実践 Linux セキュリティー講座

前回はウェブサーバーのセキュリティーに関する概念について解説しました。今回はウェブサーバーのセキュリティーの各種設定について説明します。ここではウェブサーバーの一般的な設定については触れません。あくまでもセキュリティーに関連した説明のみを扱います。したがって、多少なりともLinux（UNIX）上のウェブサーバーの知識が必要となります。

第14回 ウェブサーバーのセキュリティー（後編）

ソフトウェアコンサルタント　すずきひろのぶ

実行ユーザー権限を知る

まず、ウェブサーバーのセキュリティーについて話を進める前に、Linux（UNIX）でプログラム（プロセス）を実行するユーザーの権限（実行ユーザー権限）についての知識が必要になるので、これについて少し解説します。

Linux（UNIX）ではプロセスが実行されるとき、そのプロセスが誰の権限で実行されているかによって、プロセスがファイルなどにアクセスできる権限（アクセス権限）が違ってきます。ユーザーが普通にログインしてシェルから何かコマンド（プロセス）を実行しているときは、そのユーザーの権限でコマンド（プロセス）は動きます。また、プロセス実行中に実行時のユーザー権限を変更することもできますが、変更するにはroot権限（管理者権限）が必要になります。

RedHat Linuxをインストールして初期状態のままhttpdを立ち上げると、最初の親プロセスの実行ユーザーは「root」ですが、それ以外のプロセスの実行ユーザーは「nobody」というユーザーになるように設定されています（図❶）。

rootとnobodyが実行ユーザー

実際のプロセスがどうなっているか、psコマンドで表示してみましょう。axufというオプションを使うと親プロセスと子プロセスの関係がわかりやすく表示されます（リスト❶）。

少なくとも1つはroot権限で実行さています。これにはいくつかの理由があります。まず最初に挙げられる理由としては、httpプロトコルの初期状態のポート番号は「80」になっているからです。一般にLinux（UNIX）では、1024未満のポート番号を利用するためにはroot権限がなくてはなりません。このため、少なくとも1つのプロセスはroot権限が必要になります。ポート番号を1024以上に設定するとroot以外のユーザーでも原理的にはhttpdを実行できます。

RedHat Linuxのhttpdはrootで立ち上げることを前提条件にしているので、本連載ではこの条件のまま使うことにしますが、これは完全にApacheのサーバープログラムが安全であるということを保証するわけではありません。SOHOレベルの設定に費やす労力と今回解説するセキュリティーはバランスが取れていると考えているだけにすぎません。

子プロセスの実行ユーザーがnobodyなのは、/etc/httpd/httpd.confという設定ファイルにある「User」の設定と「Group」の設定がnobodyになっているためです。nobodyという名前に特別な意味はありません。/etc/passwdを見ればわかるのですが、nobodyというユーザー（ユーザーID：99）が定義されているのです（リスト❷）。

実行ユーザーを引き継がせる

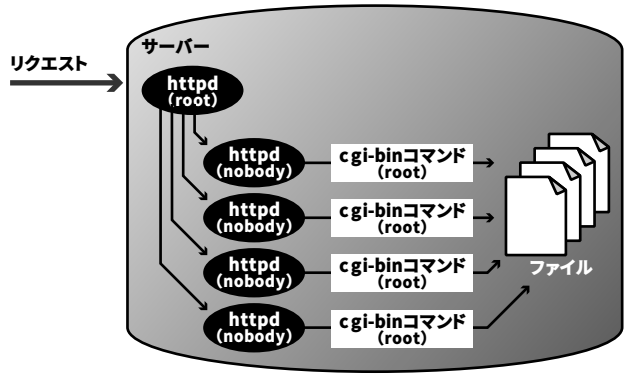
長々と実行ユーザー権限について説明してきましたが、重要なポイントはnobodyというユーザーの権限がcgi-binコマンド（プロセス）を実行するときに、プロセスの実行ユーザー権限として継承されるということです。この関係がわかるようなテスト用cgi-binコマンドを作って実験すると理解できるでしょう（図❷）。参考にこのcgi-binコマンドのコードを載せておきます（リスト❸❹）。この実験からわかるように、cgi-binで実行されるコマンドはnobodyのユーザー権限で実行されます。

この実験では以下のことがわかります。

- 初期状態ではhttpdはユーザーnobodyで実行される
- HTMLファイルはnobodyが読み込み可であるファイルにしておく必要がある
- cgi-binもnobodyの権限で実行される
- この権限は/etc/httpd/httpd.confのUserとGroupの設定で変更できる

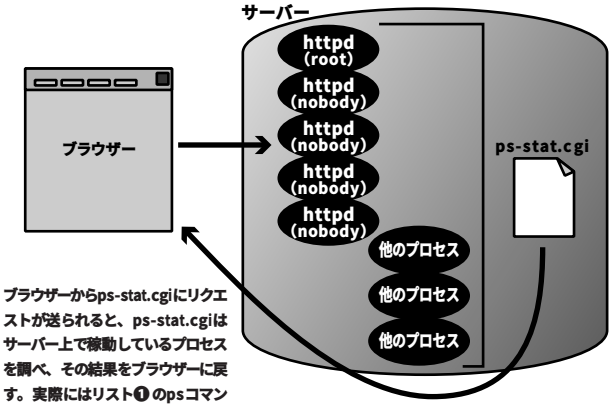
（注意）この実験の終了後直ちに利用したcgi-binコマンドを削除しておいて下さい。

図❶　httpdの実行ユーザー



起動しているhttpdプロセスの実行ユーザーはrootとnobodyとなっている。

図❷　ps-stat.cgiの動き



ブラウザーからps-stat.cgiにリクエストが送られると、ps-stat.cgiはサーバー上で稼動しているプロセスを調べ、その結果をブラウザーに戻す。実際にはリスト❶のpsコマンドと同じ内容が表示される。

リスト❶　実行ユーザーを調べる

```
$ ps axuf
USER       PID %CPU %MEM  SIZE   RSS TTY STAT START   TIME COMMAND
....
root      5550  0.0  0.8  1860  1140  ?  S    15:56   0:00 httpd
nobody    5552  0.0  0.9  1956  1264  ?  S    15:56   0:00  \_ httpd
nobody    5553  0.0  0.9  1956  1252  ?  S    15:56   0:00  \_ httpd
nobody    5554  0.0  0.9  1956  1252  ?  S    15:56   0:00  \_ httpd
nobody    5555  0.0  0.9  1932  1184  ?  S    15:56   0:00  \_ httpd
```

リスト❷　/etc/passwdの表示（部分）

```
......
nobody:x:99:99:Nobody:/:
......
```

（シャドウパスワード化されている）

リスト❸　ps-stat.cgi

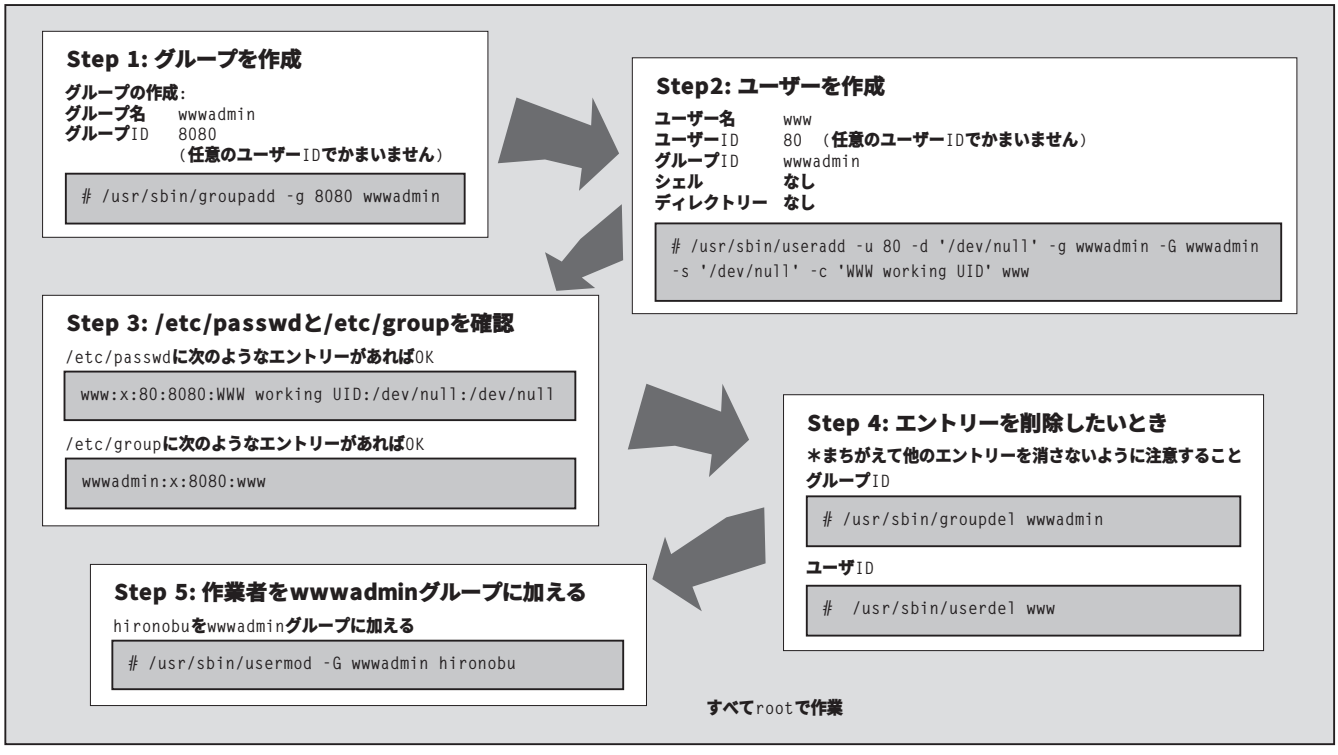
```
#!/bin/sh
echo Content-type: text/plain
echo
```

リスト❹　ps-stat.cgiの表示結果（ブラウザー）

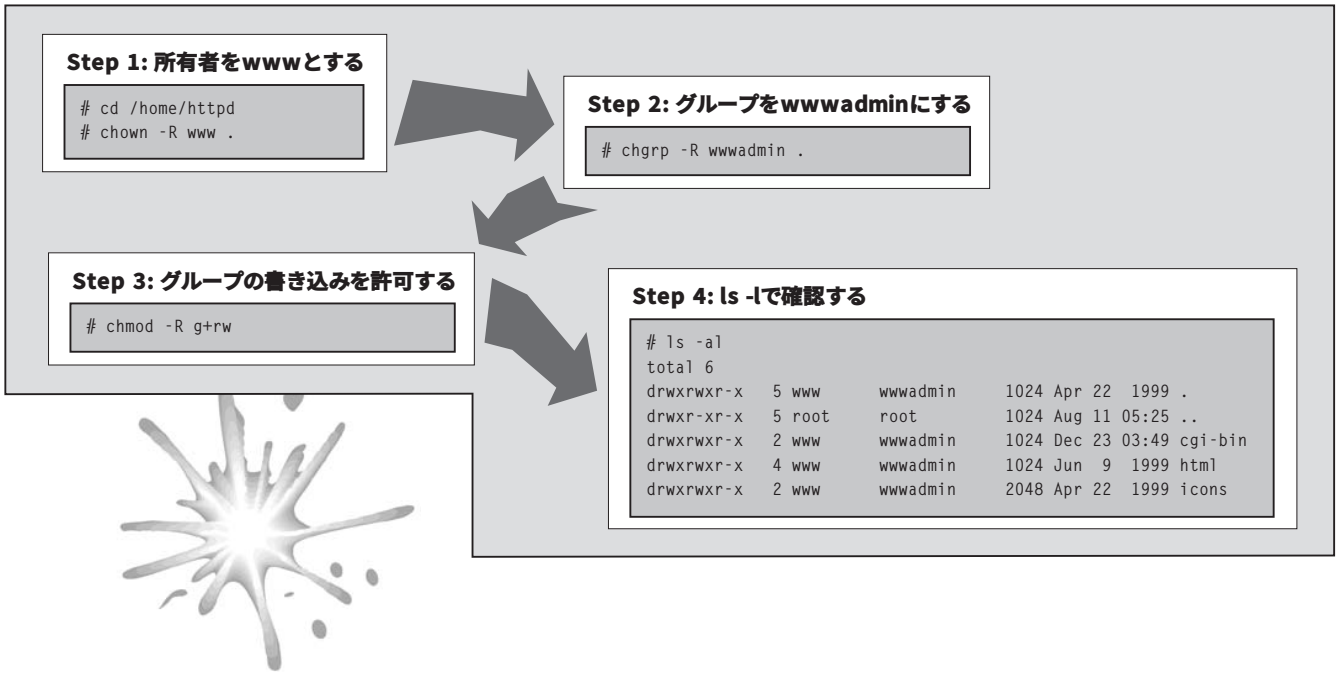
```
$ ps axf
  PID TTY STAT TIME COMMAND
...
 7042  ?  S    0:00 httpd
 7044  ?  S    0:00  \_ httpd
 7057  ?  S    0:00  |   \_ sh /home/httpd/cgi-bin/ps-stat.cgi
 7058  ?  R    0:00  |       \_ ps axf
 7045  ?  S    0:00  \_ httpd
 7046  ?  S    0:00  \_ httpd
 7047  ?  S    0:00  \_ httpd
...
```

ポート番号：送られてきたパケットの通信内容（サービス）が何なのかを区別するために使っている番号。0番から65535番までであり、特に0番から1024番はあらかじめ決められた通信用に使われている（例：WWWは80番）。

図❸　作業用ユーザーとグループの作成



図❹　httpdのディレクトリーの所有者を作業用グループに変更する



cgi（シージーアイ／common gateway interface）：ブラウザーが送った要求に対してWWWサーバー（httpd）が別のプログラム（たとえばcgi-binなど）を起動して、結果をブラウザーに返す処理を行うためのインターフェイス。

作業用のユーザーを作っておく

HTMLファイルやcgi-binコマンドをメンテナンスする場合、rootでログインして非常に煩雑な作業するようなことは、間違いも起こりやすいと言えるでしょう。このため、ウェブサーバーのメンテナンス作業用のユーザーとグループを作成しておきましょう。このユーザーは実際にはログインするわけではなく、ファイルやディレクトリーに対してユーザーのIDを割り振るために利用します。また、グループは共同作業をしやすくするために作ります。

ここでは「www」というユーザーと「wwwadmin」というグループを作成することにします（図❸）。

次に、ディレクトリー「/home/httpd」とこのディレクトリーより下にあるファイルやディレクトリーの所有者を変更します。初期状態では所有者はrootになっているはずです。そこで、ウェブサーバーのメンテナンスをしやすくするために、先ほど作成したユーザーwwwを所有者とし、所有者のグループをwwwadminにします（図❹）。

最後にグループに対する書き込み許可の設定をします（chmod g+w ファイル名）。これでユーザー「hironobu」は、HTMLファイルやcgi-binコマンドを書き込めるようになります。またディレクトリー「/home/httpd」より下にあるディレクトリーやファイルを他のユーザーが読めないように設定してしまえば、nobodyのユーザー権限で実行されている状態のhttpdからは、これらのファイルやディレクトリーにアクセスできない状態になります。

連絡先を整理しておく

ここまでは実行ユーザー権限に関する設定について解説してきました。今度はメールアドレスのエントリーを念のために設定しておきましょう。

ウェブサーバーの管理者のユーザーのメールアドレスを直接公開してしまうと、管理者が簡単に特定されてしまうので危険です。そこで、適当なメールアドレスを作り、このメールアドレスから実際の管理者にメールを転送するようにしましょう。方法としては、「/etc/aliases」というファイルにたとえば「webmaster」というエントリーを作成しておきます。

```
webmaster:   hironobu@h2np.net
```
（たとえばユーザーhironobu@h2np.netにメールを転送するとき）

このようにして、外部にはwebmasterのアドレスのみを公開します。ウェブサーバーの管理者が増えればファイル「/etc/aliases」のwebmasterのほうに追加しておきます。

またこのようなエントリーを作ったあとでは、ウェブサーバーの設定ファイル「/etc/httpd/conf/httpd.conf」のサーバー管理者のアドレスも直接誰かのアドレスを書くのではなく、webmasterにしておくといいでしょう。

```
ServerAdmin webmaster@localhost
```

加減を見ながら設定する

ウェブサーバーにしてもそうですが、時間と手間さえ惜しまなければ、徹底してセキュリティーを追求できます。しかし、徹底したセキュリティーだからいいのかというと、必ずしもそうではありません。無闇に敷居の高いセキュリティーを設定しようとすると、自分では運用できない範囲のセキュリティーになってしまうからです。常にコストとセキュリティーの現実的なバランスを考えましょう。

次回はftpとbindについてお話ししたいと思います。

参考1 cgi-binの実行ユーザー権限を変更する

今回説明した状態では、httpdから実行されるcgi-binコマンドは常にnobodyのユーザー権限で実行されてしまいます。このためcgi-bin経由でファイルに記録する場合などは、ファイルのオーナーがnobodyになってしまって不都合が起こることがあります。このような場合には、Apacheのソースコードディストリビューションに含まれている「suexec」というユーティリティーが有効です。

suexecを使うためには、Apacheをソースコードからコンパイルし、さらにsuexec用の設定をする必要があります。高機能なcgi-binコマンドを作成する場合は、このsuexecを使ってきめ細かい実行ユーザー権限の管理をするように心がけてください。

参考2 httpdを完全に一般ユーザー権限で動かす

RedHat Linuxのインストール初期状態では、少なくとも1つはrootの実行ユーザー権限で動作するhttpdが必要だと説明しました。しかし、すべてのhttpdを一般の実行ユーザー権限で動作させておいたほうが、万が一のときは安心だと思う方もいるかもしれません。その場合は、Apacheを別途コンパイルしてインストールする必要があります。最新のディストリビューションはApacheのウェブサイト Jump から入手でき、コンパイルもLinux上であれば難しくはありません。Apacheのホームとなるディレクトリーを用意し

```
./configure --prefix
```

で指定しておくといいでしょう。このほうが複数のApacheを一括管理できて楽になります。

また、ここまで徹底したセキュリティーを考えるならば、ウェブサーバーマシンではメールやネームサーバーなど一切のネットワークサービスを動かさずにウェブサービスのみに特化するほうが得策でしょう。

さらに、インターネットのルーター側でIPマスカレードを行うことによって外部に見えるIPアドレスとポート番号を変更すればセキュリティーが強化されます。

Jump www.apache.org

グループ：ユーザーに特定の権限を一括して与えるためにユーザーを1まとめにする単位。これによってユーザーの管理がしやすくなる。

［インターネットマガジン　バックナンバーアーカイブ］ ご利用上の注意

このPDFファイルは、株式会社インプレスR&D（株式会社インプレスから分割）が1994年～2006年まで発行した月刊誌『インターネットマガジン』の誌面をPDF化し、「インターネットマガジン　バックナンバーアーカイブ」として以下のウェブサイト「All-in-One INTERNET magazine 2.0」で公開しているものです。

http://i.impressRD.jp/bn

このファイルをご利用いただくにあたり、下記の注意事項を必ずお読みください。

- 記載されている内容（技術解説、URL、団体・企業名、商品名、価格、プレゼント募集、アンケートなど）は発行当時のものです。
- 収録されている内容は著作権法上の保護を受けています。著作権はそれぞれの記事の著作者（執筆者、写真の撮影者、イラストの作成者、編集部など）が保持しています。
- 著作者から許諾が得られなかった著作物は収録されていない場合があります。
- このファイルやその内容を改変したり、商用を目的として再利用することはできません。あくまで個人や企業の非商用利用での閲覧、複製、送信に限られます。
- 収録されている内容を何らかの媒体に引用としてご利用する際は、出典として媒体名および月号、該当ページ番号、発行元（株式会社インプレスR&D）、コピーライトなどの情報をご明記ください。
- オリジナルの雑誌の発行時点では、株式会社インプレス R&D（当時は株式会社インプレス）と著作権者は内容が正確なものであるように最大限に努めましたが、すべての情報が完全に正確であることは保証できません。このファイルの内容に起因する直接的および間接的な損害に対して、一切の責任を負いません。お客様個人の責任においてご利用ください。

このファイルに関するお問い合わせ先
株式会社インプレスR&D
All-in-One INTERNET magazine 編集部
im-info@impress.co.jp