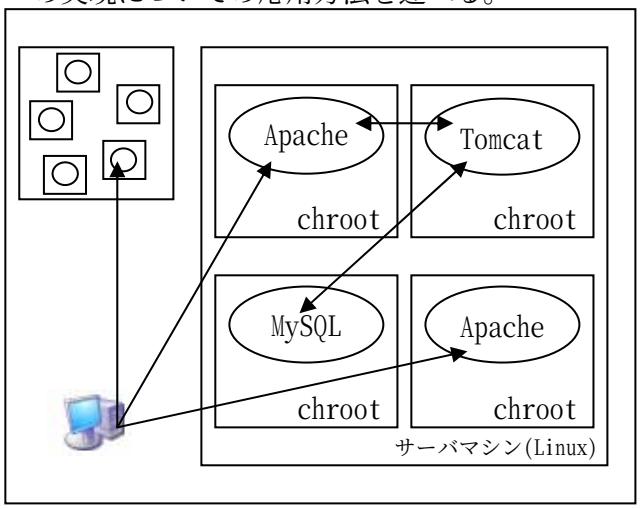
chrootを用いたセキュアなLinuxサーバシステムの構築

後藤 滋文† 堀江 夏子† 松本 大貴† 高橋 由樹† 中村 章人‡ 塚本 享治†
東京工科大学大学院メディア学研究科† 産業技術総合研究所‡

第1章 はじめに

計算機の能力を高めるための手法としてクラスタリングがある。近年盛んに研究され、既に様々なクラスタリングシステムが誕生している。しかし、これらの環境は主に研究室などの閉じた環境を想定しているため、そのままインターネットなどの開かれたネットワークにて構築するのはいささか抵抗がある。

そこで、Linux上で動作する各種プログラムをchroot環境下で動作させることにより、セキュリティホールを突かれて不正にサーバ内にアクセスされた場合でも、特定の領域外へのアクセスを防止することによって被害を最小限に防ぐサーバを構築した。また、chroot環境をクラスタリングシステムに適用するための工夫も行った。本稿では、サーバソフトとして広く用いられているApache、Tomcat、MySQLをchroot環境下で構築する方法を述べる。さらに、クラスタマシンへの応用に必要なポート番号についての考察を行い、クラスタ環境でのセキュアなサーバの実現についての応用方法を述べる。



図：chrootを用いたサーバの概念

第2章 chrootとは

今回のシステム構築の中心となるコマンドがchrootである。このコマンドを実行することにより、任意のディレクトリをルートディレクトリに設定し、それより上の階層へのアクセスができなくなる。

例：# chroot /home/chroot

この例では/home/chrootを新しいルートディレクトリと設定し、新しい“/”は大元のOSの“/home/chroot”と同一のものとなる。

このように、chrootコマンドによりプロセスをファイルシステムの特定領域に閉じ込めてしまうことによって意図しないファイルへのアクセスを防げる。

第3章 サーバ構築

実際にTurboLinux10Beta(Suzuka)を用いて、Apache(2.0.48)、Tomcat(4.1.27+J2SDK1.4.1)、MySQL(4.0.16)をchroot内で実行するサーバを構築した。その手順は下記の通りである。

1. プログラムのバイナリの準備（通常の手順にてセットアップを行う）。
2. lddコマンドを使用し、必要なライブラリを調査する。
3. プログラムのバイナリ・設定ファイル・必要なライブラリをchrootするためのディレクトリにコピーし、配置する。
4. chroot環境下で起動して、エラーが起きる(起動しない)ならエラーログもしくはstraceコマンドで原因を調査し、対処する。
5. 正常に起動するまで手順4を繰り返し、起動したらテストを行う。

次に、実際に各々のプログラムで行った作業を順を追って説明する。

3.1 Apache

1. ソースパッケージをダウンロードし、通常通りセットアップを行う。この際、全てのモジュールを有効にしてコンパイルした。
2. lddコマンドで必要なライブラリを調査。13個見つかったのでそれらをコピーし起動。
3. http: bad user name nobody と表示されたので/etc/passwdをコピー。
4. 今度はmod_unique_idのエラーが出るので、/etc/nsswich.confと/lib/libnss*をコピーし、/etc/hostsにサーバ名を記述。
5. 次にapr_proc_detachのエラーが発生したのでstraceコマンドで調査し、/dev/nullと/etc/localtimeと/dev/randomをコピー。
6. chroot環境下で起動してみるとエラーもなく起動し、別マシンからのアクセスでテストページが表示されるのを確認。

3.2 Tomcat(J2SDK)

Tomcatのインストールの前にJDKのインストールが必要である。

1. JDKのバイナリパッケージを使用し通常通りセットアップを行う。
2. lddコマンドで調査すると4つのライブラリが表示される。この4つとJDKをchroot環境下にコピーしJAVA -versionを実行。
3. libnsl.so.1が無いというエラーが出るので

Construction of secure linux system using chroot
†Shigefumi Goto , Natsuko Horie , Taiki Matsumoto , Yuki Takahashi , Michiharu Tsukamoto – Graduate School of Media Science of “Tokyo University of Technology”
‡Akihito Nakamura - National Institute of Advanced Industrial Science and Technology

コピーする。
4. libm.so.6 が無いというエラーが出るのでコピーする。
5. JAVA -version が正常に表示されるようになったので Tomcat のセットアップに移行。
6. Tomcat のバイナリパッケージを使用し、chroot 環境下へ解凍。
7. 環境変数 JAVA_HOME を設定。
8. ./bin/startup.sh を実行すると、uname と touch コマンドが無いとエラーが出るのでコピー。
9. 再び./bin/startup.sh を実行するとエラーが出ず、他マシンよりアクセスしてテストページが表示されるのを確認。

3.3 MySQL

1. バイナリパッケージを使用し通常通りにインストールする。mysql_install_db 以降は chroot 内でセットアップ作業を行う。
2. mysql ユーザを追加したので/etc/passwd をコピーする。
3. mysql_install_db 実行時に/bin/hostname が無いというエラーが出るのでコピー。
4. bin/safe_mysqld --user=mysql &を実行すると、sed・touch・chown・date・rm が無いといわれるのでコピー。
5. 再び実行すると、cat と tee が無いというエラーが出るのでコピー。
6. エラーは出なくなるが、起動しないので strace コマンドで調べると/proc 以下が読めていないようなので chroot 環境下に読み取り専用でマウントする。
7. MySQL のエラーログで ibdata1 が開けなくて落ちていることが判明し、調査すると data ディレクトリ内のファイル所有者が root になっていたので所有者を mysql に変更。
8. bin/safe_mysqld --user=mysql &にて起動に成功。bin/mysqladmin version にてバージョン表記が出ることを確認。

3.4 サーバ構築手法について

ここで実際に構築した手順を見れば分かるように、基本的には最初にあげた 5 つの手順を順番に実行すれば chroot 環境下でプログラムを動作させることが可能である。

ただし、/etc/passwd 等をコピーするときは注意が必要である。/etc/passwd の場合は必要な行以外を削除するなどの対策を行わないと、せっかく chroot でセキュリティを高めようとしているにもかかわらず母体 OS のパスワードファイル全体が漏洩してしまう可能性がある。

第 4 章　ポート番号との関係

同一のマシンで同じサーバプログラムを複数起動する場合にはポート番号の衝突が問題となる。いくつかの解決方法を検討した。

4.1 ローカルホストで完結している場合

ローカルマシン内にある MySQL にアクセスするだけの場合、MySQL にアクセスするユーザが気をつけてさえいれば MySQL のポート番号を変えるだけで対処可能である。場合によってはアクセスするためのプログラムが使用する Socket ファイルの位置等を変更する必要が出てくるが、比較的容易に解決できる。

4.2 外部からのアクセスがある場合

Apache 等の外部からアクセスがある場合はどうだろうか。Apache の場合、HTTP に割り当てられた 80 番ポートを利用する。通常、外部からアクセスする際にはこの TCP80 番ポートに対してアクセスを行う。しかし、同一マシン上で複数の Apache が起動する場合、80 番のポートを使用できるのは 1 つの Apache のみで、他の Apache は 80 番ポートを使用できない。この場合は何らかの形でクライアントにポート番号を広報するか、リバースプロキシを設置する必要がある。

4.3 クラスタリングシステムの場合

アクセスするべきプログラムが複数のマシンに分かれている場合はどうだろうか。例えば複数台のマシンで Apache を起動し負荷分散を図っている場合、外部からのアクセスを各々のマシンの各々のポートに振り分けるため、DNS ラウンドロビン等を使用した仕組みが必要となってくるだろう。

4.4 それぞれに対応するためには

一つの解決方法を提案する。サーバに何らかのデーモンを走らせておき、そのデーモンが各々のサーバプログラムのポート番号を管理する。サーバファームの入り口にプロキシのようなものを設置し、これと各サーバのポート番号管理デーモンと情報をやり取りし、アクセスがあるとプロキシが適切なサーバにアクセスを振り分ける。この方法によってクライアントからすればシームレスにアクセスが可能となる。静的にポートを転送する方法はあるが[3]、本研究の最終目的は空いているサーバリソースを動的に割り当てることであるため不足である。

また、プロキシにクラスタ管理システムを統合し、SNMP 等によって各マシンを監視することによって負荷に応じたリクエストを振り分けることが可能になる。

第5章　まとめ

本稿で示した手順に基づいて、chroot を用いてセキュアな Linux サーバシステムを構築できた。今回は手動で chroot 環境下に必要な物だけを用意する方法で構築したが、対照的な方法として OS 全体を chroot 環境下に用意する方法もある。その方法は既に一部のインターネットサービスプロバイダによって提供されている。

今後、今回の構築作業の自動化や、今回のデータを元にして複数台のマシンを管理できるクラスタリングシステムへの応用を目指す。

参考文献

[1] Dennis Sosnoski:Linux の Java サービスに対する安全性を高める(http://www-6.ibm.com/jp/developerworks/linux/030606/j_l-secjav.html)
[2] MICHEL D. BAUER, 豊福 剛/訳:Linux サーバセキュリティ,オライリー・ジャパン
[3] R.Flickenger, 山口晴広/監訳:LINUX サーバ HACKS,オライリー・ジャパン