**What is Kerberos?**

Kerberos is a protocol for [authenticating](https://searchsecurity.techtarget.com/definition/authentication) service requests between trusted hosts across an untrusted network, such as the internet. Kerberos is built in to all major operating systems, including Microsoft Windows, Apple OS X, FreeBSD and Linux.

Kerberos (Cerberus) was a three-headed dog who guarded the gates of Hades. **The three heads of the Kerberos protocol represent a**[**client**](https://searchenterprisedesktop.techtarget.com/definition/client)**, a**[**server**](https://whatis.techtarget.com/definition/server)**and a Key Distribution Center (KDC), which acts as Kerberos' trusted third-party authentication service**.

Microsoft introduced their version of Kerberos in Windows2000. It has also become a standard for websites and Single-Sign-On implementations across platforms. The [Kerberos Consortium](http://kerberos.org/)maintains Kerberos as an [open-source project](https://en.wikipedia.org/wiki/Open-source_software).

Kerberos is a vast improvement on previous authorization technologies. The strong cryptography and third-party ticket authorization make it much more difficult for cybercriminals to infiltrate your network. It is not totally without flaws, and in order to defend against those flaws, you need to first understand them.

Kerberos has made the internet and its denizens more secure, and enables users to do more work on the Internet and in the office without compromising safety.

**What is the difference between Kerberos and NTLM?**

Before Kerberos, Microsoft used an authentication technology called [NTLM](https://en.wikipedia.org/wiki/NT_LAN_Manager). NTLM stands for NT Lan Manager and is a [challenge-response authentication protocol](https://en.wikipedia.org/wiki/Challenge%E2%80%93response_authentication). The target computer or domain controller challenge and check the password, and store password hashes for continued use.

The biggest difference between the two systems is the third-party verification and stronger encryption capability in Kerberos. This extra step in the process provides a significant additional layer of security over NTLM.

NTLM systems can get hacked in a matter of hours these days: it’s simply older technology, and you shouldn’t rely upon NTLM to protect sensitive data.

[**What is a difference between security token and security ticket?**](https://stackoverflow.com/questions/14200579/what-is-a-difference-between-security-token-and-security-ticket)

The words *token* and *ticket* depend greatly on the type of system you're dealing with; and in what context you are talking. On Windows NT derivatives, the concept of a **token** is an identity. When a user or service logs in to a system, the system validates their identity once, and mints a token, which is handed to that user/service and serves as their identity. The system then doesn't need to validate identity every time a program opens a file, for example. This basically ensures a clean separation between authentication (proving a user/service is who they say they are) and authorization (determining whether a user/service can access some resource).

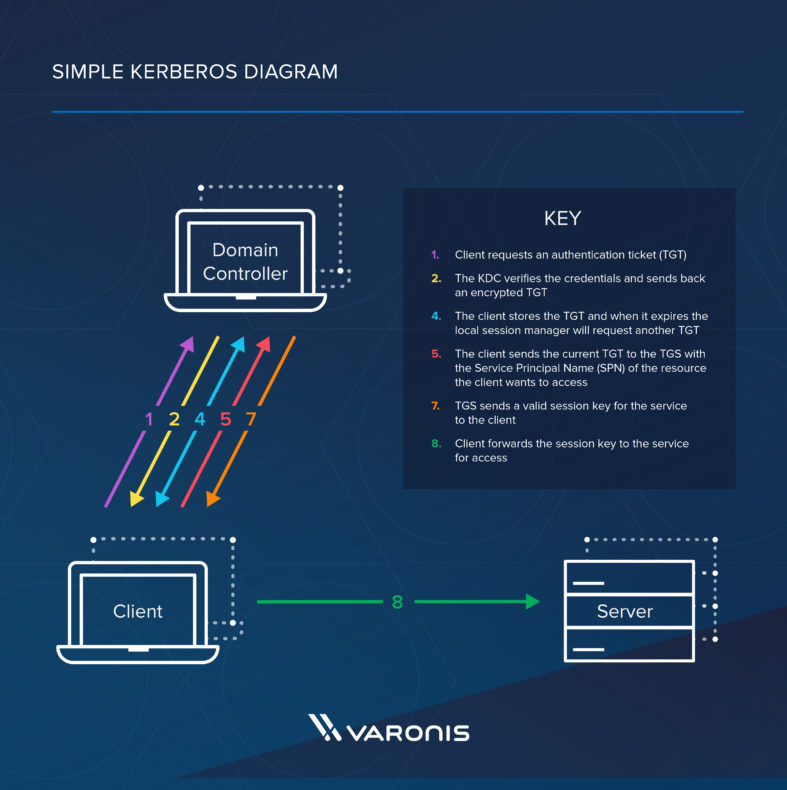
On the other hand, (again for NT derivatives) the word **ticket** usually refers to [Kerberos tickets](http://en.wikipedia.org/wiki/Kerberos_(protocol)). These are used for two machines on a domain to be able to prove each others' identity. After proving one's identity to the domain controller (with traditional means such as password or smart card), the domain controller mints a ticket which can be passed to remote machines to verify identity.

**Example**: **If one is dealing with a remote machine, then it is likely both tickets and tokens are involved. For instance, if machine A opens a file share on machine B, then machine A validates the user using it with the domain controller, thus getting a Kerberos ticket. It then uses the Kerberos ticket to verify its identity with machine B. Machine B then creates a session for machine A, minting a token, to serve as that session identity for local authorization queries on machine B**.

**Lets Learn some of the common terms used in Kerberos infrastructure.**

* **KDC**: Key distribution Centre, this will be the server which we call the middle man server or the central server arbitrator, which issues the keys for the communication.
* **REALM**: a kerberos network identified by a name, mostly this is the domain name in all caps.
* **SPN** – (Service Principal Name): this is the name used by the kerberos central server to call users, service name etc.
* **TGS**: Ticket Granting Server: this is mostly the same central server (KDC server), it grants the tickets for a service.
* **TGT**: A special ticket which contains the session key for communication between the client machine and the central KDC server.

**How do you authenticate with Kerberos?**



Here are the most basic steps taken to authenticate in a Kerberized environment.

1. Client requests an authentication ticket (TGT) from the Key Distribution Center (KDC)
2. The KDC verifies the credentials and sends back an encrypted TGT and session key
3. The TGT is encrypted using the Ticket Granting Service (TGS) secret key
4. The client stores the TGT and when it expires the local session manager will request another TGT (this process is transparent to the user)

If the Client is requesting access to a service or other resource on the network, this is the process:

1. The client sends the current TGT to the TGS with the Service Principal Name (SPN) of the resource the client wants to access
2. The KDC verifies the TGT of the user and that the user has access to the service
3. TGS sends a valid session key for the service to the client
4. Client forwards the session key to the service to prove the user has access, and the service grants access.

Note:The timestamped ticket sent by the ticket granting service allows the requesting system to access the service using a single ticket for a specific time period without having to be re-authenticated. Making the ticket valid for a limited time period makes it less likely that someone else will be able to use it later; it is also possible to **set the maximum lifetime to 0**, in which case service tickets will not expire. Microsoft recommends **a maximum lifetime of 600 minutes for service tickets**; this is the default value in Windows Server implementations of Kerberos.