[MS-NRPC]: Netlogon Remote Protocol

> **This topic lists the Errata found in [MS-NRPC] since it was last published. Since this topic is updated frequently, we recommend that you subscribe to these RSS or Atom feeds to receive update notifications.**
>
> **Errata are subject to the same terms as the Open Specifications documentation referenced.**

RSS

Atom

To view a PDF file of the errata for the previous versions of this document, see the following ERRATA Archives:

October 16, 2015 - Download

June 30, 2015 - Download

July 18, 2016 - Download

September 23, 2019 - Download

August 24, 2020 – Download

April 7, 2021 - Download

June 24, 2021 - Download

April 29, 2022 - Download

Errata below are for Protocol Document Version V40.0 2022/04/29.

Errata Published*	Description
2022/11/08	In section 3.1.1 Abstract Data Model: SealSecureChannel removed duplicate and adjusted to this setting MUST be TRUE. Changed from: TrustPasswordVersion: ... SealSecureChannel: ... StrongKeySupport: ... The Netlogon client and server variables are as follows: LocatedDCsCache: ... SealSecureChannel: A Boolean setting that indicates whether the RPC message has to be encrypted or just integrity-protected ([C706] section 13.2.5). When TRUE, the message will be encrypted; otherwise, it will be integrity-protected. Implementations SHOULD<68> persistently store and retrieve the SealSecureChannel variable.

Errata Published*	Description
	VulnerableChannelAllowList: A setting expressed in Security Descriptor Definition Language (SDDL) ([MS-DTYP] section 2.5.1) of Netlogon client allowed to not use secure bindings, see section 3.1.4.6.<69>

Changed to:

TrustPasswordVersion: ...

StrongKeySupport: ...

The Netlogon client and server variables are as follows:

LocatedDCsCache: ...

SealSecureChannel: A Boolean setting that indicates whether the RPC message has to be encrypted or just integrity-protected ([C706] section 13.2.5). This setting MUST be TRUE.

VulnerableChannelAllowList: A setting expressed in Security Descriptor Definition Language (SDDL) ([MS-DTYP] section 2.5.1) of Netlogon client allowed to not use secure bindings, see section 3.1.4.6.<69>

In section 3.1.4.6 Calling Methods Requiring Session-Key Establishment: Step 1: Replaced if…TRUE... with: Clients MUST request the Privacy authentication level. Step 4: Added RPC Integrity to the MUST deny request list. Updated product note.

Changed from:

The client and server follow this sequence of steps.<74>

1.   The client SHOULD<75> bind to the RPC server using TCP/IP.

The client and server MUST utilize a secure bind. If a secure bind is used, the client instructs the RPC runtime to use the Netlogon SSP ([MS-RPCE] section 2.2.1.1.7) for privacy/integrity of the RPC messages. If the SealSecureChannel setting is TRUE, the client requests the Privacy authentication level from the RPC runtime. If the SealSecureChannel setting is FALSE, then the authentication level requested is Integrity.

2.   …

3.   …

4.   If secure bind is not used, the server MUST deny the request unless client is in the VulnerableChannelAllowList setting.<76>

Changed to:

The client and server follow this sequence of steps.<74> |

Errata Published*	Description
	1. The client SHOULD<75> bind to the RPC server using TCP/IP.
	The client and server MUST utilize a secure bind. If a secure bind is used, the client instructs the RPC runtime to use the Netlogon SSP ([MS-RPCE] section 2.2.1.1.7) for privacy/integrity of the RPC messages. Clients MUST request the Privacy authentication level.
	2. …
	3. …
	4. If secure bind is not used or the client is using RPC Integrity instead of RPC Privacy, the server MUST deny the request unless client is in the VulnerableChannelAllowList setting.<76>
	In section 3.4.1 Abstract Data Model: RequireSignOrSeal: Added that this setting MUST be TRUE.
	Changed from:
	RequireSignOrSeal: Indicates whether the client SHOULD<87> continue session-key negotiation when the server did not specify support for Secure RPC as described in the negotiable option Y of section 3.1.4.2.
	Changed to:
	RequireSignOrSeal: Indicates whether the client SHOULD<87> continue session-key negotiation when the server did not specify support for Secure RPC as described in the negotiable option Y of section 3.1.4.2. This setting MUST be TRUE.
	In section 3.4.3 Initialization: Changed RequireSignOrSeal from SHOULD to MUST be initialized to TRUE.
	Changed from:
	RequireSignOrSeal SHOULD<92> be initialized to TRUE.
	Changed to:
	RequireSignOrSeal MUST<92> be initialized to TRUE.
	In section 3.5.1 Abstract Data Model: SignSecureChannel: Added This setting is deprecated, as SealSecureChannel MUST be TRUE.
	Changed from:
	SignSecureChannel: A Boolean variable that determines whether a domain member attempts to negotiate signing for all secure channel traffic that it initiates.
	Changed to:

Errata Published*	Description
	SignSecureChannel: A Boolean variable that determines whether a domain member attempts to negotiate signing for all secure channel traffic that it initiates. This setting is deprecated, as SealSecureChannel MUST be TRUE. In Section 3.5.3 Initialization: RejectMD5Clients, SealSecureChannel, and SignSecureChannel set to TRUE. Changed from: RejectMD5Clients SHOULD be initialized in an implementation-specific way and set to FALSE. SealSecureChannel SHOULD be TRUE. SignSecureChannel SHOULD be initialized in an implementation-specific way and set to TRUE. Any changes made to the SignSecureChannel registry keys are reflected in the ADM elements when a PolicyChange event is received (section 3.1.6). Changed to: RejectMD5Clients SHOULD be initialized in an implementation-specific way and set to TRUE. SealSecureChannel MUST be TRUE. SignSecureChannel SHOULD be initialized in an implementation-specific way and set to TRUE. Any changes made to the SignSecureChannel registry keys are reflected in the ADM elements when a PolicyChange event is received (section 3.1.6). This setting is deprecated, as SealSecureChannel MUST be true.
2022/09/20	In section 1.3.1 Pass-Through Authentication: Added little endian usage statement. Changed from: … The secure channel is achieved by encrypting the communication traffic with a session key computed using a secret key (called a server's machine account password) shared by the server and the DC. Changed to: … The secure channel is achieved by encrypting the communication traffic with a session key computed using a secret key (called a server's machine account password) shared by the server and the DC. Unless otherwise specified, MS-NRPC uses little endian for byte ordering before encryption. In section 2.2.1.3.7 NL_TRUST_PASSWORD: Added product note about little endian usage for big endian users. Changed from: . . . The NL_TRUST_PASSWORD structure is encrypted using the negotiated encryption algorithm before it is sent over the wire. Changed to: . . . The NL_TRUST_PASSWORD structure is encrypted using the negotiated encryption algorithm before it is sent over the wire.<24> <24> Section 2.2.1.3.7: Windows domain controller expects little-endian byte ordering for the encryption input. If your processor is in big endian, then both the wide-character buffer and length fields in the NL_TRUST_PASSWORD structure MUST be converted to little endian before encryption. After encryption, byte swapping to reverse the order will be needed.

Errata Published*	Description
	In section 3.4.5.2.5 Calling NetrServerPasswordSet2: Added product note about little endian usage for big endian users.
	Changed from:
	Encrypt the ClearNewPassword parameter using the negotiated encryption algorithm (determined by bits C, O, or W, respectively, in the NegotiateFlags member of the ServerSessionInfo table entry for PrimaryName) and the session key established as the encryption key.
	Changed to:
	Encrypt <98> the ClearNewPassword parameter using the negotiated encryption algorithm (determined by bits C, O, or W, respectively, in the NegotiateFlags member of the ServerSessionInfo table entry for PrimaryName) and the session key established as the encryption key.
	<98> Section 3.4.5.2.5: Windows domain controller expects little-endian byte ordering for the encryption input. If your processor is in big endian, then both the wide-character buffer and length fields in the NL_TRUST_PASSWORD structure MUST be converted to little endian before encryption. After encryption, byte swapping to reverse the order will be needed.
	In section 3.5.4.4.5 NetrServerPasswordSet2 (Opnum 30): Added product note about little endian usage for big endian users.
	Changed from:
	ClearNewPassword: A pointer to an NL_TRUST_PASSWORD structure, as specified in section 2.2.1.3.7, that contains the new password encrypted as specified in Calling NetrServerPasswordSet2 (section 3.4.5.2.5).
	Changed to:
	ClearNewPassword: A pointer to an NL_TRUST_PASSWORD structure, as specified in section 2.2.1.3.7, that contains the new password encrypted<178> as specified in Calling NetrServerPasswordSet2 (section 3.4.5.2.5).
	<178> Section 3.5.4.4.5: Windows domain controller expects little-endian byte ordering for the encryption input. If your processor is in big endian, then both the wide-character buffer and length fields in the NL_TRUST_PASSWORD structure MUST be converted to little endian before encryption. After encryption, byte swapping to reverse the order will be needed.

*Date format: YYYY/MM/DD