[MS-SAMR]: Security Account Manager (SAM) Remote Protocol (Client-to-Server)

This topic lists the Errata found in [MS-SAMR] 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

June 1, 2017 - Download

September 15, 2017 - Download

September 12, 2018 - Download

April 7, 2021 - Download

October 6, 2021 - Download

April 29, 2022 - Download

Errata below are for Protocol Document Version V45.0- 2022/04/29.

Errata Published*	Description
2023/02/27	In Section 1.3.2 Method-Based Perspective Description: Added description of new method 'SamrValidateComputerAccountReuseAttempt' to Miscellaneous category, which confirms whether client attempts to re-use a particular computer account are allowed. Changed from: • SamrCloseHandle: This method releases server resources associated with the RPC context handle that is passed as a parameter. Changed to: • SamrCloseHandle: This method releases server resources associated with the RPC context handle that is passed as a parameter. • SamrValidateComputerAccountReuseAttempt: This method validates whether a client attempt to re-use a given computer account is permitted. In section 2.2.7.15 SAMPR_REVISION_INFO_V1 Description: Updated SupportedFeatures parameter of the SAMPR_REVISION_INFO_V1 structure by adding hex value (0x00000020) to represent that the server validates client reuse of computer accounts through client calls to the SamrValidateComputerAccountReuseAttempt method.

Errata Published*	Description

	Changed from:
	0x00000010 On receipt by the client, this value, when set, indicates that the client should use AES Encryption with the SAMPR_ENCRYPTED_PASSWORD_AES structure to encrypt password buffers when sent over the wire. See AES Cipher Usage (section 3.2.2.4) and SAMPR_ENCRYPTED_PASSWORD_AES (section 2.2.6.32).
	
	Changed to:
	0x00000010 On receipt by the client, this value, when set, indicates that the client should use AES Encryption with the SAMPR_ENCRYPTED_PASSWORD_AES structure to encrypt password buffers when sent over the wire. See AES Cipher Usage (section 3.2.2.4) and SAMPR_ENCRYPTED_PASSWORD_AES (section 2.2.6.32).
	
	0x00000020 On receipt of this value by the client, when set, indicates that the server supports the validation of computer account re-use through client calls to the SamrValidateComputerAccountReuseAttempt method.
	
	In Section 3.1.1.12 ComputerAccountReuseAllowList
	Description: Created new section to define ADM element 'ComputerAccountReuseAllowList' that is used to hold trusted computer account owners.
	
	In Section 3.1.5 Message Processing Events and Sequencing Rules
	Description: Added new method to Opnum list: 'SamrValidateComputerAccountReuseAttempt' (Opnum 74)
	
	Changed from:
	SamrUnicodeChangePasswordUser4 Changes a user account password.
	Opnum 73
	
	Changed to:
	SamrUnicodeChangePasswordUser4 Changes a user account password.
	Opnum 73
	SamrValidateComputerAccountReuseAttempt Validates whether clients can re-use a computer account.
	Opnum 74
	
	In Section 3.1.5.13.8 SamrValidateComputerAccountReuseAttempt (Opnum 74)
	Description: Created new method 'SamrValidateComputerAccountReuseAttempt' (Opnum 74) that validates whether client attempts to reuse computer accounts are permitted.<pbn72>
	
	<pbn72>: ComputerAccountReuseAllowList and supporting method SamrValidateComputerAccountReuseAttempt are supported on the operating systems specified in [MSKB-5020276], each with its related KB article download installed.
	
	In Section 6 Appendix A: Full IDL
	Description: Added IDL for new method SamrValidateComputerAccountReuseAttempt Opnum 74.
	// opnum 74
	  NTSTATUS SamrValidateComputerAccountReuseAttempt(
	    [in] SAMPR_HANDLE ServerHandle,
	    [in] PRPC_SID ComputerSid,

<table>
<tr><th>Errata Published*</th><th>Description</th></tr>
<tr><td></td><td>[out] BOOL* Result

);</td></tr>
<tr><td>2022/09/20</td><td>In Section 2.2.1.18, AEAD-AES-256-CBC-HMAC-SHA512 Constants
Description: Updated AEAD-AES-256-CBC-HMAC-SHA512 constants to ensure that the value details allow an implementation to be successfully created.

Changed from:
<table>
<tr><th>Constant Name</th><th>Value</th></tr>
<tr><td>versionbyte</td><td>0x01</td></tr>
<tr><td>versionbyte_length</td><td>1</td></tr>
<tr><td>SAM_AES_256_ALG</td><td>"AEAD-AES-256-CBC-HMAC-SHA512"</td></tr>
<tr><td>SAM_AES256_ENC_KEY_STRING</td><td>"Microsoft SAM encryption key AEAD-AES-256-CBC-HMAC-SHA512 16"</td></tr>
<tr><td>SAM_AES256_MAC_KEY_STRING</td><td>"Microsoft SAM MAC key AEAD-AES-256-CBC-HMAC-SHA512 16"</td></tr>
<tr><td>SAM_AES256_ENC_KEY_STRING_LENGTH</td><td>sizeof(SAM_AES256_ENC_KEY_STRING)</td></tr>
<tr><td>SAM_AES256_MAC_KEY_STRING_LENGTH</td><td>sizeof(SAM_AES256_MAC_KEY_STRING)</td></tr>
</table>
Changed to:
<table>
<tr><th>Constant/value</th><th>Description</th></tr>
<tr><td>Versionbyte
0x01</td><td>Version identifier.</td></tr>
<tr><td>versionbyte_length
1</td><td>Version identifier length.</td></tr>
<tr><td>SAM_AES_256_ALG
“AEAD-AES-256-CBC-HMAC-SHA512”</td><td>A NULL terminated ANSI string.</td></tr>
<tr><td>SAM_AES256_ENC_KEY_STRING
"Microsoft SAM encryption key AEAD-AES-256-CBC-HMAC-SHA512 16"</td><td>A NULL terminated ANSI string.</td></tr>
<tr><td>SAM_AES256_MAC_KEY_STRING
"Microsoft SAM MAC key AEAD-AES-256-CBC-HMAC-SHA512 16"</td><td>A NULL terminated ANSI string.</td></tr>
<tr><td>SAM_AES256_ENC_KEY_STRING_LENGTH
sizeof(SAM_AES256_ENC_KEY_STRING)
(61)</td><td>The length of SAM_AES256_ENC_KEY_STRING, including the null terminator.</td></tr>
<tr><td>SAM_AES256_MAC_KEY_STRING_LENGTH
sizeof(SAM_AES256_MAC_KEY_STRING)
(54)</td><td>The length of SAM_AES256_MAC_KEY_STRING, including the null terminator</td></tr>
</table>
In Section 3.2.2.4, AES Cipher Usage
Description: Specified the format of secret plaintext for SamrUnicodeChangePasswordUser4 and SamrSetInformationUser2 when creating the content encryption key (CEK); and clarified the usage of enc_key and mac_key when encrypting the data.</td></tr>
</table>

Errata Published*	Description
	Changed from: ● For the SamrUnicodeChangePasswordUser4 method (section 3.1.5.10.4), the shared secret is the plaintext old password and the CEK is generated as specified in section 3.2.2.5. Changed to: ● For the SamrUnicodeChangePasswordUser4 method (section 3.1.5.10.4), the shared secret is the plaintext old password and the CEK is generated as specified in section 3.2.2.5. ● For SamrUnicodeChangePasswordUser4 and SamrSetInformationUser2, the secret plaintext MUST be in the format specified in section 2.2.6.32. Changed from: Let AuthData ::= HMAC-SHA-512(mac_key, versionbyte + IV + Cipher + versionbyte_length) Changed to: Let AuthData ::= HMAC-SHA-512(mac_key, versionbyte + IV + Cipher + versionbyte_length) Note that enc_key is truncated to 32-bytes and the entire 64-byte mac_key is used. In Section 3.2.2.5 Deriving an Encryption Key from a Plaintext Password Description: Clarified how a 16-byte encryption key MUST be derived. Changed from: The client MUST derive the CEK in the following manner: CEK :: = (PBKDF2(NT HASH of "OldPassword", Salt, Iteration Count, 512)) Changed to: The client MUST derive the CEK in the following manner: A 16-byte encryption key is derived using the PBKDF2 algorithm with HMAC SHA-512, the NT-hash of the users existing password, a random 16-byte Salt, and an Iteration Count. The Iteration Count MUST be between 5000 and 1,000,000 inclusive. CEK :: = (PBKDF2(NT HASH of "OldPassword", Salt, Iteration Count, 16))

*Date format: YYYY/MM/DD