HP IceWall SSO

HP IceWall技術レポート：UserExitルーチン機能特集

UserExitルーチンの概要

今回の特集は、UserExitルーチンです。

いくらWebシステムの認証が基本的には似たような構造を持っていたとしても、HP IceWall SSOを導入していただくシステムのニーズ（要件）は会社やシステムによって様々です。例えば、i-modeのコンテンツをUIDの入力なく暗証番号だけで参照したい、既存の認証データベースに大きな変更を加えることなくHP IceWall SSOを導入したい、などなど。

そのような場合に活躍するのが、今回ご紹介する「UserExitルーチン」です。HP IceWall SSOが柔軟だ、拡張性が高いといわれる大きな理由は、このUserExitルーチンにあります。

この拡張機能によって、HP IceWall SSO新規システムの構築時のみならず、既に運用されているシステムに対しての導入や、導入後の機能拡張に対しても、追加作業を最小限に抑えることが可能になります。

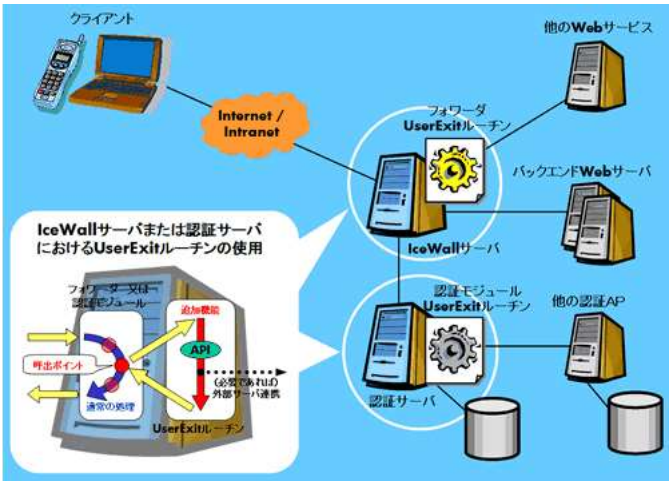
そんなUserExitルーチンを、実際の例を交えながらご紹介してまいります。

※注
UserExitルーチンはHP IceWall SSOの付加機能になりますので、以降の文章は動作環境とシステム構成や機能をご覧の上でお読みください。

1. UserExitルーチンとは？

   HP IceWall SSOには様々な機能が備わっていますが、UserExitルーチンを利用すると、さらに別の機能を追加することが可能になります。例えば、「製品サポート対象外のデータベースを使用するために、UserExitルーチンを用いてHP IceWall SSOとデータベース間の通信を実現したい。」というように、製品本体の機能を越えた要求に対しても容易にお応えすることができます。

   UserExitルーチンは、HP IceWall SSOから呼び出す外部プログラムという位置付けになります。必要な機能に対してのみ開発が必要となりますが、HP IceWall SSOでは、UserExitルーチンで利用可能な各種API（Application Program Interface）を標準オプションとして予めご用意しております。



2. UserExitルーチンの役割

   下記の2つは、UserExitルーチンによって実現する機能の代表的な例です。

   - 画面（電文）制御
     フォワーダが受け取ったリクエスト電文、及びクライアントに対する応答電文からの情報の抽出や、電文の一部追加／削除／変更。
   - オリジナル認証処理
     他のセキュリティ製品や作り込み外部認証機能の組み込み（※ HP IceWall SSOでは、フォーム認証及びクライアント証明書／ICカードによる認証が標準で実装可能です。）

   この2つを組み合わせることや、ウイルススキャン等その他の機能を持たせることも可能です。

3. UserExitルーチンの実装

   上の概要図のように、UserExitルーチンはフォワーダと認証モジュールの処理の中に組み込みます。それぞれのモジュール内部には、UserExitルーチンを呼び出すポイントがいくつかあり、任意の個所を指定することが可能です。

   ■UserExitルーチンを追加可能な個所
   フォワーダ ： バックエンドサーバへのリクエスト送信前、リクエスト受信後 など
   認証モジュール ： ユーザ認証（ログイン）直前／直後、ログアウト直前／直後 など

それでは、UserExitルーチンの活用例をご紹介します。

システム構築例その1 – フォワーダ編

UserExitルーチンをフォワーダに組み込む際の主な役割 ：電文制御

以下の例では、UserExitルーチンによって電文から任意の情報を抽出することで、オリジナルの認証システムを構築しています。

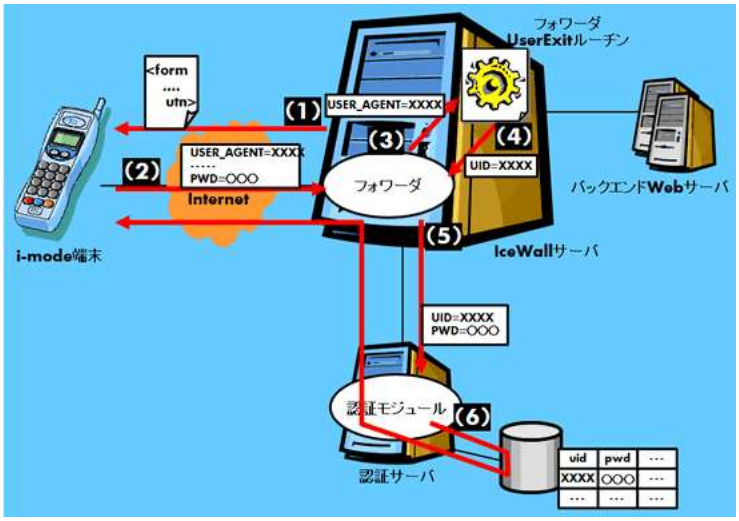
〈概要〉　i-modeサイトに対するWeb認証基盤の構築。
（※ HP IceWall SSOは標準でi-modeに対応しています。）

〈前提〉　暗証番号の入力のみでユーザ認証→アクセスを可能にする。

下記のi-mode用HTMLタグを利用したUserExitルーチンを組み込みます。

"utn"
FORMタグもしくはAタグの中に追加すると、アクセスした端末に対して製造番号をサーバに送信するよう要求する。要求を受け入れた端末は製造番号をHTTPヘッダのUSER_AGENTに埋め込む。

■ ログイン時の処理フロー



1. utnタグを埋め込んだ認証用コンテンツ（パスワード入力画面）をユーザに表示する。
2. 入力後の送信ボタン押下により、端末製造番号がUSER_AGENTに追加され、パスワードと共にフォワーダにログイン要求を送信する。
3. フォワーダ内部でログイン要求と判定した場合、認証モジュールへのログイン要求前にUserExitルーチンを呼び出す。
4. UserExitルーチン内の処理により、USER_AGENTから製造番号を抜き出し、ユーザIDとしてフォワーダに引き渡す。
5. フォワーダは、ユーザID（＝製造番号）とパスワードと共にログイン要求を認証モジュールに対して送信する。
6. 認証モジュールは、予め登録されている製造番号とパスワードをマッチングする。認証モジュールはフォワーダに、フォワーダはクライアントに対してログイン成功を通知する。
この後、ユーザはバックエンドのコンテンツを参照できる。

システム構築例その2 － 認証モジュール編

UserExitルーチンを認証モジュールに組み込む際の主な役割 ： 外部認証処理の経由

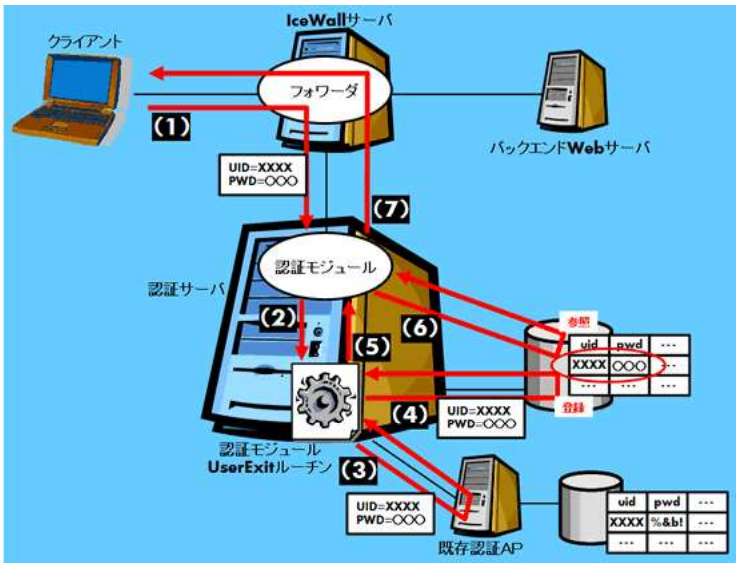
以下の例では、既存システムの認証機能をUserExitルーチンから活用することにより、従来のシステムに対して大きな変更を加えることなくシングルサインオン環境への移行を可能にしています。

〈概要〉　既存のWeb認証アプリケーションに対して、HP IceWall SSOを新しい認証基盤として導入する。

〈前提〉　・既存システムでは非可逆の独自暗号パスワードを保存している。
・ユーザに意識させることなく新システムへと移行する。
（＝新しいパスワードを発行しない。）

ユーザ情報（ユーザID/パスワード）をそのまま移行することができないため、UserExitルーチンにより既存認証システムを経由し、かつその際に新しい認証基盤に即したユーザパスワードを入手します。

■ ログイン時の処理フロー



1. クライアントはユーザIDとパスワードを入力し、フォワーダに向けてログイン要求を送信する。受け取ったフォワーダはさらに認証モジュールに向けてログインの要求を出す。
2. 認証モジュール内部でログイン要求と判定した場合、データベースへのマッチング前にUseExitルーチンを呼び出す。

3. UserExitルーチンは既存のバックエンドWeb認証アプリケーションに対してログイン要求を送信する。
   Webアプリケーションではログイン処理が実行され、成功した場合は認証モジュールに対して通知する。
4. UserExitルーチンはログインに成功したユーザID／パスワードを認証データベースへと登録する。（既に登録されていれば何も実行しない。）
   ※ ここで登録するものは、認証モジュールから受け取っているパスワード、つまりユーザが直接入力したパスワードであることがポイントとなる。
5. UserExitルーチンは認証モジュール対してログイン成功を通知する。
6. 認証モジュールはユーザID／パスワードをマッチングする。
   ※ ここでのマッチング対象は4. で登録されたものであるため、必ずログインに成功する。
7. 認証モジュールはフォワーダに、フォワーダはクライアントに対してログイン成功を通知する。
   この後、ユーザはバックエンドのコンテンツを参照できる。

3. の際に、バックエンドWebアプリケーションへのログインに失敗した場合は、4. 以降の処理は省かれ、UserExitルーチンはログイン失敗を認証モジュールに対して通知します。

（なお、本例に関しては、他にもUserExitルーチンを用いた実現方法は存在しますが、ここでは最もシンプルなパターンをご紹介しました。）

いかがでしたでしょうか？ ここで挙げたものはほんの一例ですが、実際にUserExitルーチンを活用してシステムを構築した事例は多数存在します。また、HPにはその開発ノウハウが豊富にありますし、上に記述したように、UserExitルーチンに利用可能な各種APIもご用意しております。
たとえもし、IceWall SSOではご要望を満たすことは困難であると思われた場合でも、是非一度お問い合わせください。そこには何らかの回答が用意されているはずです。

※注
ご購入のライセンスによっては、UserExitルーチンの使用に制限がございます。
詳しくはお問い合わせください。

2003.10.23 日本ヒューレット・パッカード コンサルティング事業部テクニカルコンサルタント 西谷 俊助