[MS-APDS]:
Authentication Protocol Domain Support Specification

Revision Summary

Date	Revision History	Revision Class	Comments
02/22/2007	0.1		MCPP Milestone 3 Initial Availability
06/01/2007	2.0	Major	Updated and revised the technical content.
07/03/2007	3.0	Major	Added new protocol.
07/20/2007	3.0.1	Editorial	Revised and edited the technical content.
08/10/2007	3.0.2	Editorial	Revised and edited the technical content.

Date	Revision History	Revision Class	Comments
09/28/2007	4.0	Major	Updated and revised the technical content.
10/23/2007	4.0.1	Editorial	Revised and edited the technical content.
11/30/2007	4.0.2	Editorial	Revised and edited the technical content.
01/25/2008	4.0.3	Editorial	Revised and edited the technical content.

Table of Contents

1 Introduction

This document describes the communication between a server and a **domain controller** that uses Netlogon interfaces to complete an authentication sequence. This process is called Authentication Protocol Domain Support.

Windows supports a number of authentication protocols, specifically **NT LAN Manager (NTLM)**, **Kerberos**, **Secure Sockets Layer (SSL)**/**Transport Layer Security (TLS)**, and **digest authentication**. Authentication Protocol Domain Support is used by NT LAN Manager (NTLM) and the **digest validation** protocol to perform validation of the user's credentials at the domain controller (DC). The Kerberos protocol uses Authentication Protocol Domain Support to perform the required communication for **privilege attribute certificate (PAC)** validation.

With the exception of Kerberos (which also relies on a mutually trusted third-party called **Key Distribution Center (KDC)**, as specified in [MS-KILE]), all of these protocols can be supported by any server, relying only on a local user account database. Therefore, specifications for these protocols can stand entirely on their own. However, in a **domain** context, when the server is a member of a domain and relies on the **domain account** database, the domain controller contributes to the authentication and authorization processes.

Domain members use the Netlogon Remote Protocol (as specified in [MS-NRPC]) to communicate with the domain controller for purposes of authentication and authorization.

The implementations of these authentication protocols use a variety of methods to communicate with the domain controller in the course of their executions. These methods, collectively referred to as Authentication Protocol Domain Support, are specified in this document.

1.1 Glossary

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

Active Directory (AD)
Certificate
Directory
Domain
Domain Account
Domain Controller (DC)
Kerberos
Key Distribution Center (KDC)
Machine Account
Network Logon
NT LAN Manager (NTLM)
NTOWF
Principal
Privilege Attribute Certificate (PAC)
Remote Procedure Call (RPC)
RPC Transport
Secure Sockets Layer (SSL)
Service
Transport Layer Security (TLS)
User Principal Name (UPN)

The following terms are specific to this document:

Digest Authentication: A protocol that uses a challenge-response mechanism for authentication in which clients are able to verify their identities without sending an in-the-clear password to the server. For more information, see [RFC2617] and [RFC2831].

Digest Validation: A protocol to verify the **Digest Authentication** challenge-response from a client to a server for a specified **domain** account.

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.

[ISO-8859-1] International Organization for Standardization, "Information Technology -- 8-Bit Single-Byte Coded Graphic Character Sets -- Part 1: Latin Alphabet No. 1", ISO/IEC 8859-1, 1998, http://www.iso.org/iso/en/CatalogueDetailPage.CatalogueDetail?CSNUMBER=28245&ICS1=35&ICS2=40&ICS3=

Note There is a charge to download the specification.

[MS-ADTS] Microsoft Corporation, "Active Directory Technical Specification", June 2007.

[MS-DPSP] Microsoft Corporation, "Digest Protocol Extensions", January 2007.

[MS-DTYP] Microsoft Corporation, "Windows Data Types", January 2007.

[MS-EERR] Microsoft Corporation, "ExtendedError Remote Data Structure", January 2007.

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

[MS-KILE] Microsoft Corporation, "Kerberos Protocol Extensions", January 2007.

[MS-LSAD] Microsoft Corporation, "Local Security Authority (Domain Policy) Remote Protocol Specification", June 2007.

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

[MS-NRPC] Microsoft Corporation, "Netlogon Remote Protocol Specification", March 2007.

[MS-PAC] Microsoft Corporation, "Privilege Attribute Certificate Data Structure", January 2007.

[MS-RCMP] Microsoft Corporation, "Remote Certificate Mapping Protocol Specification", September 2007.

[MS-RPCE] Microsoft Corporation, "Remote Procedure Call Protocol Extensions", January 2007.

[MS-SECO] Microsoft Corporation, "Windows Security Overview", January 2007.

[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

[RFC2617] Franks, J., Hallam-Baker, P., Hostetler, J., Lawrence, S., Leach, P., Luotonen, A., and Stewart, L., "HTTP Authentication: Basic and Digest Access Authentication", RFC 2617, June 1999, http://www.ietf.org/rfc/rfc2617.txt

[RFC2831] Leach, P. and Newman, C., "Using Digest Authentication as a SASL Mechanism", RFC 2831, May 2000, http://www.ietf.org/rfc/rfc2831.txt

[UNICODE] The Unicode Consortium, "Unicode Home Page", 2006, http://www.unicode.org/

[X509] ITU-T, "Information Technology - Open Systems Interconnection - The Directory: Public-Key and Attribute Certificate Frameworks", Recommendation X.509, August 2005, http://www.itu.int/rec/T-REC-X.509/en

Note There is a charge to download the specification.

1.2.2 Informative References

[MSDN-0SUBAUTHROUTINE9] Microsoft Corporation, "Msv1_0SubAuthenticationRoutine (9 Parameters) function [Security]", http://msdn2.microsoft.com/en-us/library/aa378752.aspx

[RFC2069] Franks, J., et al., "An Extension to HTTP: Digest Access Authentication", RFC 2069, January 1997, http://www.ietf.org/rfc/rfc2069.txt

[RFC1994] Simpson, W. "PPP Challenge Handshake Authentication Protocol (CHAP)", RFC 1994, August 1996, http://www.ietf.org/rfc/rfc1994.txt

1.3 Protocol Overview (Synopsis)

The authentication protocols supported by Windows—specifically NT LAN Manager (NTLM), Kerberos, Secure Sockets Layer (SSL)/Transport Layer Security (TLS), and digest authentication—are in turn used by a variety of higher-layer protocols to provide security **services**.

The Authentication Protocol Domain Support Protocol specifies the communication between the server and the domain controller for each of the protocols. Each of the protocols has a specific exchange with the domain controller (DC) to authenticate the client (that is, NTLM and digest), and obtain authorization information, such as group memberships (that is, NTLM, digest, and SSL/TLS), or verify the authorization information (Kerberos). All of these back-end, server-to-server protocols in turn use the Netlogon Remote Protocol (as specified in [MS-NRPC]) for their transport to the DC. Specifically, the protocols behave as follows:

- The NT LAN Manager (NTLM) Authentication Protocol (as specified in [MS-NLMP]) uses the Netlogon Remote Protocol (as specified in [MS-NRPC]) to communicate with the DC to complete the authentication of a domain account during an interactive logon or **network logon**. As user account information is maintained by the DC, only the DC can validate a user's credentials and complete the authentication sequence. The server then uses the authorization information returned by the DC to make authorization decisions.
- Kerberos uses Netlogon generic pass-through (as specified in [MS-NRPC] section 3.2) to validate the privilege attribute certificate (PAC) (as specified in [MS-PAC]) that it receives in the ticket from the client. Because PAC information can be altered by the server, the operating system may contact the DC to validate the PAC and ensure its integrity.
- The Digest Protocol Extensions (as specified in [MS-DPSP]) is used by deployments in which users are authenticated based on user name and password by using the digest authentication

mechanism. The digest authentication mechanism itself defines how the client authenticates the user to the server (by proving knowledge of the password), and optionally provides integrity and confidentiality of subsequent messages exchanged between the client and the server. Digest validation is performed between the server and the DC during the initial client/server digest–based authentication as follows:

1. The server (which does not have access to the user's password) sends a digest validation request (as specified in section 2.2.3.1) message to the domain controller by using the generic pass-through capability of the Netlogon Remote Protocol, as specified in [MS-NRPC] section 3.2.

2. The DC looks up the user's password and uses it to verify the validity of the digest input (originally generated by the digest client by using the user's password, as specified in [RFC2617] and [RFC2831]).

3. On successful validation, the domain controller returns the PAC in the DIGEST_VALIDATION_RESP message. The PAC represents the user's identity and group memberships, suitable for making authorization decisions.

- SSL/TLS and other protocols that authenticate users via the X.509 certificates (as specified in [X509]) can use the Remote Certificate Mapping Protocol (as specified in [MS-RCMP]), which relies on the generic pass-through capability of Netlogon to retrieve authorization information associated with users. This behavior is specified in [MS-RCMP].

1.4 Relationship to Other Protocols

Each of the following protocols relies on the Netlogon Remote Protocol (as specified in [MS-NRPC]) to complete the authentication sequence and retrieve or validate authorization information associated with a user. All higher-layer protocols that use one of the protocols specified in this document (for example, NT LAN Manager (NTLM) Authentication Protocol as specified in [MS-NLMP] and Kerberos Protocol Extensions as specified in [MS-KILE]) leverage the functionality specified here when in a domain environment.

1.4.1 NTLM Logon

NTLM authentication (as specified in [MS-NLMP]) uses the Netlogon **NetrLogonSamLogonEx** method (as specified in [MS-NRPC] section **3.5.4.4.1**) to authenticate the user in the domain with the domain controller during an interactive logon or a network logon.

1.4.2 Kerberos PAC Validation

Kerberos authentication (as specified in [MS-KILE]) uses the Netlogon generic pass-through (as specified in [MS-NRPC]) to validate the privilege attribute certificate (PAC) with the domain controller, when required.

1.4.3 Digest Validation Protocol

The digest validation defined in this document relies on the generic pass-through capability of the Netlogon Remote Protocol (as specified in [MS-NRPC]) as a transport for the DIGEST_VALIDATION_REQ (section 2.2.3.1) and DIGEST_VALIDATION_RESP (section 2.2.3.2) messages.

1.5 Prerequisites/Preconditions

The Netlogon Remote Protocol, which each of the following protocols relies on, assumes a secure connection is established with the domain controller (DC).<1>

1.5.1 NTLM Logon

NTLM interactive logon and NTLM network logon have all of the prerequisites and preconditions that are defined in the NT LAN Manager (NTLM) Authentication Protocol, as specified in [MS-NLMP].

1.5.2 Kerberos PAC Validation

Kerberos PAC validation assumes that the application server has a PAC that has been received in a ticket from the client. Servers MAY require PAC validation.<2>

1.5.3 Digest Validation Protocol

The digest validation protocol assumes the following:

- The digest validation server has access to the user's password.<3>
- The digest validation client possesses the digest-response message (as specified in [RFC2617] section 3.2.2) and the initial digest-challenge message (as specified in [RFC2617] section 3.2.1) used in the digest authentication protocol.

1.6 Applicability Statement

All protocol support for domains requires a domain authority to process the requests. These protocols are not applicable to any stand-alone machine that is not associated with a domain. Each protocol has additional applicability constraints.

1.6.1 NTLM Logon

The applicability of NTLM (as specified in [MS-NLMP] section 1.6) also applies when the server performing the NTLM authentication is a member of a domain. Both NTLM for interactive logon and NTLM for network logon can be performed by using a domain controller (DC).<4>

1.6.2 Kerberos PAC Validation

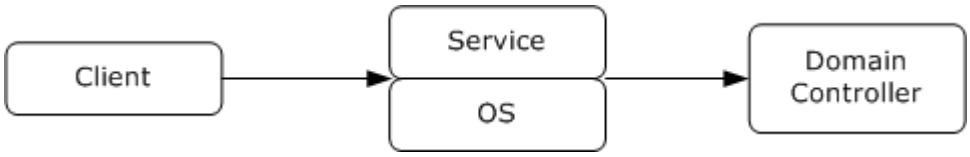


Figure 1: Kerberos PAC validation

When Kerberos PAC validation occurs, the client sends the privilege attribute certificate (PAC) to the service as a part of the Kerberos Protocol Extensions, as specified in [MS-KILE]. The operating system on which the service runs validates the PAC to prevent PAC tampering by the service. PAC tampering can result in inappropriate elevation of privileges. PAC validation is applicable for Kerberos applications that process and interpret the Windows PAC and present that authorization data to additional services. It is optional for a self-contained application because the security threat that the protocol addresses is not relevant for self-contained applications.<5>

1.6.3 Digest Validation Protocol

The digest validation protocol is appropriate for servers that are implementing digest authentication and are acting as members in an **Active Directory**-compatible domain.<6>

1.7 Versioning and Capability Negotiation

1.7.1 NTLM Logon

NTLM interactive logon and network logon do not have any versioning or capability negotiation.

1.7.2 PAC Validation

PAC validation does not have any versioning or capability negotiation.

1.7.3 Digest Validation Protocol

The DIGEST_VALIDATION_REQ and DIGEST_VALIDATION_RESP messages have a dedicated version number field. This document defines version 1 of the Digest Protocol Extensions. The Digest Protocol Extensions does not support any capability negotiation.

1.8 Vendor-Extensible Fields

1.8.1 NTLM Logon

NTLM interactive logon and network logon do not have any vendor-extensible fields.

1.8.2 PAC Validation

PAC validation does not have any vendor-extensible fields.

1.8.3 Digest Validation Protocol

The digest validation protocol does not have any vendor-extensible fields. Note that the digest validation protocol has reserved fields for future use, but these fields are not intended to carry opaque/vendor-defined data.

1.9 Standards Assignments

There are no standards assignments used in Authentication Protocol Domain Support.

2 Messages

2.1 Transport

Domain support for Windowsimplementations of authentication protocols SHOULD use Netlogon **remote procedure call (RPC)** messages in the logon interface. The Netlogon **RPC transport** is specified in [MS-NRPC].<7>

2.2 Message Syntax

For domain support, authentication protocols MUST use the **NetrLogonSamLogonEx** method (as specified in [MS-NRPC] section **3.5.4.4.1**) with parameters determined by the authentication protocol being used.

NTLM interactive and network logon use parameters that are as specified in [MS-RPCE]Appendix A when calling the **NetrLogonSamLogonEx** method. Domain controller Kerberos PAC validation and digest messages MUST be encoded as opaque blobs and transported by the generic pass-through capability of Netlogon, as specified in [MS-NRPC].

All message fields, including bitmaps, are in the following sections in little-endian format. These data structures MUST be built as if they are on a little-endian machine before transmission. On reception, the messages MUST be interpreted as little-endian and transformed into the native endianness of the implementation.

The following table shows a few of the main status codes returned by these protocols. For a complete list of status codes, see [MS-ERREF].

Symbolic name	Value	Meaning
STATUS_SUCCESS	0x00000000	Requested operation succeeded.
STATUS_LOGON_FAILURE	0xC000006D	Authentication failed, and the logon attempt should be denied by the server.
STATUS_NO_SUCH_USER	0xC0000064	Specified account does not exist.
STATUS_NO_LOGON_SERVERS	0xC000005E	None of the domain controllers are reachable to service the request.

2.2.1 NTLM Logon Message Syntax

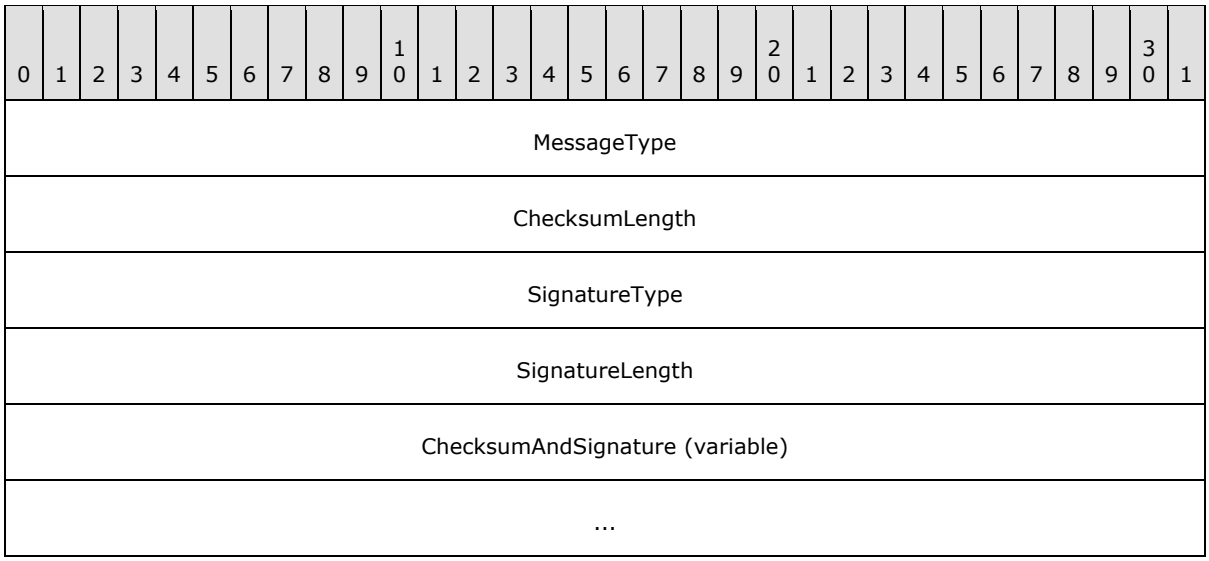
The specific message syntax for NTLM interactive logon or network logon is part of the Netlogon Remote Protocol, as specified in [MS-NRPC]. The domain support for NTLM logon is invoked by calling the **NetrLogonSamLogonEx** method (as specified in [MS-NRPC] section **3.5.4.4.1**).

2.2.2 PAC Validation Message Syntax

The privilege attribute certificate (PAC) validation request message, KERB_VERIFY_PAC_REQUEST (as specified in section 2.2.2.1), MUST be encoded as a contiguous buffer. The encoded data SHOULD be sent by using the generic pass-through mechanism, as specified in [MS-NRPC] section 3.2.4.1. The encoding of the KERB_VERIFY_PAC_REQUEST is as specified in section 2.2.2.1.<8>

2.2.2.1 KERB_VERIFY_PAC_REQUEST Message

The following diagram shows the format of the message used for PAC validation.



MessageType (4 bytes): An unsigned 32-bit value describing the message type. This member MUST be set to 0x00000003.

ChecksumLength (4 bytes): An unsigned 32-bit value that MUST contain the signature length of the signature value in the Server Checksum privilege attribute certificate (PAC) buffer, as specified in [MS-PAC] section 2.8.

SignatureType (4 bytes): An unsigned 32-bit value that MUST contain the signature type of the signature value in the Key Distribution Center (KDC) (Privilege Server) Checksum PAC buffer.

SignatureLength (4 bytes): An unsigned 32-bit value that MUST contain the signature length of the signature value in the KDC (Privilege Server) Checksum PAC buffer.

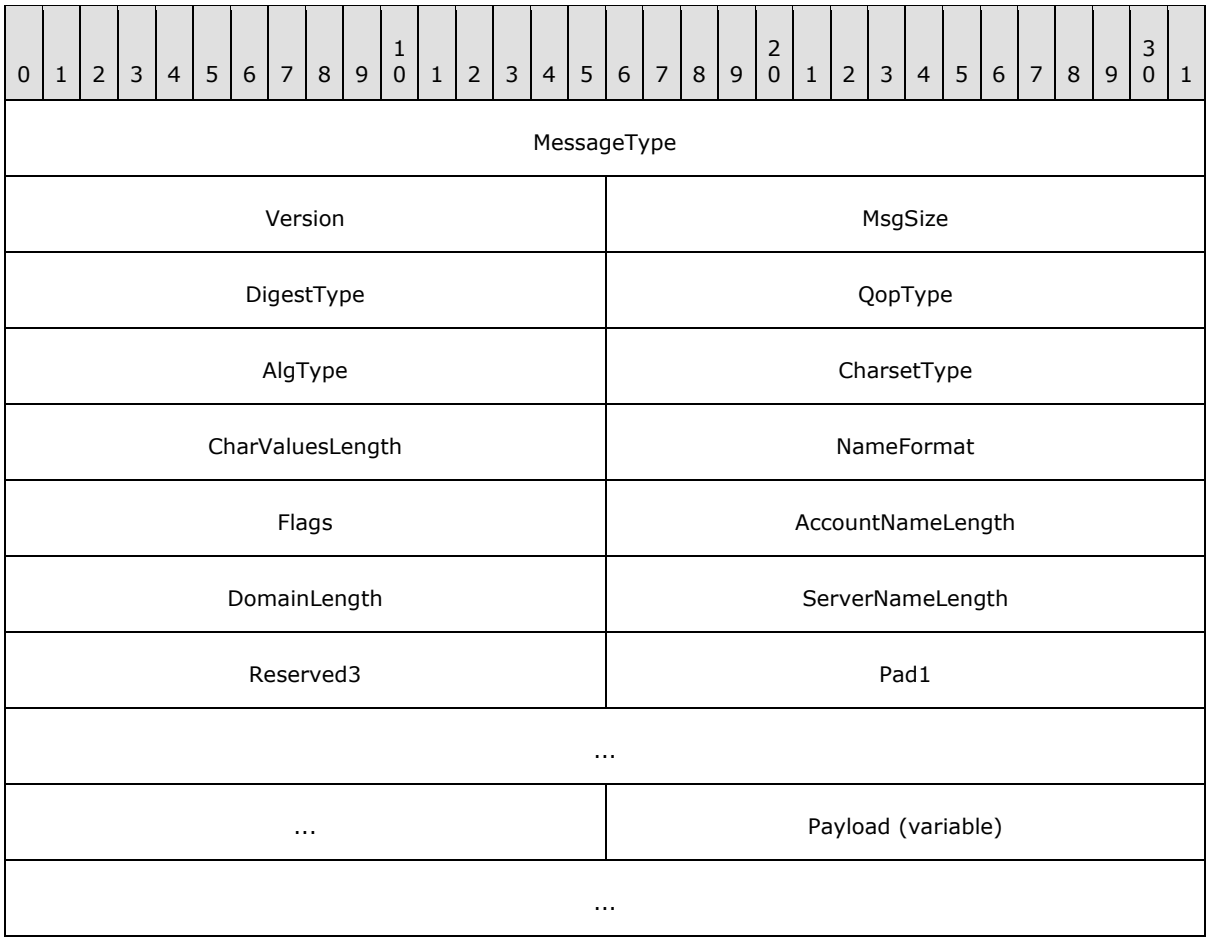
ChecksumAndSignature (variable): The signature value in the Server Checksum PAC buffer that MUST be as specified in [MS-PAC] section 2.8, followed by the signature value in the KDC (Privilege Server) Checksum PAC buffer.

2.2.3 Digest Validation Message Syntax

The digest validation protocol uses fields extracted from the digest-challenge and digest-response messages (as specified in [RFC2617] section 3.2 and [RFC2831] section 2.1) to verify the validity of the user signature (this is a hash performed with the user's password) and to retrieve the PAC for the user's account.

2.2.3.1 DIGEST_VALIDATION_REQ Message

The DIGEST_VALIDATION_REQ message defines a request to validate the input from the Digest Protocol Extensions (as specified in [MS-DPSP]) and retrieve user authorization information.



MessageType (4 bytes): A 32-bit unsigned integer that defines the digest validation message type. This member MUST be set to 0x0000001A.

Version (2 bytes): A 16-bit unsigned integer that defines the version of the digest validation protocol. The protocol version defined in this document is 1 (the value of this member MUST be 0x0001).

MsgSize (2 bytes): A 16-bit unsigned integer that MUST specify the total number of bytes in the DIGEST_VALIDATION_REQ message, as specified in section 2.2.3.1.

DigestType (2 bytes): A 16-bit unsigned integer that specifies the digest protocol used, which MUST be one of the following:

Value	Meaning
0x0003	Using the digest authentication mechanism (as specified in [RFC2617]) for the HTTP/1.1 Protocol.
0x0004	Using digest authentication as an Simple Authentication and Security Layer (SASL) mechanism, as specified in [RFC2831].

QopType (2 bytes): A 16-bit unsigned integer specifying the Quality of Protection (QoP) requested by the digest client (as specified in [RFC2617] section 3.2.1 and [RFC2831] section 2.1.2.1) that MUST be one of the following:

Value	Meaning
0x0001	The client did not specify a QoP. For backward compatibility with Digest Access authentication (as specified in [RFC2069]), digest authentication (as specified in [RFC2617]) made the QoP optional.<9>
0x0002	Authentication only. Represents auth, as specified in [RFC2617] section 3.2.1 and [RFC2831] section 2.1.1.
0x0003	Authentication and integrity protection. Represents auth-int, as specified in [RFC2617] section 3.2.1 and [RFC2831] section 2.1.1.
0x0004	Authentication with integrity protection and encryption. Represents auth-conf, as specified in [RFC2831] section 2.1.1.

AlgType (2 bytes): A 16-bit unsigned integer specifying the algorithm value specified by the digest client in the digest-challenge message (as specified in [RFC2617] section 3.2.1) that MUST be one of the following:

Value	Meaning
0x0001	MD5 value to produce the digest and checksum, as specified in [RFC2617] section 3.2.1 and [RFC2831] section 2.1.1.
0x0002	MD5-sess value to produce the digest and checksum, as specified in [RFC2617] section 3.2.1 and [RFC2831] section 2.1.1.

CharsetType (2 bytes): A 16-bit unsigned integer specifying the type of encoding used for **username** and **password** fields that MUST be one of the following:

Value	Meaning
0x0001	ISO8859-1 encoding is used for **username** and **password** fields.
0x0002	UTF-8 encoding is used for **username** and **password** fields.

CharValuesLength (2 bytes): A 16-bit unsigned integer that MUST specify the number of bytes in the **Payload** field of the DIGEST_VALIDATION_REQ message, and MUST NOT exceed the total size in MsgSize.

NameFormat (2 bytes): A 16-bit unsigned integer specifying the format of the user **AccountName** field (in the DIGEST_VALIDATION_REQ message), and MUST be one of the following:

Value	Meaning
0x0000	Digest server cannot determine the format of the user's **AccountName**.
0x0001	A format determined to be the SAM account name (as specified in [MS-SECO]).
0x0002	A format determined to be the user principal name (UPN) for the account (as specified in [MS-SECO]]).

Value	Meaning
0x0003	A format determined to be NetBIOS (as specified in [MS-SECO]).

Flags (2 bytes): A two-byte set of bit flags providing additional instructions for processing the DIGEST_VALIDATION_REQ message (as specified in section 2.2.3.1) by the DC. The **Flags** field MAY have one or more of the following flags set (with the exception of the constraint on bit C).

Note All other bits MUST be 0, and MUST be ignored on receipt.

0	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5
0	0	0	0	0	0	0	0	0	0	0	E	D	C	B	A

Where the bits are defined as:

Value	Description
A	The format of Username and Realm (carried in the **Payload** field of DIGEST_VALIDATION_REQ) MUST be determined by the DC.
B	The optional **Authzid** field (as specified in [RFC2831] section 2.1.2) is set and carried in the Payload buffer in the DIGEST_VALIDATION_REQ message (as specified in section 2.2.3.1).
C	Indicates that this request is from a server, so group memberships are to be expanded for the Account's PAC. This bit MUST NOT be set if this request is forwarded from a server's domain to user account's domain.
D	Indicates if a single backslash is found in the username value, as specified in [RFC2617] section 3.2.2.
E	Indicates the DC will attempt to validate the request with an un-escaped backslash, as specified in [MS-DPSP] section 2.2.

AccountNameLength (2 bytes): A 16-bit unsigned integer that MUST specify the length of the **AccountName** field in the Payload buffer.

DomainLength (2 bytes): A 16-bit unsigned integer that MUST specify the length of the Domain field in the Payload buffer.

ServerNameLength (2 bytes): A 16-bit unsigned integer that MUST specify the length of the **ServerName** field in the Payload buffer.

Reserved3 (2 bytes): A 16-bit unsigned integer field reserved for future use. The value of this member MUST be 0, and MUST be ignored on receipt.

Pad1 (8 bytes): An unused 64-bit unsigned integer. The value of this member MUST be 0, and MUST be ignored on receipt.

Payload (variable): A byte array that MUST contain the following strings in the following order. Note that all strings MUST be NULL terminated; strings MUST be encoded by using [ISO-8859-1], unless specified as **Unicode**. Each of the strings MUST be included. The string value MAY

be empty in which case a NULL character is used for the value. Remember that the last three strings are **Unicode** strings, so they have a Unicode NULL.

0	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	20	1	2	3	4	5	6	7	8	9	30	1
Username (variable)																															
...																															
Realm (variable)																															
...																															
Nonce (variable)																															
...																															
CNonce (variable)																															
...																															
NonceCount (variable)																															
...																															
Algorithm (variable)																															
...																															
QOP (variable)																															
...																															
Method (variable)																															
...																															
URI (variable)																															
...																															

Response (variable)
...
Hentity (variable)
...
Authzid (variable)
...
AccountName (variable)
...
Domain (variable)
...
ServerName (variable)
...

Username (variable): The user name value from the digest-response message that MUST be as specified in [RFC2617] section 3.2.2.

Realm (variable): The realm value that MUST be as specified in [MS-DPSP] section 3.1.5.4.

Nonce (variable): The nonce value from the digest-challenge message that MUST be as specified in [RFC2617] section 3.2.1.

CNonce (variable): The cnonce value from the digest-response message that MUST be as specified in [RFC2617] section 3.2.2.

NonceCount (variable): The nonce-count value from the digest-response message, that MUST be as specified in [RFC2617], section 3.2.2.

Algorithm (variable): The algorithm value from the digest-response message that MUST be as specified in [RFC2617] section 3.2.2.

QOP (variable): The QOP-value from the digest-response message that MUST be as specified in [RFC2617] section 3.2.2.

Method (variable): Method by which digest authentication information MUST be transmitted as part of the HTTP1.1 protocol. The string value is GET or PUT if digest authentication is used for the HTTP1.1 protocol, as specified in [RFC2617]. The string

value is AUTHENTICATE if digest authentication is used as an SASL mechanism, as specified in [RFC2617].

URI (variable): The digest-URI value from the digest-response message that MUST be as specified in [RFC2617] section 3.2.2.

Response (variable): The response value from the digest-response message that MUST be as specified in [RFC2617] section 3.2.2.

Hentity (variable): The H (entity-body) value that MUST be as specified in [RFC2617] section 3.2.2.3.

Authzid (variable): The Authzid value from the digest-response message that MUST be as specified in [RFC2831] section 2.1.2.

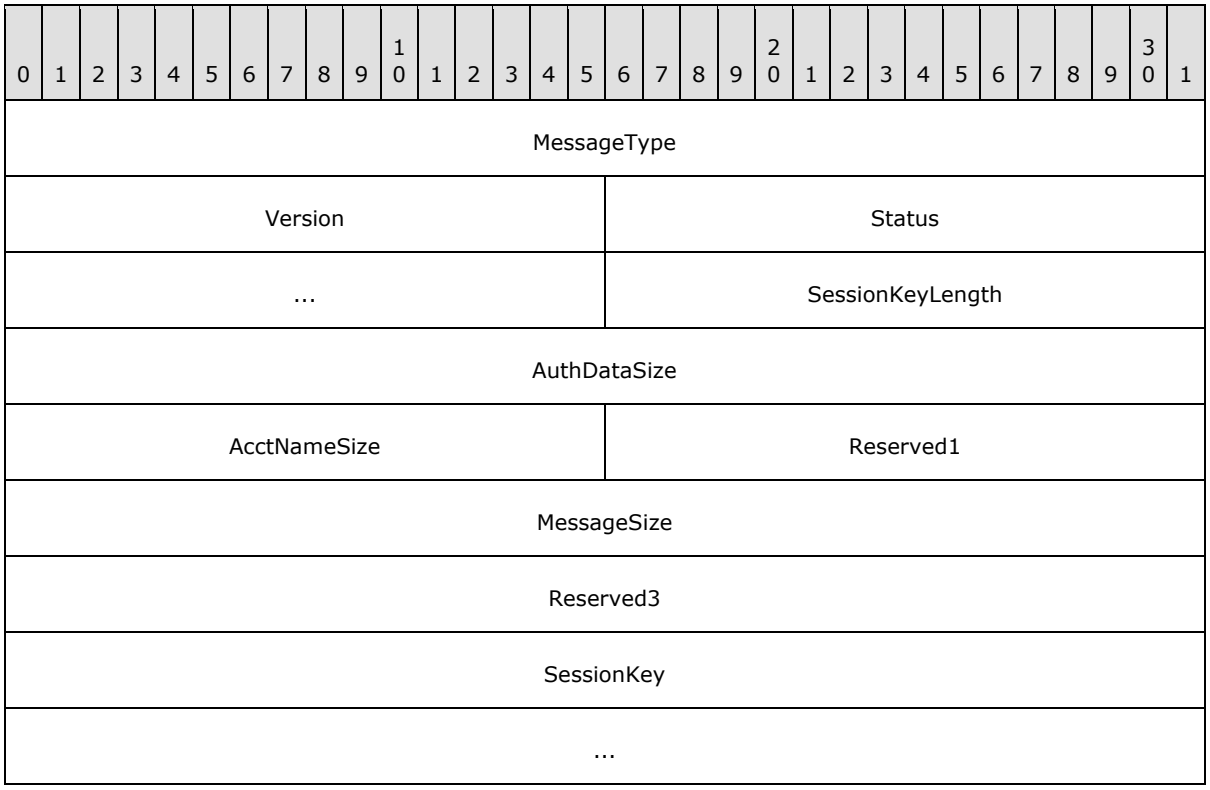
AccountName (variable): A **Unicode** string that MUST specify the user account name.

Domain (variable): A **Unicode** string that MUST specify the domain in which the user account is.

ServerName (variable): A **Unicode** string that MUST specify the NetBIOS name of the server that sent the DIGEST_VALIDATION_REQ message (as specified in section 2.2.3.1).

2.2.3.2 DIGEST_VALIDATION_RESP Message

The DIGEST_VALIDATION_RESP message is a response to a DIGEST_VALIDATION_REQ message.



...
...
...
...
...
...
Pad1
...
AuthData (variable)
...
AccountName (variable)
...

MessageType (4 bytes): A 32-bit unsigned integer that MUST specify the digest validation message type. This member MUST be 0x0A.

Version (2 bytes): A 16-bit unsigned integer that MUST specify the version of the digest validation protocol. The protocol version defined in this document is 1 (the value of this member MUST be 1).

Status (4 bytes): A 32-bit unsigned integer specifying if the digest authentication data sent in the DIGEST_VALIDATION_REQ (section 2.2.3.1) was successfully verified by the domain controller. On successful validation, the **Status** field MUST be set to STATUS_SUCCESS. On failure, it MUST be set to STATUS_LOGON_FAILURE. Both values are as specified in [MS-ERREF] section 4.

SessionKeyLength (2 bytes): A 16-bit unsigned integer that MUST specify the number of bytes of the **SessionKey** field in the DIGEST_VALIDATION_RESP message, as specified in section 2.2.3.2.

AuthDataSize (4 bytes): A 32-bit unsigned integer that MUST specify the number of bytes of the **AuthData** field in the DIGEST_VALIDATION_RESP message, as specified in section 2.2.3.2.

AcctNameSize (2 bytes): A 16-bit unsigned integer that MUST specify the number of bytes of the **AccountName** field in the DIGEST_VALIDATION_RESP message, as specified in section 2.2.3.2.

Reserved1 (2 bytes): A 16-bit unsigned integer field reserved for future use. The value of this member MUST be 0, and MUST be ignored on receipt.

MessageSize (4 bytes): A 32-bit unsigned integer that MUST specify the number of bytes in the entire DIGEST_VALIDATION_RESP message, as specified in section 2.2.3.2.

Reserved3 (4 bytes): A 32-bit unsigned integer field reserved for future use. The value of this member MUST be 0, and MUST be ignored on receipt.

SessionKey (32 bytes): A 32-byte buffer that MUST contain the Digest SessionKey, as specified in [RFC2617] section 3.2.2.2. The size of the SessionKey buffer is specified by the **SessionKeyLength** field. It MUST be less than or equal to 32.

Pad1 (8 bytes): An unused 64-bit unsigned integer. The value of this member MUST be 0, and MUST be ignored on receipt.

AuthData (variable): The **AuthData** field MUST contain the privilege attribute certificate (PAC). The length of the PAC MUST be specified by the **AuthDataSize** field. The length of this field MUST be 0 if the value of the **Status** field is STATUS_LOGON_FAILURE.

AccountName (variable): The **AccountName** field MUST contain the NetBIOS name of the user's account. The length of **AccountName** MUST be specified by the **AcctNameSize** field.

3 Protocol Details

Authentication Protocol Domain Support (specified for each of the protocols in the following sections) uses the Netlogon Remote Protocol (as specified in [MS-NRPC]) for transport. Each of the following protocols is a simple request-response exchange over this transport. The security assurances provided by the underlying Netlogon RPC layer are common to all of these protocols. All of these exchanges require that a remote procedure call (RPC) connection through the Netlogon RPC interface MUST be established with a domain controller (DC) for the domain to which the server belongs (as specified in section 1.5).

3.1 NTLM Logon Details

NT LAN Manager (NTLM) interactive logon and network logon MUST complete the authentication sequence by contacting the DC through the **NetrLogonSamLogonEx** method (as specified in [MS-NRPC] section 3.4.5.3.1), with parameters as specified in section 3.1.5.

The **NETLOGON_NETWORK_INFO** and **NETLOGON_VALIDATION_SAM_INFO4** data structures MUST be exchanged, as specified in [MS-NRPC] sections **2.2.1.4.5** and **2.2.1.4.13**. The domain controller MUST populate the **NETLOGON_VALIDATION_SAM_INFO4** with the information for the user logging on, and MUST send it back to the NTLM server. If no matching account is found, an error STATUS_NO_SUCH_USER (as specified in section 2.2) MUST be returned to the client, resulting in a logon failure.

3.1.1 Abstract Data Model

The protocol requires that the server MUST have a database or **directory** of accounts with authorization information available to it.

The NTLM abstract data model is specified in [MS-NLMP] section 3.1.1. The Netlogon abstract data model is specified in [MS-NRPC] section 3.1.1.

3.1.2 Timers

There are no timers for NTLM logon.

3.1.3 Initialization

There is no initialization that is specific to the NTLM logon protocol.

3.1.4 Higher-Layer Triggered Events

The NTLM logon message exchange MUST be triggered by a server requesting user authentication via the Netlogon remote procedure call (RPC) mechanism to the domain controller (DC).

3.1.5 Message Processing Events and Sequencing Rules

NTLM logon is a stateless protocol with request-response semantics.

The NTLM server calls the NetrLogonSamLogonEx method with the parameters defined in the following sections. Based on the account name supplied, a domain controller (DC) for the domain MUST be located, as specified in [MS-ADTS]. The NTLM server MUST establish a connection with the DC, as specified in [MS-NRPC] section 3.1.3. The NTLM server MUST invoke the **NetrLogonSamLogonEx** method, as specified in [MS-NRPC] section **3.5.4.4.1**.

The DC SHOULD attempt to validate the request and, if successful, SHOULD proceed to authenticate the user. If validation is unsuccessful, the domain controller MUST return an error. The role of the DC in the NTLM authentication sequence is specified in [MS-NLMP] section 3.3.

Upon successful validation, the user account's DC MUST send the domain global groups and universal groups (that the user is a member of) to the server's DC, and MUST follow the trust path that was used to contact the user's account DC, as specified in [MS-NRPC] section **3.5.4.4.1**. The server domain DC also MUST add the domain local groups, and then send the entire list of groups to the NTLM server to be used for authorization decisions.<10>

3.1.5.1 NTLM Interactive Logon

For NTLM interactive logons, the client MUST call **NetrLogonSamLogonEx** (as specified in [MS-NRPC] section **3.5.4.4.1**) with the following parameters (set as specified):<11>

- LogonLevel MUST be NetlogonInteractiveInformation.
- ValidationLevel MUST be either NetlogonValidationSamInfo2 or NetlogonValidationSamInfo4, as specified in [MS-NRPC] section **3.5.4.4.1**.
- LogonInformation MUST contain a reference to NETLOGON_INTERACTIVE_INFO, as specified in [MS-NRPC] section **3.5.4.4.1**.

The logon request MUST be sent to the domain controller of the user account domain that is located, as specified in [MS-NRPC] section **3.5.4.4.1**

If the domain controller for the user account is not reachable, but the user domain is one of the trusted domains, the logon MUST fail. If the user domain is not one of the trusted domains, the server's local account database MUST be used to authenticate the user.<12>

The request that is sent to the user account domain controller MUST contain the **NTOWF** of the user's password.

The domain controller MUST compare the local copy of the password to the one sent in the request. If there is a successful match, the domain controller MUST return data with ValidationInformation containing either a reference to NETLOGON_VALIDATION_SAM_INFO4 as specified in [MS-NRPC](section **3.5.4.4.1**, if the ValidationLevel in the request is NetlogonValidationSamInfo4) or a reference to NETLOGON_VALIDATION_SAM_INFO2 (as specified in [MS-NRPC] section **3.5.4.4.1**, if the ValidationLevel in the request is NetlogonValidationSamInfo2). If there is not a match, the DC MUST return the failure error code STATUS_WRONG_PASSWORD (as specified in section 2.2) with no response data.<13>

3.1.5.2 NTLM Network Logon

For NTLM network logons, the client MUST call NetrLogonSamLogonEx with the following parameters (set as specified):<14>

- LogonLevel MUST be NetlogonNetworkInformation.
- ValidationLevel MUST be NetlogonValidationSamInfo2 or NetlogonValidationSamInfo4, as specified in [MS-NRPC] sections **2.2.1.4.12** and **2.2.1.4.13**.
- LogonInformation MUST contain a reference to NETLOGON_NETWORK_INFO, as specified in [MS-NRPC].

The DC of the user account domain MUST be located as specified in [MS-NRPC] section **3.5.4.2**, and is sent the request. This request MUST contain the NTLM challenge-response pair that was

exchanged between the server and the client, as specified in [MS-NLMP] sections 2.2.1.2 and 2.2.1.3.

The DC MUST verify the response to the challenge as specified in [MS-NLMP] section 3.3 or by means of a sub-authentication package (see section 3.1.5.2.1).

Examples of usage of sub-authentication package occur with remote access authentications using the CHAP (as specified in [RFC1994]) method performed by Microsoft Windows Server platforms. In such cases, response computation of a MD5 hash value (as specified in [RFC1994]) is used instead of NTOWF.

If there is a match, the DC MUST return data with ValidationInformation containing a reference to NETLOGON_VALIDATION_SAM_INFO4 (as specified in [MS-NRPC] section **2.2.1.4.13**, if the ValidationLevel in the request is NetlogonValidationSamInfo4) or a reference to NETLOGON_VALIDATION_SAM_INFO2 (as specified in [MS-NRPC] section **2.2.1.4.12**, if the ValidationLevel in the request is NetlogonValidationSamInfo2). If there is not a match, the DC MUST return a failure error code STATUS_LOGON_FAILURE (as specified in section 2.2) with no response data. There MAY be additional causes for failure even if the match is successful, such as the account being disabled or other policy decisions. The client MUST treat all errors as resulting in a logon failure, although a server implementation MAY choose to send more descriptive error codes.<15>

3.1.5.2.1 Verifying Responses with Sub-Authentication Packages

The request to verify by a sub-authentication package is indicated by the **ParameterControl** field of the *LogonInformation* parameter.<16>

An example of sub-authentication package usage occurs with remote access authentications using the CHAP (as specified in [RFC1994]) method performed by Windows Server platforms. In such cases, response computation of a MD5 hash value (as specified in [RFC1994]) is used instead of NTOWF.

Using the NetrLogonSamLogon* family of methods MAY result in invoking a custom sub-authentication package at the **DC**, per the indication of the **ParameterControl** field of the *LogonInformation* parameter. In this case, such a sub-authentication package MUST serve as a custom response verification instead of using the method specified by section 3.3 of [MS-NLMP]. For more information on this sub-authentication package, see [MSDN-0SUBAUTHROUTINE9].

3.1.6 Timer Events

There are no timer events for NTLM logon. All associated timer events are specified in the Netlogon Remote Protocol (as specified in [MS-NRPC]) that serves as the transport.

3.1.7 Other Local Events

There are no other local events that impact the operation of this protocol.

3.2 PAC Validation Details

PAC validation MAY use the generic pass-through mechanism, as specified in [MS-NRPC] section 3.2.4.1. The KERB_VERIFY_PAC_REQUEST message (as specified in section 2.2.2.1) MUST be sent to the domain controller (DC) for privilege attribute certificate (PAC) verification. The signature verification algorithm MUST occur as specified in section 3.2.5.<17>

3.2.1 Abstract Data Model

The protocol requires that the server MUST have a database or directory of accounts with authorization information available to it.

The PAC abstract data model is specified in [MS-PAC]. The Netlogon abstract data model is specified in [MS-NRPC] section 3.2.1.

3.2.2 Timers

There are no timers for PAC validation.

3.2.3 Initialization

There is no initialization that is specific to PAC validation.

3.2.4 Higher-Layer Triggered Events

For information on higher-layer triggered events, see section 1.6.2.

3.2.5 Message Processing Events and Sequencing Rules

Kerberos PAC validation is a stateless protocol with request-response semantics.

3.2.5.1 Generating a KERB_VERIFY_PAC_REQUEST Message

The client MUST first assemble the KERB_VERIFY_PAC_REQUEST (section 2.2.2.1) structure by copying the signature values out of the privilege attribute certificate (PAC) (as specified in [MS-PAC] section 2.8) that the client is verifying. The message type field MUST be set to 0x00000003, and then the client is ready to contact the privilege attribute certificate (PAC)).

This exchange MUST be layered on top of the Netlogon generic pass-through, as specified in [MS-NRPC] section 3.2.4.1. The client MUST supply a KERB_VERIFY_PAC_REQUEST structure, packed as a single buffer, as the **GenericPassthrough.gpInput1** field. The **GenericPassthrough.gpInput2** field MUST be set to a UNICODE_STRING with a buffer of Kerberos.

If the DC cannot be reached, Netlogon (as specified in [MS-NRPC] section 3) MUST return the error STATUS_NO_LOGON_SERVERS (as specified in section 2.2). The client SHOULD fail the authentication attempt.<18>

3.2.5.2 Processing a KERB_VERIFY_PAC_REQUEST Message

On receipt of the message, the server MUST decode the KERB_VERIFY_PAC_REQUEST (section 2.2.2.1) message to locate the server checksum and the Key Distribution Center (KDC) checksum values. The server MUST verify the signature (as specified in [MS-PAC] section 2.8) and compare the result against the KDC checksum passed in the request. If the two values do not match, the server MUST return an error code, STATUS_LOGON_FAILURE (as specified in section 2.2) as the return value to the Netlogon Generic Pass-through method. If the values do match, the server MUST return STATUS_SUCCESS. There is no return message.

When the method completes, the client MAY examine the return code to determine if the PAC contents has been altered. Any non-zero return code MUST be treated by the client as a failure.

3.2.6 Timer Events

There are no timer events for PAC validation. All associated timer events are specified in the Netlogon Remote Protocol (as specified in [MS-NRPC]) that serves as the transport for PAC validation messages.

3.2.7 Other Local Events

There are no other local events that impact the operation of this protocol.

3.3 Digest Validation Details

The digest validation protocol MAY use the generic pass-through mechanism, as specified in [MS-NRPC] section 3.2.4.1. The exchanged messages are DIGEST_VALIDATION_REQ (section 2.2.3.1) and DIGEST_VALIDATION_RESP (section 2.2.3.2). The DIGEST_VALIDATION_RESP message MUST contain the status of authentication validation, and, on success, MUST also contain the user's authorization information.<19>

3.3.1 Abstract Data Model

The protocol requires that the server MUST have a database or directory of accounts with authentication and authorization information available to it. All state information, specifically the nonce challenge that foils replay attacks (as specified in [MS-DPSP] section 3.2.1) MUST be handled by the parties to the digest authentication protocol (as specified in [RFC2617] and [RFC2831]), and MUST be sent as part of the DIGEST_VALIDATION_REQ (section 2.2.3.1) message.

3.3.2 Timers

There are no timers for the digest validation protocol.

3.3.3 Initialization

There is no initialization that is specific to the digest validation protocol.

3.3.4 Higher-Layer Triggered Events

The digest validation message exchange MAY be triggered by a server that is configured to authenticate users by using the digest authentication mechanism, as specified in [RFC2617] and [RFC2831].<20>

3.3.4.1 Message Processing Events and Sequencing Rules

Digest validation is a stateless protocol with request-response semantics. The general model is:

- After the digest validation client sends the DIGEST_VALIDATION_REQ (section 2.2.3.1) message, it MUST receive either a DIGEST_VALIDATION_RESP (section 2.2.3.2) message from the digest validation server or an error status in the Netlogon generic pass-through function, as specified in [MS-NRPC] section 3.2.4.1.
- On receiving the DIGEST_VALIDATION_REQ message, the digest validation server MUST verify the keyed hash contained in the Payload buffer's **Response** field, and then send the DIGEST_VALIDATION_RESP message to the digest validation client.

3.3.4.2 Generating the DIGEST_VALIDATION_REQ Message

The client MUST construct the DIGEST_VALIDATION_REQ (section 2.2.3.1) message by using fields extracted from the digest-challenge and digest-response messages (as specified in [RFC2617] section 3.2 and [RFC2831] section 2.1). This message MUST be sent to the digest validation server to verify the validity of the user's signature (keyed hash performed with the user's password) and to retrieve the privilege attribute certificate (PAC) for the user's account. If the validation server cannot be contacted for any reason, the client SHOULD fail the authentication attempt.

The DIGEST_VALIDATION_REQ message MUST be packed as a contiguous buffer, and the encoded data MUST be sent by using the generic pass-through mechanism , as specified in [MS-NRPC] section 3.2.4.1. The encoding of the DIGEST_VALIDATION_REQ is as specified in section 2.2.3.1. The PackageName (GenericPassthrough.gpInput2) as specified in [MS-NRPC] section 3.2.4.1 in the NETLOGON_GENERIC_INFO structure as specified in [MS-NRPC] section **2.2.1.4.2** MUST be WDigest.<21>

3.3.4.3 Request Processing and Generating DIGEST_VALIDATION_RESP Message

The digest validation server MUST use the user name to look up the user's password and MUST verify the keyed hash contained in the Payload buffer's **Response** field of the DIGEST_VALIDATION_REQ (section 2.2.3.1) message . The algorithm to perform this validation MUST be as specified in [RFC2617] section 3.3.2 and [RFC2831] section 2.1.2.1.

If validation is successful, a DIGEST_VALIDATION_RESP (section 2.2.3.2) message with Status indicating successful authentication (that is, STATUS_SUCCESS), as specified in [MS-ERREF], and authorization information for the user's account (the PAC) MUST be sent back to the digest validation client. If unsuccessful, the digest validation server MUST send back the DIGEST_VALIDATION_RESP message with Status indicating failed authentication (that is, STATUS_LOGON_FAILURE), as specified in [MS-ERREF].

The digest validation response message DIGEST_VALIDATION_RESP MUST be packed as a contiguous buffer, and the encoded data MAY be sent by using the generic pass-through mechanism, as specified in [MS-NRPC] section 3.2.4.1. The encoding of DIGEST_VALIDATION_RESP is as specified in section 2.2.3.2.<22>

3.3.5 Timer Events

There are no timer events for the digest validation protocol. All associated timer events are specified in the Netlogon Remote Protocol (as specified in [MS-NRPC]) that serves as the transport for digest validation messages.

3.3.6 Other Local Events

There are no other local events that impact the operation of this protocol.

4 Protocol Examples

4.1 NTLM Interactive Logon

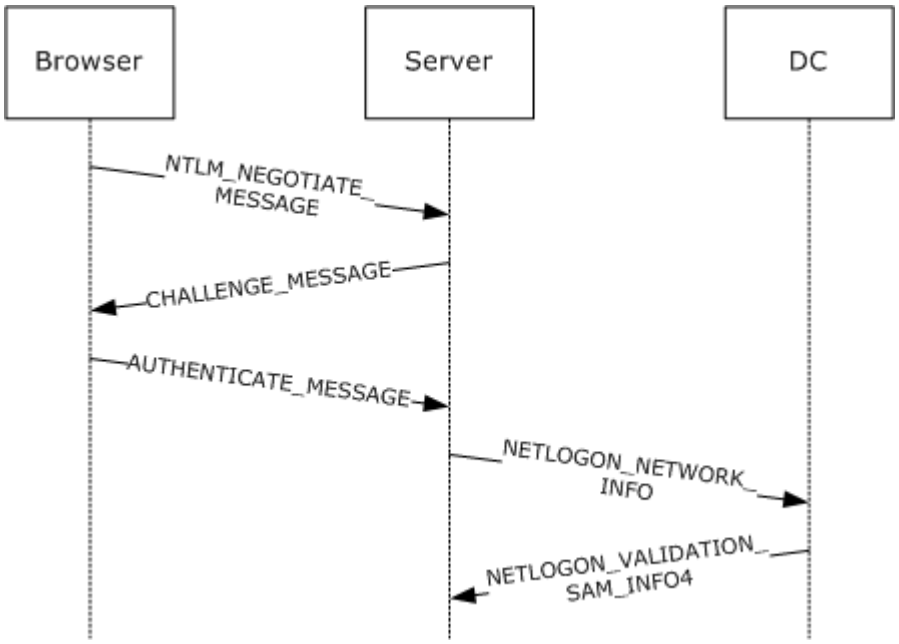


Figure 2: NTLM interactive logon

1. The user logs on to the computer desktop (labeled Client) by typing in the user name and password. Client sends an NTLM NEGOTIATE_MESSAGE to request authentication to the server.

2. The server sends a CHALLENGE_MESSAGE to the client.

3. The client responds to the challenge by signing it with its key and sending the response in an AUTHENTICATE_MESSAGE to the server.

4. The server forwards the client's response to the domain controller in a NETLOGON_NETWORK_INFO message.

5. The domain controller verifies the signature on the response, and returns the result to the server in a NETLOGON_VALIDATION_SAM_INFO4 message. If the verification is successful, the message contains the user's PAC with the authorization data. If the verification is unsuccessful, logon is denied.<23>

4.2 PAC Validation

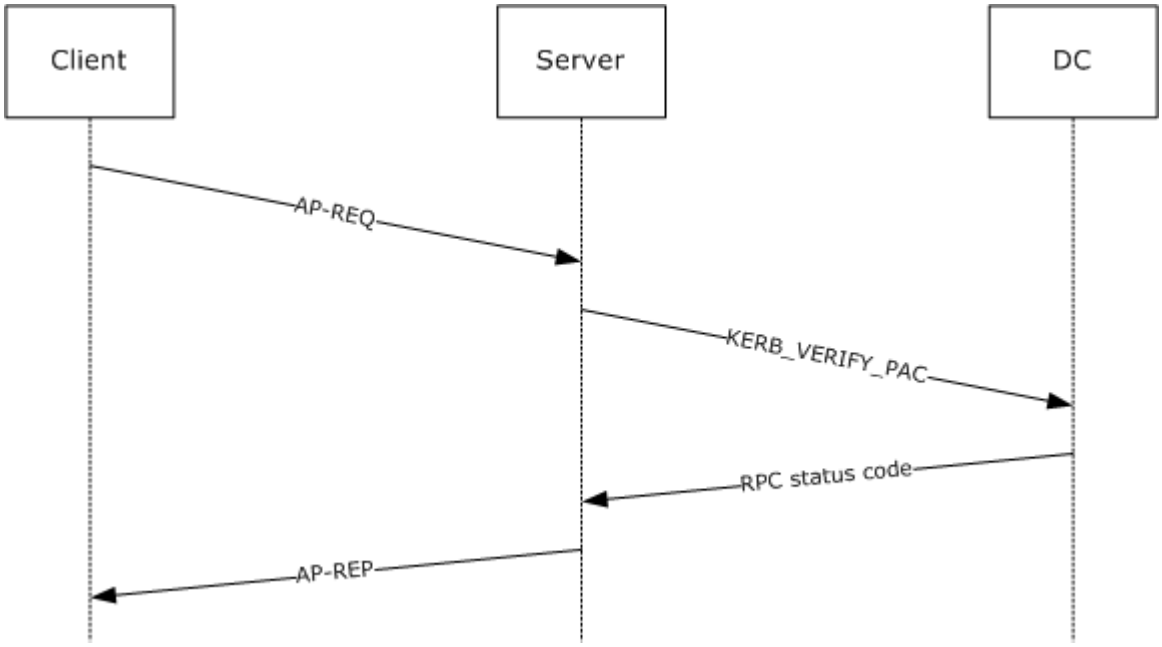


Figure 3: PAC validation

1. The client tries to access a resource requiring Kerberos authentication. The client sends an AP-REQ message to request authentication from the server.

2. The server forwards the PAC signature in the AP-REQ to the domain controller for verification in a KERB_VERIFY_PAC message.

3. The domain controller verifies the signature on the response and returns the result to the server. The error is returned as the appropriate RPC status code.

4. The server verifies the AP-REQ, and sends an AP-REP if the verification is successful.

4.3 Digest Validation Protocol

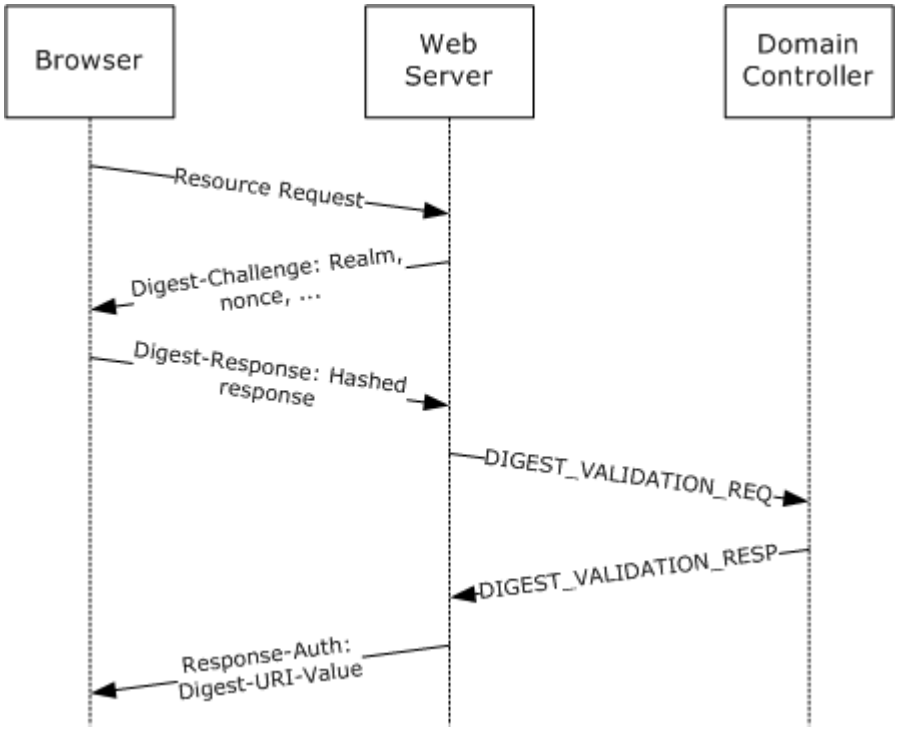


Figure 4: Digest validation protocol

1. The Web server is configured to require Digest authentication to gain access to certain documents. A user attempts to gain access to the protected document by using a Web browser. The Web server returns the Digest-Challenge message, as specified in [RFC2617] section 3.3. This challenge message includes a randomly generated nonce intended to foil replay attacks.

2. The Web browser obtains the user name and password for the user and constructs a response to the server's challenge. In the Digest-Response, the client proves knowledge of the user's password by performing a keyed hash over the user name, nonce, and other fields (the password is fed into the hash).

3. To validate the Digest-Response message, the Web server constructs the DIGEST_VALIDATION_REQ (section 2.2.3.1) message and sends it to the DC. The DIGEST_VALIDATION_REQ includes the nonce and the keyed hash value from the Digest-Response message. On receiving the DIGEST_VALIDATION_REQ message, the DC validates the message by performing the following steps:

    1. Looks up the user's password by using the user name.

    2. Recomputes the keyed hash over the clear-text fields from the Digest-Response message.

    3. Compares the resulting value to the value sent by the client.

4. If the DC's computed hash and the hash sent by the client match, the DC creates and sends back the DIGEST_VALIDATION_RESP (section 2.2.3.2) message with Status indicating successful authentication (that is, STATUS_SEVERITY_SUCCESS), as specified in [MS-ERREF] section 4), and authorization information for the user's account (the PAC). Otherwise, the DC sends back the

DIGEST_VALIDATION_RESP message with Status indicating failed authentication (that is, STATUS_LOGON_FAILURE), as specified in [MS-ERREF] section 4.

5. For mutual authentication, the server has the option to send a keyed hash over the URI that the client requested, and return it to the client in the Response-Auth message. Note that sending the Response-Auth message applies only to digest authentication when used as a Simple Authentication and Security Layer (SASL) mechanism, as specified in [RFC2831] section 2.1.3.

5 Security

5.1 Security Considerations for Implementers

Authentication Protocol Domain Support does not have any built-in security mechanisms to provide authentication or to ensure confidentiality and integrity. Instead, it relies on security mechanisms that are specified in [MS-RPCE] section 5 to protect Netlogon RPC (as specified in [MS-NRPC]), which serves as the transport for the protocols defined in this document.

5.2 Index of Security Parameters

There are no security parameters for Authentication Protocol Domain Support. All associated security parameters are specified in the Netlogon Remote Protocol (as specified in [MS-NRPC]), which provides all security for Authentication Protocol Domain Support messages.

6 Appendix A: Windows Behavior

The information in this specification is applicable to the following versions of Windows:

- Windows NT
- Windows 2000
- Windows XP
- Windows Server 2003
- Windows Vista

Exceptions, if any, are noted below. Unless otherwise specified, any statement of optional behavior in this specification prescribed using the terms SHOULD or SHOULD NOT implies Windows behavior in accordance with the SHOULD or SHOULD NOT prescription. Unless otherwise specified, the term MAY implies that Windows does not follow the prescription.

<1> Section 1.5: The server communicating with the DC is required to have a **machine account** to establish a secure connection, as specified in [MS-NRPC].

<2> Section 1.5.2: For more information, see the applicability statement in section 1.6.2.

<3> Section 1.5.3: The digest validation server is the domain controller (DC) that looks up the password from the user account that is maintained by Active Directory (AD).

<4> Section 1.6.1: NTLM interactive logon is applicable only when the domain controller is running Windows NT Server 4.0.

Network logon using NTLM is applicable under any one of the following situations:

- Client or application is explicitly configured to use NTLM.
- Client or application uses an invalid target name (the server's **principal** name) and cannot log on by using Kerberos.
- Client or application specifies a target name that cannot be resolved and, therefore, cannot log on by using Kerberos.

<5> Section 1.6.2: The following list details the behavior of each version of Windows.

- Windows 2000 Server and Windows XP do not validate the PAC when the application server is running under the local system context or has SeTcbPrivilege, as specified in [MS-LSAD] section 7, note 37 . Otherwise, Windows 2000 Server and Windows XP use Kerberos PAC validation.
- Windows Server 2003 does not validate the PAC when the application server is running under the local system context, the network service context, or has the SeTcbPrivilege. Otherwise, Windows Server 2003 uses Kerberos PAC validation.
- Windows Server 2003 SP1 does not validate the PAC when the application server is under the local system context, the network service context, the local service context, or has the SeTcbPrivilege privilege. Otherwise, Windows Server 2003 SP1 and future service packs use Kerberos PAC validation.
- Windows Vista does not validate the PAC by default for services. Windows still validates the PAC for processes that are not running as services. PAC validation can be enabled when the

application server is not running in the context of local system, network service, or local service; or it does not have the SeTcbPrivilege, as specified in [MS-LSAD] section 7, note 37.

<6> Section 1.6.3: The digest validation protocol is used to support the Digest Protocol Extensions (as specified in [MS-DPSP]) in a Windows environment to complete the initial authentication phase and retrieve the authorization information that is stored with the user's account in Active Directory. The digest authentication protocol itself offers some level of protection (that is, it does not send the user's password in the clear) but is considered weaker than Kerberos (as specified in [MS-KILE]) and public key–based authentication (for example, client-side authentication). Consequently, the Digest Protocol Extensions should be used only in environments in which these stronger mechanisms are unavailable.

<7> Section 2.1: Netlogon RPC messages are used in the logon interface for domain support for authentication protocols.

<8> Section 2.2.2: The PAC validation request message is sent by using the generic pass-through mechanism.

<9> Section 2.2.3.1: If the client did not specify the QoP, the default QoP is authentication only.

<10> Section 3.1.5: The DC attempts to validate the request and proceeds to try to authenticate the user.

The NTLM logon verification server is always a DC, and the NTLM logon verification client can be a member server of a domain or another DC. The mapped account is searched within the forest of the DC. The DC can also contact another DC in the same forest if the user's account is located in a different domain than that of the DC that receives the request.

<11> Section 3.1.5.1: In Windows NT 4.0 and Windows 2000 Server, the ValidationLevel is NetlogonValidationSamInfo2. In Windows XP and Windows Server 2003, the ValidationLevel is NetlogonValidationSamInfo4.

<12> Section 3.1.5.1: If the domain is not trusted, the server uses the local security accounts manager (SAM) database to authenticate the user. If the user account is not found in the local SAM database, the guest account can be used to authenticate the user. If the guest account has a password, that password is used to verify the challenge response; otherwise, the challenge is not verified.

<13> Section 3.1.5.1: In Windows NT 4.0 and Windows 2000 Server, the ValidationLevel is **NetlogonValidationSamInfo2** (as specified in [MS-NRPC] section **2.2.1.4.12**). In Windows XP and Windows Server 2003, the ValidationLevel is NetlogonValidationSamInfo4 (as specified in [MS-NRPC] section **2.2.1.4.13**). If the DC returns a failure code, Windows fails the logon attempt. Other failure codes are also returned based on the policy (for a list of error codes, see [MS-ERREF]), and all of them result in logon failure.

When a Windows XP NETLOGON client talks with a Windows 2000 DC, Netlogon is responsible for negotiating the minimal ValidationLevel that is supported. This negotiated ValidationLevel is used, and the corresponding validation information is returned, as specified in [MS-NRPC] section **3.5.4.4.1**. Note that the NETLOGON_VALIDATION_SAM_INFO4 structure is a superset of the NETLOGON_VALIDATION_SAM_INFO2 structure.

<14> Section 3.1.5.2: In Windows NT 4.0 and Windows 2000 Server, the ValidationLevel is NetlogonValidationSamInfo2. In Windows XP and Windows Server 2003, the ValidationLevel is NetlogonValidationSamInfo4.

<15> Section 3.1.5.2: In Windows NT 4.0 and Windows 2000 Server, the ValidationLevel is NetlogonValidationSamInfo2 (as specified in [MS-NRPC] section **2.2.1.4.12**). In Windows XP and Windows Server 2003, the ValidationLevel is NetlogonValidationSamInfo4.

When a Windows XP NETLOGON client talks with a Windows 2000 DC, Netlogon is responsible for negotiating the minimal ValidationLevel that is supported. This negotiated ValidationLevel is used, and the corresponding validation information is returned, as specified in [MS-NRPC] section **2.2.1.4.17**. Note that the NETLOGON_VALIDATION_SAM_INFO4 structure is a superset of the NETLOGON_VALIDATION_SAM_INFO2 structure.

On Windows 2000 Server and Windows Server 2003 DCs, the user information contained in the NETLOGON_VALIDATION_SAM_INFO4 structure is obtained by querying Active Directory.

<16> Section 3.1.5.2.1: The **ParameterControl** field provides an extensibility point for software providers. Windows XP, Windows Server 2003, and Windows Vista do not have subauthentication packages installed by default.

<17> Section 3.2: The validation protocol uses the generic pass-through mechanism, as specified in [MS-NRPC] section 3.2.4.1. The PAC validation server is always a DC, and the PAC validation client is always a member server or workstation.

<18> Section 3.2.5.1: If the message fails to be delivered to the DC, the client fails the logon attempt.

<19> Section 3.3: The validation protocol uses the generic pass-through mechanism, as specified in [MS-NRPC] section 3.2.4.1. The digest validation server is always a domain controller, and the digest authentication client can be a member server of a domain or another DC in the same forest. The DC can also contact another DC in the same forest by using the DIGEST_VALIDATION_REQ and DIGEST_VALIDATION_RESP exchange, if the user's account is located in a different domain than that of the DC that receives the request. Windows client products (such as Windows XP and Windows Vista) do not support digest in this manner.

<20> Section 3.3.4: During the digest authentication exchange (as specified in [RFC2617] and [RFC2831]), the digest server initiates the digest validation exchange with the DC. This transaction that is initiated by the digest server does not have direct access to the user's password and relies on the DC to validate the digest input data sent by the client.

<21> Section 3.3.4.2: Digest validation is used by the Microsoft implementation of the digest authentication protocol, as specified in [RFC2617] and [RFC2831]. The digest server contacts the domain controller by using the digest validation protocol to validate the client's authentication data and to retrieve user authorization data, as specified in section 3.3.4.3.

<22> Section 3.3.4.3: The digest validation response message is sent by using the generic pass-through mechanism, as specified in [MS-NRPC] section 3.2.4.1.

<23> Section 4.1: The user receives an error message.

7 Index