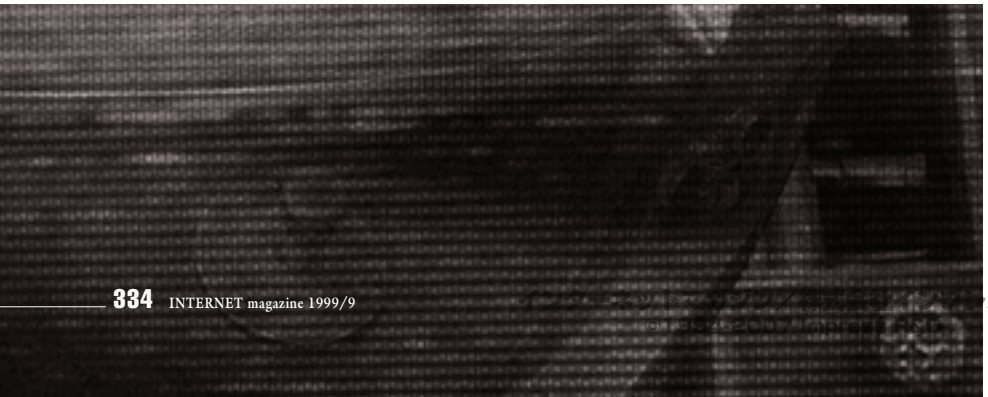
実践 Linux セキュリティー講座

今回はパスワードとユーザーアカウントの管理について考えてみます。基本の基本であるパスワードの選び方から、今後求められる認証方法などを説明します。Linuxのユーザーアカウントに対する認証などは小規模（個人）利用を想定していますが、SOHOレベルではこの仕組みを正しく利用すれば十分に安心して使えるものになるでしょう。

第8回 ユーザーアカウントを管理する

ソフトウェアコンサルタント　すずきひろのぶ



昔から使われているパスワード

世界中の誰でも知っている昔々のアラビアの物語「アリババと40人の盗賊」をちょっとだけ思い出してみましょう。そこではすでに「ひらけゴマ」というパスワードによって動作する認証システムを採用しています。それほどパスワードというのはポピュラーで、そして、そのコンセプトも昔からあったものです。パスワードの原理は、特定の人間のみが知っている情報を提示することによって、その特定の人間であると証明することです。

このパスワードに使う情報を他人が知ることがなく正しく運用されてさえいれば、これほど単純でコストがかからず、そして十分に強力な認証方法はありません。その一方で、パスワード方式は、きちんとした安全なシステムと運用方法をしなければ単純であるがゆえに簡単に破綻してしまいます。もっと短く言えば、使っているパスワードがバレてしまっては元も子もなくなってしまいます。フェイルセーフというものがありません。

パスワード方式の防御の原理原則は単純明快です。それは、使っているパスワードを「知られない／悟られない」ということです。一方、単純であるがゆえに、そのためにはいろいろな方策をとらなければなりません。

記憶するのがベスト

あるLinuxのイントール解説書には、「rootのパスワードを設定したあとで忘れると大変なのでパスワードをメモしておきましょう」と書いてあったそうです。ある意味で、本音を含んでいるので苦笑いするしかありませんが、一般論としては避けるべき行為です。

この連載を読んでいる方で、パスワードを書いた付箋紙をディスプレイに張ってあるような人はさすがにいないと信じていますが、それでいても、パソコンのようにコンピュータはコンソールの前にいる自分しか使うことがないような感覚でいる人も多いかもしれません。

筆者は自宅で利用しているLinuxのパスワ

ードをメモする行為を頭から否定する気はありません。頭の中に収めるというのがベストですが、それが不安な場合、メモを取るという非常手段に出ることもあると思います。

「メモすることは悪いことなのか」という話題は常に議論の対象となります。一番理想的なのは十分に複雑で長いパスワードを頭の中に記憶しておくことです。しかし、そうは世の中うまくいくとは限りません。「十分に複雑なパスワードを記憶しろ」と強く言ったところで、ユーザーはパスワードを忘れることが不安で簡単なパスワードを付けてしまう危険性があるのです。ならば、メモが外部に洩れないような管理をしたうえで、複雑なパスワードを選択するのも1つの選択ではないでしょうか。

このような記憶に頼るタイプのパスワードによる認証の限界は、利用するユーザーの賢さを超えることはできないと筆者は考えています。ただし一言厳しいことを言わせてもらえば、SOHO環境では数多くのシステムを同時に管理するわけではないのですから、せめて8文字程度の記号列ぐらいは記憶してほしいと思います。

システムがチェックするが…

パスワード初心者に「十分に複雑なパスワードを選択してください」と言っても、何をパスワードにしてよいかわかりません。まったく知識がないユーザーがパスワードを選ぶ場合、弱いパスワードになってしまうでしょう。

と言っても、Linuxでは、よく雑誌などに書かれている「ユーザー全体の86パーセントのパスワードが弱いパスワードである」というような事態にはなりません。この数字だけ一人歩きしていますが、もともとは1975年のUNIX System Manager's Manual（6th ed）にあるRobert MorrisとKen Thompsonが書いたドキュメント"Password Security: A Case History"に現われる数字です（図1）。筆者が想像するには、今ごろこの86パーセントという数字を出す人はこのドキュメントに目を通したことがない人たちか、あるいはドキュメントの存在すら知らない人たちではないかと思っています。

Linuxでパスワードを設定する場合、パスワードが適当であるかどうかのチェックが必ず入ります（図2）。したがって、「任意の1文字」とか「英単語辞書にある単語」というのはあり得ません。一方、誕生日の組み合わせなどは数字の無意味な羅列として認識して入力されてしまいますが、そのような数字の組み合わせは避けなければならないのは言うまでもありません。コンピュータはあなたの誕生日を知らなくとも、あなたのサーバーを攻撃する者は知っているかもしれないことを思い出すべきでしょう。

最近のLinuxはLinux-PAM（Pluggable Authentication Module for Linux）という認証を強化するセキュリティーフレームワークが追加されています。これについては、今回の話題の範囲では弱いパスワードのチェックをしてくれるものという認識で十分でしょう。

8文字の強度はどれぐらいか

パスワードをランダムな英数字の組み合わせからなる8文字の文字列にした場合、どれくらいの強度になるかをちょっと計算してみましょう。

8文字のパスワードを選ぶ場合

❶AからZまでの大文字小文字を区別して52個を組み合わせる

52^8＝53459728531456（約53兆とおり）

❷0から9までの数字10個をさらに加えて62個から組み合わせる

62^8＝218340105584896（約218兆とおり）

❸「!」や「*」など10種のキャラクターをさらに加えて72個から組み合わせる

72^8＝722204136308736（約772兆とおり）

ランダムなパスワードを破るには試行を繰り返す方法しかありません。この方法をイクゾースト攻撃あるいはブルートフォース攻撃などと呼びます。外部からパスワード認証を呼び出して試行するような手順で攻撃するならば62^8程度の組み合わせで半永久的に解けないでしょう。

パスワードファイルが盗まれた状態（パスワードファイルの構造に関しては後に説明し

図1 「ユーザー全体の86パーセントのパスワードが弱い」という根拠

```
3289ケースの中パスワードが...
      任意の1文字である ..........15ケース
      任意の2文字である ..........72ケース
      任意の3文字である ..........464ケース
      4文字のアルファベットである ..........477ケース
      5文字のアルファベットで大文字のみか小文字のみである ..........706ケース
      6文字のアルファベットで小文字のみである ..........605ケース
      英語辞書にある単語である ..........492ケース
                                トータル 2831
```

図2 passwdコマンドで弱いパスワードを入力した場合

```
# passwd
New UNIX password:                          <-- "hello"と入力
BAD PASSWORD: it is too short               <-- 短いので拒否される

# passwd
New UNIX password:                          <-- "password"と入力
BAD PASSWORD: it is based on a dictionary word  <-- 辞書にある単語だと拒否

# passwd
New UNIX password:                          <-- "1234567890"と入力
BAD PASSWORD: it is too simplistic/systematic   <-- 単純すぎるので拒否
```

ます）ではどうなのでしょうか？　パスワードファイルがあるならばパスワード生成の内部ルーチンを直接使えばいいので、高速にパスワードを試行するのが可能になります。非常に大ざっぱですが、毎秒100万回の試行が可能[注]という仮定で、全体の50パーセントを試行するにはどれくらいかかるか計算してみます。結論から先に言えば、SOHOレベルであれば3.46年程度保護できれば十分でしょう。

英字（大小区別あり）のみ
　約300日
英数字から8文字ランダムに選んだ場合
　約3.46年
加えてキャラクター10文字を加えた場合
　約11.45年

ちなみに、100万回を処理するのに必要な計算力は、現在RC5-64解読に参加しているグループのトップ10チームの合計よりも大きいものです。

distributed.net
www.distributed.net

[注]これはかなり余裕を持った計算量です。DES暗号を使っているパスワードプログラムなら毎秒2500万回ものDESの計算が必要になります。MD5ハッシュ関数を使っているものでは毎秒10億回ものMD5の計算が必要になります。

複雑なパスワードを作る

今までの解説書の多くには「ランダムな文字列を思い浮かべて利用しなさい」ぐらいの書き方がされていますが、実は慣れない人にとってはランダムな文字を思い浮かべること自体がとても難しいのです。といって、身近な所でサイコロを振って決めたとしても6^8の組み合わせ程度しかありません。

そこで、おすすめなのがLinuxのmkpasswdコマンドです。これは自動的にパスワードを生成するプログラムです。パスワードを決め兼ねているくらいなら、このコマンドを使って決めたほうが確実で安全です。方法は非常に簡単で次のコマンドを実行するだけです。

```
$ mkpasswd
lvs76SJpx ←表示されるパスワード候補文字列
```

シャドウパスワードを使う

何もしない場合、Linuxは/etc/passwdというファイルにパスワードを保存します。しかし、/etc/passwdは一般ユーザーが読み取れるファイルです。つまり、誰でも、そこからパスワードを取り出してイクゾースト攻撃ができるようになっています。それではあまりにも問題ですので、最近はrootしか参照できないシャドウパスワードファイルを利用するのが常識的な運用になっています。ユーザーすべてが丈夫なパスワードを付けているならば問題はないと思うかもしれませんが、きちんと隠しましょう。Linuxではシャドウパスワードが標準で入っています。

使い方は簡単です。root権限でpwconvコマンドを実行すると、自動的に/etc/passwdから/etc/shadowというシャドウパスワード用のファイルを生成します。また、grpconvコマンドを使えばグループのパスワードを保存している/etc/groupをシャドウ化します。pwconvを実行したあと、grpconvも実行しておきましょう（図3）。pwconv、grpconvともに一度実行したら、それ以降追加するユーザーやグループについても自動的にシャドウパスワードが有効になります。

次の例がpwconvを実行した後の/etc/passwdにあるrootのエントリーです。

```
root:x:0:0:root:/root:/bin/bash
```

パスワードがあるべき部分が"x"となっています。パスワード部分は/etc/shadowに移ります。以降、システムは自動的に/etc/shadowのパスワードを参照するようになります。

パスワードファイルの作り方

パスワードファイルに含まれるパスワードは、すでに一方向性ハッシュ関数によって撹拌されている値となってなっています（図4）。

したがって、パスワードファイルの中にあるすでに撹拌されているパスワードの値だけ持っていても、そこから元のパスワードを逆算することはできません。いろいろな入力値を一方向性ハッシュ関数に与えて、その出力結果と手元にあるパスワードファイルの内容とを比較する方法、つまり、イクゾースト攻撃でしか探すことはできません。

実際のパスワードの生成は入力された文字列にシステムが用意したSalt（塩）と呼ぶ初期値を加えたものを一方向性ハッシュ関数に与え、何度か繰り返し計算をしてビットのパターンをしつこいくらいに撹拌します。

一方向性ハッシュ関数には、暗号化するためのDESを一方向性ハッシュ関数として代用するもの（UNIXオリジナルの方法）、一方向

図3　パスワードをシャドウ化する

シャドウパスワードを使ったシステムでは、以前のUNIXのようにエディターで直接/etc/passwdにユーザーエントリーを作るようなことはできない（なぜなら/etc/shadowには反映されていないから）。すべてadduserなどの管理用コマンドを経由しなければならない。また、パスワードのチェックが必要な一般アプリケーションの場合、シャドウのAPIを介して/etc/shadowにアクセスする。

性ハッシュ関数MD5を使うものがあります。DESを使う場合は25回繰り返して計算します。MD5の場合は1000回繰り返して計算します。RedHat5.2はLinux-PAMを利用しているのでMD5を使っての撹拌が行われています。いずれも暗号学的には十分な強度が得られていると言えます。

ログインできるユーザー

サーバーにログインできるユーザーは極力最小限にしてください。ユーザーが増えれば増えるほどパスワードが洩れる危険性が高まるからです。

しかし、何らかの正当な理由でサーバーを使用する必要があるユーザーがいる場合は、利用を拒否するのではなく、きちんと教育したあとに利用させるようにしてください。ここでは説明は省きますがルールとは最終的にバランスの問題に還元されると筆者は考えるからです。

たとえば「ネットワーク経由での一切のログインを禁止しコンソールからの作業のみ許す」というルールを作るのは簡単ですが非現実的です。ネットワークでのログインには、今後連載で取り上げる予定のSSHのようなシステムを使いましょう。

サーバーでNISは使わない

NISはサン・マイクロシステムズが考案したネットワーク経由でユーザーのアカウントやネットワークホストの管理を行うシステムです。UNIXを中心としたLAN環境では広く使われているものです。LinuxにもNISのパッケージがあります。

ウェブサーバーやメールサーバーを動かすサーバーマシン（以下サーバー）上でNISを利用することはやめましょう。ネットワーク上で利用するユーザーのアカウントが一括して管理できる利点はあります。しかし、こういったサーバーのアカウント管理の問題とサーバー以外でのネットワーク全体のアカウント管理の問題が混在してしまいます。サーバーのアカウントの問題はサーバーの中だけに限定すべきです。

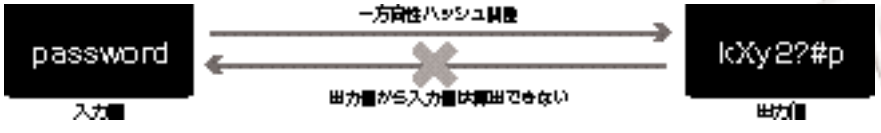
OTPはセットアップが必要

OTP（One Time Password、使い捨てパスワード）とは、ログインするごとにシステムが要求するパスワードが異なるようなパスワードのシステムです。システムに与えるパスワードは、ワンタイムパスワードを生成するプログラムや専用ICカードを使って計算します。残念ながらLinux上では広くは利用されていません。現時点の標準パッケージでは、S/KeyやOPIEのようなOTPのシステムとしてよく知られているものでもパスワード認証システムに組み込まれていません。これらを利用するとなると、独自のセットアップが必要になります。

次回

TCP WRAPPERに関しての話題を取り上げます。

図4　一方向性ハッシュ関数



セキュリティートークンカードが本当の意味でのパスワードとなる

ここで解説するのは専門的な話になります。必ずしもSOHO環境で使える話ではないのですが、知識としてぜひ持っておいてほしいことです。

筆者は「記憶に頼るタイプのパスワードによる認証の限界は利用するユーザーの賢さを超えることはできない」と指摘しました。ユーザー教育を徹底できない大規模ユーザーシステムでは、記憶に頼るパスワードはあっと言う間に破綻してしまいます。もちろん86パーセントという非現実的な数字にはなりませんが、それでいても10パーセントや20パーセントは抜け穴となる弱いパスワードを使うユーザーが現れるでしょう。

筆者は大規模ユーザーを抱えるサイトにはSecure IDやSafewordのような人間の記憶に頼らないセキュリティートークンカードの導入をすすめます。ユーザーが付けるパスワードから発生するセキュリティーホールを防ぐには、このように人間に左右されない方法が必要です。

今まで0円だった認証のための初期投資コストが1人当たり1万円を超すようになると思います。しかし、企業などで必要なセキュリティーの対コスト効果を考えれば、ほかの特殊な認証方法を使うよりも安い買い物だと筆者は考えます。人間ですから、万が一パスワードが知らぬ間に他人に洩れてしまった場合などは、そのパスワードで侵入されていても気付くのが非常に困難です。

セキュリティートークンカードの利用範囲はまだ広がってはいませんが、セキュリティーが必要なところは必ずこのような「人の賢さに左右されない」方法を利用するべきです。

SecureID
URL www.securitydynamics.com/japan/html/products/securid_tokens.html

［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