Kerberos, Active Directory's Secret Decoder Ring

By Sean Metcalf

Published: 2014-09-12 · Archived: 2026-04-05 13:00:53 UTC

Kerberos Overview

Kerberos is a protocol with roots in MIT named after the three-headed dog, Cerberus. Named because there are 3 parties: the client, the resource server, and a 3rd party (the Key Distribution Center, KDC).

Kerberos can be a difficult authentication protocol to describe, so I will attempt to simplify it as best as possible.

Kerberos authentication leverages long-term asymmetric keys (public key) and short-term symmetric keys (session keys).

Asymmetric key cryptography uses two mathematically connected keys where one key is used to encrypt and the other is used to decrypt data. This is most commonly used in Public Key encryption (PKI) where one of the keys is kept secret by the user or service (Private Key) and the other key is available to anyone who wants it (Public Key). In this manner a user can sign (encrypt a hash with the private key) data to ensure it originated from that user without modification (the receiver decrypts the hash with the public key). Also a person can use the user's public key to encrypt data so that only the user can decrypt it with the user's private key.

Symmetric key cryptography uses one key to encrypt and the same to decrypt the data. This is also referred to as a shared secret.

Since asymmetric key cryptography is more processor intensive, it is typically only used to encrypt session keys which use symmetric keys (shared secret).

Active Directory implements Kerberos version 5 in two components: the Authentication service and the Ticket-granting service.

The Authentication Service (AS) is the first contact the client has with Kerberos and is used to lookup the user's password and create the Ticket Granting Ticket (TGT). The AS also creates the session key the user will use for future communication with Kerberos.

The Ticket Granting Ticket (TGT) is the Kerberos ticket for the Ticket Granting Service (runs on the KDC) and is encrypted using the KDC key (KRBTGT domain Kerberos account), meaning that only a KDC can decrypt and read the ticket. While the user's ticket ,the TGT, is set to expire after 10 hours (AD default), it can be renewed as often as needed during the TGT renewable lifetime which is 7 days (AD default). Once the authenticating user has a TGT, it presents the TGT to the KDC to get a Service Ticket for the Ticket Granting Service (TGS) on the KDC. Most key Kerberos communication occurs over UDP port 88, though starting with Windows Vista & Windows Server 2008 now default to using TCP for Kerberos ticket requests.

There is a myth in the Windows Kerberos world that if a workstation's clock is skewed more than 5 minutes from that of the Domain Controller, Kerberos authentication wouldn't work.
Technically, all clocks in the Kerberos world must be kept closely in-sync to prevent replay attacks. By default, Microsoft Active Directory has a tolerance of 5 minutes. **Though in most cases, this doesn't mater. When a client sends a Kerberos request to a DC, the DC will reply with a "KRB_ERROR – KRB_AP_ERR_SKEW (37)" and the Windows client will update its time for the Kerberos session with the DC and resend the request.** Provided the clock skew between the client and DC is not more than the ticket lifetime (10 hours by default), the second request will be successful.

Kerberized Internet Negotiation of Keys (KINK) RFC 4430 details how this works:

> If the server clock and the client clock are off by more than the policy-determined clock skew limit (usually 5 minutes), the server MUST return a KRB_AP_ERR_SKEW. The optional client's time in the KRB-ERROR SHOULD be filled out. **If the server protects the error by adding the Cksum field and returning the correct client's time, the client SHOULD compute the difference (in seconds) between the two clocks based upon the client and server time contained in the KRB-ERROR message. The client SHOULD store this clock difference and use it to adjust its clock in subsequent messages.** If the error is not protected, the client MUST NOT use the difference to adjust subsequent messages, because doing so would allow an attacker to construct authenticators that can be used to mount replay attacks.

KB956627 also describes this behavior.
Confused yet?
Keep reading, it gets easier…

Every service that is Kerberos enabled has an entry point called a Service Principal Name (SPN) and each Kerberos user has a User Principal Name (UPN). For example, a user named Joe User in the ADSECURITY.ORG Kerberos realm aka AD domain (the Kerberos realm is always all Caps) has a UPN of JoeUser@ADSecurity.org. If Joe User initiates a connection to the share path \\server1.ADSecurity.org\Shared then Joe's workstation will lookup the computer server1.ADSecurity.org in Active Directory and read its SPN attribute (cifs/server1.ADSecurity.org). The computer SPN is used to identify the application server in the Kerberos TGS ticket request. Furthermore, when Joe opens Outlook, his workstation performs similar actions looking up the Exchange server's SPN.
Exchange 2010 has a number of registered SPNs; here are a few of them used as part of a client connection (using Outlook 2010):

- exchangeMDB
- exchangeRFR
- exchangeAB
- HTTP

KERBEROS METAPHOR

As mentioned earlier, Kerberos has 3 components, the client, the server, and the KDC (trusted 3rd party). The process is similar to when you travel to a foreign country.

1. You visit the local passport office with a birth certificate
   (get the Ticket Granting Ticket ticket from the KDC)
2. You request an entrance Visa for your passport in order to enter the country
   (get the Ticket Granting Service ticket from the KDC – ok, so you would get the Visa from the country's embassy, but you still need the passport and something authoritative the country's immigration guard will accept).
3. You travel to the country with the passport and the country's entrance Visa
   (present authoritative documentation to gain access to the resource server).

KERBEROS TICKET PROCESS OVERVIEW

Ticket Granting Ticket (aka logon ticket)

1. Joe User logs on with his Active Directory user name and password to a domain-joined computer (usually a workstation). The computer takes the user's password and runs a one way function (OWF) creating a hash of the password (typically the NTLM hash). Hashing the password is like taking a steak and running it through a meat grinder. The ground beef that is the result can never be reassembled back into the same steak we started with. This is used to handle all Kerberos requests for the user (as well as other authentication methods such as NTLM).

2. Kerberos authentication is initiated by sending a timestamp (PREAUTH data) encrypted with the user's password-based encryption key (password NTLM hash).

3. The user account (JoeUser@adsecurity.org) requests a Kerberos service ticket (TGT) with PREAUTH data (Kerberos Authentication Service Request or AS-REQ).

4. The Domain Controller's Kerberos service (KDC) receives the authentication request, validates the data, and replies with a TGT (Kerberos AS-REP). The TGT has a Privileged Attribute Certificate (PAC) which contains all the security groups in which the user is a member. The TGT is encrypted and signed by the KDC service account (KRBTGT) and only the domain KRBTGT account can read the data in the TGT.

At this point, the user has a valid TGT which contains the users group membership and is used to prove the user is who they claim to be in further conversations with a Domain Controller (KDC). The TGT is sent to the Domain Controller every time a resource ticket is requested.

Ticket Granting Service ticket (aka resource access ticket)

5. When the user wants to access an AD resource (a file share for example), the user's TGT from step 4 is presented to a Domain Controller (KDC) as proof of identity with a request for a resource ticket to a specific resource (Service Principal Name). The DC determines if the TGT is valid by checking the TGT's signature and if valid, generates a resource access ticket (TGS) signed/encrypted with the KRBTGT account and a part encrypted with the Kerberos service account's session key which the destination service uses to validate the TGS.
Note: The DC doesn't validate the user has the appropriate access to the service, it only validates the TGT and builds a TGS based on the TGT information.

6. The resource service ticket (TGS) is sent to the user by the Domain Controller and is used for authentication to the resource. At this point, all communication has been between the user's computer and the Domain Controller

(KDC).

7. The user's computer sends the user's resource service ticket (TGS) to the service on the resource computer. If the destination service is a file share, the TGS is presented to the CIFS service for access.

8. The destination service (CIFS in this example) validates the TGS by ensuring it can decrypt the TGS component encrypted with the service's session key. The service may send the TGS to a DC (KDC) to validate the PAC to ensure the user's group membership presented is accurate. The service reviews the user's group membership to determine what level of access, if any, the user has to the resource.

THE DETAILED PROCESS

Here's my example scenario to explain what occurs when a user logs on and opens Outlook to view his Exchange email. The bold text is the simple overview version while the detail follows.

1. **A user logs onto the domain ADSecurity.org on the workstation ADSecurityPC.**
2. **The user requests authentication by sending a timestamp encrypted with the users password encryption key.**
   The workstation creates an encryption key derived from the user's password (the user's password is hashed using a one way function such as MD5 = **(A)** key) to encrypt a timestamp (date/time) as an authenticator (pre-authentication is required by AD in its default configuration, so the client must send an authenticator) . The authenticator is simply a method the client uses to prove to the KDC that the user is who he claims to be (since only the user & the KDC knows his password) and protects against replay attacks. This information is sent to the KDC in an AS-REQ (Authentication Service Request) packet. This request includes the client supported encryption algorithms.Keys used:
   (A) User's password derived key
   

Packet Data:
User account (user@ADSecurity.org) requests Kerberos service ticket (TGT) with PREAUTH dataKRB5: Kerberos AS-REQ
1 Forwardable: FORWARDABLE tickets are allowed/requested
1 Renewable: This ticket is RENEWABLE
1 Canonicalize: This is a request for a CANONICALIZED ticket
1 Renewable OK: We accept RENEWED ticketsClient Name (Principal): admin
Realm: ADSECURITY.ORG

Service: krbtgt/ADSecurity.org
till: 2037-09-12 02:48:05 (UTC)
rtime: 2037-09-12 02:48:05 (UTC)
Nonce: 1976014234
Principal Name: user
HostAddress: METCORPORGDC02<20>
PAC_Request: True
Encryption Types: aes256-cts-hmac-sha1-96 aes128-cts-hmac-sha1-96 rc4-hmac rc4-hmac-exp rc4-hmac-old-exp des-cbc-md5KRB5: Kerberos AS-REP
Client Name (Principal): user
Tkt-vno: 5

3. **The Kerberos server (KDC) receives the authentication request, validates the data, and replies with a TGT.**
   The KDC receives the AS-REQ, decrypts the authenticator (encrypted with key **(A)**), and validates the timestamp is within the time skew limits set by the domain (5 minutes by default). If the KDC is satisfied the request is a valid user request, the KDC responds with an AS-REP packet which includes the TGT. The TGT can only be decrypted by a KDC (using the **(B)** key) and is used to authenticate the user to the Kerberos server so it doesn't have to look up the user's password (long-term key) again. The KDC also includes a session key (**(C)**key) for use in future communication with the KC.Keys used:
   (B) Kerberos account's password derived key
   (C) User's Kerberos service (KDC) session key NOTE: The TGT is encrypted with the KRBTGT account password so only a valid Kerberos server can decrypt it. In an environment with RODCs, each RODC has its own krbtgt account with a unique password. This means that if a user presents a TGT received from a RODC to a writable DC, the DC dumps the TGT and generates a new one.

Packet Data:
The KDC replies with the TGT and session key
KRB5: Kerberos AS-REP
Client Name (Principal): User
Ticket (Tkt-vno): 5

Realm: ADSECURITY.ORG
Server Name: krbtgt/ADSecurity.org
enc-part aes256-cts-hmac-sha1-96
[Encrypted Key]
enc-part rc4-hmac
[Encrypted Key]

4. **The user opens Outlook which locates the user's mailbox server and requests a TGS ticket for access.**
   The workstation locates the Exchange mailbox server containing the user's mailbox (MetcorpEXMB02.ADSecurity.org) and reads the ServicePrincipalName attribute on the computer account in AD (ExchangeMDB/ADSecurityEXMB02.ADSecurity.org – there are a bunch, so I will just use this one for the example).
   The client then sends a TGS-REQ to the KDC requesting a TGS for access to the Exchange service running on the MetcorpEXMB02 Exchange server. The TGS request includes the target server SPN, the user's TGT (encrypted with the **(B)** key), and an authenticator (encrypted with the **(C)**key).Keys used:
   (B) Kerberos account's password derived key
   (C) User's Kerberos service (KDC) session key

Packet Data:
User account requests service ticket (TGS) for MetcorpEXMB02 Exchange service access
KRB5: Kerberos TGS-REQ
1 Forwardable: FORWARDABLE tickets are allowed/requested
1 Renewable: This ticket is RENEWABLE
1 Canonicalize: This is a request for a CANONICALIZED ticket
Realm: ADSECURITY.ORG
Server Name: ExchangeMDB/MetcorpEXMB02.ADSecurity.org
till: 2037-09-12 02:48:05 (UTC)
Nonce: 1976014234

Encryption Types: aes256-cts-hmac-sha1-96 aes128-cts-hmac-sha1-96 rc4-hmac rc4-hmac-exp rc4-hmac-old-exp des-cbc-md5KRB5: Kerberos AS-REP

5. **The KDC validates the TGS request and replies with the TGS.**
   The KDC replies with a TGS-REP packet to the client which includes 2 session tickets (TGS) (2?). One of the session tickets is encrypted with the user's (KDC) session key (**(C)** key) and the second one is encrypted with the target server's (KDC) session key (**(D)** key). The second TGS also includes the user's group membership & associated SIDs which provides the server information used to determine authorization and help the server determine: Is the user allowed to access the server's resource?
   Both session tickets include a new session key (**(E)**key) for exclusive use in communication between the Exchange server and the user.Keys used:
   (C) User's Kerberos service (KDC) session key
   (D) Server's Kerberos service (KDC) session key
   (E) User-Exchange service session key
   

   Packet Data:
   The KDC replies with the service ticket (TGS) for MetcorpEXMB02 Exchange service access
   KRB5: Kerberos TGS-REP
   Client Name (Principal): User
   Ticket (Tkt-vno): 5
   Realm: ADSECURITY.ORG
   Server Name: krbtgt/ADSecurity.org
   enc-part aes256-cts-hmac-sha1-96
   [Encrypted Key]
   enc-part rc4-hmac
   [Encrypted Key]
6. **The client authenticates to the Exchange server with the session ticket.**
   The client sends the target server (MetcorpEXMB02.ADSecurity.org) an AP-REQ packet containing the TGS it received from the KDC encrypted with the server's session key (**(D)** key) and an authenticator encrypted with the user-Exchange server session key (**(E)** key) . This lets the Exchange server know that the user was authenticated to the Kerberos domain (realm) and that the TGS is valid (assuming the Exchange server is able to decrypt it). The client also sends the server an authenticator (timestamp)

encrypted with the session key (**(E)**key) it received from the KDC in Step 5. The Exchange server decrypts the TGS, extracts the user's group information, extracts the session key, and uses the session key to decrypt the authenticator. This provides the server enough information to make an authorization decision. If the user is authorized to connect to the server, it sends a reply.Keys used:
(C) User's Kerberos service (KDC) session key
(D) Server's Kerberos service (KDC) session key
(E) User-Exchange service session key

7. **The Exchange server replies that authorization to the service is granted.**
   The Exchange server sends the client an AP-REP packet which includes its own authenticator encrypted with the user-Exchange service session key (**(E)** key)**.** This assumes the client requested mutual authentication which is the default configuration.Keys used:
   (E) User-Exchange service session key

Note:
This is a simplified explanation of Kerberos and doesn't cover everything involved in this process.

Kerberos Key Storage Locatons

Workstation Keys:

- User Key
- Ticket-Granting Ticket
- Ticket-Granting Service Session Key
- Service Ticket
- Session Key

Domain Controller:

- User Key
- Ticket-Granting Service Key
- Service Key

Server:

- Service Key
- Session Key

TICKETING

There are different "tickets" that are used to authenticate a client to a server's resource. The client can be a user or a computer.

The Ticket Granting Ticket (TGT) is the first ticket given to the requester (user or computer.

The TGT is comprised of the following fields:

- Ticket Version Number

- Realm: The AD domain name in CAPITAL LETTERS
- Server Name: The KDC
- Flags: Kerberos Flag options
- Key
- Client Realm: The client's AD domain name in CAPITAL LETTERS
- Client Name: The user name
- Transited: If the user is in a different domain than the resource, Kerberos tickets have transited.
- Authentication
- Time
- Start Time
- End Time
- Renew Till
- Client Address
- Authorization Data

Ticket Flags:

- FORWARDABLE
- FORWARDED
- PROXIABLE
- PROXY
- MAY-POSTDATE
- POSTDATED
- INVALID
- RENEWABLE
- INITIAL
- PRE-AUTHENT
- HW-AUTHENT

Supported Encryption Algorithms & Key Lengths:

- AES 128
  AES128-CTS-HMAC-SHA1-96 (0x08)
- AES 256
  AES256-CTS-HMAC-SHA1-96 (0x10)
- RC4-HMAC 128
  RC4-HMAC (0x04)
- DES-CBC-CRC 56
  DES-CBC-CRC (0x01)
- DES-CBC-MD5 56
  DES-CBC-MD5 (0x02)

DEFAULT AD KERBEROS POLICY SETTINGS

- **Enforce user logon restrictions:** Enabled
- **Maximum lifetime for service ticket:** 600 minutes (10 hours)
- **Maximum lifetime for user ticket:** 600 minutes (10 hours)
- **Maximum lifetime for user ticket renewal :** 7 days
- **Maximum tolerance for computer clock synchronization:** 5 minutes

References:

- Kerberos Explained
- Kerberos for the Busy Admin
- [MS-PAC]: Privilege Attribute Certificate Data Structure
- The Kerberos Protocol (Wikipedia)
- Microsoft Kerberos Protocol (MSDN)
- What is Kerberos Authentication
- Kerberos Authentication Technical Reference
- How the Kerberos Version 5 Authentication Protocol Works
- Microsoft's Kerberos Extensions – RFC 3244: "Microsoft Windows 2000 Kerberos Change Password and Set Password Protocols"
- Kerberos Technical Supplement for Windows
