■ A+ Lab: Time Drift & Log Correlation Breakdown

Generated: 2025-08-03 00:09 UTC

■ Goal

Demonstrate how time mismatches between systems affect log correlation, Kerberos authentication, and troubleshooting.

■ Lab Setup

3 Ubuntu VMs (Vagrant): logserver, client1, client2.
- client1: normal time
- client2: time drifted (e.g., +5 mins)
- Logs sent to logserver via rsyslog

■ Observable Symptoms

- Logs appear out of order
- Authentication failures if using Kerberos
- TLS/SSL errors if time is skewed badly

■ What to Look For

- Compare timestamps in /var/log/client1/* vs client2/*
- Look for sudo, sshd, rsyslog events with mismatched times

■ Recovery Steps

- Use `timedatectl` or `w32tm` (Windows) to re-enable NTP
- Confirm with `timedatectl status` or `w32tm /query /status`

■ Teaching Prompts

- How do systems detect time drift?
- What breaks in enterprise environments?
- How can a SIEM normalize timestamps?
- What role does NTP play in incident response?

■ Student Challenges

- Write a script to compare timestamps across logs
- Fix the time on a drifted system
- Explain why log correlation is unreliable with unsynced clocks
- Recreate Kerberos login failure using Windows

■ Extra Activities

- Add custom log tags with `logger`
- Simulate TLS failures (visit https site with wrong clock)
- Use `ntpq -p` or `chronyc sources` to view time peers