さらに進んだ インターネット・セキュリティ

サーバー構築/運用術

第5回 使い捨てパスワード

いまだに大半のLinuxディストリビューションが，パスワード認証を採用しています。しかし，毎回同じパスワードを入力するこの認証方法は再生攻撃を受ける恐れがあり，セキュリティ上かなり問題です。1度使うと2度と使うことができない「使い捨てパスワード」を用いることにより，再生攻撃を防ぐことができます。

（ケイ・ラボラトリー　仙石 浩明）

最近は，携帯電話やPDA（携帯情報端末）の普及によりずいぶん便利になりました。移動中でも携帯電話にメールを着信させることができますし，書きかけの原稿をPDAへ転送して満員電車の中で執筆を続けることもできます*1。

こうなってくると次第に欲が出てきて，いつでもリモート・ログインできる端末を持ち歩きたくなってきます。適当なモバイルPCがないかと調べたのですが，ノートPCは最軽量のものでも1kg前後で，とても常に持ち歩ける重さではありません。私が欲しいと思うPCまたはPDAの条件を列挙してみると次のようになります。

- 重量400g以下，厚み2cm以下
- PCカード
- キーボード一体型
- 端末エミュレータ
- Webブラウザ

PDAというとコンパクト・フラッシュしか使えないものも多いのですが，手持ちのLANカードやPHSカードを使いまわしたいので，PCカード対応は譲れないところです。キーボードが外付けのPDAはいくつかありますが，出先での使用を考えると一体型でないと使いにくそうです。端末エミュレータは，VT100互換で，日本語表示が可能であれば何でも構いません。もちろん，sshに対応していると一番良いのですが。

IBM PC互換機でこれらの条件を満たすのは，まだしばらく難しいのは仕方がないとして，PDAなどでいまだにこういった機種が発売されないのはなぜなのでしょう。HP 200LXの発売から5年以上もたっているというのにです。

PalmがはやるとPalmやPalmサイズのPDAだらけになり，またノートPCでも1kg前後のものが主流になるとそれまで800gのA5サイズのノートPCを作っていたメーカーまでが一回り大きいノートへ鞍替えしてしまうという，右へならえ体質はなんとかならないのでしょうか。どこのメーカーの製品も代わり映えせず，PCを選ぶ楽しみが無くなってしまいます。

Dynabook J-3100SS001（初代ダイナブック）とか，Libretto 20とか，Cassiopeia FIVA MPC-10xなどといったような個性的なマシン*2がもっと出てくることを望みます。

＊1　私は外出するときは常に2台の携帯電話とTRGpro（Palm OS互換機）を持ち歩いています。
＊2　いずれも，その個性に引かれて購入してしまいました。

再生攻撃

正規のユーザーがサーバーへ送信したデータを記録しておいて，ちょうどテープ・レコーダを再生するように同じデータをサーバーへ送り付けることにより正規ユーザーになりすます攻撃を，再生攻撃と呼びます。例えば，telnetでログインするとき，ユーザーは毎回同じユーザーIDとパスワードをサーバーに送信しますから，この送信データを「録音」することに成功すれば，後は「再生」するだけでログインすることが可能になってしまいます。

「録音」すなわち盗聴ですが，現在のインターネットは1次プロバイダの回線をバックボーンとして構成されていますから，公衆回線と同程度の安全性はあります＊3。従って，十分信頼できる技術力に定評のあるプロバイダだけを常に利用するのであれば，盗聴を心配する必要はあまり無いでしょう。

しかしながら，インターネットに接続するために使用している端末，あるいはLANが信頼できるとは限りません。端末に細工してあってキー入力をすべて記録されているかも知れませんし，LANのどこかにパケットを記録する装置がつながっている可能性もあるでしょう。出張先などで端末を借りる場合は要注意です。

WWW（World Wide Web）の普及により，LAN上のPCから自由にインターネットへアクセスできることが当り前になってしまいましたから，内部LANは安全，インターネットは危険，という図式は全く成り立たなくなってしまいました。LAN内のいずれかのマシンが侵入されていて盗聴のためのソフトウエアを仕掛けられている可能性を否定できません。

同じパスワードを複数のマシンで使っていると，危険はさらに高まります。安全でないLANで一度でもパスワードを流してそれが盗聴されれば，同じパスワードを使っているすべてのマシンへ侵入されてしまいます。

つまり再生攻撃が可能なプロトコルを利用する限り，端末やLANの安全性に常に細心の注意を払わなければならず，そんな注意を払うくらいだったら，再生攻撃が不可能なプロトコルを用いたほうがよっぽど楽と言えるでしょう。

シャドウ・パスワード

Linuxディストリビューションの大半は，標準でシャドウ・パスワードを実装しています。つまり，だれでも参照可能な/etc/passwdにパスワードを暗号化して登録する代わりに，root権限を取得した者だけが参照可能な/etc/shadowにパスワードを登録するようにし，/etc/passwdには（そのファイル名に反して）パスワードを登録しない方法です。

root権限がなければパスワード・ファイルを参照できないので，一見安全性が高まったように感じられるのですが，認証方法自体は依然として再生攻撃が可能であることに注意してください。すなわち一度でもパスワードを盗聴されれば一巻の終りです。

再生攻撃が可能であるという本質的な問題点を放置したまま，パスワード・ファイルを一般ユーザーが読めないようにするというささいな対策だけを行っている点で，シャドウ・パスワード方式はナンセンス＊4と言えるでしょう。

通信路暗号化

ssh（Secured Shell）＊5などを使って，サーバーと端末間の暗号化を行えば，LAN上での盗聴を完全に防ぐことができます。しかし，端末は安全でしょうか？ほかの人が使う可能性が全く無い個人

＊3　「インターネットは盗聴の危険があるため，企業間通信では専用のプライベート・ネットワークを構築すべき」と主張する記事をみかけることがありますが，あまり賛成できません。
＊4　まあ，無いよりましではあるのですが，「シャドウ・パスワードを使っているから安全」という印象をユーザーに与えているとしたら，有害と言えます。
＊5　日経Linux2000年12月号から2001年2月号の連載「実践で学ぶ，一歩進んだサーバ構築・運用術（第9回，第10回，第11回）」の「ssh」を参照。

専用の端末か，あるいは複数の人が使うPCならば各個人のデータの秘密が保たれるOSおよびシステムを使う必要があります。

前者は，例えば常に持ち歩いているノートPCなどです。紛失や盗難に備えてパスワード・ロックが自動的にかかる＊6ような仕掛けにしておくべきです。後者のPCは，サーバーと同程度に安全になるようセキュリティ対策を行わなければなりません。sshでお互いを信頼し合うマシン群は一蓮托生（いちれんたくしょう）ですから，どちらかがぜい弱であれば他方もその影響を受けてしまいます。

このほかの場合，例えば出張先などで端末を借りる場合など，端末の安全性を十分確認できない場合は，端末にパスワードを入力するべきではありません。sshには秘密かぎを用いる認証方式のほかに，パスワードを使った認証方式もサポートしていますが，もし端末に細工がしてあって，このパスワードが盗み読みされてしまったらサーバーを危険にさらすことになります。このような危険を防ぐため，sshサーバーの設定ファイルでパスワード認証方式を禁止すべきです（**図1**）。

使い捨てパスワード

では，信頼できない端末を使ってサーバーへアクセスしたい場合はどうすべきでしょうか?信頼できない端末に秘密かぎをインストールすることはできませんから，sshのような秘密かぎを用いる認証方式は利用できません。従ってパスワードで認証することになりますが，再生攻撃を防ぐには毎回パスワードを変更する必要があります。つまり一度使うと二度と使うことができない「使い捨てパスワード」（OTP，One Time Password）です。

OTP計算機

使い捨てパスワード方式には，その都度新しいパスワードを表示する携帯型の専用ハードウエアを利用する方法と，サーバーから受信した「チャレンジ」から「レスポンス」を計算するOTP計算機を利用する方法があります。

前者が専用ハードウエアを常に持ち運ぶ必要があるのに対し，後者はさまざまなプラットフォーム上で動作するOTP計算機（のソフトウエア）を利用できます。Linux，Windowsなど各種OS用はもちろん，Javaで動作するものもありますから，Webブラウザ上で使い捨てパスワードを計算することも可能です。

```
RhostsAuthentication            no
RhostsRSAAuthentication         no
PasswordAuthentication          no
PermitEmptyPasswords            no
RSAAuthentication               yes
```

図1　sshサーバーの設定でパスワード認証を禁止する

ただし，OTP計算機にパスフレーズを入力するときは，盗み読みされないよう十分に注意してください。ネットワークに接続されているPCは利用しない方が無難かも知れません。PDAを常に持ち歩いている方は，PDAにOTP計算機をインストールしておくと便利でしょう。Palm OS互換機ならば，pilOTP＊7が利用できます。サーバーから受信したチャレンジ「otp-md5 470 as5266」（図5参照）に対し，パスフレーズを「Phrase:」へ入力して，レスポンス「WOK MOP GAY HAM CUP VAN」を計算した実行例を**写真1**に示します。

チャレンジ「otp-md5 470 as5266」のうち，「md5」はハッシュ関数の種別，「470」はシーケンス番号，「as5266」はシード（種）を表わします。

図2に，OTP計算機が使い捨てパスワードを計算する方法を示し，以下順を

＊6　大抵のノートPCでは，電源ON時にパスワードを要求するように設定することが可能です。さらに，ハード・ディスクを外して別のマシンで内容を解析されてしまう場合に備えて，sshの秘密かぎは必ずパスフレーズで保護するようにしましょう。
＊7　http://astro.uchicago.edu/home/web/valdes/pilot/pilOTP/を参照。

追って説明します。

(1) まずシードとパスフレーズ「hiroaki sengoku」*8をつなげて1つの文字列「as5266hiroaki sengoku」を作り，これをハッシュ関数MD5で変換します。

写真1　Palm OS互換機で利用可能なOTP計算機
文字が化けるのはご愛敬。日本語に対応していないだけで動作に支障はない。

MD5は任意の文字列を入力とし，128ビットの数値（ハッシュ値）を出力する一方向関数です。すなわち逆変換（あるハッシュ値に対応する文字列の1つを求めること）が極めて困難な関数です。

(2) 変換で得られた128ビットのハッシュ値「BB78 405C 4281 42AE 5AE0 24FD 8304 B88F」*9の第0～31ビットの数値「bb78 405c」と，第64～95ビットの数値「5ae0 24fd」のXOR（排他的論理和）を計算すると32ビットの数値「E198 64A1」になります。同様に，ハッシュ値の第32～63ビットと第96～127ビットのXORを計算すると「C185 FA21」になります。こうして求めた2つの32ビット数値をつなげて64ビット数値「E198 64A1 C185 FA21」を求めます。

(3) この64ビット数値を(1)と同様にMD5で変換して128ビット数値を得，さらに(2)と同様にXORを計算すると64ビット数値が得られます。この操作を，シーケンス番号の回数（470回）だけ繰り返します。こうして得られた64ビット数値「45A5 2C59 0C60 C886」を「(16進数) 数値で表現した使い捨てパスワード」と呼びます。

(4) 数値で表現した使い捨てパスワードを2ビットずつ区切って2ビットのパリティを求め，この2ビットを追加して66ビット数値を求めます。

(5) 66ビット数値を11ビットづつ区切って6個の数値（0～2047）を求めます。

(6) 変換表（**表1**にその一部を示します）を用いて，6個の数値を6個の英単語に変換し，それらを空白でつなげた文字列を，「英単語で表現した使い捨てパスワード」と呼びます。

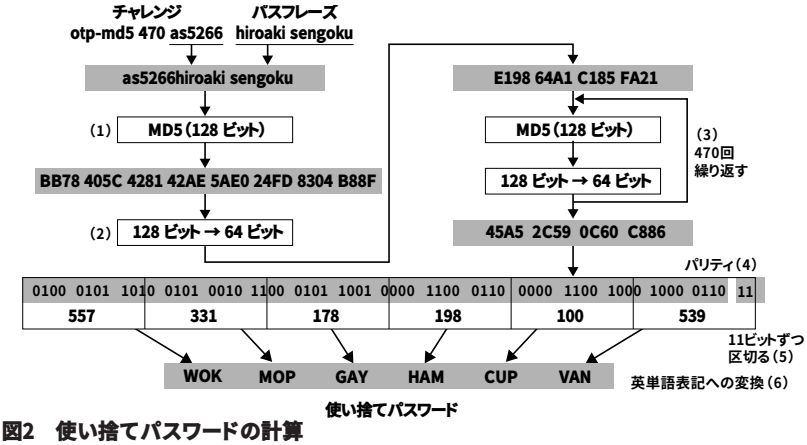

図2　使い捨てパスワードの計算

OPIE

困ったことにLinuxのディストリビュー

＊8　説明のためにあえて単純なパスフレーズを使っています。実際には，第三者が予測することが困難なパスフレーズを使うようにしてください。
＊9　見やすくするために16ビットずつ区切って，それぞれ4桁の16進数で表現しています。これ以降の説明も同様です。

ションの大半が，標準では使い捨てパスワードをサポートしていないようです。telnetやftp，suで使い捨てパスワードを使うには，OPIE（One-time Passwords In Everything）＊10をインストールすると良いでしょう。

OPIEをインストールすると，/bin/login，/bin/su，そしてftpサーバーが，使い捨てパスワード対応のものに置き換えられます。また，ユーザーがパスフレーズなどを設定・変更するためのツールopiepasswdや，OTP計算機のopiekey＊11がインストールされます。

使い捨てパスワードで認証を行うには，まずopiepasswdを使ってユーザーごとの認証情報を設定する必要があります。opiepasswdの実行例を**図3**に示します。ここでは，シーケンス番号を「471」，シードを「as5266」に設定していますが，どちらも省略することができて，その場合デフォルトの値が設定されます。

「Response: 」で入力する使い捨てパスワードは，各ユーザーが自分のOTP計算機を使って算出する必要があります。「-c」オプションを指定すれば，opiepasswdがOTPの計算を行ってくれるので，パスフレーズを入力するだけで済みますが，盗聴される危険がないか十分に確認しなければなりません。

表1　数値（0〜2047）から英単語（1〜4文字）への変換

数値	英単語
0	A
1	ABE
2	ACE
3	ACT
4	AD
	～
	中略
	～
2045	YELL
2046	YOGA
2047	YOKE

```
asao:/ # opiepasswd -n 471 -s as5266 sengoku↵
Adding sengoku:
You need the response from an OTP generator.
New secret pass phrase:
      otp-md5 471 as5266
      Response: RAIL PAN MAKE KITE DEEM MAP↵

ID sengoku OTP key is 471 as5266
RAIL PAN MAKE KITE DEEM MAP   ↵このマークで改行
```

図3　opiepasswdを使って認証情報を設定する

シーケンス番号は「使い捨て」ごとに1ずつ減っていき，0になると認証できなくなります。opiepasswdは，減ってしまったシーケンス番号を戻すためにも使われます（後述）。

図3を実行すると，/etc/opiekeysに**図4**に示すような行が追加されます。先頭から順に，ユーザーID，シーケンス番号，シード，数値で表現した使い捨てパスワード，最終更新時刻です。この後，telnetでログインしたときの例を**図5**に示します。

このときにサーバー側で行われている認証の方法を**図6**に示し，順を追って説明します。

(1) ログインしようとしているユーザーのシーケンス番号（470番）およびシードを，/etc/opiekeysから取り出します。

(2) 470番から1を引いたシーケンス番号469番をチャレンジとしてユーザーへ送信します。

(3) ユーザーがOTP計算機を使ってシーケンス番号469番の使い捨てパスワードを計算し，英単語表現に変換したものをレスポンスとしてサーバーへ送信します。

(4) 英単語表現を数値表現へ変換して，シーケンス番号469番の使い捨てパスワードであるとユーザーが主張する数値を得ます。前述した使い捨てパスワードの計算方法（図2）によれば，469番の使い捨てパスワードは，図2中の(3)の変換操作を469回繰り返し，470番は470回繰り

＊10　US Naval Research Laboratoryが開発しました。http://www.inner.net/pub/opie/から最新版がダウンロードできます。
＊11　キー入力が盗み読みされる可能性がある環境下では使ってはいけません。コンソールでの使用に限定した方が無難です。

```
sengoku 0471 as5266          cbe63ed953971a4e  Jun 03,2001 17:26:36
```

図4　認証情報

```
% telnet asao.gcd.org
Trying 210.145.125.162...
Connected to asao.gcd.org.
Escape character is '^]'.
login: sengoku
otp-md5 470 as5266 ext
Response: WOK MOP GAY HAM CUP VAN
Linux 2.2.19.

asao:/home/sengoku %
```

図5　使い捨てパスワードを使ってログイン

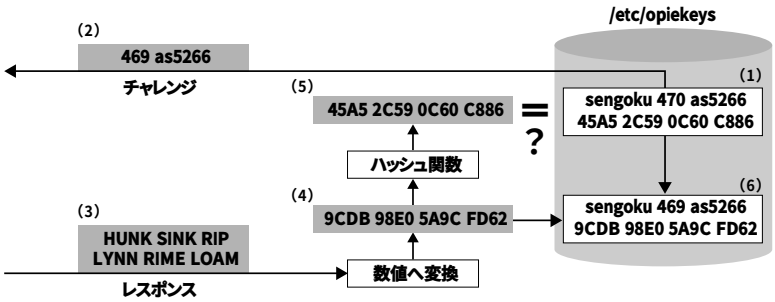

図6　使い捨てパスワードによる認証

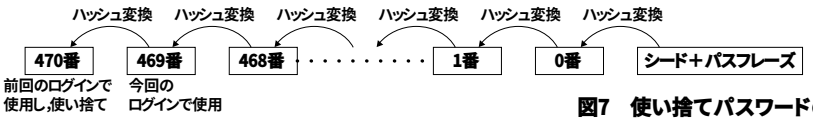

図7　使い捨てパスワードの列

```
asao:/home/sengoku % opiepasswd
Updating sengoku:
You need the response from an OTP generator.
Old secret pass phrase:
      otp-md5 17 as5266 ext
      Response: LINE MADE HOLD ALOE DIAL YELL
New secret pass phrase:
      otp-md5 499 as5267
      Response: BREW PAM CAB WINK NIBS CAKE

ID sengoku OTP key is 499 as5267
BREW PAM CAB WINK NIBS CAKE
```

図8　シーケンス番号とシードの再設定

返すことによって得られます(**図7**)から，469番の使い捨てパスワードに，もう一回この変換操作を繰り返せば，470番の使い捨てパスワードが得られるはずです。

(5)従って，ユーザーが入力した使い捨てパスワードが正しいものであれば，ハッシュ変換操作（図2の(3)の操作）後の(5)(図6)は，シーケンス番号470番の使い捨てパスワードに一致するはずです*12。そこで，(5)と(1)(図6)の使い捨てパスワードと一致している場合は，レスポンスが正しいと見なして，ユーザーのログインを許可します。

(6)(4)で得られた，シーケンス番号469番の使い捨てパスワードを/etc/へ書き込みます。

シーケンス番号は，ログインするたびに減っていきます。0になってしまうとログインができなくなり，システム管理者に認証情報を再設定してもらう必要があります。シーケンス番号が小さくなってきたら早めにopiepasswdを使ってシーケンス番号を設定し直したほうが良いでしょう。実行例を**図8**に示します。このとき同時にパスフレーズを変更することもできます。

＊12　当然ですが，逆は成立しません。つまり一致したとしてもユーザーが入力した使い捨てパスワードが正しいことの証明にはなりません。しかし，ハッシュ関数の性能が十分であれば，変換操作後の値がたまたま一致してしまう危険は実用上無視して構わないでしょう。