

# [MS-SSEAN]: Simple Mail Transfer Protocol (SMTP) AUTH Extension for SPNEGO

---

## 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 [Open Specification Promise](#) or the [Community Promise](#). 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](mailto: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. For a list of Microsoft trademarks, visit [www.microsoft.com/trademarks](http://www.microsoft.com/trademarks).
- **Fictitious Names.** The example companies, organizations, products, domain names, email addresses, logos, people, places, and events depicted in this documentation are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

**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
10/25/2012	1.0	New	Release new document.
01/31/2013	1.0	No change	No changes to the meaning, language, or formatting of the technical content.
08/08/2013	2.0	Major	Significantly changed the technical content.

# Contents

<b>1</b>	<b>Introduction</b>	<b>4</b>
1.1	Glossary	4
1.2	References	4
1.2.1	Normative References	4
1.2.2	Informative References	5
1.3	Overview	5
1.4	Relationship to Other Protocols	6
1.5	Prerequisites/Preconditions	7
1.6	Applicability Statement	7
1.7	Versioning and Capability Negotiation	7
1.8	Vendor-Extensible Fields	7
1.9	Standards Assignments	7
<b>2</b>	<b>Messages</b>	<b>8</b>
2.1	Transport	8
2.2	Message Syntax	8
2.2.1	SASL Mechanism Name	8
<b>3</b>	<b>Protocol Details</b>	<b>9</b>
3.1	Client Details	9
3.1.1	Abstract Data Model	9
3.1.2	Timers	9
3.1.3	Initialization	9
3.1.4	Higher-Layer Triggered Events	9
3.1.5	Message Processing Events and Sequencing Rules	9
3.1.5.1	Initiating the "GSSAPI" Mechanism	9
3.1.6	Timer Events	9
3.1.7	Other Local Events	10
3.2	Server Details	10
3.2.1	Abstract Data Model	10
3.2.2	Timers	10
3.2.3	Initialization	10
3.2.4	Higher-Layer Triggered Events	10
3.2.5	Message Processing Events and Sequencing Rules	10
3.2.5.1	Receiving an AUTH GSSAPI Command	10
3.2.6	Timer Events	10
3.2.7	Other Local Events	10
<b>4</b>	<b>Protocol Examples</b>	<b>11</b>
4.1	Server Successfully Authenticating Client	11
<b>5</b>	<b>Security</b>	<b>13</b>
5.1	Security Considerations for Implementers	13
5.2	Index of Security Parameters	13
<b>6</b>	<b>Appendix A: Product Behavior</b>	<b>14</b>
<b>7</b>	<b>Change Tracking</b>	<b>15</b>
<b>8</b>	<b>Index</b>	<b>17</b>

# 1 Introduction

The Simple Mail Transfer Protocol (SMTP) AUTH Extension for SPNEGO enables **SMTP** clients and servers to negotiate a mutually supported authentication mechanism by using the **Simple and Protected GSS-API Negotiation Mechanism (SPNEGO)**.

Sections 1.8, 2, and 3 of this specification are normative and can contain the terms MAY, SHOULD, MUST, MUST NOT, and SHOULD NOT as defined in RFC 2119. Sections 1.5 and 1.9 are also normative but cannot contain those terms. All other sections and examples in this specification are informative.

## 1.1 Glossary

The following terms are defined in [\[MS-GLOS\]](#):

**base64**  
**Kerberos**  
**NT LAN Manager (NTLM) Authentication Protocol**  
**SASL**  
**Simple and Protected GSS-API Negotiation Mechanism (SPNEGO)**  
**Simple Authentication and Security Layer (SASL)**  
**Simple Mail Transfer Protocol (SMTP)**  
**SMTP**

The following terms are specific to this document:

**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

References to Microsoft Open Specifications documentation do not include a publishing year because links are to the latest version of the documents, which are updated frequently. References to other documents include a publishing year when one is available.

A reference marked "(Archived)" means that the reference document was either retired and is no longer being maintained or was replaced with a new document that provides current implementation details. We archive our documents online [\[Windows Protocol\]](#).

### 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](mailto: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-SPNG] Microsoft Corporation, "[Simple and Protected GSS-API Negotiation Mechanism \(SPNEGO\) Extension](#)".

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

[RFC2222] Myers, J., "Simple Authentication and Security Layer (SASL)", RFC 2222, October 1997, <http://www.ietf.org/rfc/rfc2222.txt>

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

[RFC4752] Melnikov, A., Ed., "The Kerberos V5 ("GSSAPI") Simple Authentication and Security Layer (SASL) Mechanism", RFC 4752, November 2006, <http://www.rfc-editor.org/rfc/rfc4752.txt>

## 1.2.2 Informative References

[IANA-SASL] IANA, "Simple Authentication and Security Layer (SASL) Mechanisms", September 2012, <http://www.iana.org/assignments/sasl-mechanisms>

[MS-GLOS] Microsoft Corporation, "[Windows Protocols Master Glossary](#)".

[MS-NLMP] Microsoft Corporation, "[NT LAN Manager \(NTLM\) Authentication Protocol](#)".

[RFC2743] Linn, J., "Generic Security Service Application Program Interface Version 2, Update 1", RFC 2743, January 2000, <http://www.ietf.org/rfc/rfc2743.txt>

[RFC4121] Zhu, L., Jaganathan, K., and Hartman, S., "The Kerberos Version 5 Generic Security Service Application Program Interface (GSS-API) Mechanism: Version 2", RFC 4121, July 2005, <http://www.ietf.org/rfc/rfc4121.txt>

[RFC4178] Zhu, L., Leach, P., Jaganathan, K., and Ingersoll, W., "The Simple and Protected Generic Security Service Application Program Interface (GSS-API) Negotiation Mechanism", RFC 4178, October 2005, <http://www.ietf.org/rfc/rfc4178.txt>

[RFC5321] Klensin, J., "Simple Mail Transfer Protocol", RFC 5321, October 2008, <http://rfc-editor.org/rfc/rfc5321.txt>

## 1.3 Overview

The **Simple Mail Transfer Protocol (SMTP)** [\[RFC5321\]](#) is commonly used to transfer email between computers, and securing such transfers requires authentication. The ability for a client and server to negotiate authentication mechanisms for SMTP is specified in [\[RFC2554\]](#), which allows the use of any mechanism that is defined as a **Simple Authentication and Security Layer (SASL)** mechanism [\[RFC2222\]](#). A server advertises a list of supported **SASL** mechanism names, from which the client selects one.

Separately, the Generic Security Services API (GSS-API) [\[RFC2743\]](#) defines a set of abstract functions that allow security mechanisms to be defined in a way that can be used by any protocol that is defined to use GSS-API. Examples of GSS-API security mechanisms include **Kerberos** [\[RFC4121\]](#) and the **NT LAN Manager (NTLM) Authentication Protocol** [\[MS-NLMP\]](#).

SMTP requires authentication mechanisms to be defined as SASL mechanisms rather than GSS-API mechanisms. In an attempt to support the use of GSS-API mechanisms by protocols that are defined to use SASL mechanisms, [\[RFC2222\]](#) section 7.2 specified how to map all GSS-API mechanisms to SASL mechanisms. However, it specified a single SASL mechanism name ("GSSAPI") to be used for all GSS-API mechanisms, which made it impossible to determine which GSS-API mechanism to use when the "GSSAPI" SASL mechanism name is negotiated. As a result, protocols such as SMTP that negotiate SASL mechanism names were unable to agree on which GSS-API mechanism was referred to by "GSSAPI". [\[RFC2222\]](#) is rendered obsolete by [\[RFC4752\]](#), which specifies "GSSAPI" as specifically referring to Kerberos.

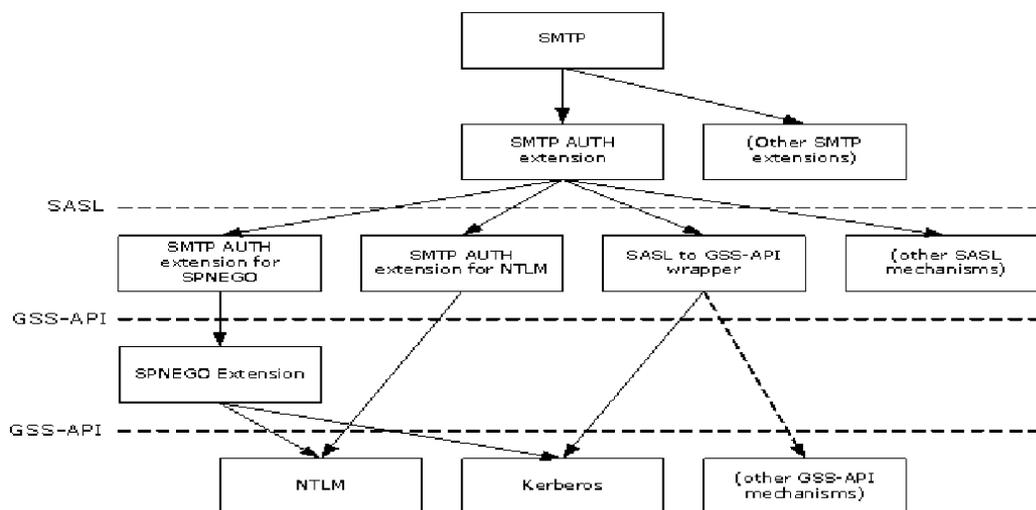
The Simple and Protected Generic Security Service Application Program Interface (GSS-API) Negotiation Mechanism (SPNEGO) [RFC4178] is defined for use by protocols that use GSS-API to negotiate security mechanisms. This allows such protocols to negotiate the use of GSS-API mechanisms such as Kerberos and NTLM. It does so by exposing itself as a single GSS-API mechanism that performs the negotiation and is layered atop other GSS-API mechanisms. It does not, however, define any SASL mechanism name for use by protocols defined to use SASL mechanisms rather than GSS-API mechanisms.

Using the SMTP AUTH Extension for SPNEGO, SMTP clients and servers map the SASL mechanism name "GSSAPI" to the SPNEGO Extension GSS-API mechanism specified in [MS-SPNG], rather than to Kerberos. The SMTP AUTH extension for SPNEGO conforms to the problematic [RFC2222] section 7.2, but it resolves the issue of the ambiguous SASL mechanism name "GSSAPI" differently from the later [RFC4752].

In summary, the SMTP client and server use the SMTP AUTH Extension for SPNEGO to negotiate the use of the SPNEGO Extension by using the "GSSAPI" SASL mechanism name and then use SPNEGO to further negotiate a specific GSS-API mechanism underneath, which does not need to have its own SASL mechanism name.

### 1.4 Relationship to Other Protocols

This extension uses the SPNEGO Extension specified in [MS-SPNG], which differs slightly from SPNEGO [RFC4178]. The following diagram illustrates the relationships between its related protocols and mechanisms.



**Figure 1: Relationship to other protocols**

The SMTP **AUTH** extension specified in [RFC2554] is defined to use SASL mechanisms. The SMTP AUTH Extension for SPNEGO is one SASL mechanism. The SASL to GSS-API wrapper defined in [RFC2222] section 7.2 is another SASL mechanism; however, it does not define a SASL mechanism name for any particular GSS-API mechanism (thus the diagram contains a dotted line from "SASL to GSS-API wrapper" to "(other GSS-API mechanisms)"). To use a particular GSS-API mechanism in SMTP requires that the mechanism have a SASL mechanism name defined for it. [RFC4752] specifies the SASL to GSS-API wrapper over Kerberos. The SMTP **AUTH** extension for NTLM specified in [MS-SMTP] is another SASL mechanism.

The SPNEGO Extension specified in [MS-SPNG] is both a GSS-API mechanism (as used in this document) as well as a protocol that uses other GSS-API mechanisms, including NTLM and Kerberos.

Since both the SMTP AUTH Extension for SPNEGO and the SASL to GSS-API wrapper over Kerberos defined in [RFC4752] use the same SASL mechanism name, they cannot be used in the same environment. However, the SASL mechanism name for the SMTP **AUTH** extension for NTLM, as well as other SASL mechanisms, are different and therefore can be used in the same environment.

## 1.5 Prerequisites/Preconditions

The SMTP AUTH Extension for SPNEGO is only invoked after an SMTP client with this extension has initiated an SMTP connection and is using the SMTP **AUTH** extension [RFC2554].

It is assumed that the client knows that the server supports the SMTP AUTH Extension for SPNEGO (as opposed to the SASL to GSS-API wrapper specified in [RFC4752]) before the client determines to use this extension.

## 1.6 Applicability Statement

The SMTP AUTH Extension for SPNEGO is used when implementing an SMTP client and SMTP server that negotiate a mutually supported authentication mechanism using SPNEGO.

Since this extension and the SASL to GSS-API wrapper defined in [RFC4752] use the same SASL mechanism name, this extension is not applicable to clients in environments with servers that implement the Kerberos V5 ("GSSAPI") SASL Mechanism specified in [RFC4752]. A client in such an environment will use the SPNEGO Extension specified in [MS-SPNG] when the server advertises "GSSAPI", and authentication will fail.

However, the SMTP AUTH Extension for SPNEGO can be used by servers that support Kerberos with clients that use either this extension or the Kerberos V5 ("GSSAPI") SASL Mechanism. When the server advertises "GSSAPI", the former will use the SPNEGO Extension and the latter will use Kerberos directly, and both can succeed.

## 1.7 Versioning and Capability Negotiation

The SMTP AUTH Extension for SPNEGO introduces no new versioning mechanisms.

Negotiation of SMTP options is specified in [RFC5321] sections 2.2 and 3.2. Negotiation of authentication mechanisms using SASL mechanism names is specified by [RFC2554]. Negotiation of GSS-API mechanisms is specified by [MS-SPNG].

## 1.8 Vendor-Extensible Fields

None.

## 1.9 Standards Assignments

This extension reuses the following assignment.

Parameter	Value	Reference
SASL mechanism name	GSSAPI	[IANA-SASL]

This use is consistent with the problematic specification in [RFC2222] section 7.2.

## 2 Messages

### 2.1 Transport

The SMTP AUTH Extension for SPNEGO does not establish transport connections. Instead, its messages are encapsulated in SMTP **AUTH** commands and responses as specified in [\[RFC2554\]](#).

### 2.2 Message Syntax

[\[RFC2554\]](#) section 4 specifies messages that are usable by any SASL mechanism that defines a mechanism name, an optional initial response to be encoded by using **base64**, and a reply also to be encoded by using base64.

[\[RFC2222\]](#) section 7.2 specifies the mapping of the initial response and the reply to specific GSS-API functions. In the SMTP AUTH Extension for SPNEGO, the GSS-API calls are directed to the SPNEGO Extension, so the calls use the syntax specified in [\[MS-SPNG\]](#).

#### 2.2.1 SASL Mechanism Name

The SASL mechanism name is defined as "GSSAPI".

## 3 Protocol Details

### 3.1 Client Details

#### 3.1.1 Abstract Data Model

This section describes a conceptual model of possible data organization that an implementation maintains to participate in this protocol. The described organization is provided to facilitate the explanation of how the protocol behaves. This document does not mandate that implementations adhere to this model as long as their external behavior is consistent with that described in this document.

This extension specifies no additional state beyond what is already required by [\[RFC2554\]](#) and [\[MS-SPNG\]](#). The abstract state specified by [\[RFC2554\]](#) and used by this protocol is as follows.

For each SMTP connection:

**Authentication Mechanism:** The choice of authentication mechanism being used on that connection.

#### 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

##### 3.1.5.1 Initiating the "GSSAPI" Mechanism

When the SMTP AUTH extension [\[RFC2554\]](#) determines to use the "GSSAPI" SASL mechanism, the client MUST then associate the SPNEGO Extension [\[MS-SPNG\]](#) GSS-API mechanism with the connection. The way the client determines which of the SASL mechanisms to use is implementation-specific, but the client SHOULD use the SPNEGO Extension only if "GSSAPI" is returned in the list of SASL mechanisms in the EHLO exchange specified in [\[RFC2554\]](#) and SHOULD prefer the SPNEGO Extension over any other such SASL mechanisms supported.

As a result of the client associating the SPNEGO Extension with the connection, the rest of the exchange will proceed as specified in [\[RFC2554\]](#) for **AUTH** command processing, in [\[RFC2222\]](#) for mapping the SASL mechanism to specific GSS-API function calls, and in [\[MS-SPNG\]](#) for the behavior of those GSS-API function calls.

#### 3.1.6 Timer Events

None.

### 3.1.7 Other Local Events

None.

## 3.2 Server Details

### 3.2.1 Abstract Data Model

This section describes a conceptual model of possible data organization that an implementation maintains to participate in this protocol. The described organization is provided to facilitate the explanation of how the protocol behaves. This document does not mandate that implementations adhere to this model as long as their external behavior is consistent with that described in this document.

This extension specifies no additional state beyond what is already required by [\[RFC2554\]](#) and [\[MS-SPNG\]](#). The abstract state specified by [\[RFC2554\]](#) and used by this protocol is as follows.

**List of SASL Mechanisms:** The list of SASL mechanism names to be returned in an EHLO response.

For each SMTP connection:

**Authentication Mechanism:** The choice of authentication mechanism being used on that connection.

### 3.2.2 Timers

None.

### 3.2.3 Initialization

When the server is initialized, it MUST place "GSSAPI" in its **List of SASL Mechanisms** abstract data model element.

### 3.2.4 Higher-Layer Triggered Events

None.

### 3.2.5 Message Processing Events and Sequencing Rules

#### 3.2.5.1 Receiving an AUTH GSSAPI Command

When the server receives an **AUTH GSSAPI** command from a client over a given SMTP connection, the server MUST set that connection's **Authentication Mechanism** to the SPNEGO Extension [\[MS-SPNG\]](#) and thus route all subsequent GSS-API function calls for that connection to the SPNEGO Extension.

### 3.2.6 Timer Events

None.

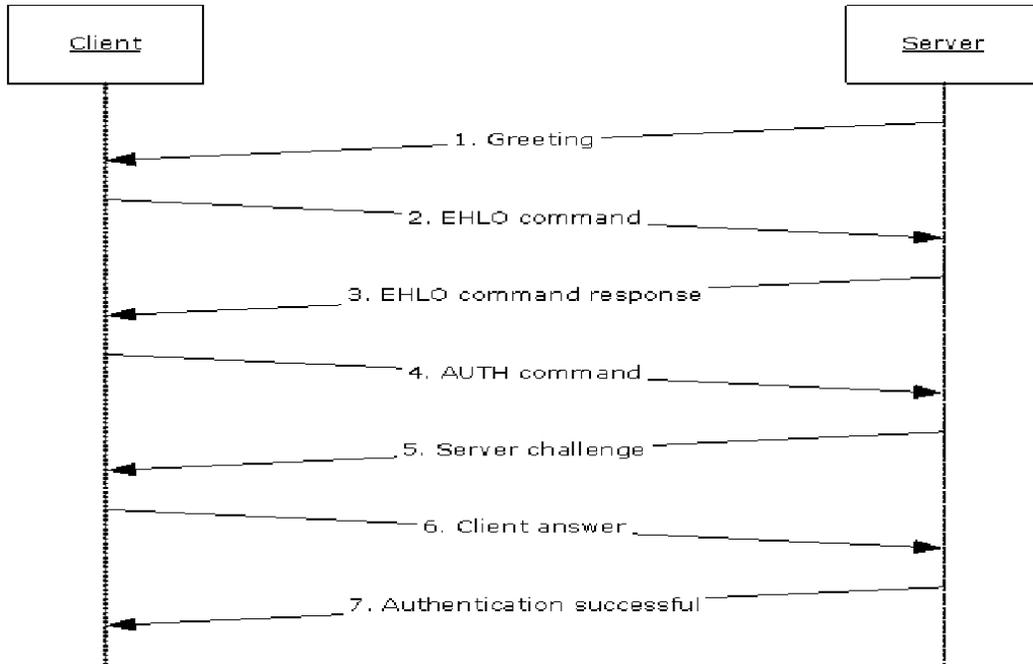
### 3.2.7 Other Local Events

None.

## 4 Protocol Examples

### 4.1 Server Successfully Authenticating Client

In this example a server successfully authenticates a client by using the SMTP AUTH Extension for SPNEGO. The following sequence diagram shows the flow of command requests and responses in a successful authentication negotiation.



**Figure 2: Server successfully authenticates client**

1. When the client connects to the SMTP port on the server, the server responds with a greeting.

```
220 server.contoso.com Authenticated Receive Connector
```

2. The client sends an **EHLO** command as described in [\[RFC5321\]](#).

```
EHLO client.contoso.com
```

3. The server sends a response that includes an **AUTH** keyword containing a list of SASL mechanism names that it supports, including "GSSAPI".

```
250-server-contoso.com Hello [203.0.113.1]
250-AUTH GSSAPI NTLM
250 OK
```

4. The client examines the SASL mechanism names after the **AUTH** keyword and selects one that it supports. In this example, the client selects "GSSAPI" and begins to use the SPNEGO Extension. The SMTP AUTH Extension for SPNEGO then uses GSS-API to ask the SPNEGO Extension for an

initial response. As specified in [MS-SPNG], this is a **NegTokenInit** message that contains the client's list of requested GSS-API mechanisms, which the SMTP **AUTH** extension specified in [RFC2554] then base64-encodes.

```
AUTH GSSAPI <base64-encoded NegTokenInit>
```

5. The server determines that the client is requesting "GSSAPI" and begins to use the SPNEGO Extension. The SMTP AUTH Extension for SPNEGO then base64-decodes the initial response and uses GSS-API to supply the **NegTokenInit** message and to request a reply from the SPNEGO Extension. As specified in [MS-SPNG], this will be a **NegTokenResp** message, which in this example includes a server challenge for the selected GSS-API mechanism. The SMTP **AUTH** extension base64-encodes the message and returns it in a 334 reply.

```
334 <base64-encoded NegTokenResp>
```

6. The client base64-decodes the reply and passes the result to the SPNEGO Extension. Depending on the GSS-API mechanism used by the SPNEGO Extension, there might or might not be another response to send to the server. In this example, the SPNEGO Extension returns another response to send, which the SMTP **AUTH** extension again base64-encodes.

```
<base64-encoded client answer>
```

7. The server base64-decodes the response and passes the result to the SPNEGO Extension. Depending on the GSS-API mechanism used by the SPNEGO Extension, there might or might not be another challenge reply to send to the client. In this example, the SPNEGO Extension determines that the authentication succeeded, so the SMTP **AUTH** extension reports success back to the client.

```
235 2.7.0 Authentication successful
```

## 5 Security

### 5.1 Security Considerations for Implementers

As with all other SMTP authentication mechanisms, the security considerations described in [\[RFC2554\]](#) section 9 and [\[RFC2222\]](#) section 9 apply. Since this extension uses the SPNEGO Extension, the security considerations described in [\[MS-SPNG\]](#) section 5.1 also apply.

### 5.2 Index of Security Parameters

None.

## 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 .NET Framework 2.0
- Microsoft .NET Framework 3.5
- Microsoft .NET Framework 4.0
- Microsoft .NET Framework 4.5
- Microsoft Exchange Server 2003
- Microsoft Exchange Server 2007
- Microsoft Exchange Server 2010
- Microsoft Exchange Server 2013
- Microsoft Office Outlook 2003
- Microsoft Office Outlook 2007
- Microsoft Outlook 2010
- Microsoft Outlook 2013
- Windows 2000 Professional operating system
- Windows 2000 Server operating system
- Windows XP operating system
- Windows Server 2003 operating system
- Windows Server 2008 operating system
- Windows 8 operating system
- Windows Server 2012 operating system
- Windows Server 2012 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.

## 7 Change Tracking

This section identifies changes that were made to the [MS-SSEAN] protocol document between the January 2013 and August 2013 releases. Changes are classified as New, Major, Minor, Editorial, or No change.

The revision class **New** means that a new document is being released.

The revision class **Major** means that the technical content in the document was significantly revised. 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.
- The removal of a document from the documentation set.
- Changes made for template compliance.

The revision class **Minor** means that the meaning of the technical content was clarified. Minor changes do not affect protocol interoperability or implementation. Examples of minor changes are updates to clarify ambiguity at the sentence, paragraph, or table level.

The revision class **Editorial** means that the language and formatting in the technical content was changed. Editorial changes apply to grammatical, formatting, and style issues.

The revision class **No change** means that no new technical or language changes were introduced. The technical content of the document is identical to the last released version, but minor editorial and formatting changes, as well as updates to the header and footer information, and to the revision summary, may have been made.

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

- New content added.
- Content updated.
- 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 are always classified with the change type **Editorially updated**.

Some important terms used in the change 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.

The changes made to this document are listed in the following table. For more information, please contact [protocol@microsoft.com](mailto:protocol@microsoft.com).

<b>Section</b>	<b>Tracking number (if applicable) and description</b>	<b>Major change (Y or N)</b>	<b>Change type</b>
<a href="#">6 Appendix A: Product Behavior</a>	Added Windows Server "vNext" to the list of applicable product versions.	Y	Content updated.

## 8 Index

### A

Abstract data model  
[client](#) 9  
[server](#) 10  
[Applicability](#) 7

### C

[Capability negotiation](#) 7  
[Change tracking](#) 15  
Client  
[abstract data model](#) 9  
[higher-layer triggered events](#) 9  
[initialization](#) 9  
[other local events](#) 10  
[timer events](#) 9  
[timers](#) 9

### D

Data model - abstract  
[client](#) 9  
[server](#) 10

### E

Examples  
[server successfully authenticating client](#) 11

### F

[Fields - vendor-extensible](#) 7

### G

[Glossary](#) 4

### H

Higher-layer triggered events  
[client](#) 9  
[server](#) 10

### I

[Implementer - security considerations](#) 13  
[Index of security parameters](#) 13  
[Informative references](#) 5  
Initialization  
[client](#) 9  
[server](#) 10  
[Introduction](#) 4

### M

[Message syntax](#) 8  
Messages

[SASL Mechanism Name message](#) 8  
[syntax](#) 8  
[transport](#) 8

### N

[Normative references](#) 4

### O

Other local events  
[client](#) 10  
[server](#) 10  
[Overview \(synopsis\)](#) 5

### P

[Parameters - security index](#) 13  
[Preconditions](#) 7  
[Prerequisites](#) 7  
[Product behavior](#) 14

### R

References  
[informative](#) 5  
[normative](#) 4  
[Relationship to other protocols](#) 6

### S

[SASL Mechanism Name message](#) 8  
Security  
[implementer considerations](#) 13  
[parameter index](#) 13  
Server  
[abstract data model](#) 10  
[higher-layer triggered events](#) 10  
[initialization](#) 10  
[other local events](#) 10  
[timer events](#) 10  
[timers](#) 10  
[Server successfully authenticating client example](#) 11  
[Standards assignments](#) 7

### T

Timer events  
[client](#) 9  
[server](#) 10  
Timers  
[client](#) 9  
[server](#) 10  
[Tracking changes](#) 15  
[Transport](#) 8  
Triggered events - higher-layer  
[client](#) 9

[server](#) 10

## **V**

[Vendor-extensible fields](#) 7

[Versioning](#) 7