す．

この場合，PAMの認証機能の中の，su認証情報設定を変更する必要があります．具体的には，/etc/pam.d/suのpam_wheel.soを設定します．

6行目のコメントをはずします．

```
(変更前)#auth        required      /lib/security/$ISA/pam_wheel.so use_uid
(変更後)auth         required      /lib/security/$ISA/pam_wheel.so use_uid
```

そして，グループ・ファイルの/etc/groupのwheelグループにsuの使用を許可するユーザuser1を追加します．

```
# vigr  ←グループ・ファイルの変更
11行目 wheel:x:10:root,user1
        ←user1をwheel　グループに追加する(空白なしの[,]のみ))
```

シャドウ・グループ・ファイルの変更も同様です．

```
11行目 wheel:::root,user1
        ←user1をwheelグループに追加する(空白なしの[,]のみ)
```

この後，user1でsuを実行すればsuになれますが，ほかのユーザでは，不可能になります．なお，デフォルトでは誰でもがsuになれる設定なので危険です．

6.2 ネットワーク侵入検出システム

ネットワーク侵入検出システム(NIDS)，ネットワークやシステムへの不審な侵入を検知し，警告を発する運用管理ツールです(*6.37)．

ここでは，ネットワークのトラフィックを監視して不審な侵入を検知するsnort(*6.38)と，ファイル・システムの改ざんを監視するtripwire(*6.39)を取り上げます．

(*6.37) NIDS：Network Intrusion Detection System；ネットワーク侵入検出システム．

(*6.38) **snortホームページ**：http://www.snort.org/
パッケージ=http://www.snort.org/dl/binaries/linux/snort-2.8.2.2-1.RH5.i386.rpm

(*6.39) **tripwireホームページ**：http://www.tripwire.org/
＊RHEL4用rpm：tripwire-2.3.1-22.el4.i386
ftp://ftp.pbone.net/mirror/ftp.silfreed.net/repo/rhel/4/i386/silfreednet/RPMS/tripwire-2.3.1-22.el4.i386
＊sourceforge：tripwire-2.4.1.2-src
http://sourceforge.net/projects/tripwire/
http://sourceforge.net/project/showfiles.php?group_id=313

6.2.1　ネットワーク侵入検出システム：snort

snortはポート・スキャンなどを検出するNIDSの代表例です．snortのログはtcpdumpなどで見るようになりますが，WWWブラウザから見るようにできるインターフェース(*6.40)もあります．

snortのインストールと初期テスト

snortのrpmパッケージをインストールした後，すぐに「スニファ・モード」でテストを行います．これは，snortのFAQ(How do I run snort? (*6.41))の記述にあるものです．

```
/usr/sbin/snort -n 3 -dvi eth0 -c /etc/snort/snort.conf
```

スニファ・モードで3パケット(-n 3)を監視して終了するコマンドで，eth0をアプリケーション・レイヤのダンプでコンソール表示(-dvi eth0)します．例えば，telnet処理で3パケット送受信する分析ができます．

snortの設定変更

設定ファイルsnort.confの実際の変更点は，ネットワーク・アドレスの設定やログをとる設定ルール・ファイル(*.rules)がデフォルトでは一部しか使用されない設定になっているのでほかも有効にします．ping(ICMP-echo)などはicmp-info.rulesに記述がありますが，snort.confでは記録されない設定(#でコメント化)になっているので，先頭の#をはずすだけです．

設定確認テスト

設定ファイルの変更をテストで確認します．コマンドは，

```
/usr/sbin/snort -u snort -g snort -b -i eth0 -A fast -c /etc/snort/
snort.conf
```

として，alertファイルをパケット付きで出力する確認テストとします(*3.42)．

alertファイルを1パケットにつき1行で出力する(-A fast)設定でsnortを動かし，snortが待ち状態に入った時点でWindowsクライアントからtelnetで接続してわざとログインを失敗してみます(alertテストもFAQで記述されている)．そして，このテストを「Ctrl」+「C」キーで強制終了します．snortのログ・ディレクトリ/var/log/snort内のalertファイルにはパケット一つにつき1行単位のログが作成されています．また，tcpdump形式のログsnort.log.XXXXXXXXXXは「-b」オプションで生成されたもので，

(*6.40) SnortSnarf：http://www.snort.org/dl/contrib/data_analysis/snortsnarf/

(*6.41) snort-FAQ：http://www.snort.org/docs/faq.html

(*6.42) この実行までにrootで何らかのsnort処理を行ってしまうと，root(所有者/グループがroot)のみrw属性のalertが作成されてしまい，ユーザsnortがこのalertに追加書き込みできずに「/var/log/snort/alert: Permission denied」になって，以降何もできない．対処としては，このroot属性のalertファイルを削除すれば正常に戻る．