インターフェイスの街角 (96) — マイ認証

増井俊之

2001 年 7 月号や 2002 年 1 月号で、パスワード認証の問題を解決する〝なぞなぞ認証〟を紹介しました。この方式では専用のサイトの構築が必要で、残念ながら一般の Web ページには適用できません。今回は、通常のパスワード認証を使うサイトで、なぞなぞ認証などの任意の認証手法を利用する方法を考えてみましょう。

パスワード認証の問題点

パスワードによる認証システムには長い歴史があり、ひろく使われてきました。その結果、システムの構築方法や安全性に関する多くの知見が蓄積され、計算機にログインしたり Web 上のサービスを利用する場合の認証手法の定番となっています。しかし、ユーザーにとってはひどく使いにくい面があります。

辞書に載っている単語や固有名詞の組合せによるパスワードは、最近では比較的簡単に破られてしまいます。したがって、安全性を保つには、なるべくランダムで長い文字列を使う必要があります。ところが、長いパスワードは入力に手間がかかるだけでなく、暗記も難しいため、けっきょくどこかに書き留めることになり、かえってセキュリティの問題がひろがってしまいます。最近は、多くの Web ブラウザにユーザー名やパスワードを記録する機能があるので、これを利用すれば Web 上のサービスでパスワードを入力する必要は少なくなります。とはいっても、パスワードを記憶したノート PC を盗まれたりすると、あらゆるサービスに自由にアクセスされるため、重要なサービスでこのような機能を使うのは危険です。

数多くのサービスごとに異なるパスワードを設定すると、たとえ文字列が短くてもふとした拍子に忘れてしまい、サービス提供者に問い合わせる結果になります。提供者側からみると、いくらきちんとした認証システムを構築しても、パスワードを忘れた場合の処理に無駄なコストがかかることになります。パスワードと生体認証などの装置を併用し、ユーザーの便宜と安全性を両立させるシステムもあります。しかし、複数のシステムを組み合わせると問題が増えることも多く、注意が必要です。このような認証システムの問題点については、画像認証システムを開発しているニーモニックの國米氏が詳細に指摘しています[1]。

パスワード認証には数多くの実装があり、攻撃手法やそれへの対策などもよく知られています。ですから、上に述べたような問題はあっても、適切に運用すれば安全だろうと考える人が多く、今後もすたれることはないでしょう。

なぞなぞ認証や、ニーモニックが考案した画像認証方式「あわせ絵」システム [1] のように、パスワードよりも効果的と思われる手法はいろいろあります。しかし、それらを利用するにはシステム全体に手を入れる必要があり、現実のサービスで運用するのは容易ではありません。認証システムをゼロから実装するのは大変ですし、新しいシステムの場合、バグなどによるセキュリティ・ホールが存在する確率は、パスワード認証などよりもはるかに高くなります。また、新しいシステムに戸惑うユーザーも多いでしょう。認証システムはサービスの中核をなすものではなく、システムを開発/運用する側もこういった面倒は避けたいはずですから、優れた認証システムを知っていても〝認証は、とりあえずパスワードで〟ということになるのは目に見えています。

原理的に優れていても、コストや危険性が高くユーザー

1 http://www.mneme.co.jp/data/thesis.html

にとって扱いにくい認証システムが急速に普及するとは思えません。その意味では、パスワード認証システムを使いやすくする方法を考えたほうが現実的です。

パスワード認証システムの有効利用

さきほど述べたように、パスワード認証の最大の問題は、入力に手間がかかり、暗記も難しい点にあります。したがって、パスワードを入力したり暗記する方法をやめ、別の手段でパスワードを生成して入力できるようにすれば、かなり使いやすくなる可能性があります。自分しか知らない属性をなんらかの方法でパスワード文字列に変換する方式なら、従来のパスワード認証システムを変更せずに、異なる認証システムが使えます。

たとえば、指紋や鍵の形状を文字列に変換するシステムがあれば、その文字列を用いて、通常のパスワード認証をおこなうサービスが利用できるでしょう。

このように、パスワードを中間言語のようなものと捉え、なんらかの方法でパスワード文字列を生成することで、任意の認証手法をパスワード認証システムで使えるようになります。その場合、ユーザーは実際のパスワード文字列を知る必要すらありません。

このような〝中間言語方式〟には、いろいろな実装があるでしょう。たとえば、USB キーなどのハードウェア、あるいはマウス操作やなぞなぞ認証といったソフトウェアでパスワードを生成する手法などが考えられます。

Greasemonkey でマイ認証

1 月号で紹介した Greasemonkey を応用すれば、パスワード認証を利用するサービスに前述の方式を適用し、パスワード入力の代わりに自前の認証システムが使えます。

自前の認証をおこなう CGI と、認証結果を取得する CGI スクリプトを用意し、以下の機能をもつ Greasemonkey スクリプトから両者を呼び出します。

- password 属性をもつフォームをページ内で検索する。
- そのようなフォームがあれば新しいウィンドウを開き、自前の認証を実行する CGI を呼び出す。
- 同時に認証結果を取得する CGI を動かし、得られた結果をフォームに貼り付ける。

図 1　Greasemonkey の設定

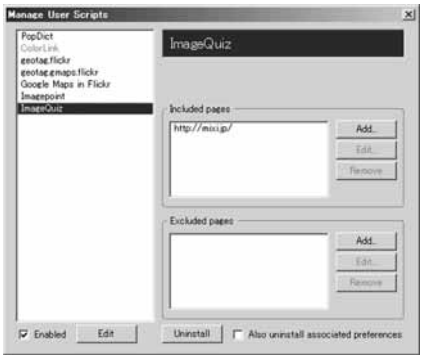
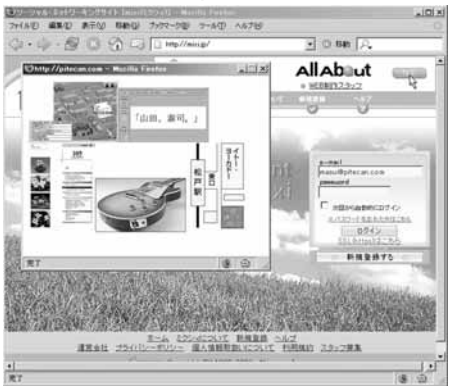


図 2　Mixi のログイン画面にアクセス



画像なぞなぞ認証の実装

Greasemonkey にリスト 1 のスクリプト (imagequiz.user.js) をインストールして図 1 のように設定し、Mixi[2] にアクセスします。すると、パスワード入力欄の色が変わって imagequiz.cgi が呼び出され、図 2 のなぞなぞ認証画面が表示されます。表示されている 5 枚の画像のうち、私と関連のあるものは 1 つだけです。これをクリックして選択すると、別の写真が表示されるので、ふたたび私に関連のあるものをクリックします。これに 5 回連続成功するとパスワードがパスワード欄に入力されます。

この操作ではユーザーはキーボードに触れていませんが、それでもパスワード入力を求めるサイトにログインできます。

画像上の操作を利用する認証

なぞなぞ認証の場合はクリックを 5 回繰り返すという手順があるので、無限に操作すればたまたま認証に成功する

2 http://mixi.jp/

リスト 1　なぞなぞ認証呼出しスクリプト (imagequiz.user.js)

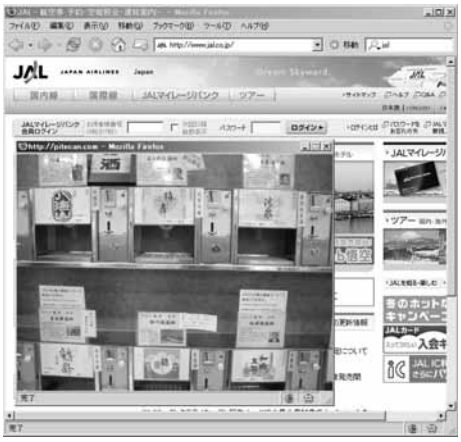
```
// imagequiz.user.js
// $Date: 2005-12-21 14:44:53 +0900 (Wed, 21 Dec 2005) $
// Copyright (c) 2005, Toshiyuki Masui (http://pitecan.com)
// Released under the GPL license
//
// ==UserScript==
// @name ImageQuiz
// @namespace http://pitecan.com/MyPassword
// @replace <input type="password"> with user password
// @include http://*
// ==/UserScript==

function getpassword(te){
  // マイ認証用のウィンドウを開いてマイ認証を実行する。
  // authcgiでマイ認証をおこなった結果をgetcgiで取得。
  // getcgiは認証の終了まで待つ。timeoutでタイムアウト。

  authcgi = "http://pitecan.com/MyPassword/programs/imagequiz.cgi";
  getcgi = "http://pitecan.com/MyPassword/programs/getpassword.cgi";
  timeout = 20000;

  d = new Date;
  id = d.getTime();
  w = window.open(authcgi+"?url="+encodeURIComponent(location.href)+
      "&id="+id,"","width=400,height=400");

  setTimeout(function(){ w.close(); },timeout);
  setTimeout(function(){ te.value = ''; },2000); // 最初にパスワードをクリア

  GM_xmlhttpRequest({
    method:"GET",
    url:getcgi+"?id="+id,
    onload:function(details){
      te.value = details.responseText;
    }
  });
}

inputs = document.getElementsByTagName("input");
for (var i = 0; i < inputs.length; i++) {
  te = inputs[i];
  if(te.type == "password"){
    te.style.backgroundColor = "#ffff80"; // マイ認証の色
    te.value = '';
    getpassword(te);
  }
}
```

ことも考えられます。しかし、どんな操作でログインするかが分からなければ、第三者が認証をすり抜ける確率はほぼゼロになるでしょう。

そこで、画像の選択を繰り返すのではなく、画像上でマウスを操作し、それにもとづいて認証をおこなう方法を実装してみました。まず、imagequiz.user.js を imagepoint.user.js という名前でコピーし、imagepoint.cgi を呼び出すように書き換えます。そして、imagepoint.user.js を Greasemonkey に登録し、JAL のサイト (http://www.jal.co.jp/) にアクセスすると、このスクリプトが呼び出されるようにします。

JAL のサイトにアクセスすると、1 枚の画像が表示されます (図 3)。この写真は、越後湯沢駅にある新潟の清酒の試飲システムで、いろいろな酒を 1 杯 100 円で飲めるというものです。この画像の上でなんらかのマウス操作をおこなうとパスワードが生成され、さきほどと同様にパスワード欄に貼り付けられます。操作は、銘柄を順にクリックするのかもしれませんし、〝八海山〟の部分を何回もクリック

図 3　JAL のログイン画面にアクセス



するのかもしれません。いずれにせよ操作に関する手がかりは皆無なので、それを知らない人がパスワード生成に成功する確率はかなり低いでしょう。なぞなぞ認証よりは操作が憶えにくく、忘れてしまう可能性もありますが、なぞなぞ認証よりレベルを強くすることも弱くすることも簡単です。

安全の工夫

今回の実装手法では、サーバーから送られるパスワードを getpassword.cgi で取得しているため、タッピングなどでパケットを盗み見られるとパスワード文字列が漏れてしまいます。ユーザーの操作をもとにローカルにパスワードを生成すればよいのですが、そうすると操作方法が分かってしまい、画像なぞなぞ認証の実現は難しくなります。文字列を暗号化して送る方法も考えましたが、パスワードをサーバーのどこかに記録しておくかぎり、危険はなくなりません。実際の運用には、もうひと工夫が必要でしょう。

楽しい認証

パスワードを思い出して入力する作業は煩わしいものですが、画像なぞなぞ認証のような、自分の知識や能力にもとづく方式ならどうでしょうか。認証に使う写真を選んだり、自分しか解けない問題や技術を工夫するのは楽しいでしょう。知的なトリックを使ったり、ゲーム的な要素をもたせることも可能です。自分だけが使える認証手法を考えるのは簡単ではありませんが、憶えにくいパスワードをむりやり暗記するよりもはるかによいのではないでしょうか。

パスワード認証では、入力しているところを他人から見られないように隠すのが普通です。しかし、傍目には何をしているのか分からない方法なら、手順を人に見せて煙に巻くこともできるでしょう。

認証手法は無限にありますが、誰にとってもベストな手法を考えるのは難しいので、楽しいもの、強力なもの、簡単なもの、おもしろいものなど、状況に適した方法を利用するほうがよいでしょう。たとえば、以下のようなものが考えられます（括弧内は必要な知識や能力です）。

- 与えられた地名を地図上でクリックする（地名の知識）。
- 詰め将棋の問題を解く（将棋の実力）。
- 表示された楽譜を演奏する（楽器と譜面を読む技術）。
- 聞こえる和音をそのまま弾く（絶対音感）。

パスワードによる認証結果には成功/失敗の 2 通りしかありませんが、この手法なら、知識や能力に応じて認証レベルを変えることができるでしょう。全問即答なら問題はありませんが、回答に時間がかかった場合は、それなりに注意してサービスを提供するといった配慮が必要です。

おわりに

今回紹介した方法の有効性は、ふだん利用しているさまざまなサイトで検証できます。新たなシステムの使いやすさは、短期間の実験ではなかなか判定できません。毎日、さまざまな認証手法を使っていくうちに、より効果的なアイデアが頭に浮かぶ可能性も高くなるでしょう。

今回はソフトウェアによる実装だけ紹介しましたが、デスクトップ PC などの据置きシステムで利用する場合は、自宅の鍵の形状をパスワード文字列に変換するといった、特別なハードウェアの利用も実用的かもしれません。今後、各種のハードウェアやソフトウェアを用いた認証の実験をおこなおうと思っています。

（ますい・としゆき　産業技術総合研究所）

[参考文献]

[1] 高田哲司、小池英樹「あわせ絵：画像登録と利用通知を用いた正候補選択方式による画像認証方式の強化法」、情報処理学会論文誌、Vol.44、No.8、2003 年 8 月