Kerberos Explained

blog.netwrix.com/what-is-kerberos

In Greek mythology, Kerberos is a multi-headed dog that guards the gates of the underworld. The Kerberos meaning in technology is analogous: Kerberos is an authentication protocol guards the network by enabling systems and users to prove their identity to one another before access to resources is granted. Read on to learn how Kerberos authentication works and get valuable tips for avoiding issues.

Kerberos Structure and Operation

Kerberos was named after the three-headed dog because of the three different actors in the protocol:

- **Client**: The entity seeking to provide its identity
- **Application Server (AP)**: The service that the client or user wants to access
- **Key Distribution Center (KDC)**: A trusted third party that issues tickets

Support for Kerberos is found in almost every operating system, including Apple OSX/iOS and many UNIX and Linux distributions. However, Microsoft Active Directory is the most widely consumed Kerberos implementation. It is based on Kerberos Network Authentication Service (V5).

Handpicked related content:
[Free Guide] Active Directory Security Best Practices

Microsoft expanded upon the base protocol specification, adding a number of extensions to implement features specific to Active Directory and the Windows Server operating systems.

In Active Directory, each domain controller acts as a KDC and provides two core services:

- **Authentication Service (AS)** — Authenticates clients and issues them tickets
- **Ticket Granting Service (TGS)** — Accepts authenticated clients and issues them tickets to access other resources

The tickets utilize symmetric encryption technology. Certain user passwords are used to encrypt and sign specific tickets, but the root of the Kerberos security is a key known only to the trusted third party that issues the tickets.

Kerberos Authentication Process

Each step of Kerberos authentication employs cryptography to protect packets from being altered or read and provide mutual authentication. A client requests a ticket for a user from the KDC, using the user's password to encrypt the request. If the KDC can decrypt the request with the user's password it has stored, it knows the client has supplied the correct password for the user. The KDC creates a ticket granting ticket (TGT) for the user, encrypts it with the user's password, and returns it to the client. If the client can decrypt that ticket with the user's password, it knows that the KDC is legitimate.

A client requests a ticket for a service from the KDC by presenting its TGT and a ticket-granting service (TGS) request that includes the service principal name for the service it would like to access. The KDC creates a service ticket (TGS) that is encrypted with the service's password hash (TGS secret key), encrypts the ticket and authenticator message with the shared ticket-granting service session key, and finally sends the TGS back to the client.

A client requests access to an application server (service) by presenting the service ticket it obtained from the KDC to the application server, which decrypts the message using its own password hash. If it successfully decrypts the TGS, the application server grants access to the client.

Kerberos Authentication Steps

Figure 1: Kerberos Authentication Flow

1. **KRB_AS_REQ: Request TGT from Authentication Service (AS)**

The client’s request includes the user’s User Principal Name (UPN) and a timestamp. It is encrypted using the user’s password hash.

2. **KRB_AS_REP: TGT Received from Authentication Service**

The KDC uses the UPN to look up the client in its database and uses the user’s password hash to attempt to decrypt the message.

AS generates a TGT containing the client ID, client network address, timestamp, lifetime and a session key (SK1).

If the KDC successfully decrypts the TGT request and if the timestamp is within the KDC’s configured time skew, the authentication is successful.

A TGT and a TGS session key are sent back to the client. The TGS session key is used to encrypt subsequent requests.

3. **KRB_TGS_REQ: Present TGT and TGS request**

The client presents its TGT along with a request that includes the SPN for the service it wants to access. The TGS request is encrypted with the TGS session key.

4. **KRB_TGS_REP: Receive TGS from KDC**

The KDC attempt to validate the TGT; if successful, it generates a TGS that contains information about the requestor, such as their SID and group memberships, and is encrypted with the service's password hash.

The TGS and the service session keys are encrypted with the TGS session key and sent back to the client.

5. **KRB_AP_REQ: Present TGS to Application Server for Authorization**

The client sends the TGS that it received from the KDC to the application server, along with an authenticator message that is encrypted with the service session key.

6. **KRB_AP_REP: Grant Client Access to the Service**

The client receives the message and decrypts it with the service session key.

The Application Server extracts the Privilege Attribute Certificate (PAC) from the service ticket to verify its contents with a domain controller.

Validation of the ticket and PAC happens only when the TGT is older than 20 minutes.

Factors Affecting Kerberos Operation

There are a handful of factors that problems if not sufficiently provided for.

- **Replication is required between domain controllers.**
  If multiple domain controllers (and therefore multiple KDCs) are deployed, then replication must be enabled and happen in a timely manner. Should replication fail or be delayed, authentication failures are possible when a user changes their password.
- **Clients and KDCs must use NetBIOS and DNS name resolution.**
  Kerberos Service Principal Names normally include NetBIOS and DNS addresses, which means both the KDC and client must be able to resolve those names the same way. In certain situations, IP addresses may also be used in Service Principal Names.
- **Clients and KDCs must have their clocks synchronized.**
  Accurate measurement of time is important to prevent replay attacks. Kerberos supports a configurable time skew (5 minutes by default), outside of which client authentication will fail.
- **Clients and KDCs must be able to communicate on the network.**
  Kerberos traffic occurs on TCP and UDP port 88, which must be accessible from all clients to at least one KDC.
- **Clients, users and services must have unique names.**
  Duplicate credentials for computers, users or Service Principal Names can cause unexpected Kerberos authentication

Kerberos vs LDAP

When reading about the Kerberos protocol, you'll frequently see mentions of Lightweight Directory Access Protocol (LDAP). Kerberos and LDAP are commonly used together (including in Microsoft Active Directory) to provide a centralized user directory (LDAP) and secure authentication (Kerberos) services.

LDAP stores information about users, groups and other objects (like computers) in a central location. It can also provide simple authentication; however, this protocol, unlike Kerberos, generally requires the user's secret (i.e., password) to be transmitted over the network. Each resource the user wants to access must handle the user's password and separately authenticate the user to the directory.

Unlike LDAP, Kerberos provides for single sign-on functionality. Once a user has authenticated to the KDC, no other service (like an intranet site or file share) needs the user’s password. The KDC is responsible for issuing tickets that each service trusts.

The combination of LDAP and Kerberos provides centralized user management and authentication, and in larger networks, Kerberos provides substantial security benefits.

How can I see my Kerberos tickets?

It is easy to see your Kerberos tickets. On a Microsoft Windows computer, you can use the klist.exe program to enumerate them by opening a command prompt or PowerShell and running the **klist tickets** command. In the example below, you can see that Joe has a ticket for the CIFS service, which is file share access, to a server called fileserver1.

```
PS C:Windowssystem32> klist tickets
Current LogonId is 0:0xe67df
Cached Tickets: (4)
#0>     Client: Joe @ domain.local
        Server: cifs/fileserver1.domain.local/domain.local @ DOMAIN.LOCAL
        KerbTicket Encryption Type: AES-256-CTS-HMAC-SHA1-96
        Ticket Flags 0x60a10000 -> forwardable forwarded renewable pre_authent name_canonicalize
        Start Time: 7/10/2020 12:33:49 (local)
        End Time:   7/10/2020 22:32:13 (local)
        Renew Time: 7/17/2020 12:32:13 (local)
        Session Key Type: AES-256-CTS-HMAC-SHA1-96
        Cache Flags: 0x40 -> FAST
        Kdc Called: DC1.domain.local
```

Conclusion

Kerberos is a well-known and widely used authentication protocol. Because it lies at the heart of Microsoft Active Directory, it has become one of the protocols most targeted for abuse by adversaries of all shades. Netwrix is dedicated to helping enterprises protect against and detect attack on Active Directory. To learn more, visit the Netwrix Attack Catalog or visit our website to explore our solution portfolio.