Decrypting the Selection of Supported Kerberos Encryption Types

techcommunity.microsoft.com/t5/core-infrastructure-and-security/decrypting-the-selection-of-supported-kerberos-encryption-types/ba-p/1628797

undefined

In recent months Microsoft support has received a lot of questions regarding disabling RC4 for the encryption of Kerberos tickets. If I had to guess the CIS L1 Baseline and RFC 8429 guidance to disable RC4 is likely responsible for much of that interest. While RC4 has not been formally deprecated in Active Directory, the evolution of an attack known as Kerberoasting provides a compelling reason to upgrade given RC4 encryption uses the weak NTLM hash as the key for encryption. To date tickets encrypted with AES keys are not susceptible to Kerberoasting.

As with many hardening settings, the decision to eliminate RC4 for Kerberos ticket encryption is not entirely cut and dry. Let's a take a look at the considerations and then you can decide how you want to move forward with improving your security posture in this area.

First a little history

When Active Directory was first introduced, DES and RC4 were all the rage. In time computational advancements made it possible to brute force attack DES encrypted tickets in a short amount of time and RFC 6649 called for the retirement of DES. Even before RFC 6649 was formally published, Microsoft disabled (by default) DES with the release of server 2008 R2 Windows 7. If you were supporting Active Directory in 2009, you most likely did not even notice DES had been disabled by your newly upgraded domain controllers because Active Directory is designed to select the highest level of encryption that is supported by the client and target of a Kerberos ticket. Support for AES ticket encryption was introduced with the release of Server 2008 / Windows 7 but it was not automatically enabled on domain accounts in order to ensure backward compatibility.

Kerberos 101 Refresher

Before we dive into the compatibility concerns, we need to make sure we are not being too generic with our terminology. Here is a quick refresher.

Authenticator – Even back with Windows for Workgroups (Where are my 3.11 people?) it was uncool to send a clear password over the wire. Active Directory avoids that by encrypting the system time with a derived version of the password. The output of that function produces what is called the authenticator (aka pre-auth data). When the DC receives the authenticator, it looks up the account password (aka Long-Term Key), decrypts the authenticator and compares the result to its own time. If the timestamps match within 5 minutes, it knows the correct password was used and that a replay attack is very unlikely.

Ticket Granting Ticket (TGT) – The domain controller will return a TGT to the account once the authenticator has been validated. Inside the TGT is the SID of the account, SIDs of the account's groups and a **session key**, along with some other security stuff. The TGT is only read by domain controllers from the domain where it was issued. To keep it private the TGT is encrypted with the password of the **KRBTGT** domain account. As a result, the contents of the TGT cannot be read by the client.

Session Key – When the account receives the TGT it also receives a copy of the session key (symmetric). To keep the key safe while crossing the network it is encrypted with the account's password. Once decrypted the session key is placed in LSA (Local Security Authority) memory along with the TGT. Going forward the account's password is no longer required. When the client makes subsequent ticket requests it will present the TGT and creates a new authenticator using the session key and the system timestamp. The domain controller will then use the KRBTGT password to decrypt the TGT, extract the session key then decrypt the authenticator. To be clear, every ticket has a unique session key and the domain controller does not attempt to remember each session key. Once it is done with a session key it will discard it. When it needs the key again it will repeat the process of extracting it from the presented TGT.

Service Ticket – When an account wants to access a resource it will request a service ticket from the domain controller by providing the name of the resource, its copy of the TGT and an authenticator generated based on the TGT session key. Assuming the authenticator is valid, and the requested name can be matched to a security

principle, the domain controller will construct the requested service ticket by copying the account's SIDs from the TGT, a new session key and encrypt it with a derived password of the security principle. In some cases, the password will be a computer account password. In other cases, it will be the password of a service account used to host the resource. Like with the TGT, the client will not be able to read the service ticket and will be securely sent a copy of the session key for the ticket.

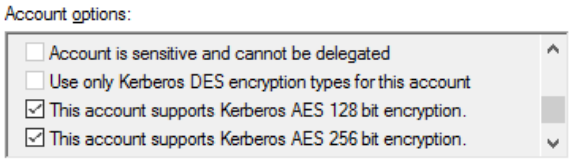
Referral Ticket – When a user is attempting to access a resource in another domain, a service ticket from a domain controller in the resource's domain must be acquired. That is accomplished by submitting a referral ticket request to a domain controller of the user's domain. The client provides its TGT, a fresh authenticator and the FQDN of the remote resource. The FQDN will let the domain controller know in which trusted domain the resource resides. It will then create the referral ticket which contains the user's SIDs and a session key. The referral ticket is then encrypted with a key derived from the domain trust password and returned to the client. The client forwards the referral ticket to a domain controller in the remote domain and requests a service ticket for the resource. If everything is correct a service ticket is returned to the client along with a session key associated with that ticket.

Bringing it all together

Now that the Kerberos flow is fresh in our minds, we can break down the considerations for disabling RC4.

Authenticator encryption type – Sometimes a client will include an authenticator with the initial TGT request (KRB_AS_REQ) in which case it will simply declare which encryption it decided to use base on the configuration of the OS. Other times the client will ask for a TGT without providing an authenticator to which the domain controller will respond with a KDC_ERR_PREAUTH_REQUIRED message along with a list of encryption types it supports. Either way the client and domain controller must be able to agree on a supported encryption type. As documented in this article, Server 2000, Server 2003 and XP do not support either version of AES. Therefore, if you have those legacy operating systems still in your domain you are not ready to remove RC4 support from your domain controllers.

TGT encryption type – As mentioned before, a TGT is only read by domain controllers in the issuing domain. As a result, the encryption type of the TGT only needs to be supported by the domain controllers. Once your domain functional level (DFL) is 2008 or higher, you KRBTGT account will always default to AES encryption. For all other account types (user and computer) the selected encryption type is determined by the **msDS-SupportedEncryptionTypes** attribute on the account. You can modify the attribute directly or you can enable AES using the checkboxes in the Account tab.



The **msDS-SupportedEncryptionTypes** attribute uses a single HEX value to define which encryption types are supported. You could calculate the value based on this article or you could use the following decoder ring:

Decimal Value	Hex Value	Supported Encryption Types
0	0x0	Not defined - defaults to RC4_HMAC_MD5
1	0x1	DES_CBC_CRC
2	0x2	DES_CBC_MD5
3	0x3	DES_CBC_CRC, DES_CBC_MD5
4	0x4	RC4

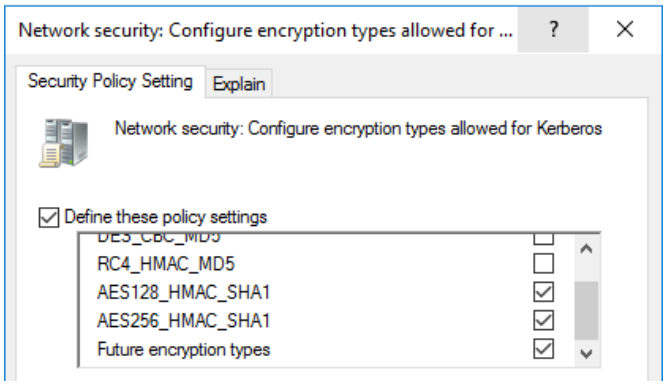
5	0x5	DES_CBC_CRC, RC4
6	0x6	DES_CBC_MD5, RC4
7	0x7	DES_CBC_CRC, DES_CBC_MD5, RC4
8	0x8	AES 128
9	0x9	DES_CBC_CRC, AES 128
10	0xA	DES_CBC_MD5, AES 128
11	0xB	DES_CBC_CRC, DES_CBC_MD5, AES 128
12	0xC	RC4, AES 128
13	0xD	DES_CBC_CRC, RC4, AES 128
14	0xE	DES_CBC_MD5, RC4, AES 128
15	0xF	DES_CBC_CBC, DES_CBC_MD5, RC4, AES 128
16	0x10	AES 256
17	0x11	DES_CBC_CRC, AES 256
18	0x12	DES_CBC_MD5, AES 256
19	0x13	DES_CBC_CRC, DES_CBC_MD5, AES 256
20	0x14	RC4, AES 256
21	0x15	DES_CBC_CRC, RC4, AES 256
22	0x16	DES_CBC_MD5, RC4, AES 256
23	0x17	DES_CBC_CRC, DES_CBC_MD5, RC4, AES 256
24	0x18	AES 128, AES 256
25	0x19	DES_CBC_CRC, AES 128, AES 256
26	0x1A	DES_CBC_MD5, AES 128, AES 256
27	0x1B	DES_CBC_MD5, DES_CBC_MD5, AES 128, AES 256
28	0x1C	RC4, AES 128, AES 256
29	0x1D	DES_CBC_CRC, RC4, AES 128, AES 256

30	0x1E	DES_CBC_MD5, RC4, AES 128, AES 256
31	0x1F	DES_CBC_CRC, DES_CBC_MD5, RC4-HMAC, AES128-CTS-HMAC-SHA1-96, AES256-CTS-HMAC-SHA1-96

If you enable AES on the KRBTGT account and find your TGTs are still issued with RC4 encryption you may need to manually reset the password of the KRBTGT account. That is due to the fact that the KRBTGT password does not automatically rotate. As a result, the current password may have been set back in the 2003 days when AES key generation was not supported. If you need to update your password I recommend you leverage this script. In fact, it is recommended to reset it a second time after waiting a minimum of 10 hours (default TGT lifetime) so there is an AES key in the password history attribute.

Session Key encryption type – The client supported encryption type is similar to the authenticator encryption type in that it is dependent on the configuration of the client OS and is declared during the ticket request (KRB_AS_REQ). The session key selected for the TGT must be compatible with the client and the domain controllers of the issuing domain. The session key selected for a service ticket must be compatible with the client and the server hosting the resource. When selecting a compatible session key the KDC will evaluate the client request and the **msDS-SupportedEncryptionTypes** attribute of the target account.

Service Ticket encryption type – When a service ticket is requested, the domain controller will select the ticket encryption type based on the **msDS-SupportedEncryptionTypes** attribute of the account associated with the requested SPN. As mentioned before, this may be a computer object, or it could be a service account that is being used to host the resource on the network. If the attribute has no value defined, the domain controller will encrypt the ticket with RC4 to ensure compatibility. By default, user accounts do not have a value set so unless you have manually enabled AES on them, tickets for service accounts will be encrypted with RC4. For computer objects you can directly update **msDS-SupportedEncryptionTypes** or you apply a GPO to define the supported encryption types. Once the computer processes that policy it will update the attribute on its own computer object.



Referral Ticket encryption type – The encryption used for a referral ticket and session key is determined by the trust properties and the encryption types supported by the client. If you select **The other domain supports AES Encryption**, referral tickets will be issued with AES. Otherwise the referral ticket will be encrypted with RC4. By default, trusts (including inter-forest trusts) do not have AES support enabled. When deciding to enable AES on a trust keep in mind the client does not read the contents of the referral ticket, but it does need a common session key encryption type. If you are considering disabling RC4 over a trust please first review KB4492348. As pointed out by Daniele's blog , enabling AES with the Active Directory Domains and Trusts GUI will disable RC4 across the trust but using **ksetup** will allow you to add AES support without disabling RC4.

*** **Update** *** The November 2022 update changed the logic for referral ticket encryption. As a result it is no longer necessary to manually enable AES for trusts. For more details see Active Directory Hardening Series - Part 4 – Enforcing AES for Kerberos - Microsoft Community Hub

Auditing for encryption type

In my role as Sr Customer Engineer I find the fear of the unknown to be the primary reason security hardening recommendations are not embraced. Moving forward with enforcing AES for Kerberos will require analysis and one of the best inputs for that assessment are 4769 events from the domain controller security log which show the encryption type (Ticket Encryption Type field) of issued service tickets. Event 4768 will show the same information for issued TGTs. If you have the luxury of having centralized log collection and analysis tool, then getting a quick handle on your ticket encryption types will be achievable. Without such a solution you are facing a tough challenge. The table below maps the values in the events to the encryption type of the issued tickets.

Type Value	Encryption Type Used
0x1	DES-CBC-CRC
0x3	DES-CBC-MD5
0x11	AES128-CTS-HMAC-SHA1-96
0x12	AES256-CTS-HMAC-SHA1-96
0x17	RC4-HMAC
0x18	RC4-HMAC-EXP

Event ID 16 can also be useful when troubling scenarios where a service ticket request failed because the account did not have an AES key.

Do's and Don'ts of RC4 disablement for Kerberos Encryption Types

That was a lot of information on a complex topic. Here is a quick summary to help you determine your next move.

- **Don't** disable RC4 across your domain without performing a thorough assessment unless you have recently updated your resume.
- **Don't** confuse this information with guidance and settings for disabling RC4 for TLS\SSL (Schannel). See this MSRC blog if you still need to disable RC4 in TLS.
- **Don't** wait for RC4 disablement to be forced on you. Start making sure AES has been fully enabled on your computers, accounts and trusts. Once that is done leverage central log collection and analyze your 4769 events to determine if RC4 tickets are still being issued
- **Do** enable AES on service accounts which have a SPN set. Keep in mind that a null value for **msDS-SupportedEncryptionTypes** will cause the DC to issue service tickets and session keys with RC4
- **Do** reset service account passwords for accounts which do not have AES keys. Passwords set before 2008 do not have AES keys. Pro Tip: The domain group **Read-only Domain Controllers** creation date will tell you when the first domain controller newer than 2003 was promoted in your domain. Using PowerShell, search your domain for user accounts with a SPN set that have **pwdLastSet** older than when your group Read-only Domain Controllers was created
- **Do** confirm your TGTs are encrypted with AES. If they are still being issued with RC4 check the **pwdLastSet** attribute on the KRBTGT account and determine if it is newer than the created date of your **Read-Only Domain Controllers** group.
- **Do** understand that Kerberoasting makes it trivial for an attacker to determine your weak service account passwords when issued a service ticket encrypted with RC4. Prioritize your privileged service accounts when setting strong passwords and enabling AES for ticket encryption. Kerberoasting can be performed offline once a service ticket has been acquired so this is not an area to rely on your EDR solution.
- **Don't** forget about remediating your KeyTab files. When you enable AES on a service account used with an existing KeyTab file, it may be necessary to generate new file. Unfortunately, many organizations do not have a good inventory of issued KeyTab files so remediating them could be challenging
- **Do** use **4768** events to identify devices that are dependent on RC4. For more details check out my follow-up article Active Directory Hardening Series - Part 4 – Enforcing AES for Kerberos - Microsoft Community Hub

- **Do** learn how to use **klist** to view the encryption type used for tickets and session keys. If you have UAC enabled you will see different results depending on if you launched the command prompt with elevation. That is because technically you have two sessions running which means two different sets of tickets. If you want to the view tickets issued to the system rather than your account run **klist –li 0x3e7** from an elevated prompt.
- **Do** remember that ticket encryption only needs to be compatible with the account opening the ticket. The session key selected needs to be compatible with both sides of the connection.
- **Do** retire legacy operating systems (Server 2003 and older) which are not compatible with AES encrypted tickets

Thanks for reading. I hope this information helps you move forward with eliminating RC4 encryption without unexcepted impacts.

Jerry Devore , Sr Customer Engineer

24 Likes
Like

54 Comments

I thought i knew something about kerberos and then this blog happened.

But this is super helpful as we have undertaken a project to get rid of all accounts which are still requesting for Service tickets using RC4. Thanks.

3 Likes
Like

@JerryDevore brilliant post and is actually pretty succinct for such a complex topic. Been through parts of this process when upgrading a large domain from FFL/DFL 2003 to 2008. It was six months of planning, communicating, monitoring, reporting & testing. But ultimately successful.

Observations & recommendations:

1. Don't forget about non-Windows clients that maybe using AD for Kerberos auth. e.g. Linux, NAS & Network gear
2. Centralised logging is worth the effort and will ultimately save you time for this and other projects - have a look at native WEF (windows Event Forwarding) and Power BI analysis, you can use this blog post by Jessica Payne as a starter - Monitoring what matters - Windows Event Forwarding for everyone (even if you already have a SIEM.) |...
3. Don't forget your blank root domains (remember when those were best-practice)
4. For a FFL/DFL upgrade there really is no viable fall-back you will need to fix and move forward; you need to explain this to management
5. Following on from the last point whilst in practice it is highly unlikely it would be invoked you should know how to do a forest level authoritative restore
6. If you an MS TAM engage with them, have PFE workshops and share the schedule so that a rapid response engineer can be briefed if the worst happens
7. Document, document, document; use this as an exercise to really understand your environment, build relations with app owners and inventory (you'll probably uncover practices that you weren't aware and will scare you)

   Paul

3 Likes
Like

The link to the krbtgt script is broken.

The link to Daniele's blog is also broken, but I was able to at least find it here: https://techcommunity.microsoft.com/t5/itops-talk-blog/tough-questions-answered-can-i-disable-rc4-et...

0 Likes

Like


JerryDevore
Microsoft

Thanks for the heads up Chris. The links have been fixed.

0 Likes

Like

Thanks for an amazing article! After reading it, it was simple to make a plan and implement it without any problems (lucky to not use keytabs in that environment). Now really small number of RC4 are left.

My question: Once we hunt them down and see that AES is used for all TGT and TGS everything to AES, how do we disable RC4 completely? What will happen if we configure **msDS-SupportedEncryptionTypes** to only support AES on **krbtgt** account?

The reason for the questions is obvious - now we have cleaned up everything and it looks good. Even if we configure all existing accounts to only support AES and no RC4, there's no guarantee that some time later someone would create a new service account without specifying **msDS-SupportedEncryptionTypes** attribute or just joins another non-Windows device into domain, which may allow all encryption algorithms by default. Do we need to constantly monitor for such accounts, or can we turn off RC4 once and for all?

Also maybe this fresh article would be nice to mention in your post, as it describes some border cases which may be useful for troubleshooting:
https://syfuhs.net/lessons-in-disabling-rc4-in-active-directory
1 Like
Like

RossUA - Congrats on nearly eliminating RC4. To ensure RC4 encryption for Kerberos does not make a comeback in your environment you should disable support for it in the operating system of your devices. Group policy is one way you can do that using the Network security: Configure encryption types allowed for Kerberos setting. That setting will modify HKLM\Software\Microsoft\Windows\CurrentVersion\Policies\System\Kerberos\Parameters\SupportedEncryptionTypes. If you still have legacy clients that do not support AES you would not want to apply that setting domain wide but you could use it harden modern devices that do not interact with the legacy systems.

The krbtgt account is a bit of a special case in that when it does not have a value defined for **msDS-SupportedEncryptionTypes** the KDC will still encrypt TGTs with AES. Configuring krbtgt to only support AES before you eliminated the legacy devices will be problematic given those devices are dependent on RC4 session keys.

1 Like
Like

** Bonus Material **

Here is a quick query you can use to determine the **msDS-SupportedEncryptionTypes** attribute value for all accounts with a SPN set (i.e. Kerberos enabled service accounts). msDS-SupportedEncryptionTypes is not a Global Catalog attribute by default so if you have a multi-domain forest you will need to run the query against a DC in each domain. If you have a more eloquent query feel free to share it via a comment.

get-aduser -filter {(objectclass -eq 'user')} -property servicePrincipalName,pwdLastSet,description,displayName,msDS-UserPasswordExpiryTimeComputed,userAccountControl,msDS-PrincipalName,msDS-SupportedEncryptionTypes,lastLogonTimestamp | where-Object {$PSItem.ServicePrincipalName -ne $null} | select-

object servicePrincipalName,userPrincipalName,displayName,distinguishedName,description,pwdLastSet,msDS-UserPasswordExpiryTimeComputed,userAccountControl,msDS-PrincipalName,msDS-SupportedEncryptionTypes,lastLogonTimestamp | Export-Csv -Path .\SPNEncryptionData.csv -NoTypeInformation

5 Likes
Like

Shajeer
Copper Contributor

Brilliant... this doc is great

1 Like
Like

KQL query if using Sentinel or some other log collector:

SecurityEvent | where EventID in (4768, 4769) | where EventData contains '<Data Name="TicketEncryptionType">0x17</Data>' or EventData contains '<Data Name="TicketEncryptionType">0x18</Data>'

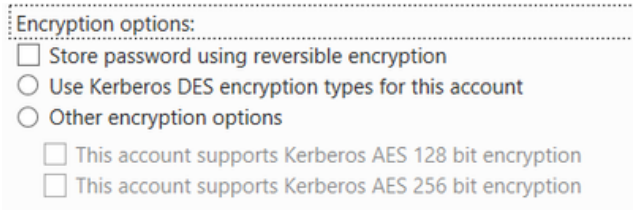
0 Likes

Like

Great article, thank you.

I'm troubleshooting the use of RC4. The account and device have **msDS-SupportedEncryptionTypes** values supporting AES.

The KRBTG account does not have enabled AES.



Would checking those boxes block the use of RC4 (not ready for that) or allow AES. Meaning, could something break from this, or am I allowing the use of newer encryption options in addition to the RC4. Would be grateful for any input.

0 Likes

Like

Hi @sintra3000 - Enabling AES on the KRBTGT account would not disable RC4 in your environment. From my testing it does not appear the KDC references the msDS-SupportEncryptionTypes of the KRBTGT account when issuing TGTs. If the account has a AES key (password reset since 2008) it will issue AES encrypted TGTs. That behavior could depend on the DFL but I has not researched that.

If you are unable to acquire an AES service ticket for an account that has AES enabled in the msDS-SupportEncryptionTypes attribute I would first confirm the password on the account was not last set prior to the elimination of 2003 (and earlier) domain controllers.

1 Like
Like

Hi Jerry,

Thank you your reply, that cleared some things up. I have verified that the account has a recent pwdLastSet.

The cloud app security report still is reporting the account as using weak cipher.

0 Likes

Like

if your domain controllers are server 2019 or higher, it is not necessary to modify **msDS-SupportedEncryptionTypes settings anywhere.** as soon as RC4 is disabled on the domain controller (Network security: Configure encryption types allowed for Kerberos) no RC4 tickets will be issued, ever. it does not matter if the checkbox "this account supports aes 256...." checkboxes are set or not, and it does not matter if computer accounts have their bit flags updated or not.

here is a klist example in a hardenend domain (encryption types allowed for kerberos set to AES on domain controllers only), no checkbox set on service account, i.e. RC4 is enabled, AES is not enabled

still, we receive aes tickets only:

#1> Client: darr @ LAB.LOCAL
Server: MSSQLSvc/lab004s.lab.local:1433 @ LAB.LOCAL
KerbTicket Encryption Type: AES-256-CTS-HMAC-SHA1-96
Ticket Flags 0x40a10000 -> forwardable renewable pre_authent name_canonicalize
Start Time: 10/21/2021 18:52:09 (local)
End Time: 10/22/2021 4:52:09 (local)
Renew Time: 10/28/2021 18:52:09 (local)
Session Key Type: AES-256-CTS-HMAC-SHA1-96
Cache Flags: 0
Kdc Called: LAB001DC.lab.local

this is different in previous generations (server 2016, server 2012R2). we have verified this in multiple lab domains, customer domains and our own domain. this fact is also documented here: https://dev-2null.github.io/Kerberoasting-AES-Encryption-Protected-Users-Group-and-gMSA/

we also ran rubeus downgrade attacks against nonhardenend and hardened environments and confirm the same observation.

so in short, you do NOT need to worry about modifying service accounts or computer accounts to harden them against kerberos rc4. just harden the DC with the security setting provided above and all is fine.

0 Likes

Like

may i also point out that this statement is incorrect:

"To date tickets encrypted with AES keys are not susceptible to Kerberoasting."

kerberos aes 256 tickets can be kerberoasted just fine with hashcat since march 2019

https://github.com/hashcat/hashcat/pull/1955

... it just takes a lot longer (about 750times when switching from RC4 to AES256)

0 Likes

Like

@robertro-sit - Thanks for sharing your testing and research. Many enterprises have pockets of legacy OSs (2003\XP) as well as old keytab files which were not created to support AES. If RC4 support was removed from the DCs those devices would be impacted which is why I recommended leveraging audit logs to uncover and resolve

RC4 use rather than completely disabling it at the domain level. In environments without legacy dependencies you could certainly save considerable time by removing RC4 support from the DCs if the organization is willing to accept some risk of impact.

I was not aware that hashcat can support kerberoasting AES tickets. I will edit my phrasing. That bit of information highlights the importance of setting long and complex (non-guessible) passwords on service accounts in addition to enabling them for AES. Too often that is not the case because service accounts have been made exempt from password changes and as a result the current passwords were set when the organization had much more lax password standards. Personally I am fan of leveraging fine grained password policy to hold service accounts to a higher standard than the average user account. Of cource the PSO would not be enforced until the service account passwords are cycled.

Your correct about server 2019 addressing the issue with ticket down grade attacks. That improvement alone is great justification to get domain controllers up to 2019 (or 2022) as soon as possible. If the DC upgrades cannot happen in the near term, Microsoft Defender for Identity is a good mitigation given it will alert you when such ticket ecryption downgrade are occurring.

2 Likes
Like

It seems the "decoder ring" table contains some errors. For example, for value of 1 it says the encryptions is "**DES**_DES_CBC_CRC" (extra DES there), for value of 15 (0xF) "DES_CBC_MD5" appears twice (I assume one of them should be DES_CBC_CRC), etc..

2 Likes
Like

Thanks Jerry, i am still watching this topic and the conversation. Its interesting observation from @robertro-sit . In summary, to identify the legacy RC4 traffic we leverage the Audit logs and once its clean the settings can be applied Network security: Configure encryption types allowed.

Open question:

1. Should this settings be applied on DCs only or other computer objects as well (Servers & Computers)

2. Does a keytab file leveraging RC4 is identified or captured in the Auditlog 4768, 4769 some way?

Thanks

shajeer

0 Likes

Like

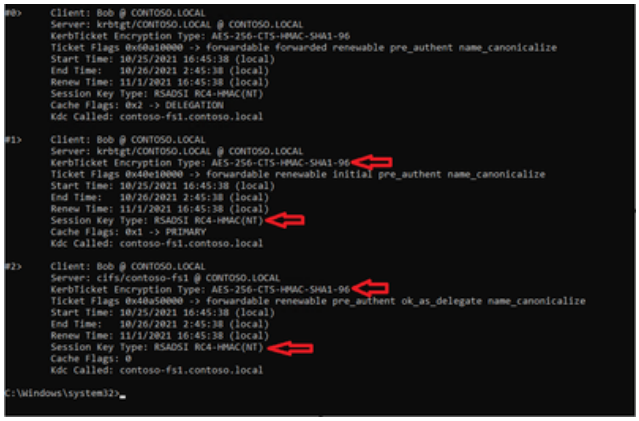
Great catch Shnitzel! - I have made the table corrections.

Shaz_Blog - Once logging confirms tickets are no longer issued with RC4 you will want to use the "**Configure encryption types allowed**" policy to remove RC4 support for all domain members and not just the domain controllers. After a Windows device processes that policy it will dynamically update the **msDS-SupportedEncryptionTypes** attribute on its own computer account.

If the keytab file is only being used to consume a service ticket (no authentication back to the domain), the 4768/4769 events will be no help in identifying keytabs which only support RC4. However, if the keytab is being used to acquire TGTs from the KDC, the **Ticket Encryption Type** field in the **4768** will reflect encryption type used to perform the pre-authentication rather than the encryption used for the TGT. I know that might not seem right so here is an example from my lab.

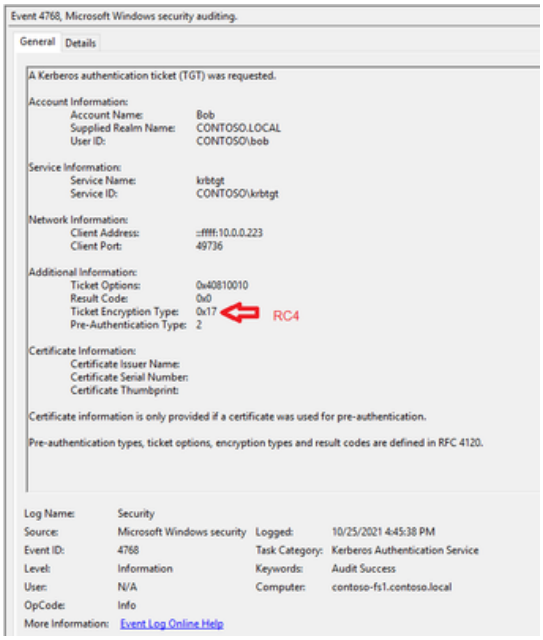
I used the "**Configure encryption types allowed**" policy to only allow RC4 on a Windows 10 computer. When Bob logged on, he received an AES encrypted TGT but because his device only supported RC4 during pre-authentication the session key for the TGT was RC4. As you can see the 4768 showed the Ticket Encryption Type

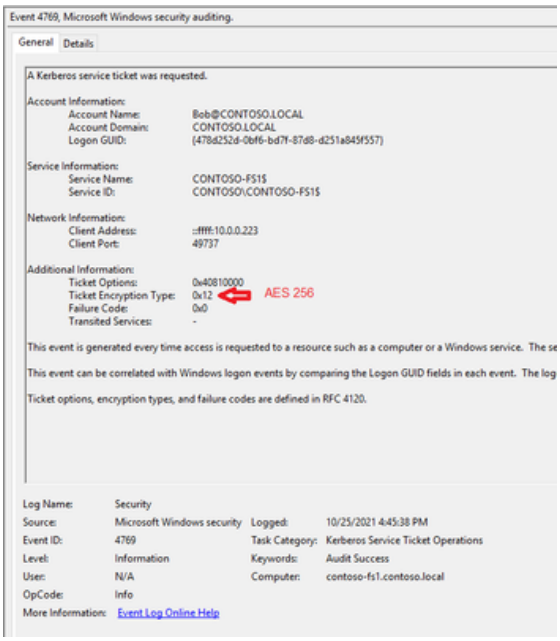
field logged RC4 (session key\pre-auth) rather than AES which was the encryption type of the TGT. However, the 4769 actually reflected the encryption type of the Service Ticket rather than the session key encryption type.

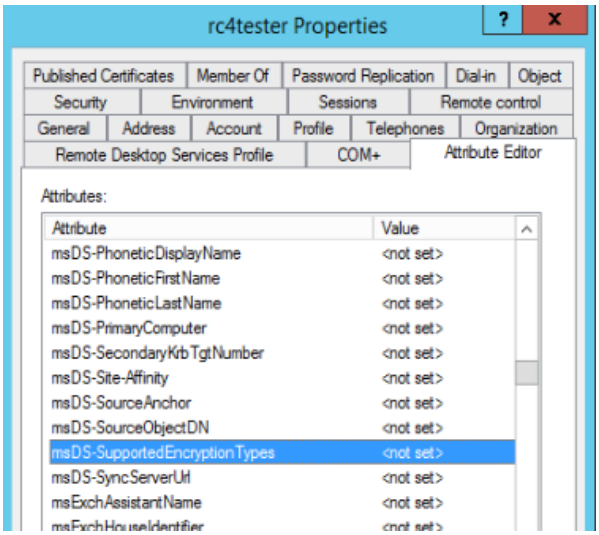


3 Likes
Like

Thanks again @JerryDevore .. Helpful.
In this whole excersice of eliminating RC4 from the environment, we are dealing with Computer Objects (via Policy), TDO update etc... There is this AD User objects also carrying the similar attribute (**msDS-SupportedEncryptionTypes**). By default a user object being created is having value <**not set**>, meaning that it could accept RC4 only. However, i see that tickets are getting generated on AES for those accounts. In the whole process of eliminating RC4, Is there any action to be done on the user objects?

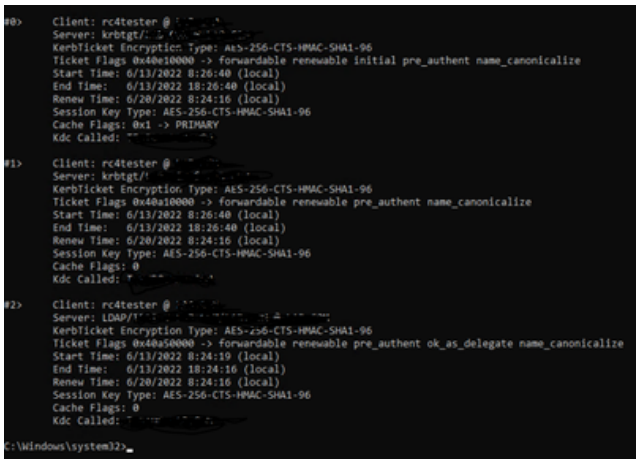






Decimal Value	Hex Value	Supported Encryption Types
0	0x0	Not defined - defaults to RC4_HMAC_MD5

Just logged in as rc4tester and i am getting tickets on AES when checked using klist.



2 Likes
Like

Hi Shaz_Blog - The ticket encryption type (TGT and Service Tickets) will be determined by the account linked to the tickets so their selected ecryption type has no bearing on the configuration of the user account requesting the tickets. The session key used with those tickets are determined by configuration of the device where the inital pre-authentication originated from along with the encryption type supported by the account associated with the ticket. The KDC will issue the session keys based on the highest supported encryption type of those two. The KDC does not factor in the **msDS-SupportedEncryptionTypes** of the account requesting the tickets. In your example **rc4tester** logged on to a device that is capable of AES so the session keys it recieved were AES.

Bottom line: You don't need to worry about the value of **msDS-SupportedEncryptionTypes** on user accounts that do not have a SPN. If it would make your security team feel better if normal user accounts were configured for AES you could set a value on the attribute but it will not make any difference.

3 Likes
Like

@JerryDevore I've applied the "**Configure encryption types allowed**" policy to remove RC4 support for all the domain controllers. On the Domain Controllers container.

Did Gpupdate /force, restarted and waited for days on end yet the **msDS-SupportedEncryptionTypes** wasn't updated on the DC computer account object.

But when I did it on the Default Domain policy it worked.

Why do you suppose is the issue here and why wasn't it updated?

(Tried on 2 different domains) Running Server 2016. Same thing happened for some reason.

Edit : Works now, comment can be deleted.

1 Like
Like

@JerryDevore

Congratulations on the content presented.

But I was left with a doubt, my goal at the company I work for is just to disable the DES encryption algorithm. (we have Windows XP machines - Windows Server 2003) so I can't disable RC4.

I pulled a report and identified 208 machines with msDS-SupportedEncryptionTypes attribute with value 0x1F(31)

performing a comparison with my station the value is 0x1C (28)

To disable only the DES algorithm, I suggest you create a GPO allowing only RC4, AES ?

thanks

0 Likes

Like

Hi @GuilhermeFranklin

Using a GPO to only allow RC4 (disable DES) on your legacy devices is a good strategy. Once the devices have processed the GPO you will notice the **msDS-SupportedEncryptionTypes** attribute is automatically updated on the respective computer objects. The policy setting for that GPO is "**Network security: Configure encyrption types allowed for Kerberos**" which you will find under Local Policies \ Security Options.

You should also confirm that DES has not been enabled on any user accounts in your domain. A quick way to perform that check is to run **Get-ADUser -Filter {UserAccountControl -band 0x200000}** form a PowerShell prompt. For more information on that step check out Remove the highly insecure DES encryption from User accounts (recommended) | Microsoft Learn

Jerry

1 Like
Like

Following on from the advice in KB5021131: How to manage the Kerberos protocol changes related to CVE-2022-37966 (microsoft.com) about the November 2022 Kerberos changes for RC4 and the "quick query" posted earlier, this is a method to find accounts with the DES-only account flag or RC4 encryption type set explicitly (remembering that, at present, RC4 is still the default type when msDS-SupportedEncryptionTypes is set to **0**).

```
#query properties into array for convenience
$props = "servicePrincipalName","pwdLastSet","description","displayName","distinguishedName","msDS-
UserPasswordExpiryTimeComputed","userAccountControl","msDS-PrincipalName","msDS-
SupportedEncryptionTypes","lastLogonTimestamp"
# query only users with SPNs, and with either DES-only UAC flag or RC4 specified in encryptiontypes
get-aduser -filter 'sAMAccountType -eq 805306368 -and servicePrincipalName -like "*" -and
(useraccountcontrol -band 0x200000 -or msDS-SupportedEncryptionTypes -band 0x4)' -properties $props |
select-object $props
```

To break it down a bit:

- Rather than returning ALL user accounts and discarding the ones without an SPN in the Where clause , the **servicePrincipalname -like "*"** filter clause returns *only* accounts from LDAP with an SPN.
- **useraccountcontrol -band 0x200000** filters for accounts with the "Use only Kerberos Des encryption types" flag
- **msDS-SupportedEncryptionTypes -band 0x4** filters accounts that have RC4 specifically enabled. Since it's a bitwise filter, the query will find accounts with only RC4, or RC4 along with other encryption types.

This is the output from a test account with RC4, AES128 and AES256 encryption types enabled (this example account doesn't have an SPN, so I omitted the '-and servicePrincipalName -like "*"' filter clause).

```
servicePrincipalName : {}
pwdLastSet : 133124575903507250
description :
displayName : test crypt
distinguishedname : CN=test crypt,OU=Generic,DC=dev,DC=example,DC=com
msDS-UserPasswordExpiryTimeComputed : 133202335903507250
userAccountControl : 512
msDS-PrincipalName : DEV\test_crypt
msDS-SupportedEncryptionTypes : 28
lastLogonTimestamp :
```

Note that msDS-SupportedEncryptionTypes is **28** on the account, which indicates the AES128 and AES256 types as well as RC4 (refer to the table in the article). If we change the bitwise filter to **msDS-SupportedEncryptionTypes -band 0x10** (for AES256), the same account will also be found.

Here is the list of bitwise values to query each encryption type (from Ldapwiki: MsDS-SupportedEncryptionTypes 😞

```
0x01 - DES-CBC-CRC
0x02 - DES-CBC-MD5
0x04 - RC4-HMAC
0x08 - AES128-CTS-HMAC-SHA1-96
0x10 - AES256-CTS-HMAC-SHA1-96
```

If you want to search combos of these, such as accounts that have AES128 **and** AES256 enabled, the hex codes for those in the table in the main article will work too, in this instance **-band 0x18** (this also finds my test account, which has both those encryption types enabled along with RC4).

0 Likes

Like

The Nov 8 update added a new registry key DefaultDomainSupportedEncTypes to define what msDS-SupportedEncryptionType undefined now is, which is 0x27 which enables DES-CBC-CRC,DES-CBC-MD5,DES-CBC-MD5 plus something else, but does not enable AES128/256. This means those of us who hardened to set only AES128/256 via SupportedEncryptionTypes have broken trusts now.

Edit: I was wrong about that new bit, there was an errata which has added AES256-CTS-HMAC-SHA1-96-SK (assuming SK stands for session Keys). From the sounds of it there's a bug somewhere else

0 Likes

Like

Hi guys. This is an excellent article and series of comments. I have a strange situation related to the msDS-SupportedEncryptionTypes attribute on the AD object of one of our Windows Servers joined to AD. I discovered through an audit that this sever had a value of 31. The desired value is 28. I changed it, but within half an hour it had changed back to 31. This happened multiple times. After I turned the server off this behavior stopped, which means it has to be something on the server itself that is changing it. Anyone have an idea on why this is happening? Thanks again!

0 Likes

Like

Hi Morgan,

Potentially your computer object settings are overwritten caused by the GPO. The msDS-SupportedEncryptionTypes attribute is impacted caused by the GPO.
https://learn.microsoft.com/en-us/windows/security/threat-protection/security-policy-settings/networ...

0 Likes

Like

@Shaz_Blog Thanks but would not a GPO affect settings on the machine itself, typically in the registry? What I am seeing is a change on the AD attribute of the computer object. Is there a known case where a local setting on a computer would cause it to change this attribute on its AD computer object? Also, I have no GPOs that apply to this particular server that change the **Network security: Configure encryption types allowed for Kerberos** policy.

0 Likes

Like

Yes, the GPO has an impact on the attribute of the Computer Object. @JerryDevore has explained that in this article itself... My guess is still towards a GPO, perhaps a GPResult can tick that out.

1 Like
Like

We have many user objects that have a 0 in it.

Also my user account - if i type "klist" i get as session and encryption type= AES-256-CTS-HMAC-SHA1-96

Do i have to act on these settings? Where may they come from? It seems like that these are the "older" AD objects.

0 Likes

Like

Hi @StephanGee - The session key selection is determined by which encryption type was used during the initial authentication (KRB_AS_REQ) which depends on the client devices setting and not the value of the user account's **msDS-SupportedEncryptionTypes**. The ticket's encryption type will be determine by the **msDS-SupportedEncryptionTypes** attribute for the account associated with the ticket (e.g. service account with a SPN set). Keep in mind the session key needs to be compatible with the client device and the device being authenticated to using the ticket. The ticket need to be compatible with the device consuming those tickets (the service you are accessing). The accounts with SPNs set (i.e. service accounts) are the user accounts you want to focus on.

0 Likes

Like

@JerryDevore So as i understand. At the moment i do not have to act on the things and fill in a 28 or even 24 into it? I will have a look into our SPNs

0 Likes

Like

Hi @JerryDevore ,

thanks for this very helpful post and your participation in the comments sections. Maybe you or someone else can bring some light to me on the following situation.

We've patched one of our DC's (Server 2019) with the 2022 November Update including the OOB update provided. Since the original update of the DC's users in this site are facing issues accessing on old NetApp filestore. Users from others sites, with unpatched 2019 DC's, are not facing issues accessing the file shares.

Looking at the kerberos tickets issued, we noticed, all DC's except the patched one issue Keberos tickets for this service with a RSADSI RC4-HMAC(NT) session key, whereas the patched DC is always issuing AES-256-CTS-HMAC-SHA1-96.

We've looked at the different registry keys and AD attributes and all are the same for all DC's.

Not working Ticket (2019 DC patched including OOB)

> #1> Client: XXX @ XXX
> Server: cifs/XXX@ XXX
> KerbTicket Encryption Type: RSADSI RC4-HMAC(NT)
> Ticket Flags 0x40a10000 -> forwardable renewable pre_authent name_canonicalize
> Start Time: 11/23/2022 2:41:54 (local)
> End Time: 11/23/2022 12:40:57 (local)
> Renew Time: 11/30/2022 2:40:57 (local)
> Session Key Type: AES-256-CTS-HMAC-SHA1-96
> Cache Flags: 0
> Kdc Called: s001dc01.XXX

Working ticket (2019 DC unpatched)

> #1> Client: XXX@ XXX
> Server: cifs/XXX @ XXX
> KerbTicket Encryption Type: RSADSI RC4-HMAC(NT)
> Ticket Flags 0x40a10000 -> forwardable renewable pre_authent name_canonicalize
> Start Time: 11/22/2022 19:36:37 (local)
> End Time: 11/23/2022 5:33:52 (local)
> Renew Time: 11/29/2022 19:33:52 (local)
> Session Key Type: RSADSI RC4-HMAC(NT)
> Cache Flags: 0
> Kdc Called: s049dc05.xxx

Any idea where can have a look to bring the DC back to issuing RC4 sessions keys?

Thanks in advance for your help.

0 Likes

Like

@FelixF we are troubleshooting an integration of NetApp with AD. After enabling AES encryption on NetApp (there is an article on how to do that), we noticed that TGT tickets are still encrypted with RC4 (we have 2016 DCs), but Service Tickets that NetApp is requesting are using AES encryption. Even after we set **msDS-SupportedEncryptionTypes** on krbtgt account to only support AES - it didn't have any impact.

According to the people managing NetApp, the article on NetApp page says "Even when AES encryption for Kerberos-based communication is enabled on vserver, Advertising the RC4 encryption type cannot be disabled". My guess is that by "advertising" the RC4 encryption type, they mean setting **msDS-SupportedEncryptionTypes** attribute on the computer object for NetApp storage. We have seen that after password reset, NetApp is writing **msDS-SupportedEncryptionTypes**attribute. We tried to manually change this attribute on the object, but it didn't help either.

In your case, please check that you have enabled AES encryption on NetApp storage. When you run the commands on NetApp, you need to have permissions to write password and **msDS-SupportedEncryptionTypes** attribute among others on the computer object for NetApp storage.
https://docs.netapp.com/us-en/ontap/smb-admin/enable-disable-aes-encryption-kerberos-task.html

If that config is already done, I'm curious what is the present value of **msDS-SupportedEncryptionTypes**attribute on the computer object for NetApp storage.

0 Likes

Like

@FelixF - The November update (11B.22) is causing the updated DCs to default to AES session keys. In some cases you can resolve the issue by explictly setting RC4 in **msDS-SupportedEncryptionTypes** for the computer objects of the target (NetApp server in this case). However, the recommended approach would be to enable AES on the NetApp server as @RossUA mentioned.

@RossUA - " TGT tickets are still encrypted with RC4" - Are you seeing the ticket encryption reported as RC4 for the TGT or just the session key? As you noticed setting **msDS-SupportedEncryptionTypes** for the KRBTGT account will not change the behavior for issued TGTs. If you are seeing TGT tickets encrypted with RC4 the most likely explanatoin is that the KRBTGT account has not been reset since AES was supported by the domain (2008R2).

0 Likes

Like

It would be unrestricted for Kerberos if the GPO for the "Network security: Configure encryption types allowed for Kerberos" policy is not defined.
Also, the msDS-SupportEncryptionTypes attribute for the KRBTGT not set(nothing).

Will the KRBTGT's encryption type remain the same as RC4 since we deployed the AD forest when it was Windows 2000 and raised the FL to the current? If so. What will happen to the KRBTGT's encryption with the following few months' patches since MS gradually changes to netlogon and Kerberos encryption? Will the patch change the encryption type to AES?

0 Likes

Like

Thank you @JerryDevore and @RossUA . Your help is much appriciated.

After changing **msDS-SupportedEncryptionTypes** to 0x4 (RC4_HMAC_MD5) , we're able to access the NetApp again. The session key now is RC4.

Thank you for your help, we'll check how to get rid of this configuration by either getting rid of the old NetApp or changing configuration of the NetApp to AES if possible.

0 Likes

Like

@JerryDevore we are only observing RC4 used for TGT tickets obtained by NetApp storage. It is reported in the event 4768 as Ticket Encryption Type 0x17. The vast majority of other computer and user accounts are using AES encryption for TGT tickets for a long time already. And we have changed the password for krbtgt several times

during last 5 years.

Can you confirm if my understanding is correct. If the goal is to prevent Kerberoasting attack, then RC4 encryption in TGT is not a problem, since it is TGS ticket which is being attacked. And as long as all sensitive accounts with SPNs are configured with **msDS-SupportedEncryptionTypes** that only allows AES encryption and strong password, kerberoasting attack should not be successful. Does it make sense?

0 Likes

Like

I have noticed that when I run the script I get a report that 8 accounts do not have the msDS-SupportedEncryptionTypes set. However, when I manually spot check, NONE of the user accounts have this field set. How should I interpret this?

0 Likes

Like

Hi @RossUA - The Ticket Encryption Type reported in 4768 events is a bit misleading. Through lab testing I have found that the field reflects the either the session key or the authenticator type used to acquire the TGT. When you see 0x17 for a 4768 it does not mean the ticket itself was encrypted with RC4. A 4769 (Service Ticket) will however correctly report the ticket encryption in the Ticket Encryption Type field.

Kerberoasting is targeted at Service Tickets but if the TGTs were issued with RC4 encryption and the KRBTGT account had a weak password TGTs would in theory be susceptible to the same attack. Fortunately, the automatically generated passwords for KRBTGT are very strong and as long as the KRBTGT account has an AES key the KDC will issue TGT with AES.

0 Likes

Like

Hi @MDZCOB - The script I posted (Comment on April 29 2021) is only looking for user accounts with SPNs set. When you did your spot check were you looking at SPN enabled account or any user accounts. The **msDS-SupportedEncryptionTypes** value for user accounts without a SPN does not matter given those user accounts do not "consume" service tickets and session key encryption type is determined by the configuration of the device and not the user account.

0 Likes

Like

Hi All - Looking for some suggestions after the November 2022 changes to the KDC service. Unfortunately I still have the need for some XP machines to function on our domain. I understand that the KDC service has logic to look at the operating system attributes and won't issue RC4 session keys to pre 2008 devices even when the msds-supportedencryptiontypes value is 4. A hack I found is to clear the XP OS attribute on the computer object. This allows the KDC to issue an RC4 session key for the XP service.

The last roadblock that I have is joining or rejoining a pre 2008 device to the domain. Would anyone have any creative suggestions for the initial join to work?

0 Likes

Like

MAybe the Part about "Support for AES ticket encryption was introduced with the release of Server 2008 / Windows 7 but it was not automatically enabled on domain accounts in order to ensure backward compatibility." should be rewritten with the NOV2022 Security updates, where AES is now the new default.

0 Likes

Like

Hi,

What about user accounts with no password expiration? It's supposed that we have many accounts of that type that were created before we rotated the krbtgt password, so they still don't have AES keys generated. I think those accounts will fail once the patches are deployed. We can't easily change the password for that many accounts.

On the other hand, will Windows Server 2003 and Windows XP systems fail once the patches are deployed?

Thank you.

0 Likes

Like

Running "klist" on my machine, I can see 2 (TGT?) tickets with the following information: Server: krbtgt/DOMAIN.COM @ DOMAIN.COM and KerbTicket Encryption Type: RSADSI RC4-HMAC(NT).

I understand that RC4 is deprecated, and all my other tickets are listed with AES256. The use of RC4-HMAC raises concerns for me, especially considering that Microsoft plans to turn off support for deprecated encryption with the July patch.

Is it because the krbtgt AD account has not been reset? Is there a risk associated with applying the July patch before resetting the krbtgt AD account?

0 Likes

Like

First off all, great article! Thanks!

I'm actually convinced that the last row of Supported Encryption Types, 31 (0x1F), has an (Excel range?) typo.

31 0x1F DES**+A1:C33**_CBC_MD5, DES_CBC_MD5, RC4, AES 128, AES 256

It should be: DES_CBC_CRC, DES_CBC_MD5, RC4-HMAC, AES128-CTS-HMAC-SHA1-96, AES256-CTS-HMAC-SHA1-96

or perhaps somewhat shorter: DES_CBC_CRC, DES_CBC_MD5, RC4, AES128, AES256

For a few years now I've seen many scripts passing by using your Supported Encryption Type list, and all of them copy-pasted the exact same descriptions, including the typo which is actually funny though 😉

Also, slightly off-topic, the https://docs.microsoft.com/en-us/archive/blogs/petergu/interpreting-the-supportedencryptiontypes-reg... link is out-dated. Perhaps you should use the following link: [MS-KILE]: Supported Encryption Types Bit Flags | Microsoft Learn

0 Likes

Like

@azuser -

When the account you are using is enabled for delegation you will see two TGTs in your cache. One will be your **Primary** TGT and the other will be marked as **Forwardable**. Microsoft recommends that you check the "**Account is sensitive and cannot be delegated**" box for privileged accounts in order to prevent exposing a reusable copy of a TGT to a remote device.

Active Directory does not store the clear text password for any account by default. That is why accounts which have not changed passwords since 2008R2 cannot have AES256 keys automatically generated after the domain has been upgraded to support AES. The KRBTGT password does not rotate automatically It is possible yours

lacks AES keys. This post provides some good details on performing that reset - https://techcommunity.microsoft.com/t5/core-infrastructure-and-security/faqs-from-the-field-on-krbtg...

BTW – The July changes will not remove support for RC4 with Active Directory but the November update did change how the encryption type of session keys are selected. The 5840 Event ID (The Netlogon service created a secure channel with a client with RC4) will warn you if RC4 is used with Netlogon secure channels to domain controller but those connections will not be blocked once the Netlogon changes are in enforcement mode.

@fedayn1 -

The AES key derived for a user account are independent of the KRBTGT account. It is ultimately dependent on what the domain controllers supported when the account password was last changed.

2003 and XP systems will continue to work given RC4 support is not going away (including for Netlogon Secure Channel connections as mentioned above). However, you do have to factor in how the November 2022 update changed the defaults for session key encryption type. Session key encryption for service tickets to Windows devices is now determined by the encryption type supported by the device used to request the service ticket. As a result, you can end up with a service ticket for a 2003 \ XP device that is encrypted RC4 but the **session key** for the ticket is AES. If that happens authentication will fail. For example, a Windows 10 device will by default receive a AES session key with a service ticket for a 2003 server because it is capable of negotiating AES during the initial AS-REQ.

@San1978 - Great catch on the table. The post has been up for nearly 3 years and nobody else has pointed that out. I also upated the URL per your recommendation.

1 Like
Like

Wow! Just now, rereading the post and all the comments, besides some other misunderstandings, the following became clear to me. The November update 2022 has no affect on the encryption type of service tickets. It only affects session keys.... and here I was wondering why my IIS running as a service user (with empty msDS-SupportedEncryptionTypes value) was showing up as RC4 in klist.... are you aware of any plans on forcing AES for service tickets as well?

0 Likes

Like

@David_Trevor - The changes from Nov 22 (and Dec 22) where rough for organizations with legacy devices. You are correct that it changed the logic for session key selection but not for the ticket encryption type. I found many customers are tempted to configure msDS-SupportedEncryptionTypes on regular user accounts but that has no impact on the selection of the session key for keys aquired by the user.

To my knowledge there is no plan to change the ticket encryption logic to make AES is the default when msDS-SupportedEncryptionTypes is blank. Changing that logic would be a bigger impact than change to session key selection.

0 Likes

Like

Great information in the article and the comments; great to know there are such complete references out there.

One question @JerryDevore , we are trying to update all the supported ciphers on our internal domain and want to know which should come first, Remove the RC4 ciphers 1st, or set the msDS-SupportedEncryptionType value to AES only 1st?

0 Likes

Like

- Previous

-
    - 1
    - 2
- Next