msDS-SupportedEncryptionTypes – Episode 1 - Computer accounts

learn.microsoft.com/en-us/archive/blogs/openspecification/msds-supportedencryptiontypes-episode-1-computer-accounts

- Article
- 09/12/2009

Introduction

In order to be concise with this article, I need to assume that the reader is familiar with Kerberos and Active Directory.

If not, then I can quickly think of two scenarios:

1) Your favorite search engine (Bing in my case) took you here as a misunderstanding.

2) You came because you stumbled upon the name “msDS-SupportedEncryptionTypes” somewhere and you really really want to understand what it is related to, even if you need to learn about Kerberos and Active Directory as a pre-requisite.

Let’s assume (for the sake of the posting) that the option you fall into is #2 and that you are eager to know where to find the docs that are inherent to this article.

Here are the links:

The Kerberos Network Authentication Service (V5): https://www.ietf.org/rfc/rfc4120.txt

[MS-KILE]: Kerberos Protocol Extensions: https://msdn.microsoft.com/en-us/library/cc233855.aspx

[MS-ADA2]: Active Directory Schema Attributes M: https://msdn.microsoft.com/en-us/library/cc220154.aspx

[MS-ADA3]: Active Directory Schema Attributes N-Z: https://msdn.microsoft.com/en-us/library/cc220699.aspx

[MS-ADTS]: Active Directory Technical Specification: https://msdn.microsoft.com/en-us/library/cc223122.aspx

Juicy information

In order for the KDC to be able to generate tickets that the target server can read, there has to be some mean of communicating what type of encryptions the involved actors can understand.

For quite a while, that was not an issue because the older versions of Windows that had Kerberos5 implementations (Windows 2000 all flavors, Windows XP and Windows 2003 all flavors) only supported DES (RFC3961) and RC4 (RFC4757) as the methods of encryption.

However, Windows Vista, Windows Server 2008, Windows 7 and Windows Server 2008 R2 incorporated the newer and more secure algorithm AES (RFC3962) (128 AND 256). With this new addition, and with so many machines running previous versions of Windows, it was imperative to have a way to inform which algorithms each particular account could handle and to make sure that when newer algorithms should become available they would not necessarily represent many changes.

msDS-SupportedEncryptionTypes came up as the solution. This AD attribute (defined in MS-ADA2, section 2.324) is present in the Computer, User and Trust objects for Schema version 44 (Windows 2008) and later. Its sole purpose is to hold the values of the encryption types that the account owner supports.

Well, I guess that you could have deducted that from the name of the attribute so; I better go a little deeper.

Its size is 4 bytes, its type is Unsigned Integer and its format is a Bit Mask. The values that it can accept, are defined in the following table (and originally in MS-KILE section 3.1.5.4):

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

0	0	0	0	0	0	0	0	0	0	0	0	0	0	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	DES-CBC-CRC
B	DES-CBC-MD5
C	RC4-HMAC
D	AES128-CTS-HMAC-SHA1-96
E	AES256-CTS-HMAC-SHA1-96

This is an example of how it looks like in AD (formatted per RFC2989):

CN=server2008,CN=Computers,DC=testdomain,DC=com

…

msDS-SupportedEncryptionTypes: 0x1F = (DES_CBC_CRC | DES_CBC_MD5 | RC4_HMAC_MD5 | AES128_CTS_HMAC_SHA1_96 | AES256_CTS_HMAC_SHA1_96);

Although this attribute is present in all the computer objects of the domain regardless of the version of the OS the physical machines have installed, not all of them are aware of its existence hence, older versions (2003 and earlier) do not populate it at any time.

Legacy systems leave it Blank, NULL, Zeroed or Empty at all times. However, more recent versions of the OS do specify a value that could be any combination of the first 5 bits being 31 (0x1F) the default. The initial set up of the value happens at domain join time with subsequent updates via LDAP should the algorithms that the machine supports change for any reason.

When the KDC checks the attribute to decide what encryption algorithm to use in order to encrypt the ticket, it could find basically two scenarios:

1) The attribute is populated

2) The attribute is empty

If the attribute is populated, then the deal is done since the KDC can determine the best common algorithm to encrypt the ticket with the value present.

However, if the attribute is empty then the KDC will have to work harder being the next step to check another attribute. This attribute is defined in MS-ADA3 (section 2.341) and described in MS-ADTS (section 2.2.15) and it's called userAccountControl. This attribute is also a 4 byte Bit Mask that defines many aspects of the account but the only one the KDC is interested in is the DK (ADS_UF_USE_DES_KEY_ONLY) bit.

This bit defines what legacy encryption method will be used:

1) If the bit is set, then only DES will be used

2) If the bit is NOT set, then DES and RC4 can be used

This check is especially relevant in domains that have Win7 and Windows Server 2008 R2 machines joined because those two newer OSs disable their bit by default so older DES is not an option for them.

Conclusion

Windows 2008, Vista, Windows 7 and Windows 2008 R2 have expanded the options available when securing resources and communications on the network.

Having the msDS-SupportedEncryptionTypes attribute is a good starting point to further incorporating newer and more secure encryption algorithms in the future.

Comments

- **Anonymous**
  April 07, 2011
  I've read that Windows Server 2008 R2 and Windows 7 machines objects have the msDS-SupportedEncryptionTypes attribute set to 0x1C (RC4_HMAC_MD5 | AES128_CTS_HMAC_SHA1_96 | AES256_CTS_HMAC_SHA1_96). However, in my VM environment, on a Windows Server 2008 R2 domain controller in a Windows Server 2008 R2 domain functional level the attribute is set to 0x1F = (DES_CBC_CRC | DES_CBC_MD5 | RC4_HMAC_MD5 | AES128_CTS_HMAC_SHA1_96 | AES256_CTS_HMAC_SHA1_96). Also, the HKLMSoftwareMicrosoftWindowsCurrentVersionPoliciesSystemKerberosparameters – SupportedEncryptionTypes value is not present on the registry. However, it looks like the server does not list DES as a supported etype: Kerberos TGS-REQ Record Mark: 1649 bytes Pvno: 5 MSG Type: TGS-REQ (12) padata: PA-TGS-REQ KDC_REQ_BODY rc4-hmac Padding: 0 KDCOptions: 40810000 (Forwardable, Renewable, Canonicalize) Realm: DEV.DOM Server Name (Service and Instance): cifs/ChildDC02.Dev.Dom Name-type: Service and Instance (2) Name: cifs Name: ChildDC02.Dev.Dom till: 2037-09-13 02:48:05 (UTC) Nonce: 1803734880 Encryption Types: aes256-cts-hmac-sha1-96 aes128-cts-hmac-sha1-96 rc4-hmac rc4-hmac-exp rc4-hmac-old-exp Encryption type: aes256-cts-hmac-sha1-96 (18) Encryption type: aes128-cts-hmac-sha1-96 (17) Encryption type: rc4-hmac (23) Encryption type: rc4-hmac-exp (24) Encryption type: rc4-hmac-old-exp (-135) Encryption type: rc4-hmac (23) enc-authorization-data: ac3a97d047efab59693ad96ef2781c5398784b446ceae4b3... Is the value of the msDS-SupportedEncryptionTypes attribute not taken into account?

- **Anonymous**
  April 07, 2011
  Hi, If you need assistance on Windows-based Kerberos configuration, the Directory Services forums would be the best place to help: social.technet.microsoft.com/.../threads However, the following blog post could complement and help answer your questions. Encryption Type Selection in Kerberos Exchanges blogs.msdn.com/.../encryption-type-selection-in-kerberos-exchanges.aspx Our team owns this blog and supports issues related to open specifications documentation available at: msdn.microsoft.com/.../dd208104.aspx. Currently, there are two options available to engage our team and get support on the open specifications:

1. either you send the question to dochelp <-at-> microsoft <-dot->com;
2. or you post the question on one of the open specifications forums under social.msdn.microsoft.com/.../openspecifications Thanks, Edgar

- **Anonymous**
  July 08, 2011
  Why is this attribute necessary however? Doesnt the Kerberos protocol itself have a built-in mechanism to negotiate acceptable encryption mechanisms: - the client specifies what it can support in it's AS-REQ - the server picks up the highest one it supports out of that list, returns KRB5KDC_ERR_ETYPE_NOSUPP if none of these are supported? thanks, -Ravi

- **Anonymous**
  October 02, 2012
  @Ravi, the client indicates to the KDC, but this thing is for the target server.