●アプリケーション難読化の必要性

ここでは、貴社が考慮するべき大変重要な事項について記載いたします。

それは、.NET アプリケーションの難読化の必要性です。

.NET 用に作成したアプリケーションのアセンブリ(.EXE や.DLL など)は、中間言語(IL)で作成されているので、簡単にリバースエンジニアリングをされてしまいます。
.NET アプリケーションの逆コンパイルは、無償の逆コンパイラなどを利用していとも簡単に実行できます。
逆コンパイルにより、単にコードの内容を知られるだけにとどまらず、パスワードなどのログイン情報やレジストリ名などの文字列も明らかになります。これらを放置するとその危険性がシステム全体におよび、貴社のソフトウェアビジネスに大きな損害を与えかねません。

これらのことは、「認証レスキュー！」に関わらず、一般的な問題なのですが、「認証レスキュー！」の場合を考えてみます。
「認証レスキュー！」には、次の 3 つの部分のプロジェクトソースがあることは既に説明しました。

1. **「Web」プロジェクト**
   Web サービスを利用した Web 開発部分
2. **「Package プロジェクト」**
   パッケージに組み込み、ユーザの PC 上で動作するライセンス認証インターフェイス部分
3. **「InsideSystem」プロジェクト**
   認証登録状況の表示やラベルの印刷、電話での認証などを受け付ける社内システム部分

貴社が、この 3 つの部分のプロジェクトソースをカスタマイズされご利用になった場合、完成したアプリケーションが動作する場所を考えて見ましょう。
1 の「Web」プロジェクトは、既に説明したように Web サービスとして Web サーバー上に発行され、その場所で動作します。
3 の「InsideSystem」プロジェクトを利用して作成した EXE ファイルは、社内システムですので社内 LAN 内の PC などにセットアップされるでしょう。
つまり、1 と 3 はいづれも社外に出るアプリケーションではありません。

ところが、2 の「Package プロジェクト」を利用して作成した EXE ファイルは、パッケージアプリケーションとしてユーザの手に渡ります。
さらに、今までの説明で見てきたように「Package プロジェクト」には、暗号・復号処理や Web サービスのログイン情報があり結果としてアセンブリ(.EXE)に含まれユーザに提供されることになります。
そのままでは、悪意を持ったユーザ(または第三者)がそのアプリケーションを逆コンパイルすることを妨げる方法はありません。

そこで、そのアプリケーションを難読化すれば、悪意を持ったユーザ(または第三者)が逆コンパイルしてもパスワードなどのログイン情報やレジストリ名などが露呈することはありません。

不正逆コンパイル対策のために弊社の難読化ツールなどで.NET アプリケーションを「難読化」されることを強く推奨いたします。

当製品には弊社の難読化ツール「Spices.Obfuscator 5J」が同梱されているお得な「難読化セット」もございます。
「難読化ツール」の詳細につきましては、弊社の「Spices.NET」Web サイトをご覧ください。