Linuxによるセキュリティ入門（6）
── ntpで時刻を合わせる ──

西　村　竜　一

Ⅰ．はじめに

この原稿を書いているのは3月ですので，これが世に出るころにはもう過ぎてしまっているのですが，この3月で情報連携基盤センターの長谷川先生がご栄転により名古屋大学を退職されるそうです。私が名古屋大学を卒業してすでに5年を経過しますので，最近の事情は知りませんが，この原稿を書きながら昔のことを思い出しています。先生は，大学の巨大ネットワークを維持するという重責の下，当時から献身的な活動をされていました。私と当時の友人にはネットワークに異様な愛情？を持った若僧が何人かいて，ああするべきだ，とか，こうするべきだ，とか，生意気な意見を先生にぶつけて，いちいち聞いていただいたことを覚えています。その際には，先生には余計な時間を費やしていただくことになり，ご面倒をおかけしたのではないかと思います。この場をお借りして御礼いたします。ありがとうございました。

そして，心配しています。名古屋大学のネットワークはこれから大丈夫でしょうか？今後のセキュリティが心配だと言っているのではありません。噂では，名古屋大学でもセキュリティに関する対策が整いつつあると聞いてますし，きっと大丈夫でしょう（完璧はどこの組織でも無理ですし）。でも，規則やファイヤウォールで厳しくしばられたネットワークってやっぱり楽しくありませんよね。これからも名古屋大学から楽しい人材を輩出するためにも，後任の先生には，決してしばるのではなく，長谷川先生のように愛情を持ったネットワーク運営をお願いしたいです。正しい知識と愛情を持った専門家が機能的に活躍され，皆が幸せかつ健全にインターネットを利用できる環境を整えてくださると期待しております。長谷川先生，楽しいNICEをありがとうございました。

それでは本題に入ります。Linuxとセキュリティなんてタイトルを付けながら地味な話題ばかりをとり上げているこの連載ですが，それにはそれなりの理由があります。まず，流行りの目立つ部分のセキュリティネタに関する解説は書籍やWebなどにすでに多くあること。つぎに，システムの安定した運営の実現は，管理者の愛情に満ちた，常日頃の地味なメンテナンスに寄ることが多いと著者は信じているためです。どんなに工夫しても，結局のところ地味な作業はシステム管理からはなくなりません（現状では，そう信じています）。その地味な作業を少しでも楽にできたら良いなといった発想でネタを選んでいるので，内容も地味になりがちです。今回も地味な話に徹して，Linux上の時計の設定を解説します。楽に高い精度の時刻情報を得ようということで，NTP（Network Time Protocol）プログラムを導入してネットワーク上の時刻サーバを使った自

動時刻設定を紹介します[1]。

Ⅱ．Linuxと時計

前回までで解説した，システムのログには，アクセスされた時刻やアクセス元のIPアドレスなどが記録されています。最近生じた個人情報流出事件では，ログを一週間分しか保存していなかったため，肝心の犯行日時の手がかりが残っていないというお粗末なものもありましたが，それ以前の問題として，ログに残っている情報が正確でないとやはり意味がありません。ログに記録される時刻には，計算機の内部時計を元にした時刻が用いられます。しかし，この内部時計は精度の高いものではなく，不正確な時刻を保持していることが多々あります。これではログに記録された情報も価値がありません。時計を正しく設定しましょう。

Linuxに限らず，UNIX系のOSでは時計を設定するには`date`コマンドを用います。まず，現在の時刻を確認してみます。

```
% date
```

と，引数を付けないで実行することで，現在保持している時刻が表示されます。つぎに，時計の設定にも同じ`date`コマンドを用います。root権限が必要なので，`sudo`を併用します。

```
% sudo date MMDDhhmmYYYY
```

ここで`MM`は月，`DD`は日，`hh`は時，`mm`は分，`YYYY`は西暦を指定します。例えば，2004年４月１日10時00分に時計を設定するときは，

```
% sudo date 040110002004
```

となります。

これで時計の設定は完了としたいところです。しかし，少し面倒ですが，ハードウェアクロックとシステムクロックというLinuxシステム内に存在する２種類の時計のことも，理解しておきましょう。ハードウェアクロックはRTCやBIOSクロック，CMOSクロックなどとも呼ばれ，計算機内のLSIに搭載された時計です。計算機の電源が切れた状態でもバックアップ電池を用いて時を刻み続けます。一方で，システムクロックはソフトウェア的にLinuxカーネル内部に実装された時計です。先ほど，`date`コマンドで表示したり，設定したのは，このシステムクロックになります。ハードウェアクロックとシステムクロックの関係ですが，Linuxではシステムクロックがすべての

1　前回の最後では，メールを使って…と書いたのですが，予定を変更いたしました。期待されていた方がいらっしゃいましたら申し訳ございません。御意見や御要望等ありましたら，文末のアドレスまでメールでご連絡ください。

プログラムにおいて基準となります。普段はハードウェアクロックが参照されることはありません。ハードウェアクロックの役割はシステムが停止中にも時を刻み続けることであり，システムが起動時に一度，ハードウェアクロックを参照し，それを元にシステムクロックが設定されます。その後は，システムクロックのみを使うため，２つの時計は同期すらしていません。そのため，dateコマンドで設定した時刻も，このままではシステムを再起動するとハードウェアクロックの時刻に戻ってしまいます。これを防ぐために，設定した時刻をハードウェアクロックにも適用する必要があります。

それを実現するコマンドが，Debianの場合，hwclockです。Debianには，hwclockをさらに簡単に実行するためのスクリプト/etc/init.d/hwclock.shも用意されています。このスクリプトが起動時に実行され，システムクロックがハードウェアクロックをもとに設定されます。また，システムの再起動やシャットダウン時にも/etc/init.d/hwclock.shは実行され，システムクロックの時刻がハードウェアクロックに書き出されます。こうすることでシステムが停止している間も時刻は引き継がれ，ユーザは，ハードウェアクロックとシステムクロックの違いを意識することなく利用できるようになっています。

システムクロックの時刻をハードウェアクロックにもすぐに反映するには，

```
% sudo /etc/init.d/hwclock.sh reload
```

と上記のスクリプトを実行します。dateコマンドでシステムクロックを変更しても，クラッシュなどでhwclockがシステム停止時に正しく実行されないと，再起動後には古い時刻設定に戻ってしまいます。dateコマンドによる時刻の設定の後にこのスクリプトを実行しておくと，これを防ぐことができます。

Ⅲ．NTPを使ってみよう

さて，このように時刻設定をしても，ハードウェアクロック，システムクロックともにあまり精度が高くないため，やっぱり時刻は徐々にずれていきます。原子時計や電波時計，GPSの信号等を利用することで正確な時刻を得ることはできるので，これらの信号を参照する機械を用意すれば良いのですが，専用のハードウェアが必要でお手軽とはいえないので，本連載の主旨に沿いません。

そこで登場するのがNTP（Network Time Protocol）です[2]。NTPは，その名が示すとおり，ネットワーク上で時刻情報を伝えるためのプロトコルです。正確な時刻を持ったNTPサーバがネットワーク上に一台あれば，他の計算機がその情報を参照することで正確な時刻をネットワーク上で共有することが可能になります。また，これから紹介するNTPデーモンであるntpdを利用すれば，クライアント自身もNTPサーバとして動作します。つまり，NTPサーバから時刻を受信しながら，他の計算機に時刻情報を配信することができるのです。正確な時刻を配信しているNTPサ

2　http://www.ntp.org/

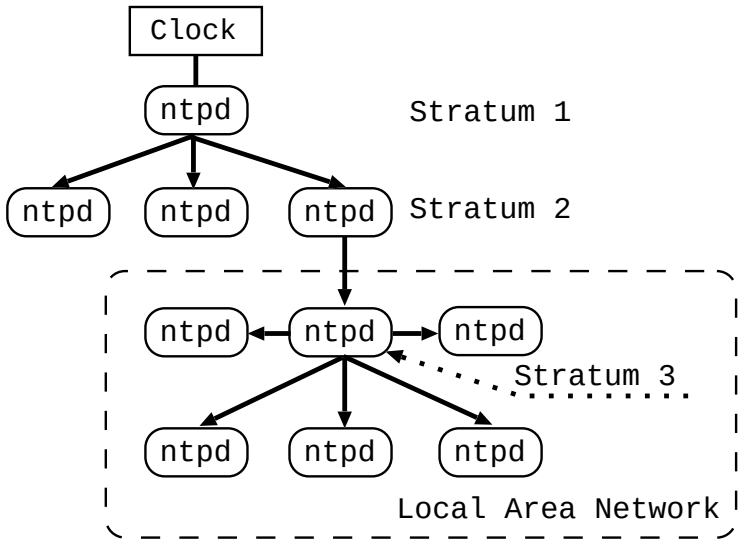

図１　NTPの階層構造：この例ではStratum２のntpdがインターネット上に公開されているNTPサーバに該当します。LAN内で他の計算機に時刻を配信するのはStratum３のntpdです。Stratum１のサーバが公開されているケースもあります。

ーバは，インターネット上で現在もいくつか公開されています。組織内で一台の計算機がそれらのNTPサーバから時刻情報を受信すれば，組織内の別の計算機は，その一台の計算機に対して時刻を取得すれば良く，インターネット上に無駄な通信を増やすことなく運用することが可能です。図１にNTPの階層構造のイメージを示してみました。すべての計算機上では`ntpd`が動いており，クライアントとしてのみでなくサーバとしても動作しているものとします。時計（Clock）を直接参照して時刻を得る計算機が階層構造の最も上位であり，NTPではこれをStratum１と呼びます。Stratum１を参照するのがStratum２といった感じにツリー状のネットワークが構成されていきます。この例では，LAN上には外部のNTPサーバから時刻を取得している計算機が一台あり（Stratum３），他の計算機はそれを参照するようになっています。ここで，組織内のすべての計算機が外部のNTPサーバを参照するような設定にしてはいけません。公開されているNTPサーバにはただでさえ多くのアクセスが集中します。可能な限り負荷を下げることができるように，一つの組織内からは限られたサーバだけ参照するように設定するのがマナーです。

Ⅳ．ntpdのインストール

それではNTPを利用するための必要なプログラムをインストールします。Debianでは，`ntpd`は`ntp-simple`，関連するプログラムは`ntp`というパッケージ名で提供されています。依存関係で`ntp`をインストールすれば必要なものはすべてインストールされるようになっているので，いつものように`apt-get`を使ってインストールします。

```
% sudo apt-get install ntp
```

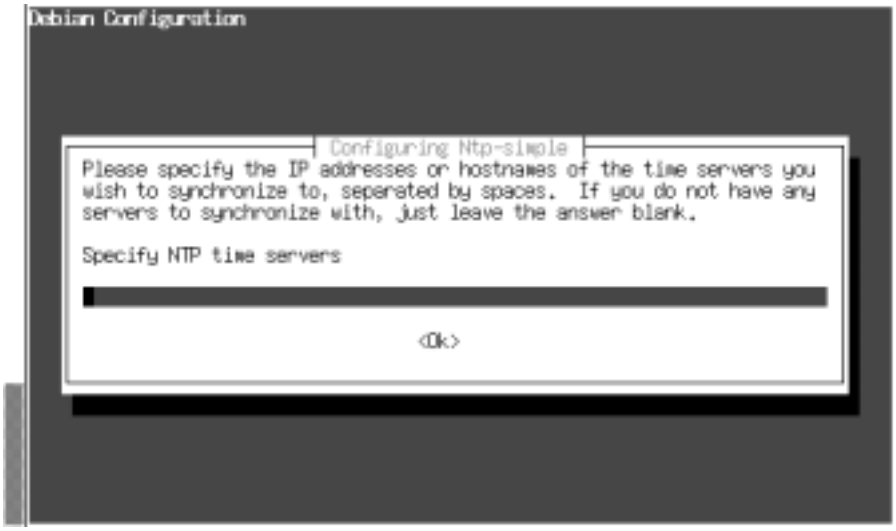

図２　ntp-simpleの設定（１）：無駄な負荷を生じさせないよう，最寄りのNTPサーバを使いましょう。

図３　ntp-simpleの設定（２）：はじめてntpを入れたときはYesを，/etc/ntp.confを手で修正している人はNoを選択してください。

続いて，図２と図３の設定になります。この二つの設定は`ntp-simple`パッケージが必要としているもので，

```
% sudo dpkg-reconfigure ntp-simple
```

で，再設定することができます。

表１　NTPサーバのリスト

ホストネーム（FQDN）	IPアドレス
ntp1.jst.mfeed.ad.jp	210.173.160.27
ntp2.jst.mfeed.ad.jp	210.173.160.57
ntp3.jst.mfeed.ad.jp	210.173.160.87
nu104.cc.nagoya-u.ac.jp	133.6.1.9

おのおのの設定の中身を見てみます。図２は時刻情報を参照するNTPサーバを指定する設定になります。もし使って良い手頃なNTPサーバをあなたが知らなかったら，周囲のネットワーク管理者の人にNTPサーバの存在をまず聞いてみてください。意外と近くにNTPサーバが運用されていることは良くあります。それでも使えるNTPサーバがない場合，公開されているPublic NTPサーバを使わせてもらうことになります。現在，公開されているPublic NTPサーバの中で，主に日本国内のネットワークからもっとも広く使われているのが独立行政法人通信総合研究所のグループにより実験的に公開されているStratum２サーバです[3]。ここで提供されているNTPサーバは，日本標準時を刻む原子時計に直結しているStratum１サーバを参照しており，非常に高精度な時刻情報を得ることができます。また，名古屋大学内ではnu104.cc.nagoya-u.ac.jpがこれとは別にStratum２の時刻情報を配信していますので，そちらを利用することもできます。表１にNTPサーバのFQDNとIPアドレスを示します[4]。図２のNTPサーバの指定では，これらの中から適当なものを設定するようにしてください。名古屋大学内からは，nu104.cc.nagoya-u.ac.jpを指定するのが良いようです。

つぎの図３は，`ntpd`の設定ファイルである`/etc/ntp.conf`を上書きして良いかの確認です。とりあえず，最初のインストールの時はYesを選んでください。後から直接ファイルを編集して設定の変更を行いますが，`/etc/ntp.conf`が上書きされては困る時はNoを選びます。

以上で，NTPの基本的なインストールは完了です。NTPではネットワークの遅延をキャンセルできるため，高精度の時刻情報をこのように手軽に利用できるのが特徴です。`date`コマンドで表示される時間は正確なものになりましたか？

Ⅴ．ntpdの動作確認

と，書きましたが，NTPはネゴシエーションに少し時間を必要とするので，ntpd起動後のすぐには動作を確認することはできません。ntpdの動作の確認には，`ntpq`コマンドを使います。

```
% ntpq -p
```

と実行して，

3　http://www.jst.mfeed.ad.jp/

4　これらの時刻配信のサービスは原稿執筆時に動作を確認したものであり，皆さんが利用する際には運用を終了している可能性もあります。

```
     remote           refid      st t when poll reach   delay offset jitter
==============================================================================
*nu104.cc.nagoya ntp.nc.u-tokyo.  2 u   82  128  377  19.567 -18.827 2.669
```

といった感じの表示がされれば動作は正常です。一番左端の*（アスタリスク）が，NTPサーバと時刻を同期していることを示しています。ここでremoteは参照しているNTPサーバ名，refidはNTPサーバが参照している情報源（不明のときは0.0.0.0），stは階層（Stratum）数を示します。Stratum数は，あまり大きくなると精度が悪くなるので４程度になるようにNTPサーバの参照の仕方を考えてください。紙面の都合と筆者の理解不足から，残りの表示内容については説明を省略します。

```
% sudo apt-get install ntp-doc
```

で，NTPに関するドキュメント（HTMLファイル）が/usr/share/doc/ntp-doc/以下に展開されるので，詳しくはそちらを参照してください。

ntpq -pを実行した後，

```
    remote           refid     st twhen poll reach   delay   offset  jitter
==============================================================================
nu104.cc.nagoya 0.0.0.0         0 u    -   64    0     0.000    0.000 4000.00
```

といった感じの表示になったときは，まだ同期は確立できていません。30分程度待っても同期しないようだったらサーバの指定が間違っていることが考えられるので，設定を見直しましょう。

また，自分自身と参照先のNTPサーバとの時刻が大きく異なると同期は確立できず，ntpdは異常終了してしまいます。そんな場合は，システムクロックを修正してからntpdを再起動する必要があります。dateコマンドで修正しても良いのですが，NTPプロトコルを使用して時刻を強制的に修正するntpdateというコマンドもあるのでそちらを使ってみます。

```
% sudo apt-get install ntpdate
```

と，apt-getを使ってntpdateをインストールした後，

```
% sudo /etc/init.d/ntpdate start
```

を実行することで，先ほどのntp-simpleの設定で指定したNTPサーバに合わせてシステムクロ

ックを修正してくれます。このとき，NTPサーバには/etc/default/ntp-serversに記述されているものが使われます。これはdpkg-reconfigure ntpdateで他のサーバに変更することが可能です。最後に,

```
% sudo /etc/init.d/ntp restart
```

のようにntpdを再起動すれば完了です。

ntpqとならんでNTPに動作確認に便利なのが，ntptraceコマンドです。実行すると,

```
% ntptrace

localhost:stratum 3,offset 0.000094,synch distance 0.97183
nu104.cc.nagoya-u.ac.jp:stratum 2,offset-0.001855,synch distance 0.02119
ntp.nc.u-tokyo.ac.jp:stratum 1,offset 0.003071,synch distance 0.00259,refid 'GPS'
```

このようにNTPの参照関係をStratum 1までトレースした結果を表示してくれます。

VI．NTPの少しだけ高度な設定

ntpdの設定ファイルは，/etc/ntp.confです。高度な設定をする際には，直接このファイルを編集する必要があります。編集した後は，ntpdを再起動するのを忘れないようにしてください。

図４を例にとって簡単に設定ファイルの中身を見てみましょう。#（シャープ）で，はじまる行

```
# /etc/ntp.conf,configuration for ntpd

# ntpd will use syslog() if logfile is not defined
#logfile /var/log/ntpd

driftfile /var/lib/ntp/ntp.drift
statsdir /var/log/ntpstats/

statistics loopstats peerstats clockstats
filegen loopstats file loopstats type day enable
filegen peerstats file peerstats type day enable
filegen clockstats file clockstats type day enable

### lines starting 'server' are auto generated,
### use dpkg-reconfigure to modify those lines.

server nu104.cc.nagoya-u.ac.jp
```

図４　/etc/ntp.confファイルの例

はコメントです。ここでは最後の行を確認してください。serverに続いてNTPサーバが記述されています。このNTPサーバの指定は複数のものを列挙することができ，その場合，ntpdがネットワークの遅延などを考慮して，同期するNTPサーバを与えられたものの中から選択します。もし一つのNTPサーバとなんらかの原因で通信できなくなっても，他のものが自動的に使われるので，いくつか列挙しておくと良いでしょう。以下のようになります。

```
server nu104.cc.nagoya-u.ac.jp
server ntp1.jst.mfeed.ad.jp
server ntp2.jst.mfeed.ad.jp
server ntp3.jst.mfeed.ad.jp
```

また，これらの中から使用したいサーバを明示するときは，preferというオプションを付けて以下のようにします。

```
server nu104.cc.nagoya-u.ac.jp prefer
server ntp1.jst.mfeed.ad.jp
server ntp2.jst.mfeed.ad.jp
server ntp3.jst.mfeed.ad.jp
```

preferは，ネットワーク的に最も近いサーバに対して設定するのが良いでしょう。

serverは上位のStratumに対して使用しますが，もう一つのNTPサーバの設定方法として，peerを使って同じStratum同士を互いに参照させることができます。例えば，nu104.cc.nagoya-u.ac.jpを上位Stratumに持つhost1と，ntp1.jst.mfeed.ad.jpを上位に持つhost2があるとします。このとき，host1とhost2を互いにpeerすることで，もし上位との通信になんらかの障害が生じたときにも，高い精度で時刻を維持することができます。host1とhost2の/etc/ntp.confファイルは以下のとおりになります。

・host1の設定

```
server nu104.cc.nagoya-u.ac.jp
peer host2
```

・host2の設定

```
server ntp1.jst.mfeed.ad.jp
peer host1
```

同じ上位Stratumを持つNTPサーバ同士をpeer設定すると，互いが同期して時刻のずれが生じます。高い時刻精度を維持するためには，同じ上位Stratumを持つNTPサーバ同士はpeerしてはならないといわれています。また，自分より上位や下位のStratumのNTPサーバとのpeer設定をしてもいけません。マナーを守ってNTPをご利用ください。

Ⅶ. おわりに

以上でntpdの基本的な設定は終わりです。みなさんのところでも正確な時間を得ることはできたでしょうか？次回は，セキュリティに関する小ネタ集をお届けしたいと思います。

（にしむら　りゅういち：奈良先端科学技術大学院大学情報科学研究科）
（nisimura@linux.or.jp）