optional behavior in this specification 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 product does not follow the prescription.

<1> Section 1.3: POP3 servers support at least one authentication mechanism. Outlook 2003 supports USER and PASS verbs as defined in [RFC1939]. Outlook 2007 supports the AUTH verb as defined in [RFC1734]. Outlook 2003 and Outlook 2007 do not support APOP. Both Outlook 2003 and Outlook 2007 support AUTH NTLM.

<2> Section 1.4: In Exchange 2007, the maximum POP3 command size is 45 characters. In Exchange 2010, the maximum POP3 command size is 512 characters.

<3> Section 1.7: Exchange 2003, Exchange 2007, Exchange 2010 and Outlook clients mutually support [RFC1939], [RFC1734], and [RFC2449].

<4> Section 1.7: Exchange 2010 is not [RFC822]-compliant by default. Exchange 2010 can be made [RFC822]-compliant by setting EnableExactRFC822Size to TRUE.

<5> Section 2.2.1: In Exchange 2003, Exchange 2007, and Exchange 2010, the AUTH command issued without arguments will list the supported authentication mechanisms. This is not part of [RFC1734].

<6> Section 2.2.2: In Exchange 2003, messages received by using MAPI are converted to MIME the first time they are retrieved and subsequently stored. The MIME size can be different before the Message is retrieved than after it is converted to MIME. This is needed because NTLMmessages contain data outside the ASCII character range, whereas POP3 only supports ASCII characters.

<7> Section 2.2.2: In Exchange 2010, messages are not stored in MIME format. messages are converted from MAPI to MIME before being sent to the client. When the client requests the size of the Message before retrieving the actual Message itself, the MIME size provided is the size associated with the Message as a MAPI **property**. When the client retrieves the Message, the Message is converted from MAPI to MIME and the Message size calculated thereafter can be different than the size calculated from the MAPI property. Additionally, if the Message is modified by MAPI, the size is likely to change again and thus the corresponding MIME size after actual

conversion from MAPI to MIME.In Exchange 2003, the MIME**stream** is preserved. In Exchange 2007 and Exchange 2010, the MIMEstream is not preserved. Only content of the **best body** as specified in [MS-OXBBODY] is preserved. The order in which the best **body part** is selected is as follows:a.Enriched Text Formatb.**HTML** c.plain textTherefore the MIMEstream is regenerated every time a Message is retrieved. The alternative body parts are regenerated on demand as the Message is retrieved.In Exchange 2010, the MIME headers and body parts of messages are stored if the headers existed when the Message was delivered.

7 Change Tracking

This section identifies changes made to [MS-OXPOP3] protocol documentation between July 2009 and November 2009 releases. Changes are classed as major, minor, or editorial.

Major changes affect protocol interoperability or implementation. Examples of major changes are:

- A document revision that incorporates changes to interoperability requirements or functionality.
- An extensive rewrite, addition, or deletion of major portions of content.
- A protocol is deprecated.
- The removal of a document from the documentation set.
- Changes made for template compliance.

Minor changes do not affect protocol interoperability or implementation. Examples are updates to fix technical accuracy or ambiguity at the sentence, paragraph, or table level.

Editorial changes apply to grammatical, formatting, and style issues.

No changes means that the document is identical to its last release.

Major and minor changes can be described further using the following revision types:

- New content added.
- Content update.
- Content removed.
- New product behavior note added.
- Product behavior note updated.
- Product behavior note removed.
- New protocol syntax added.
- Protocol syntax updated.
- Protocol syntax removed.
- New content added due to protocol revision.
- Content updated due to protocol revision.
- Content removed due to protocol revision.
- New protocol syntax added due to protocol revision.
- Protocol syntax updated due to protocol revision.
- Protocol syntax removed due to protocol revision.
- New content added for template compliance.
- Content updated for template compliance.

- Content removed for template compliance.
- Obsolete document removed.

Editorial changes always have the revision type "Editorially updated."

Some important terms used in revision type descriptions are defined as follows:

Protocol syntax refers to data elements (such as packets, structures, enumerations, and methods) as well as interfaces.

Protocol revision refers to changes made to a protocol that affect the bits that are sent over the wire.

Changes are listed in the following table. If you need further information, please contact protocol@microsoft.com.

Section	Tracking number (if applicable) and description	Major change (Y or N)	Revision Type
3.2.5.2.3 NTLM Returns Status, Indicating User Name or Password Was Incorrect	48690 Modified the POP3_AUTH_NTLM_Failed_Response status to the correct identifier.	N	Protocol syntax updated.
7 Change Tracking	53353 Updated title.	N	Content update.

8 Index