Day 18

Exploitation Analyst

Hacking NTP Protocol:

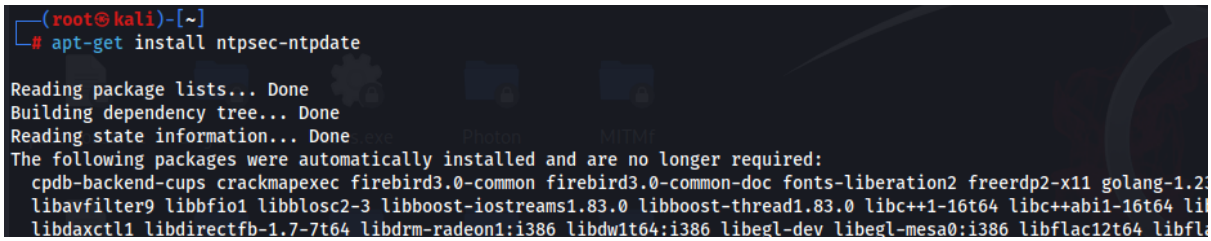
Why NTP is not secure?

1. **Lack of Authentication by Default**
   - Standard NTP does **not verify** the identity of time servers.
   - Attackers can **spoof time servers** and send fake timestamps.
   - Can disrupt:
     - Certificate validity
     - Log timelines
     - Time-based authentication systems (e.g., Kerberos, 2FA)
2. **Vulnerable to MITM (Man-in-the-Middle) Attacks**
   - NTP uses **UDP (port 123)** – connectionless and easily spoofable.
   - Attackers can intercept or **modify NTP replies** in transit.
3. **DDoS Amplification Attacks**
   - Older NTP servers supported **monlist**, which reveals recent clients.
   - Attackers exploited it to send **large replies** to spoofed victim IPs.
   - Resulted in massive **DDoS amplification attacks**.
4. **Implementation Vulnerabilities (CVE Examples)**
   - Various bugs have been found:
     - **CVE-2013-5211**: monlist abuse for DDoS
     - **CVE-2014-9295**: Buffer overflow in control mode
     - **CVE-2020-11868**: Authentication bypass
     - **CVE-2015-7704**: Spoofing via crafted packets
5. **No Built-in Encryption**
   - NTP traffic is **unencrypted**, exposing it to:
     - Packet sniffing
     - Replay or modification attacks

NTP Enumeration

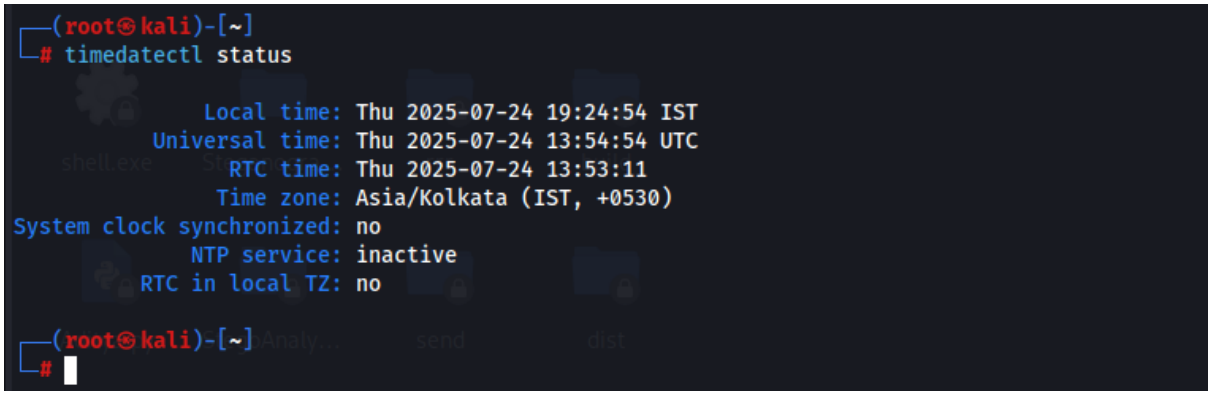
Steps:

First install the ntpdate tool:



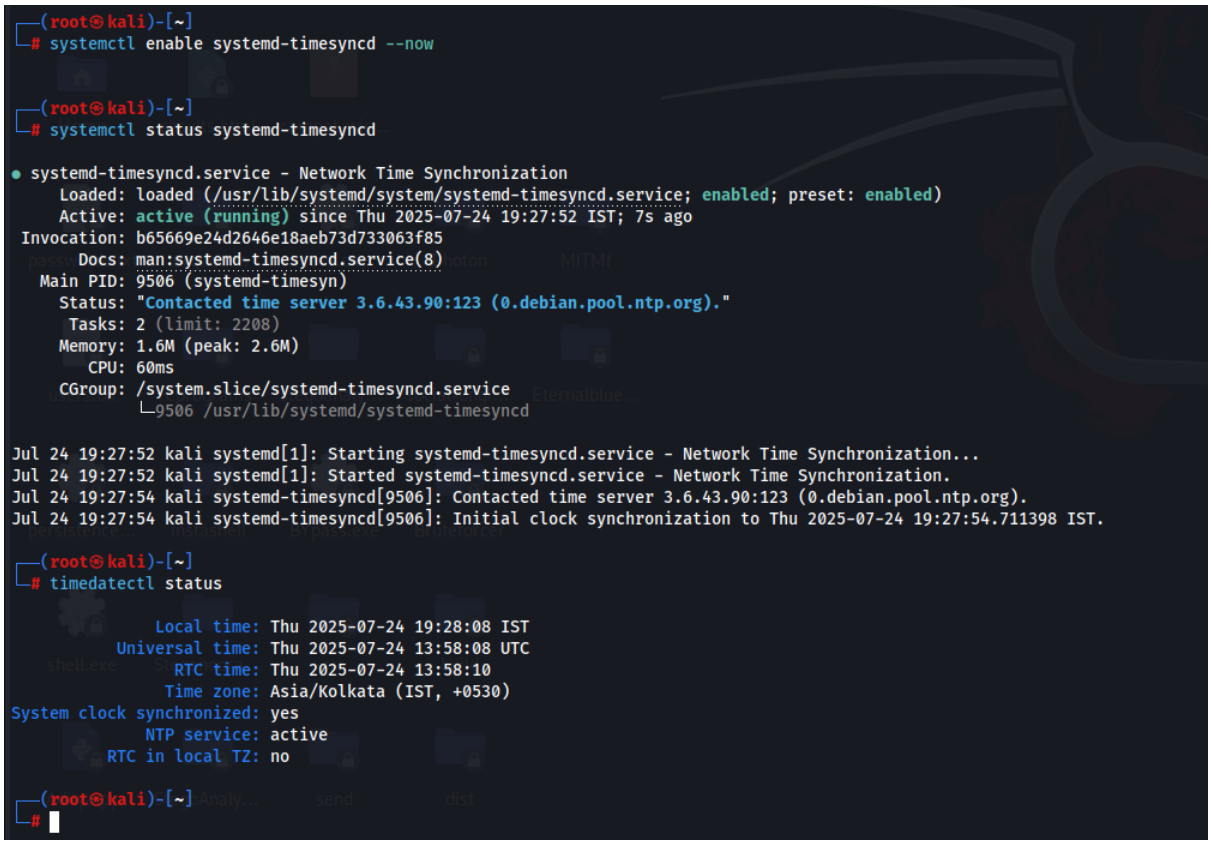
This command installs the ntpsec-ntpdate utility, a modernized and more secure replacement for the legacy ntpdate tool. It is part of the NTPsec project, which enhances security and reduces attack surfaces in time synchronization tools. Installing this allows you to manually query or update the time from an NTP server using ntpdate-like syntax.

Then check status of the NTP: it is inactive now.



The timedatectl status command displays the current system time, time zone, real-time clock (RTC), and NTP synchronization status. It is a key diagnostic tool to verify whether your Linux system is currently synchronized with an NTP source. It helps confirm if NTP is active and if the system clock is aligned with UTC.

To make it active use the following commands:



Now, look if the target allows the use of ntpdate: yes it does.

```
┌──(root㉿kali)-[~]
└─# ntpdate -q 192.168.1.103

2025-07-24 19:48:35.054798 (+0530) -0.002011 +/- 0.001393 192.168.1.103 s4 no-leap

┌──(root㉿kali)-[~]
└─#
```

This command queries the NTP server at IP 192.168.1.103 to retrieve time information without actually setting the local system time (-q for query only). It helps validate if the remote NTP server is responding correctly and whether it's eligible as a time source. It's commonly used for NTP testing or troubleshooting.

Now, we will run command to see the client's information: It gave us the MAC address. But no useful information obtained.

```
┌──(root㉿kali)-[~]
└─# nmap -sU -pU:123 -Pn -n --script=ntp-monlist 192.168.1.103
Starting Nmap 7.95 ( https://nmap.org ) at 2025-07-24 19:56 IST
Nmap scan report for 192.168.1.103
Host is up (0.00092s latency).

PORT    STATE SERVICE
123/udp open  ntp
MAC Address: 08:00:27:AC:56:B6 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 5.29 seconds

┌──(root㉿kali)-[~]
└─#
```

This advanced Nmap command performs a UDP scan (-sU) on port 123 (used by NTP) of the target server while skipping host discovery (-Pn) and DNS resolution (-n). The --script=ntp-monlist runs a specific Nmap script that checks if the NTP server allows monlist queries—a feature once used for monitoring but now considered a vulnerability as it can be abused in amplification DDoS attacks. If successful, it would list the IPs that have recently queried the server.

How to protect NTP?

1. Disable monlist or Use ntpsec / chrony

- The monlist feature in older NTP daemons is vulnerable to abuse (CVE-2013-5211), allowing reflection attacks.
- To avoid this, prefer using more secure implementations like ntpsec or chrony, which do not support monlist by default.
- If using the classic ntpd, disable monlist by adding the following line to /etc/ntp.conf:

  disable monitor

2. Use a Firewall to Restrict Access

- Limit access to NTP (UDP port 123) using a firewall to prevent unauthorized or external traffic.

- Only allow trusted IP addresses to send NTP queries.
- Example command using UFW:

sudo ufw deny in on eth0 to any port 123 proto udp

3. Rate Limit NTP Requests

- Use rate limiting to prevent abuse and reduce the risk of denial-of-service attacks through excessive UDP queries.
- Example using iptables to limit NTP traffic:

iptables -A INPUT -p udp --dport 123 -m limit --limit 5/second -j ACCEPT

4. Prevent Spoofing and Reflection

- Do not allow the server to respond to unauthenticated or spoofed queries, which can be used in reflection attacks.
- Modify your /etc/ntp.conf to restrict access using:

restrict default noquery notrap nomodify

restrict 127.0.0.1

restrict ::1

5. Use Authentication

- NTP supports authentication using symmetric keys or public key infrastructure (Autokey).
- To enable authentication:
    - Add keys to /etc/ntp.keys
    - Reference the keys in /etc/ntp.conf:

keys /etc/ntp.keys

trustedkey 1

6. Keep NTP Software Updated

- Regularly update your NTP software (ntp, ntpsec, or chrony) to ensure you are protected against known vulnerabilities and exploits.

--The End--