

特集
1-3

実践! スマート クライアント

ここが重要!
スマートクライアントの勘どころ
よりセキュアなアプリケーション構築とつまづかないための方策

出口 俊輔 株式会社ハローシステム <http://www.hellosystem.co.jp/>
DEGUCHI, Shunsuke



前項で作成したWebサービスですが、何か問題はないでしょうか？

そう、セキュリティがまったく考慮されていません。もし、このまま公開サーバーに配置した場合は、データは盗られ放題、更新／削除され放題になってしまいます。

WebサービスはURLさえわかれば、ブラウザから簡単にアクセスできてしまいます。本稿で取り上げているような、限定アプリケーションで使用するWebサービスは簡単にブラウザからアクセスされては困ります。

そこで本稿では、いかにWebサービスをアクセスできなくするかを考えます。本稿で解説する対応策は複合的に行なうとより効果があります。また、Webクライアントと同じようにSSLを使用すれば通信の暗号化ができます。インターネット経由での通信を行なう場合はSSLを使用することをお勧めします。



Visual Studio .NETで作成したWebサービスはIIS上で動作しているため、Webクライアントと同じ認証を使用できます。つまり「基本認証」や「チャレンジレスポンス」、「クライアント証明書」などの認証方法を使用できます。

「基本認証」を使う

では「基本認証」を使用する例を紹介しましょう。

先ほど作成したWebサービスに基本認証機能を追加するため、IISマネージャを起動します。

作成したWebサービスアプリケーションを右クリックし、プロパティを選択します (図1)。

表示されたプロパティダイアログボックスから「ディレクトリセキュリティ」タブを選択し、「認証とアクセス制御」の「編集」ボタンをクリックします (図2)。

「匿名アクセス」のチェックをはず

図1: WSTest1のプロパティを表示する

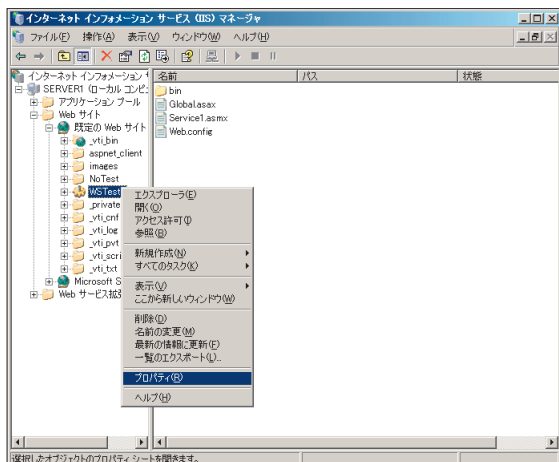
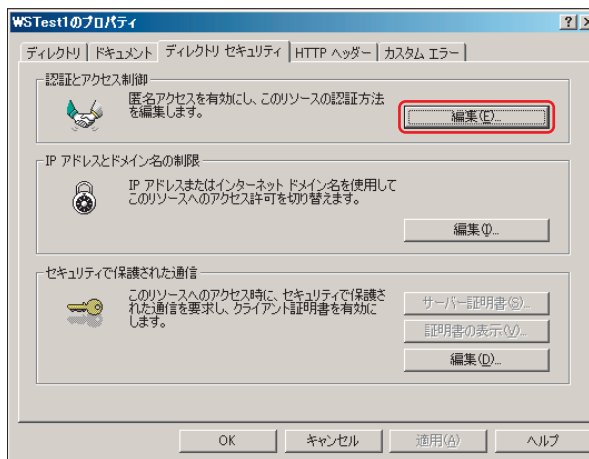


図2: Test1のプロパティダイアログボックス



し、「基本認証」をチェックします。その他項目にチェックがある場合はチェックをはずします (図3)。

準備はこれだけです。

ではこのWebサービスに直接アクセスしてみます。URLは「http://server1/WSTest1/Service1.asmx」です。ブラウザからこのURLを入力し、アクセスすると認証画面 (図4) が表示されます。

これで、ブラウザからWebサービスに直接アクセスできなくなりました。

次に作成したWindowsアプリケーションを認証に対応するように修正します。

フォームのロードイベントの最初にリスト1を追加してください。このとき、「User」と「Password」はサーバーに設定されているユーザーを使用してください。

ビルドしてから、Webサーバーに配置して実行してみてください。正常に動作することが確認できます。

ここで重要な問題があります。

実際にアプリケーションを作成する場合はUserとPasswordは必ず入力フォームを作成し、入力されたデータを使用するようにしてください。当たり前の話ですが、プログラム中に直接記述してはいけません。.NETはコンパイル結果を中間コードに保存するため、アセンブリから簡単にソースコードを再現することができます。ソースの中にサーバーのユーザーIDとパスワードを記述することは非常に危険です。

本稿では「基本認証」を使用する方法を紹介しましたが、まだまだいろいろな認証方法があります。セキュリティ度を高めたければ、設定は面倒ですが、「クライアント証明書認証」を使用するのがお勧めです。

図3: 「基本認証」だけチェックする

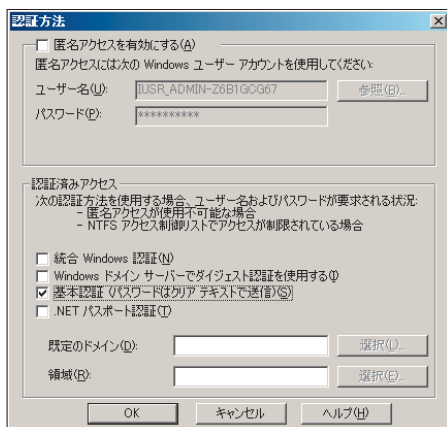
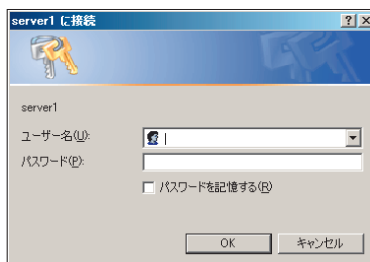


図4: おなじみの認証画面



リスト1: UserとPasswordを設定

```
wstest.Credentials = _
New System.Net.NetworkCredential(_
User, Password)
```

アプリケーションで認証を行なう

仕組みをまとめる

通常、アプリケーションを作成する場合は、一番最初にユーザーIDやパス