[MS-ADTS]: Active Directory Technical Specification

This topic lists the Errata found in the MS-ADTS document 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

March 20, 2017 - Download

September 15, 2017 - Download

December 1, 2017 - Download

March 16, 2018 - Download

September 12, 2018 - Download

March 13, 2019 - Download

March 4, 2020 - Download

August 24, 2020 – Download

April 7, 2021 - Download

Errata below are for Protocol Document Version 54.0 2021/06/25.

<table>
<tr><th>Errata Published*</th><th>Description</th></tr>
<tr><td>2022/02/28</td><td>Section 3.1.1.3.4.6 LDAP Policies

The MaxValRangeTransitive policy enables implementations to use LDAP limits to configure the maximum number of objects retrievable with msds-TokenGroup* family constructed attributes in a single LDAP search. Added Note indicating the operating systems that support this feature.

Changed From:

<table>
<tr><th>Policy name</th><th>Default value</th><th>Description</th></tr>
</table>
</td></tr>
</table>

Errata Published*	Description

MaxValRangeTransitive	4500	The maximum number of values that can be retrieved from one of the following multivalued, constructed attributes in a single search request: 3.1.1.4.5.19 - tokenGroups, tokenGroupsNoGCAcceptable 3.1.1.4.5.42 - msds-tokenGroupNames, msds-tokenGroupNamesNoGCAcceptable 3.1.1.4.5.43 - msds-tokenGroupNamesGlobalAndUniversal 3.1.1.4.5.20 - tokenGroupsGlobalAndUniversal This policy is effective only when the fTreatTokenGroupsAsLDAPTransitiveAttribute dsHeuristic is TRUE (section 6.1.1.2.4.1.2). This policy is significant on Windows 10 v19H1 and later and Windows Server v19H1 and later. It otherwise has no significance.

Changed To:

Policy name	Default value	Description
MaxValRangeTransitive	4500	The maximum number of values that can be retrieved from one of the following multivalued, constructed attributes in a single search request: 3.1.1.4.5.19 - tokenGroups, tokenGroupsNoGCAcceptable 3.1.1.4.5.42 - msds-tokenGroupNames, msds-tokenGroupNamesNoGCAcceptable 3.1.1.4.5.43 - msds-tokenGroupNamesGlobalAndUniversal 3.1.1.4.5.20 - tokenGroupsGlobalAndUniversal This policy is effective only when the fTreatTokenGroupsAsLDAPTransitiveAttribute dsHeuristic is TRUE (section 6.1.1.2.4.1.2). Note: The ability to use LDAP limits to configure the maximum number of objects retrievable by the msds-TokenGroup* family constructed attributes, is supported in Windows 11 v22H2 and later, and in the operating systems specified in [MSKB-5011543], [MSFT-KB-5011551], [MSKB-5011558],

Errata Published*	Description

		and [MSKB-5011563], each with the corresponding KB package installed. This policy is significant on Windows 10 v19H1 and later and Windows Server v19H1 and later. It otherwise has no significance.

Section 6.1.1.2.4.1.2 dSHeuiristics

The dSHeuristic 27 'fTreatTokenGroupsAsLDAPTransitiveAttribute' enables implementations to use LDAP limits to configure the maximum number of objects retrieved by the msds-TokenGroup* family constructed attributes in a single LDAP search request. Added Note indicating the operating systems that support this feature.

Changed From:

Character number	Character name	Description
27	fTreatTokenGroupsAsLDAPTransitiveAttribute	If this character is "0" (or not set), then the fTreatTokenGroupsAsLDAPTransitiveAttribute heuristic is false; otherwise, the fTreatTokenGroupsAsLDAPTransitiveAttribute heuristic is true. This heuristic applies to Windows 10 v19H1 and later and Windows Server v19H1 and later. This heuristic also applies only to the number of values returned by the following constructed attributes: 3.1.1.4.5.19 - tokenGroups, tokenGroupsNoGCAcceptable 3.1.1.4.5.42 - msds-tokenGroupNames, msds-tokenGroupNamesNoGCAcceptable 3.1.1.4.5.43 - msds-tokenGroupNamesGlobalAndUniversal 3.1.1.4.5.20 - tokenGroupsGlobalAndUniversal If fTreatTokenGroupsAsLDAPTransitiveAttribute is false, then the number of values returned is defined by the "MaxValRange" LDAP policy, as defined in section 3.1.1.3.4.6. If fTreatTokenGroupsAsLDAPTransitiveAttribute is true, then the number of values returned is defined by

<table>
<tr><th>Errata Published*</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><td></td><td></td><td>the “MaxValRangeTransitive” LDAP policy, as defined in section 3.1.1.3.4.6.</td></tr>
</table>

Changed To:
<table>
<tr><th>Character number</th><th>Character name</th><th>Description</th></tr>
<tr><td>27</td><td>fTreatTokenGroupsAsLDAPTransitiveAttribute</td><td>If this character is "0" (or not set), then the fTreatTokenGroupsAsLDAPTransitiveAttribute heuristic is false; otherwise, the fTreatTokenGroupsAsLDAPTransitiveAttribute heuristic is true.

This heuristic applies to Windows 10 v19H1 and later and Windows Server v19H1 and later. This heuristic also applies only to the number of values returned by the following constructed attributes:

3.1.1.4.5.19 - tokenGroups, tokenGroupsNoGCAcceptable

3.1.1.4.5.42 - msds-tokenGroupNames, msds-tokenGroupNamesNoGCAcceptable

3.1.1.4.5.43 - msds-tokenGroupNamesGlobalAndUniversal

3.1.1.4.5.20 - tokenGroupsGlobalAndUniversal

If fTreatTokenGroupsAsLDAPTransitiveAttribute is false, then the number of values returned is defined by the “MaxValRange” LDAP policy, as defined in section 3.1.1.3.4.6.

If fTreatTokenGroupsAsLDAPTransitiveAttribute is true, then the number of values returned is defined by the “MaxValRangeTransitive” LDAP policy, as defined in section 3.1.1.3.4.6.

Note: The ability to use LDAP limits to configure the maximum number of objects returned by the msds-TokenGroup* family constructed attributes, is supported in Windows 11 v22H2 and later, and in the operating systems specified in [MSKB-5011543], [MSKB-5011551], [MSKB-</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Errata Published*</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><td></td><td></td><td>5011558], and [MSKB-5011563] each with the corresponding KB package installed.</td></tr>
</table>
</td></tr>
<tr><td>2022/02/08</td><td>
Section 3.1.1.3.4.6 LDAP Policies

Updated the LDAP policy table by revising the MaxValRangeTransitive policy and description and specifying the operating system applicability.

Changed from:

<table>
<tr><th>Policy name</th><th>Default value</th><th>Description</th></tr>
<tr><td>MaxValRangeTransitive</td><td>none</td><td>This policy has no significance.</td></tr>
</table>

Changed to:

<table>
<tr><th>Policy name</th><th>Default value</th><th>Description</th></tr>
<tr><td>MaxValRangeTransitive</td><td>4500</td><td>The maximum number of values that can be retrieved from one of the following multivalued- constructed attributes in a single search request:

3.1.1.4.5.19 - tokenGroups, tokenGroupsNoGCAcceptable
3.1.1.4.5.42 - msds-tokenGroupNames, msds-tokenGroupNamesNoGCAcceptable
3.1.1.4.5.43 - msds-tokenGroupNamesGlobalAndUniversal
3.1.1.4.5.20 - tokenGroupsGlobalAndUniversal

This policy is effective only when the fTreatTokenGroupsAsLDAPTransitiveAttribute dsHeuristic is TRUE (section 6.1.1.2.4.1.2).

This policy is significant on Windows 10 v1903 operating system and later and Windows Server v1903 operating system and later. It has no significance otherwise.</td></tr>
</table>

Section 6.1.1.2.4.1.2 dSHeuristics

Updated the Character table in this section by adding heuristic characters fLoadV1AddressBooksOnlySetting (26) and fTreatTokenGroupsAsLDAPTransitiveAttribute (27) to the Heuristics string. Also, specified the operating system applicability.

Changed from:
</td></tr>
</table>

Character number	Character name	Description
24-25	MinimumGetChangesReplyVersion	A hex value, ranging from "00" to "FF". This value controls the minimum version of the DRS_MSG_GETCHGREPLY* structures the DC will send or accept. If the value is not set, the value "00" is used. When the value is "00", no restriction is enforced. See [MS-DRSR] section 4.1.10.5.20.

Changed to:

Character number	Character name	Description
24-25	MinimumGetChangesReplyVersion	A hex value, ranging from "00" to "FF". This value controls the minimum version of the DRS_MSG_GETCHGREPLY* structures the DC will send or accept. If the value is not set, the value "00" is used. When the value is "00", no restriction is enforced. See [MS-DRSR] section 4.1.10.5.20.
26	fLoadV1AddressBooksOnlySetting	If this character is "0", then the fLoadV1AddressBooksOnlySetting heuristic is false; otherwise, the fLoadV1AddressBooksOnlySetting heuristic is true. If fLoadV1AddressBooksOnly is true, then the hierarchy table used to support the MAPI address book is calculated using V1 attributes only, which means ignoring the V2 attributes "addressBookRoots2" and "templateRoots2". If fLoadV1AddressBooksOnly is false, then those V2 attributes are used. This heuristic applies to Windows 10 v1903 operating system and later and Windows Server v1903 operating system and later.
27	fTreatTokenGroupsAsLDAPTransitiveAttribute	If this character is "0" (or not set), then the fTreatTokenGroupsAsLDAPTransitiveAttribute heuristic is false; otherwise, the fTreatTokenGroupsAsLDAPTransitiveAttribute heuristic is true. This heuristic applies to Windows 10 v1903 and later and Windows Server v1903 and later. This heuristic also applies only to the number of values

Errata Published*	Description
	<table><tr><td></td><td></td><td>returned by the following constructed attributes: 3.1.1.4.5.19 - tokenGroups, tokenGroupsNoGCAcceptable 3.1.1.4.5.42 - msds-tokenGroupNames, msds-tokenGroupNamesNoGCAcceptable 3.1.1.4.5.43 - msds-tokenGroupNamesGlobalAndUniversal 3.1.1.4.5.20 - tokenGroupsGlobalAndUniversal If fTreatTokenGroupsAsLDAPTransitiveAttribute is false, then the number of values returned is defined by the "MaxValRange" LDAP policy defined in section 3.1.1.3.4.6. If fTreatTokenGroupsAsLDAPTransitiveAttribute is true, then the number of values returned is defined by the "MaxValRangeTransitive" LDAP policy defined in section 3.1.1.3.4.6.</td></tr></table>
2022/01/11	The following sections were changed. Please see the diff document for the details. In section 1.2, added informative reference to [MSFT-CVE-2022-21857], "NTLM Pass-through Authorization Vulnerability." In section 3.1.1.6, Background Tasks, added information about querying and persisting data about trusted forests. Changed from: • Maintain security descriptor requirements (see Security Descriptor Requirements in section 6.1.3). Changed to: • Maintain security descriptor requirements (see Security Descriptor Requirements in section 6.1.3). • Query and persist domain information about trusting forests (see section 3.1.1.6.4). In section 3.1.6.4, PDC Forest Trust Update, added new section. In section 3.1.6.4.1, Informative Overview, added new section. In section 3.1.6.4.2, Logical Processing, added new section.

<table>
<tr><th>Errata Published*</th><th>Description</th></tr>
<tr><td></td><td>
In section 6.1.5.4, PDC Emulator FMSO Role, added text about the need to periodically query state on trusted forests.

Changed from:

● The PDC emulator FSMO also fulfills the role of the PDC in the NetLogon Remote Protocol methods described in [MS-NRPC] section 3. Therefore, the PDC emulator FSMO MUST support and perform all PDC specific functionality specified in that section. Every DC, other than the PDC emulator FSMO, MUST NOT perform this functionality.

Changed to:

● The PDC emulator FSMO also fulfills the role of the PDC in the NetLogon Remote Protocol methods described in [MS-NRPC] section 3. Therefore, the PDC emulator FSMO MUST support and perform all PDC specific functionality specified in that section. Every DC, other than the PDC emulator FSMO, MUST NOT perform this functionality.
● The PDC emulator periodically queries state about trusting forests and stores it in the msdsForestTrustInfo attribute (see section 3.1.1.6.4).

In section 6.1.6.9.3.1, Record, added two new record type values, and added packet diagram and processing rules for the ForestTrustScannerInfo record type. See the diff doc.

In section 6.1.6.9.6.2, PDC Forest Trust Scanning, added new section.
</td></tr>
<tr><td>2022/01/04</td><td>
In Section 6.1.1.2.4.1.2 dSHeuristics

Description: In the characters table for the dSHeuristics string, changed index number 27 for character name 'AttributeAuthorizationOnLDAPAdd' to index number 28 and changed index number 28 for character name 'BlockOwnerImplicitRights' to index number 29.

Changed From:

<table>
<tr><td>Character number</td><td>Character name</td></tr>
<tr><td>27</td><td>AttributeAuthorizationOnLDAPAdd</td></tr>
<tr><td>28</td><td>BlockOwnerImplicitRights</td></tr>
</table>

Changed To:

<table>
<tr><td>Character number</td><td>Character name</td></tr>
<tr><td>28</td><td>AttributeAuthorizationOnLDAPAdd</td></tr>
<tr><td>29</td><td>BlockOwnerImplicitRights</td></tr>
</table>
</td></tr>
<tr><td>2021/11/11</td><td>
▪ Updated UserAccountControl attributes for computer and set default to UF_WORKSTATION_TRUST_ACCOUNT, if not, Add operation returns ERROR_DS_SECURITY_ILLEGAL_MODIFY.
</td></tr>
</table>

Errata Published*	Description
	▪ Added additional authorization checks for LDAP add-object and modify-object operations. ▪ Addded information on improved UPN and SPN uniqueness checks, including SPN alias checks. **Section 3.1.1.3.4.1 LDAP Extended Controls** **Changed From**: LDAP_SERVER_SD_FLAGS_OID — Instructs the **DC** which portions of a **Windows security descriptor** to retrieve during an **LDAP** search operation. **Changed To**: LDAP_SERVER_SD_FLAGS_OID — Instructs the **DC** which portions of a **Windows security descriptor** to either retrieve during an **LDAP** search operation or to set during an **LDAP** modify operation.[1] [1] The **Extended Control Name** LDAP_SERVER_SD_FLAGS_OID impacts the portions of the Windows security descriptor to retrieve during an LDAP search or to set during an LDAP modify operation, as supported on the operating systems specified in **[MSFT-CVE-2021-42291]**; each with its related MSKB article download installed. **Section 3.1.1.3.4.1.11 LDAP_SERVER_SD_FLAGS_OID** Revised to clarify that the LDAP_SERVER_SD_FLAGS_OID control is used with LDAP Modify requests, while not used with LDAP Add requests. **Changed From**: "It is also used with LDAP Add and Modify requests to control the portion of a Windows security descriptor to modify. The DC modifies only the specified portion of the security descriptor." **Changed To**: "It is also used with LDAP Modify requests to control the portion of a Windows security descriptor to modify.[2] The DC modifies only the specified portion of the security descriptor. [2]Clarified the use of the LDAP_SERVER_SD_FLAGS_OID control with respect to LDAP Modify requests on the operating systems specified in **[MSFT-CVE-2021-42291]**, each with its related MSKB article download installed. **Section 3.1.1.5.1.3 Uniqueness Constraints** Updated the user Principle Name (UPN) and service principle name (SPN) uniqueness checking feature along with the adding new DS Heuristics characters. **Changed From**: "... The following additional considerations for uniqueness checking are relevant for winblue_server_1 with [MSKB-3070083] and winthreshold_server and later: • userPrincipalName uniqueness is not checked if the DoNotVerifyUPNAndOrSPNUniqueness character of the dsHeuristics attribute (see section 6.1.1.2.4.1.2) is set to 1. • servicePrincipalName uniqueness is not checked if the DoNotVerifyUPNAndOrSPNUniqueness character of the dsHeuristics attribute is set to "2". • Neither userPrincipalName nor servicePrincipalName uniqueness is checked if the DoNotVerifyUPNAndOrSPNUniqueness character of the dsHeuristics attribute is set to 3. • userPrincipalName and servicePrincipalName uniqueness is checked if the DoNotVerifyUPNAndOrSPNUniqueness character of the dsHeuristics attribute is set to any value other than 1, 2, or 3." **Changed To**: "... The following additional considerations for uniqueness checking are relevant:[3] • userPrincipalName uniqueness is checked only if bit 0 of the DoNotVerifyUPNAndOrSPNUniqueness dsHeuristic attribute (section 6.1.1.2.4.1.2) is set to 1.

Errata Published*	Description
	• servicePrincipalName uniqueness is checked only if bit 1 of the DoNotVerifyUPNAndOrSPNUniqueness dsHeuristic attribute value (section 6.1.1.2.4.1.2) is set to 1. • servicePrincipalName alias uniqueness is checked only if bit 2 of the DoNotVerifyUPNAndOrSPNUniqueness dsHeuristic attribute value (section 6.1.1.2.4.1.2) is set to 1 and if the current user is not admin or local system. **Note**: sPNMappings are defined in [MS-ADA3] section 2.276. The format of an entry is x=a,b,c. In this context, a, b, and c are all aliases of x. The first part of a servicePrincipalName is the SERVICE, for example SERVICE/foo. When the servicePrincipalName alias uniqueness feature is on the new value SERVICE, the name must be unique, including its aliases. For example if CIFS is an alias of HOST, then setting the servicePrincipalName to CIFS/foo will actually check uniqueness for both CIFS/foo and HOST/foo. [3] The uniqueness checking additions are relevant to Windows Server 2012 R2 with **[MSKB-30700083]** installed and to the operating systems specified in **[MSFT-CVE-2021-42282]**, each with its related MSKB article download installed. **Section 3.1.1.5.2.1 Security Considerations** Updated to indicate that security considerations include satisfying the constraints specified in section 3.1.1.5.2.2. **Changed From**: "For regular object creation, the requester must have RIGHT_DS_CREATE_CHILD on the parent object for the objectClass of the object being added." **Changed To**: "For regular object creation, the requester must have RIGHT_DS_CREATE_CHILD on the parent object for the objectClass of the object being added, and MUST also satisfy the constraints specified in section 3.1.1.5.2.2.[4] [4] Specified additional constraints to apply to regular object creation, as supported by the operating systems specified in **[MSFT-CVE-2021-42291]**, each with its related MSKB article download installed. **Section 3.1.1.5.2.1.1 Per Attribute Authorization for Add Operation** Created new topic to describe how to authorize attributes for the Add operation. **Changed From**: "" **Changed To**: "If AttributeAuthorizationOnLDAPAdd is 0 or 2, this check succeeds with no further processing. If AttributeAuthorizationOnLDAPAdd is 1, processing proceeds as follows: 1. If the requester is a member of either Domain Administrators (section 6.1.1.6.5) and Enterprise Administrators (section 6.1.1.6.10), this check succeeds with no further processing. 2. If the objectClass being added is neither computer or a class derived from computer, this check succeeds with no further processing, otherwise proceed. [5] 3. Let DefaultSD be a security descriptor created per the algorithm specified in section 3.1.2. If the requester submitted an nTSecurityDescriptor attribute as part of the add request, that attribute MUST be excluded for the purpose of creating DefaultSD. 4. Check if the requester is granted explicit WRITE_DAC permission on DefaultSD. Explicit means that WRITE_DAC must be granted due to the presence of least one access-allowed ACE in SD, and not due to the requester being an Owner in DefaultSD. 5. If the requester is granted explicit WRITE_DAC permission on DefaultSD, this check succeeds with no further processing. 6. If the requester is not granted explicit WRITE_DAC permission on DefaultSD, and the requester submitted an nTSecurityDescriptor attribute as part of the add request, and implicit Owner rights are blocked as specified in section 6.1.3.4, the server returns an error. 7. Let A be the set of attributes included in the requester's add request. Remove from A any attributes which are configured in the schema as either systemMustContain or mustContain attributes for the object class being created.

Errata Published*	Description
	8. Remove from A the unicodePwd or userPassword attributes if present. 9. If A is empty, this check succeeds with no further processing. 10. If A is non-empty, perform an access check operation against DefaultSD as if the requester was trying to modify the attributes contained in A, using the steps specified in section 3.1.1.5.3.1. If this access check fails, the server returns an error. 11. If processing reaches this point with no server errors, the check succeeds. [5] This new process for authorizing attributes for the Add operation is supported by the operating systems specified in **[MSFT-CVE-2021-42291]**, each with its related MSKB article download installed. **Section 3.1.1.5.2.2 Constraints** Clarified the constraints that apply when a computer object is being created and the requester has RIGHT_DS_CREATE_CHILD access. **Changed From**: "... If the attribute is servicePrincipalName and its value does not conform to the requirements stated in section 3.1.1.5.3.1.1.4, the Add operation returns ERROR_DS_INVALID_ATTRIBUTE_SYNTAX." **Changed To**: "... If the attribute is servicePrincipalName and its value does not conform to the requirements stated in section 3.1.1.5.3.1.1.4, the Add operation returns ERROR_DS_INVALID_ATTRIBUTE_SYNTAX. If the object being created is a computer object and the requester has RIGHT_DS_CREATE_CHILD access, the following constraints apply:[6] • If the userAccountControl attribute is not specified, then the default bit will be set to UF_WORKSTATION_TRUST_ACCOUNT. • If the userAccountControl attribute is specified and does not contain UF_USER_NORMAL_ACCOUNT, UF_USER_INTERDOMAIN_TRUST_ACCOUNT, UF_USER_WORKSTATION_TRUST_ACCOUNT, or UF_USER_SERVER_TRUST_ACCOUNT, then the default bit will be set to UF_WORKSTATION_TRUST_ACCOUNT. • If the userAccountControl attribute is specified and does not contain UF_WORKSTATION_TRUST_ACCOUNT or UF_SERVER_TRUST_ACCOUNT, the Add method returns ERROR_DS_SECURITY_ILLEGAL_MODIFY. [6] The constraints that apply when a computer object is being created and the requester has RIGHT_DS_CREATE_CHILD access, are supported by the operating systems specified in **[MSFT-CVE-2021-42278]**, each with its related MSKB article download installed. **Section 5.1.3.3.1 Null vs. Empty DACLs** Describes the conditions under which specific operating systems MUST ignore the implicit WRITE_DAC grant. Added behavior note to specify that Windows Server 2008 SP2 and later operating systems, as specified in [MSFT-CVE-2021-42291], MUST ignore the implicit WRITE_DAC grant for purposes of the authorization check." **Changed From**: "An empty DACL, on the other hand, is a properly allocated and initialized DACL containing no ACEs. An empty DACL in the nTSecurityDescriptor attribute of an object grants no access to the object. Note that even with an empty DACL, some rights are implied. For example, the current OWNER of an object is implicitly granted RIGHT_READ_CONTROL and RIGHT_WRITE_DAC access. If the user possesses the SE_TAKE_OWNERSHIP_PRIVILEGE, then RIGHT_WRITE_OWNER access is implied." **Changed To**: "An empty DACL, on the other hand, is a properly allocated and initialized DACL containing no ACEs. An empty DACL in the nTSecurityDescriptor attribute (2) of an object (5) grants no access to the object (5). Note that even with an empty DACL, some rights are implied. For example, the

Errata Published*	Description
	current OWNER of an object (5) is implicitly granted RIGHT_READ_CONTROL and RIGHT_WRITE_DAC access. When BlockOwnerImplicitRights is set to 1 and the requester is a member of neither the Domain Administrators (section 6.1.1.6.5) or Enterprise Administrators (section 6.1.1.6.10) group. The implicit grant of WRITE_DAC MUST be ignored for purposes of the authorization check.[7] If the user possesses the SE_TAKE_OWNERSHIP_PRIVILEGE, then RIGHT_WRITE_OWNER access is implied." [7]Under the stated conditions, the implicit WRITE_DAC grant MUST be ignored for purposes of the authorization check on computers running the operating systems specified in **[MSFT-CVE-2021-42291]**, each with the related MSKB article download installed. **Section 6.1.1.2.4.1.2 dSHeuristics** "Updated character 21 'DoNotVerifyUPNAndOrSPNUniqueness' in dsHeuristics table to specify how bit values of this heuristic determine whether UPN and SPN are checked for uniqueness in AD LDS and AD DS. **Changed From**: "In AD LDS, if this character is anything other than "0", AD LDS will not check values of userPrincipalName for uniqueness. See section 3.1.1.5.2.2. In AD LDS, this heuristic applies to Windows Server 2003 and later." In AD DS, if this character is "1", "2" or "3", AD DS will not check values of userPrincipalName or servicePrincipalName for uniqueness. See section 3.1.1.5.1.3. In AD DS, this heuristic applies to Windows Server 2012 R2 with **[MSKB-3070083]** and Windows Server 2016 and later." **Changed To**: "In AD LDS, if this character is anything other than "0", AD LDS will not check values of userPrincipalName for uniqueness. See section 3.1.1.5.2.2. In AD LDS, this heuristic applies to Windows Server 2003 and later. The following applies to AD DS only: This heuristic value is converted to an unsigned integer and the result is interpreted as a bitwise OR. This heuristic applies to Windows Server 2012 R2 operating system with [MSKB-3070083] installed.[8] Bit 2 is supported with values between 0 and 7. Otherwise, only Bit 0 and 1 are supported, meaning supported values are between 0 and 3. The heuristic value is interpreted as follows, with Bit 0 as the lower bit: Bit 0: AD DS will check values of userPrincipalName (UPN) for uniqueness only if this bit is set (section 3.1.1.5.1.3). Bit 1: AD DS will check values of servicePrincipalName (SPN) for uniqueness only if this bit is set (section 3.1.1.5.1.3). Bit 2: AD DS will check values of SPN (1) for alias uniqueness only if this bit is set (section 3.1.1.5.1.3). [8]In AD DS, the DoNotVerifyUPNAndOrSPNUniqueness heuristic also applies to the operating systems specified in **[MSFT-CVE-2021-42282]**, each with its related MSKB article download installed. **Section 6.1.1.2.4.1.2 dsHeuristics** Added new dsHeuristic Characters 'AttributeAuthorizationOnLDAPAdd' (27) and 'BlockOwnerImplicitRights' (28) and descriptions to the dsHeuristics table to support the procedure in section 3.1.1.5.2.1.1. **Changed From**: "" **Changed To**:

27	AttributeAuthorizationOnLDAPAdd	If this character is "0", "1", or "2", the AttributeAuthorizationOnLDAPAdd heuristic[9] is set to the equivalent numeric value (0, 1, or 2). If this character is not set, the AttributeAuthorizationOnLDAPAdd heuristic defaults to 0. If this character has any other value, the AttributeAuthorizationOnLDAPAdd heuristic defaults to 1. See section 3.1.1.5.2.1.1.
28	BlockOwnerImplicitRights	If this character is "0", "1", or "2", the BlockOwnerImplicitRights heuristic[10] is set to the equivalent numeric value (0, 1, or 2). If this character is not set, the BlockOwnerImplicitRights heuristic defaults to 0. If this character has any other value, the BlockOwnerImplicitRights heuristic defaults to 1. See sections 3.1.1.5.2.1.1 and 3.1.1.5.3.1.

[9, 10] These heuristics are supported by the operating systems specified in **[MSFT-CVE-2021-42291]**, each with the related MSKB article download installed.

Section 6.1.3.4 Blocking Implicit Owner Rights

Created new section to describe the conditions when implicitly granted rights are blocked to the owner of a security descriptor.

Changed From:

""

Changed To:

"The Owner of a security descriptor is implicitly granted READ_CONTROL and WRITE_DAC rights by default. For servers running specific operating systems[11], these implicit rights are blocked when the following are true:

- The BlockOwnerImplicitRights dsHeuristic is set to 1 (section 6.1.1.2.4.1.2).
- The requester is a member of neither the Domain Administrators (section 6.1.1.6.5) or the Enterprise Administrators (section 6.1.1.6.10) group.
- The objectClass being added or modified is either of type computer or is derived from type computer.

[11] For servers running the operating systems specified in **[MSFT-CVE-2021-42291]**, each with the related MSKB article download installed, implicit rights granted by default to the owner of the security descriptor are blocked when the specified conditions are true.

Section 6.1.3.5 Security Considerations

Clarified requirements that MUST be satisfied when a DACL value is written according to SD flags.

Changed From:

"When an add operation is processed, the client is allowed to specify any SD value, subject to some constraints to the OWNER field, specified in this section.

When a modify operation is processed, the following security checks are applied to the requester's security context. If the requester does not pass the check, then accessDenied is returned.

1. If the DACL value is written (according to SD flags), then one of the following requirements must be satisfied:

    1. RIGHT_WRITE_DAC is granted to the requester on the object.

    2. The OWNER SID in the SD value is one of the SIDs in the requester's token (either as user SID or group SID)."

Changed To:

Errata Published*	Description
	"When an add operation is processed, the client is allowed to specify any SD value, subject to some constraints to the OWNER field, specified in this section and in section 3.1.1.5.2.1.1. When a modify operation is processed, the following security checks are applied to the requester's security context. If the requester does not pass the check, then accessDenied is returned. 1. If the DACL value is written (according to SD flags), then one of the following requirements MUST be satisfied:[12] • Explicit RIGHT_WRITE_DAC is granted to the requester on the object. • The OWNER SID in the SD value is one of the SIDs in the requester's token (either as user SID or group SID), in which case, implicit Owner rights are not blocked, as specified in section 6.1.3.4. [12] Clarified requirements that MUST be satisfied when a DACL value is written according to SD flags, as supported on operating systems that are specified in **[MSFT-CVE-2021-42291]**, each with the related MSKB article download installed.

*Date format: YYYY/MM/DD