A **keytab** (short for *key table*) is a file used in **Kerberos authentication** that stores **pairs of Kerberos principals and their encrypted secret keys**. These keys are used by services or users to authenticate themselves to the Kerberos Key Distribution Center (KDC) without having to manually enter a password.

**Key Features:**

* **Used in Kerberos-based environments** (e.g., Active Directory, MIT Kerberos).
* **Enables non-interactive authentication**, especially useful for services and scripts.
* **Encrypted format**, making it more secure than storing plaintext credentials.
* Typically managed with tools like ktutil, kadmin, or klist.

**Common Use Cases:**

* Running a **service (like HTTP, SSH, Hadoop, etc.)** that needs to authenticate Kerberos.
* **Automated scripts** that need Kerberos access without human input.
* **Single sign-on (SSO)** scenarios in enterprise environments.

**Example:**

If a service is running as http/host.example.com@EXAMPLE.COM, the corresponding keytab might contain an entry like:

bash

CopyEdit

Principal: HTTP/host.example.com@EXAMPLE.COM

So when the service starts, it reads the keytab and uses the stored key to authenticate to Kerberos.

**🔧 Part 1: Creating a Keytab File**

You can create a keytab file in several ways depending on your Kerberos environment (e.g., MIT Kerberos, Active Directory). Here’s how to do it with **MIT Kerberos tools**.

**✅ Option 1: Using kadmin**

1. **Log in to kadmin**:

bash

kadmin -p admin/admin

1. **Add a principal (if not already created)**:

kadmin

CopyEdit

addprinc -randkey http/webserver.example.com@EXAMPLE.COM

1. **Export the keytab**:

kadmin

CopyEdit

ktadd -k /etc/http.keytab http/webserver.example.com@EXAMPLE.COM

This will save the keys for the http/webserver.example.com principal into the file /etc/http.keytab.

**✅ Option 2: Using ktutil (manual)**

You can manually create a keytab with ktutil, but this is less common and only works if you already have the password or raw key.

Example:

bash

CopyEdit

ktutil

ktutil: addent -password -p username@REALM -k 1 -e aes256-cts-hmac-sha1-96

ktutil: wkt /tmp/username.keytab

**🚀 Part 2: Using a Keytab File**

Once you have a keytab file, you can use it to authenticate without entering a password.

**🔑 Authenticate with kinit using a keytab:**

bash

kinit -kt /path/to/keytab username@REALM

🔹 Example:

kinit -kt /etc/http.keytab http/webserver.example.com@EXAMPLE.COM

Check that authentication succeeded:

klist

You should see a valid Kerberos ticket.

A complete step-by-step example showing how to generate a keytab file on Windows using Active Directory and ktpass, then use it on a Linux server to authenticate a service via Kerberos.

**🖥️ Scenario**

* You're using Active Directory as your Kerberos KDC.
* You want a Linux service (e.g. Apache, Kafka, Hadoop) to authenticate using a service account with a keytab.
* Realm: EXAMPLE.COM
* Service principal: HTTP/linuxserver.example.com@EXAMPLE.COM
* AD service account: svc\_http

**✅ Step 1: Create the Service Account in Active Directory**

1. On a domain controller or AD admin console:
   * Create a user account, e.g., svc\_http
   * Set the password and select:
     + ✅ "Password never expires"
     + ✅ "User cannot change password"
2. **✅ Step 2: Register the SPN**

On the domain controller, open **Command Prompt as Administrator** and run:

setspn -A HTTP/linuxserver.example.com svc\_http

Check it:

setspn -L svc\_http

1. **✅ Step 3: Generate the Keytab Using ktpass**
2. Still in the command prompt:

ktpass -princ HTTP/linuxserver.example.com@EXAMPLE.COM ^

-mapuser svc\_http ^

-pass [YourPassword] ^

-out C:\keytabs\http.keytab ^

-ptype KRB5\_NT\_PRINCIPAL ^

-crypto AES256-SHA1

Replace [YourPassword] with the actual password of the svc\_http account.

Ensure the keytab is generated using a strong encryption type (e.g., AES256-SHA1). Older types like RC4 are deprecated.

**✅ Step 4: Copy the Keytab to the Linux Server**

On your Linux machine:

bash

scp administrator@domaincontroller:C:/keytabs/http.keytab /etc/http.keytab

Set proper permissions:

chown root:root /etc/http.keytab

chmod 600 /etc/http.keytab

**✅ Step 5: Test with kinit on Linux**

**bash**

kinit -kt /etc/http.keytab HTTP/linuxserver.example.com@EXAMPLE.COM

Verify:

klist

You should see a valid ticket.

**✅ Step 6: Configure Your Service (e.g., Apache, Hadoop)**

Set your service to use the principal and keytab. For example, in krb5.conf or service-specific configs:

ini

principal = HTTP/linuxserver.example.com@EXAMPLE.COM

keytab = /etc/http.keytab

An **SPN (Service Principal Name)** is a **unique identifier** for a service instance in a **Kerberos** authentication system. It's used by **clients to request access to a specific service** and by the Key Distribution Center (KDC) to locate the corresponding credentials.

**🔑 Key Concepts:**

* An SPN **maps a service** (like a website or database) to the **account running it** (usually a domain account).
* It ensures the correct **Kerberos ticket** is issued to the right service.
* Without a correctly registered SPN, **Kerberos authentication will fail** and may fall back to less secure methods (like NTLM).

**🔍 SPN Format:**

The format of an SPN is:

php-template

<service>/<hostname>[:<port>]

Examples:

* HTTP/webserver.example.com
* MSSQLSvc/dbserver.example.com:1433
* HOST/appserver.example.com

**🛠 Common Uses:**

* **Web servers** (IIS, Tomcat, etc.)
* **Database services** (SQL Server, Oracle, etc.)
* **File shares**, LDAP, or other network services using Kerberos

**🔐 SPN & Keytab:**

* When a service starts, it uses a **keytab** containing keys for its SPN.
* A Kerberos ticket issued to an SPN proves that the client is allowed to talk to that service.

Here's how to **list and register (or set) SPNs in Active Directory** using the Windows command-line tool setspn.

**✅ 1. List Existing SPNs**

To list all SPNs registered to a specific account (typically a service account):

bash

setspn -L <AccountName>

🔹 **Example**:

setspn -L svc\_http

This will show all SPNs currently associated with the svc\_http account.

**✅ 2. Register (Add) a New SPN**

To manually add a new SPN:

bash

setspn -A <SPN> <AccountName>

🔹 **Example**:

setspn -A HTTP/webserver.example.com svc\_http

This associates the SPN HTTP/webserver.example.com with the domain account svc\_http.

🛑 **Important**: Only one account in the domain should have a given SPN. Duplicate SPNs cause Kerberos failures.

**✅ 3. Delete an SPN**

If needed, you can remove an SPN like this:

bash

CopyEdit

setspn -D <SPN> <AccountName>

**🧰 Useful Tip – Check for Duplicate SPNs**

To search for an SPN across the domain and check for duplicates:

bash

setspn -Q <SPN>

🔹 **Example**:

bash

CopyEdit

setspn -Q HTTP/webserver.example.com

Top of Form