

Ralf Spenneberg

Linux-Firewalls mit iptables & Co.

Sicherheit mit Kernel 2.4 und 2.6
für Linux-Server und -Netzwerke



 ADDISON-WESLEY

An imprint of Pearson Education

München • Boston • San Francisco • Harlow, England
Don Mills, Ontario • Sydney • Mexico City
Madrid • Amsterdam



11 Zentrale Zeitsynchronisation

Wenn Sie Ihre Firewall-Protokolle später auswerten möchten und vielleicht auch noch mit Protokollen Ihres Webservers und Ihres Intrusion-Detection-Systems korrelieren möchten, ist es wichtig, dass die Protokolle über eine einheitliche Zeitbasis verfügen. Um dies zu erreichen, führt kein Weg an einem zentralen Zeitserver vorbei, der die Systeme mit der korrekten Uhrzeit versorgt. Dieses Kapitel zeigt Ihnen, wie Sie einen Zeitserver aufsetzen und Ihre Systeme damit synchronisieren.

11.1 Das Zeitsynchronisationsprotokoll NTP

Viele Systemadministratoren verschwenden keinen Gedanken an zentrale Zeitsynchronisation. Dabei ist die zentrale Zeitsynchronisation sämtlicher Netzwerkkomponenten im Falle eines sicherheitsrelevanten Ereignisses (Incident) von größter Bedeutung. Wie wollen Sie die Protokolle auswerten, wenn die Zeiten in den Protokollen des angegriffenen Webservers, der Datenbank, der Firewall und den Intrusion-Detection-Systems nicht zusammenpassen? Wollen Sie raten, was zuerst passiert ist? Auch die normale Auswertung von Protokollen ist bei nicht synchroner Zeit schwierig. Wie möchten Sie die Laufzeit einer E-Mail über zwei oder drei Mail-Relays ermitteln, wenn alle Systeme eine unterschiedliche Zeitbasis verwenden?

Um dieses Problem zu lösen, ist bereits vor langer Zeit das Network Time Protocol (NTP) geschaffen worden. Aktuell ist die Version 4 des Protokolls, das mit der Version 3 (RFC 1305) und den Versionen 1 (RFC1059) und 2 (RFC1119) kompatibel ist. Die wesentliche Neuerung des Protokolls Version 4 ist die Unterstützung von asymmetrischer Kryptographie und IPv6. Die Version 4 ist bisher nicht in einem RFC beschrieben worden. Es existiert lediglich ein Draft (<http://tools.ietf.org/wg/ntp/draft-ietf-ntp-ntp4-proto/draft-ietf-ntp-ntp4-proto-00.txt>), der das Protokoll beschreibt. Alternativ zu dem NTP-Protokoll existiert auch das Simple-Network-Time-Protocol (SNTP), das dem NTP-Protokoll ähnlich ist, aber weniger Funktionen besitzt und einfacher zu implementieren ist.

Das NTP-Protokoll verwendet zum Transport der Informationen den UDP-Port 123. Ungewöhnlicherweise verwendet das Protokoll diesen Port sowohl als Client wie auch als Server. Die Zeitinformationen können per Unicast, Multicast und Broadcast verteilt werden. Leider bietet das UDP-Protokoll keinerlei Schutz vor Spoofing oder Modifikation der transportierten Informationen. Hierfür kann ab der Version 4 die asymmetrische Kryptographie zum Schutz der Integrität und Authentizität eingesetzt werden.

Das NTP-Protokoll wird beim Zugriff eines NTP-Clients auf einen NTP-Server eingesetzt. Als NTP-Server kann jedes System eingesetzt werden, das über eine genaue Zeit verfügt. Die Zeitquelle kann eine DCF-77-Uhr sein, wie zum Beispiel die Expert Mouseclock von GUDE ANALOG- und DIGITALSYSTEME GmbH (<http://www.gude.info>). Alternativ können Sie auch eine GPS-Maus nutzen. Achten Sie nur darauf, dass das Produkt über Linux-Unterstützung verfügt. Natürlich können Sie auch einen oder mehrere andere NTP-Server als Zeitquelle nutzen. Eine Liste von öffentlich frei verfügbaren NTP-Servern finden Sie auf <http://www.pool.ntp.org/> und <http://ntp.isc.org/bin/view/Servers/WebHome>. Die Server werden in Stratum 1,2,3- etc. Server eingeteilt. Ein Stratum-1-Server ist direkt mit einer Hochpräzisionszeitquelle (z.B. Atomuhr) verbunden. Ein Stratum-3-Server synchronisiert sich mit einem Stratum-2-Server, der sich wieder mit einem Stratum-1-Server synchronisiert. Je weiter weg sich ein Server von einem Stratum-1-Server befindet, desto ungenauer ist seine Zeit. Wenn Sie selbst einen Server mit DCF-77-Uhr aufsetzen, ist dies ein Stratum-1-Server. Alle Systeme, die diesen Rechner zur Synchronisation verwenden, sind Stratum-2-Systeme.

Wenn Sie für die Synchronisation Ihres NTP-Servers weitere NTP-Server im Internet verwenden, achten Sie bitte darauf, dass Sie mehrere global verteilte Systeme verwenden. Ansonsten besteht die Gefahr, dass beim Ausfall eines Systems oder bei einer falschen Uhrzeit auf diesem System Ihre Zeitsynchronisation in Mitleidenschaft gezogen wird. Alle Systeme können mehrere Zeitserver zur Synchronisation einsetzen!

11.2 Der ntpd-Zeitserver

Die am häufigsten eingesetzte Software für den Aufbau eines Zeitserver auf der Basis von Linux ist der `ntpd` (früher `xntpd`). Er ist in den meisten Distributionen enthalten und wird von dem Internet Software Consortium entwickelt und gepflegt (<http://ntp.isc.org>). Eine weitere Möglichkeit für den Aufbau eines NTP-Servers ist die Software `openntp` von dem OpenBSD-Projekt (<http://www.openntp.org>). Dieser Zeitserver ist wesentlich einfacher in seinen Funktionen und in vielen Konfigurationseinstellungen kompatibel. Er unterstützt bisher jedoch nicht NTP Version 4 und damit auch keine Authentifizierung.

Da der `ntpd` in allen aktuellen Distributionen enthalten ist, spare ich mir hier die Beschreibung der Installation aus den Quellen. Sie finden eine Installationsanleitung in dem Quellpaket, falls Sie tatsächlich das Paket manuell installieren möchten. Ansonsten stellen Sie bitte sicher, dass das `ntp`-Paket Ihrer Distribution installiert ist.

Zunächst betrachten wir die Konfiguration des NTP-Servers als Client und anschließend als Server.

11.2.1 Der Client

Das `ntp`-Paket enthält zwei Möglichkeiten, um ein System als Client mit einem Zeitserver zu synchronisieren. Der Kommandozeilenbefehl `ntpdate` führt eine einmalige

Synchronisierung durch. Der Server `ntpd` kann auch als Client eine ständige Synchronisierung des Clients mit einem NTP-Server ermöglichen. Sinnvoll ist daher der Einsatz des `ntpd` als Client, um die ständige Synchronisierung zu gewährleisten. Der `ntpdate`-Befehl wird jedoch in zukünftigen Versionen der Software entfernt werden, da er bereits jetzt durch den Aufruf `ntpd -q` ersetzt werden kann.

Hinweis



Wenn die Zeit des Clients um mehr als 1000 Sekunden von der Zeit des Servers abweicht, weigert sich der `ntpd`, die Zeit zu synchronisieren. Daher wird häufig der Befehl `ntpdate` vor dem Start des `ntpd` aufgerufen. Wenn Sie diesen Befehl durch `ntpd -q` ersetzen möchten, müssen Sie hier zusätzlich `-g` angeben. Diese Option schaltet die 1000-Sekunden-Prüfung ab.

Die Konfiguration des `ntpd` als Client ist sehr einfach. Erzeugen Sie lediglich die folgende Konfigurationsdatei `/etc/ntp.conf`:

```
# Erlaube per Default niemandem die Modifikation, die Abfrage oder
# das Monitoring des Zeitserverns
restrict default ignore

# Erlaube alle Funktionen über das Loopback-Interface
restrict 127.0.0.1

# Lokale Drift-Datei, muss schreibbar sein.
driftfile /var/lib/ntp/drift

# Expert Mouseclock
# Generiere Link: ln -s /dev/ttyS0 /dev/refclock-0
# server 127.127.8.0 mode 5

# Server im Internet
server 0.pool.ntp.org
server 1.pool.ntp.org
server 2.pool.ntp.org

# Lokale Uhr
server 127.127.1.0 # local clock
fudge 127.127.1.0 stratum 10
```

Ausgestattet mit dieser Konfigurationsdatei, können Sie Ihren NTP-Server bereits probierhalber starten. Natürlich sollten Sie die eingetragenen Server im Internet prüfen und bei Bedarf durch andere ersetzen. Um den NTP-Server probierhalber zu starten, geben Sie auf der Kommandozeile zunächst den Befehl `ntpd -q -g` ein. Dies

führt eine einmalige Synchronisation durch. Dieser Vorgang kann einige Sekunden dauern:

```
# ntpd -q -g
ntpd: time set -6.640906s
```

Anschließend geben Sie das Kommando `ntpd` auf der Kommandozeile ein. Es sollte sich scheinbar sofort beenden. Prüfen Sie, ob der Ntpd richtig im Hintergrund läuft, indem Sie sich die laufenden Prozesse anzeigen lassen.

Sie können die Funktion des Ntpd nun mit dem Befehl `ntpd` auf dem lokalen System überprüfen.

```
# ntpdc
ntpd> peers
  remote          local      st poll reach  delay  offset  disp
=====
=216.154.195.60  192.168.255.100  3  64   3  0.15869  0.003562  1.98438
=LOCAL(0)       127.0.0.1       10  64   3  0.00000  0.000000  1.98436
=gatekeeper.no-s 192.168.255.100  1  64   3  0.23141  0.000760  1.98444
=frigg.interstro 192.168.255.100  3  64   3  0.07040  0.001728  1.98438
ntpd>
```

Der Befehl `ntpd` erlaubt die komplette Administration des Ntpd-Servers. Sie können Server hinzufügen und entfernen, den aktuellen Zustand betrachten, Rechte ändern etc. Die Manpage gibt Ihnen weitere Auskünfte. Der Befehl `peers` ist der häufigste Befehl, den Sie wahrscheinlich brauchen. Sie können diesen Befehl auch direkt mit `ntpd -p` aufrufen. Dieser Befehl zeigt Ihnen die aktuellen Peers des Servers und ihre Zustände an. Die erste Spalte gibt Ihnen Auskunft über den Zustand des Peers:

- +: Symmetrisch aktiv. Der Rechner sendet regelmäßig Nachrichten an die Adresse und zeigt seine Synchronisationsfähigkeit.
- -: Symmetrisch passiv. Der Rechner empfängt symmetrisch aktive Nachrichten.
- =: Der Peer wird im Clientmode abgefragt.
- ^: Broadcasts werden an die Adresse versandt.
- ~: Broadcasts werden von dieser Adresse empfangen.
- *: Mit diesem Server erfolgt aktuell die Synchronisation.

Sobald Sie ein `*` in der ersten Spalte erkennen können, erfolgt eine aktive Zeitsynchronisation mit dem entsprechenden System.

```
# ntpdc -p
  remote          local      st poll reach  delay  offset  disp
=====
*216.154.195.60  192.168.255.100  3  64   37 0.15869  0.003562  0.66310
=LOCAL(0)       127.0.0.1       10  64   37 0.00000  0.000000  0.66202
```

11.3 Sicherheit

```
=gatekeeper.no-s 192.168.255.100 1 64 37 0.23141 0.000760 0.66327
=frigg.interstro 192.168.255.100 3 64 37 0.07040 0.001728 0.66351
```

Sobald der NTP-Server als Client zu Ihrer Zufriedenheit läuft, können Sie ihn mit `kill` beenden und über die Startskripten Ihrer Distribution starten. Prüfen Sie bitte, ob der Server anschließend auch tatsächlich läuft. Viele Distributionen verwenden einen eigenen Benutzer für den Betrieb des NTP-Servers und starten ihn in einem Chroot (siehe »Sicherheit«, 11.3). Dabei kann es zu Rechteproblemen kommen. Prüfen Sie die Protokolle Ihrer Distribution, um mögliche Fehlermeldungen zu finden.

11.2.2 Der Server

Die bisher erstellte Konfigurationsdatei erlaubt den Betrieb eines NTP-Servers als Client. Dieser Client wird sich nun ständig mit den verfügbaren Zeitquellen synchronisieren. Häufig möchten Sie aber auch einen eigenen Zeitserver betreiben, so dass sich weitere Clients mit diesem Zeitserver synchronisieren können.

Die bisherige Konfiguration erlaubt es keinem Client, Synchronisationsanfragen an diesen Server zu schicken, und der Server versendet auch keine Broadcast-Pakete. Die Verwendung des Broadcast- oder Multicast-Transports kann ich auch nur empfehlen, wenn Sie gleichzeitig eine Authentifizierung aktivieren (siehe Abschnitt 11.3). Die Gefahr eines gespoofen NTP-Angriffs ist ansonsten zu hoch.

Um einen Broadcast-Server zu konfigurieren, müssen Sie die Konfigurationsdatei nur um eine Zeile ergänzen:

```
broadcast 192.168.0.255
```

Wenn ein Client diesen Broadcast-Server nutzen soll, tragen Sie auf dem Client die folgenden beiden Zeilen ein:

```
broadcastclient
broadcastdelay 0.008
```

Um Unicast-Clients zu unterstützen, fügen Sie eine zusätzliche Zeile mit dem `restrict`-Parameter in der Konfigurationsdatei hinzu. Achten Sie darauf, dass Sie bei der Angabe den Clients nur die Abfrage der Synchronisationsinformationen erlauben.

```
restrict 192.168.0.0 mask 255.255.255.0 nomodify notrap
```

Nun dürfen Clients in dem Netzwerk 192.168.0.0/24 diesen NTP-Server als Zeitserver zur Synchronisation nutzen.

11.3 Sicherheit

In einem Firewall-Buch muss beim Einsatz von NTP auch über die Sicherheit des Protokolls, der Server und des Clients gesprochen werden. Dabei müssen zwei Aspekte betrachtet werden:

- Der NTP-Server verwendet den Port 123/udp. Dies ist ein privilegierter Port (< 1224), und er steht daher nur dem Benutzer root zur Verfügung. Der NTP-Server muss daher mit root-Rechten gestartet werden!
- Das NTP-Protokoll nutzt das UDP-Protokoll für den Transport der Informationen. Dieses Protokoll bietet keinen Schutz der Integrität und Authentizität. Ein Spoofing-Angriff ist leicht möglich.

Um die Sicherheit des Servers auf Port 123 zu erhöhen, ist der `ntpd` in der Lage, nach seinem Start die root-Privilegien abzugeben und in ein Chroot-Verzeichnis zu wechseln. Hiermit reduzieren Sie mögliche Gefahren bei einem Angriff auf den Dienst enorm.

Damit der `ntpd` nach dem Start die root-Privilegien abgibt, müssen Sie beim Start mit der Option `-u ntp:ntp` einen unprivilegierten Benutzer und eine unprivilegierte Gruppe übergeben. Sobald sich der NTP-Server an den Port 123/udp gebunden hat, wird er seinen Benutzerkontext entsprechend ändern.

Damit der NTP-Server in einem Chroot-Verzeichnis arbeitet, müssen Sie das Verzeichnis vorbereiten. Hierzu müssen Sie alle Dateien, die der `ntpd` für seine Funktion benötigt, in das Verzeichnis kopieren. Hierbei handelt es sich mindestens um die Datei `ntp.conf` und die Drift-Datei. In Abhängigkeit der Konfiguration sind auch noch die Verzeichnisse `/var/run`, `/var/log` und das `/dev`-Verzeichnis mit lokalen Zeitquellen erforderlich.

Anschließend starten Sie den NTP-Server mit der Option `-T /var/chroot-dir`. Diese Funktion erhöht die Sicherheit nur, wenn der Prozess auch die root-Privilegien abgibt!

Zum Schutz vor direkten Angriffen auf das Protokoll wie Spoofing unterstützt der NTP-Server in der Version 4 des NTP-Protokolls kryptographische Methoden. Um diese zu nutzen, müssen Sie zunächst Schlüssel erzeugen.

Am einfachsten ist die Authentifizierung mit symmetrischen Schlüsseln. Daher soll diese Authentifizierung hier zuerst besprochen werden.

11.3.1 Symmetrische Authentifizierung

Bei der symmetrischen Authentifizierung verfügen beide Authentifizierungspartner über identische Schlüssel. Dies ist sowohl ein Vorteil, denn die Verteilung ist sehr einfach, aber auch gleichzeitig ein Nachteil, da der Schlüssel vertraulich transportiert und gespeichert werden muss und der Austausch mit Dritten problematisch sein kann.

Es gibt vier verschiedene Arten von Schlüsseln:

1. A: Einen ASCII-Schlüssel aus maximal 8 Zeichen
2. M: Einen ASCII-Schlüssel mit maximal 31 Zeichen
3. S: Einen 64-Bit-Wert mit dem niedrigstwertigsten Bit pro Byte als Parität (DES)
4. N: Einen 64-Bit-Wert mit dem höchstwertigsten Bit pro Byte als Parität

Die Schlüssel A und M sind am einfachsten zu handhaben. Diese Schlüssel müssen nun in der Datei `/etc/ntp/keys` abgespeichert werden:

```
1 A ABCDEFGH
2 M qwertzuiopasdfghjklxyxcbnmqwer
```

Jeder Schlüssel in dieser Datei erhält eine Nummer und einen Typ. Über die Nummer des Schlüssels werden diese nun in der Konfigurationsdatei referenziert. Damit ein Client einen Schlüssel für die Verbindung zu einem Server nutzt, muss er zunächst dem Schlüssel vertrauen (`trust`) und wissen, welchen Schlüssel er für welchen Server verwenden soll:

```
trustedkey 1 2
server 192.168.0.5 key 1
server 192.168.0.7 key 2
```

Der Server benötigt identische Schlüssel und muss ebenfalls den Schlüsseln vertrauen. Wenn Sie die Zeit per Broadcast verteilen möchten, müssen Sie auf dem Server den Schlüssel in der `broadcast`-Zeile definieren und ebenfalls auf dem Client und dem Server den Schlüsseln vertrauen:

```
trustedkey 1 2
broadcast 192.168.0.255 key 1
```

Auf dem Client genügen dann die folgenden Zeilen:

```
broadcastclient
trustedkey 1 2
keys /etc/ntp/keys
```

11.3.2 Asymmetrische Authentifizierung

Die asymmetrische Authentifizierung ist in vielen Umgebungen die bessere Lösung. Sie müssen nun nur die öffentlichen Schlüssel (Public Keys) sämtlicher Systeme untereinander austauschen. So ist es auch einfach möglich, zu dritten Instanzen sichere Verbindungen aufzubauen.

Die asymmetrische Authentifizierung wird beim `ntpd` als *Autokey* bezeichnet. Bei der Konfiguration wird zwischen Broadcast- und Multicast-Autokey und Unicast-Autokey unterschieden. Während das Unicast-Autokey-Verfahren auf dem Client konfiguriert wird, wird das Broadcast- und Multicast-Autokey-Verfahren auf dem Server konfiguriert.

Für das Broadcast-Autokey-Verfahren tragen Sie auf dem Server folgende Zeile ein:

```
broadcast 192.168.0.255 autokey
```

Bei dem Unicast-Autokey-Verfahren tragen Sie auf dem Client folgende Zeile ein:

```
server 192.168.0.5 autokey
```

Zusätzlich benötigen beide Systeme noch die folgenden Zeilen:

```
crypto pw (client|server)password  
keysdir /etc/ntp
```

Dann müssen Sie noch die Schlüssel erzeugen und verteilen. Autokey unterstützt drei verschiedene Identitätsschemata: IFF, GQ und MV.

- **IFF.** Bei dem IFF-Identitätsschema werden für jeden Server spezifische Schlüssel erzeugt. Sie müssen diesen Schlüssel für jeden Client exportieren. Dabei kann der Schlüssel wieder mit einem Client-Passwort verschlüsselt transportiert werden.

Um die IFF-Parameter zu erzeugen, verwenden Sie:

```
cd /etc/ntp  
ntp-keygen -T -I -p serverpassword
```

- **GQ.** Bei dem GQ-Identitätsschema wird ein Schlüssel erzeugt, der von allen Systemen in der Gruppe genutzt wird. Die Erzeugung erfolgt mit:

```
cd /etc/ntp  
ntp-keygen -T -G -p serverpassword
```

- **MV.** Bei dem MV-Schema erzeugen Sie einen Schlüssel für den Server und N-1 Schlüssel für die Clients.

```
cd /etc/ntp  
ntp-keygen -V N -p serverpassword.
```

Für die Erzeugung der Parameter werden ein Server Key und ein Zertifikat erzeugt. Diese sind nur für ein Jahr gültig und müssen anschließend neu erzeugt werden. Dies erfolgt mit `cd /etc/ntp; ntp-keygen -T -q serverpassword`.

Auf dem Client müssen Sie zunächst auch einen Schlüssel und ein Zertifikat erzeugen. Anschließend müssen Sie die entsprechenden Schlüssel der Server nun auch auf dem Client importieren und installieren. Um den Schlüssel und das Zertifikat für den Client zu erzeugen, verwenden Sie:

```
cd /etc/ntp  
ntp-keygen -H -p clientpassword
```

- **IFF.** Bei den IFF-Gruppenschlüsseln müssen Sie diese auf den Servern zunächst exportieren. Dazu verwenden Sie den folgenden Befehl:

```
ntp-keygen -e -q serverpassword -p clientpassword
```

Diesen so mit dem Client-Passwort verschlüsselten Schlüssel können Sie nun per E-Mail verschicken oder per Diskette auf den Client transportieren. Speichern Sie den Schlüssel auf dem Client in dem Verzeichnis `/etc/ntp` ab, und erzeugen Sie einen symbolischen Link:

```
cd /etc/ntp
ln -s ntpkey_IFFkey_server.3301264563 ntpkey_iff_server
```

- **GQ.** Kopieren Sie den GQ-Gruppen-Schlüssel sicher auf den Client, und erzeugen Sie ebenfalls eine Verknüpfung.

```
d /etc/ntp
ln -s ntpkey_GQpar_server.3301145293 ntpkey_gq_server
```

- **MV.** Kopieren Sie den auf dem Server erzeugten Client-MV-Schlüssel auf den Client, und erzeugen Sie ebenfalls die Verknüpfung.

```
cd /etc/ntp
ln -s ntpkey_MVkey1_server.3301144193 ntpkey_mv_server
```

Nach einem Neustart der `ntpd`-Daemons auf den Servern und Clients sollten Sie mit dem Kommando `ntpq -p` den Zustand der Synchronisation beobachten.

Mit `ntpq -c` erhalten Sie Informationen über die Authentifizierung:

```
# ntpq -c
nd assID status conf reach auth condition last_event cnt
=====
1 26132 f694 yes yes ok sys.peer reachable 9
```

