The **maximum lifetime** of a Kerberos ticket is controlled by policy settings in Active Directory (or the KDC, in general). There are two key time settings:

**1. Maximum lifetime for user ticket (TGT)**

* This is the **maximum number of days a ticket-granting ticket (TGT)** can be valid.
* Default in Active Directory: **10 hours**
* **Maximum possible value**: **99,999 hours** (~4,166 days) — but this is **strongly discouraged** and almost never used.

**2. Maximum renewable lifetime**

* This defines how long the ticket can be **renewed** without re-authentication.
* Default in Active Directory: **7 days**
* Maximum allowed: **in theory up to 99,999 hours**, but **practical limits are usually set by domain policy**, typically up to 30 days or less.

**Where to Set This (Admins):**

* In **Group Policy** → Computer Configuration > Policies > Windows Settings > Security Settings > Account Policies > Kerberos Policy:
  + **Maximum lifetime for user ticket**
  + **Maximum lifetime for service ticket**
  + **Maximum lifetime for user ticket renewal**

**Example:**

If you want a Kerberos ticket to last **30 days**, you'd need:

* TGT lifetime = 30 days
* Renewable lifetime ≥ 30 days
* The requesting client must explicitly ask for the longer duration (e.g., kinit -l 30d in Linux)

**Important:**

* Even if configured, **service tickets and TGTs are typically short-lived for security**. Long-lived tickets increase risk in case of credential compromise.
* To keep access alive over long periods, use **renewable tickets with automation**, not long static tickets.

Let me know your environment (Linux, Windows, AD domain, etc.) and whether you're using a service account — I can help tailor the config.

**You said:**

Linux, Windows, AD domain using a service account

**ChatGPT said:**

Thanks — you're using **Linux to authenticate to a Windows AD domain via Kerberos using a service account**. Here's what you need to know about **max ticket lifetime** and how to **extend it safely**.

**🔒 Kerberos Ticket Lifetime Limits in AD**

On the **Active Directory (Windows)** side, the max lifetime for Kerberos tickets is controlled by domain policy:

| **Setting** | **Default** | **Max Allowed (theoretical)** |
| --- | --- | --- |
| Max lifetime for user ticket (TGT) | 10 hours | ~4166 days (not recommended) |
| Max lifetime for service ticket | 600 minutes (10 hours) | ~4166 days |
| Max lifetime for ticket renewal | 7 days | Up to 365 days (practical limit) |

These are configurable in **Group Policy** on the AD domain controller:  
GPMC.msc → Computer Configuration → Policies → Windows Settings → Security Settings → Account Policies → Kerberos Policy

**✅ What You Should Do**

For a **Linux service account authenticating to AD**, the cleanest approach is:

**1. Set renewable tickets**

Ask for a **renewable TGT** with a long renewal window:

bash

CopyEdit

kinit -l 10h -r 30d service\_account@YOUR.DOMAIN.COM

This requests:

* -l 10h: 10-hour TGT (same as default)
* -r 30d: renewable for 30 days without password reentry

**2. Ensure AD allows this**

Make sure the **Kerberos policy** in the domain allows:

* TGT lifetime ≥ -l duration
* Renewable lifetime ≥ -r duration

**3. (Optional) Modify service account settings**

On the Windows side (in ADUC or PowerShell):

* Set the service account to:
  + **Trusted for delegation** (if required)
  + **Password never expires**
  + Ensure Account is sensitive and cannot be delegated is **unchecked** (if delegation is needed)

**4. Auto-renew tickets on Linux**

Use a cron job or systemd timer to auto-renew:

bash

CopyEdit

kinit -R

(Only works within the renewable lifetime)

**🔎 To Check Current Ticket Lifetime**

bash

CopyEdit

klist -c /tmp/krb5cc\_{UID}

It will show:

* Valid starting
* Expires
* If renewable: renew until

**🛑 Don’t Try to Hack Long Static Tickets**

Kerberos is designed to **limit long-lived static credentials**. Instead of requesting a 30-day ticket, **use renewable tickets and rotate them**.

Would you like a sample Linux script to maintain Kerberos auth over time using your service account?

**You said:**

yes

**ChatGPT said:**

Here’s a **Linux script** that will:

* Request a **renewable Kerberos TGT** using your **service account**
* Periodically **renew the ticket** before it expires (via cron or systemd)
* Optionally log activity for monitoring/debugging

**✅ 1. Setup Script: /opt/kerberos/renew\_kerberos.sh**

bash

#!/bin/bash

# CONFIGURATION

KEYTAB="/opt/kerberos/service\_account.keytab"

PRINCIPAL="service\_account@YOUR.DOMAIN.COM"

CCACHE="/tmp/krb5cc\_service\_account"

LIFETIME="10h"

RENEWABLE\_LIFETIME="30d"

LOGFILE="/var/log/kerberos\_renew.log"

# Ensure directory exists

mkdir -p "$(dirname "$CCACHE")"

mkdir -p "$(dirname "$LOGFILE")"

# Get new ticket if none exists or is expired

if ! klist -c "$CCACHE" -s; then

echo "$(date) [INFO] Requesting new TGT" >> "$LOGFILE"

kinit -k -t "$KEYTAB" -c "$CCACHE" -l "$LIFETIME" -r "$RENEWABLE\_LIFETIME" "$PRINCIPAL"

else

# Try renewing ticket

echo "$(date) [INFO] Renewing TGT" >> "$LOGFILE"

kinit -R -c "$CCACHE"

if [ $? -ne 0 ]; then

echo "$(date) [WARN] Renewal failed, requesting fresh TGT" >> "$LOGFILE"

kinit -k -t "$KEYTAB" -c "$CCACHE" -l "$LIFETIME" -r "$RENEWABLE\_LIFETIME" "$PRINCIPAL"

fi

fi

###############################################################################

Make it executable:

chmod +x /opt/kerberos/renew\_kerberos.sh

**✅ 2. Create Service Account Keytab (once)**

From a Windows DC:

ktpass -out service\_account.keytab -princ service\_account@YOUR.DOMAIN.COM -mapuser DOMAIN\service\_account -pass {password} -ptype KRB5\_NT\_PRINCIPAL -crypto AES256-SHA1 -kvno 0

Move securely to /opt/kerberos/service\_account.keytab on Linux.

**✅ 3. Automate Renewal (cron)**

Edit crontab:

bash

crontab -e

Add:

\*/30 \* \* \* \* /opt/kerberos/renew\_kerberos.sh >/dev/null 2>&1

This runs every 30 minutes.

**✅ 4. Use the Credential Cache**

If a client (e.g., sqlcmd, FreeTDS, Python, etc.) uses Kerberos, point it to:

bash

CopyEdit

export KRB5CCNAME=/tmp/krb5cc\_service\_account  
  
===========---------------------------------------------------------------------------------===========

**🔐 What’s Service Account–Specific**

| **Component** | **Unique per service account?** | **Notes** |
| --- | --- | --- |
| service\_account.keytab | ✅ Yes | Tied to specific AD user (mapuser) and principal |
| PRINCIPAL="service@DOMAIN.COM" | ✅ Yes | This is your Kerberos identity |
| CCACHE="/tmp/krb5cc\_..." | ✅ Recommended to be unique | Avoid collisions between accounts/sessions |
| Cron job or systemd task | ✅ Typically per account | Unless you use a central script to loop over multiple accounts |
| Ticket lifetime / renewal config | ✅ Yes, based on account policy | Could differ if AD applies different policies to each account |

✅ Option 1: One Script per Account

Create a version of the script *per account:*

/opt/kerberos/renew\_kerberos\_<account1>.sh

/opt/kerberos/renew\_kerberos\_<account2>.sh

*Each uses its own:*

Keytab

Principal

Cache file

Set up *individual cron jobs or systemd timers for each*.

**🔁 Option 2: Single Script for All Accounts**

Create one reusable script that loops over all service accounts:

bash

#!/bin/bash

SERVICE\_ACCOUNTS=(

"account1@DOMAIN.COM:/opt/kerberos/account1.keytab:/tmp/krb5cc\_account1"

"account2@DOMAIN.COM:/opt/kerberos/account2.keytab:/tmp/krb5cc\_account2"

)

for entry in "${SERVICE\_ACCOUNTS[@]}"; do

IFS=":" read -r PRINCIPAL KEYTAB CCACHE <<< "$entry"

if ! klist -c "$CCACHE" -s; then

echo "$(date) [INFO] New TGT for $PRINCIPAL"

kinit -k -t "$KEYTAB" -c "$CCACHE" -l 10h -r 30d "$PRINCIPAL"

else

echo "$(date) [INFO] Renewing TGT for $PRINCIPAL"

kinit -R -c "$CCACHE" || kinit -k -t "$KEYTAB" -c "$CCACHE" -l 10h -r 30d "$PRINCIPAL"

fi

done

--++++++++++++++++++++++++++++---

Here's a **production-grade Kerberos renewal script** that supports:

* 🔧 **Parameter input**: principal, keytab, cache file
* 📜 **Logging** to file (with optional rotation support)
* 🛠️ **Error handling**
* 🔄 **Systemd service and timer** templates

**✅ 1. Reusable Script: /opt/kerberos/renew\_kerberos.sh**

bash

CopyEdit

#!/bin/bash

set -euo pipefail

# ---------------------------

# Parameters

# ---------------------------

PRINCIPAL="${1:-}"

KEYTAB="${2:-}"

CCACHE="${3:-}"

LIFETIME="${4:-10h}"

RENEW\_LIFETIME="${5:-30d}"

LOGFILE="${6:-/var/log/kerberos\_renew.log}"

# ---------------------------

# Validation

# ---------------------------

if [[ -z "$PRINCIPAL" || -z "$KEYTAB" || -z "$CCACHE" ]]; then

echo "Usage: $0 <principal> <keytab> <ccache> [lifetime] [renew\_lifetime] [logfile]" >&2

exit 1

fi

mkdir -p "$(dirname "$CCACHE")"

mkdir -p "$(dirname "$LOGFILE")"

# ---------------------------

# Logging Helper

# ---------------------------

log() {

echo "$(date '+%Y-%m-%d %H:%M:%S') [${1}] $2" >> "$LOGFILE"

}

# ---------------------------

# Main Logic

# ---------------------------

if ! klist -c "$CCACHE" -s; then

log "INFO" "No valid TGT. Getting new TGT for $PRINCIPAL"

if kinit -k -t "$KEYTAB" -c "$CCACHE" -l "$LIFETIME" -r "$RENEW\_LIFETIME" "$PRINCIPAL"; then

log "SUCCESS" "New TGT acquired for $PRINCIPAL"

else

log "ERROR" "Failed to obtain new TGT for $PRINCIPAL"

exit 2

fi

else

log "INFO" "Attempting TGT renewal for $PRINCIPAL"

if kinit -R -c "$CCACHE"; then

log "SUCCESS" "TGT renewed for $PRINCIPAL"

else

log "WARN" "Renewal failed; requesting new TGT for $PRINCIPAL"

if kinit -k -t "$KEYTAB" -c "$CCACHE" -l "$LIFETIME" -r "$RENEW\_LIFETIME" "$PRINCIPAL"; then

log "SUCCESS" "New TGT acquired for $PRINCIPAL"

else

log "ERROR" "Failed to obtain new TGT after renewal failure for $PRINCIPAL"

exit 3

fi

fi

fi

Make executable:

bash

CopyEdit

chmod +x /opt/kerberos/renew\_kerberos.sh

**✅ 2. Systemd Service and Timer**

Create these files under /etc/systemd/system/

**kerberos-renew-<account>.service**

ini

CopyEdit

[Unit]

Description=Kerberos ticket renewal for service account <account>

[Service]

Type=oneshot

ExecStart=/opt/kerberos/renew\_kerberos.sh service\_account@YOUR.DOMAIN.COM /opt/kerberos/service\_account.keytab /tmp/krb5cc\_service\_account 10h 30d /var/log/kerberos\_renew\_<account>.log

**kerberos-renew-<account>.timer**

ini

CopyEdit

[Unit]

Description=Runs Kerberos renewer for <account> every 30 min

[Timer]

OnBootSec=1min

OnUnitActiveSec=30min

Persistent=true

[Install]

WantedBy=timers.target

Enable and start:

bash

CopyEdit

systemctl daemon-reexec

systemctl daemon-reload

systemctl enable --now kerberos-renew-<account>.timer

**✅ 3. Logrotate (optional)**

Create /etc/logrotate.d/kerberos\_renew

conf

/var/log/kerberos\_renew\_\*.log {

weekly

rotate 4

compress

missingok

notifempty

create 0640 root adm

}

**✅ 1. Shared Multi-Account Renewal Script**

Update /opt/kerberos/renew\_all.sh to loop through multiple service accounts:

bash

CopyEdit

#!/bin/bash

set -euo pipefail

# ---------------------------

# Configuration: One line per service account

# Format: principal:keytab:ccache:lifetime:renew\_lifetime:logfile

# ---------------------------

ACCOUNTS=(

"svc\_sql@DOMAIN.COM:/opt/kerberos/svc\_sql.keytab:/tmp/krb5cc\_svc\_sql:10h:30d:/var/log/kerberos\_renew\_svc\_sql.log"

"svc\_report@DOMAIN.COM:/opt/kerberos/svc\_report.keytab:/tmp/krb5cc\_svc\_report:10h:30d:/var/log/kerberos\_renew\_svc\_report.log"

)

RENEW\_SCRIPT="/opt/kerberos/renew\_kerberos.sh"

for entry in "${ACCOUNTS[@]}"; do

IFS=':' read -r PRINCIPAL KEYTAB CCACHE LIFETIME RENEW\_LIFETIME LOGFILE <<< "$entry"

"$RENEW\_SCRIPT" "$PRINCIPAL" "$KEYTAB" "$CCACHE" "$LIFETIME" "$RENEW\_LIFETIME" "$LOGFILE" || \

echo "$(date '+%Y-%m-%d %H:%M:%S') [ERROR] Renewal failed for $PRINCIPAL" >> "$LOGFILE"

done

Make it executable:

bash

CopyEdit

chmod +x /opt/kerberos/renew\_all.sh

**✅ 2. Systemd Unit Files for All Accounts**

**/etc/systemd/system/kerberos-renew.service**

ini

CopyEdit

[Unit]

Description=Kerberos TGT renewal for multiple service accounts

[Service]

Type=oneshot

ExecStart=/opt/kerberos/renew\_all.sh

**/etc/systemd/system/kerberos-renew.timer**

ini

CopyEdit

[Unit]

Description=Run Kerberos ticket renewal every 30 minutes

[Timer]

OnBootSec=1min

OnUnitActiveSec=30min

Persistent=true

[Install]

WantedBy=timers.target

Enable and start:

bash

CopyEdit

systemctl daemon-reexec

systemctl daemon-reload

systemctl enable --now kerberos-renew.timer

**✅ 3. Logrotate Support**

bash

CopyEdit

cat <<EOF > /etc/logrotate.d/kerberos\_renew

/var/log/kerberos\_renew\_\*.log {

weekly

rotate 4

compress

missingok

notifempty

create 0640 root adm

}

EOF

**🔄 Result**

This setup will:

* Renew Kerberos tickets for all service accounts listed
* Log separately per account
* Automatically start at boot and run every 30 minutes
* Rotate logs weekly

Let me know if you want it to email on failure or retry more aggressively.

To enable **email notifications on failure** in your multi-account Kerberos renewal setup, follow this guide. We’ll configure:

* 📨 Email alerts when **any account renewal fails**
* 📥 Uses mail or mailx (installed via mailutils or bsd-mailx)
* 🔧 Minimal logic changes to /opt/kerberos/renew\_all.sh

**✅ 1. Install mail Utility**

On most Linux distros:

**Debian/Ubuntu:**

bash

CopyEdit

sudo apt update && sudo apt install mailutils

**RHEL/CentOS/Rocky:**

bash

CopyEdit

sudo yum install mailx

**✅ 2. Set Notification Email**

At the top of /opt/kerberos/renew\_all.sh, define who gets alerted:

bash

CopyEdit

ADMIN\_EMAIL="admin@example.com"

You can also make this configurable via environment variable if needed.

**✅ 3. Update /opt/kerberos/renew\_all.sh with Email on Failure**

bash

CopyEdit

#!/bin/bash

set -euo pipefail

ADMIN\_EMAIL="admin@example.com"

RENEW\_SCRIPT="/opt/kerberos/renew\_kerberos.sh"

ACCOUNTS=(

"svc\_sql@DOMAIN.COM:/opt/kerberos/svc\_sql.keytab:/tmp/krb5cc\_svc\_sql:10h:30d:/var/log/kerberos\_renew\_svc\_sql.log"

"svc\_report@DOMAIN.COM:/opt/kerberos/svc\_report.keytab:/tmp/krb5cc\_svc\_report:10h:30d:/var/log/kerberos\_renew\_svc\_report.log"

)

for entry in "${ACCOUNTS[@]}"; do

IFS=':' read -r PRINCIPAL KEYTAB CCACHE LIFETIME RENEW\_LIFETIME LOGFILE <<< "$entry"

if ! "$RENEW\_SCRIPT" "$PRINCIPAL" "$KEYTAB" "$CCACHE" "$LIFETIME" "$RENEW\_LIFETIME" "$LOGFILE"; then

ERROR\_MSG="$(date '+%Y-%m-%d %H:%M:%S') [ERROR] Renewal failed for $PRINCIPAL"

echo "$ERROR\_MSG" >> "$LOGFILE"

# Send email

echo -e "$ERROR\_MSG\n\nSee log: $LOGFILE\nHost: $(hostname)" | \

mail -s "Kerberos Renewal Failed: $PRINCIPAL on $(hostname)" "$ADMIN\_EMAIL"

fi

done

**✅ 4. Optional: Use Mail Relay (Postfix or ssmtp)**

If your server can't send mail directly, configure it to relay via your mail server:

**Example: /etc/ssmtp/ssmtp.conf**

ini

CopyEdit

root=admin@example.com

mailhub=smtp.example.com:587

AuthUser=alert@example.com

AuthPass=YourAppPassword

UseTLS=YES

FromLineOverride=YES

Or use postfix with relay host config.

**✅ 5. Test Email Manually**

bash

CopyEdit

echo "Kerberos test email body" | mail -s "Kerberos Test" admin@example.com

**🧪 Result**

* Every 30 minutes, Kerberos renews for all service accounts
* If any fail:
  + A clear message is logged
  + An **email alert** is sent with principal, host, and log location