HP IceWall SSO

HP IceWall技術レポート：エージェントモジュール特集（1）

これまでIceWallといえばリバースプロキシ型の製品だ、とお考えになっている方がほとんどではないでしょうか。ところが、シングルサインオン製品のカテゴリの中で、もう一つの認証方式であるエージェントモジュールも、HP IceWall SSO ver.7.0からオプションとしてラインナップに加わっております。今回は、まずエージェントモジュール型とリバースプロキシ型の違い、エージェントモジュール型の処理フローについてご説明し、さらに二つの方式を併用するのに適した環境について二例ご紹介いたします。

エージェントモジュール型とリバースプロキシ型

HP IceWall SSOでご提供するSSO方式は、アーキテクチャにより「エージェントモジュール型」と「リバースプロキシ型」に分類されます。

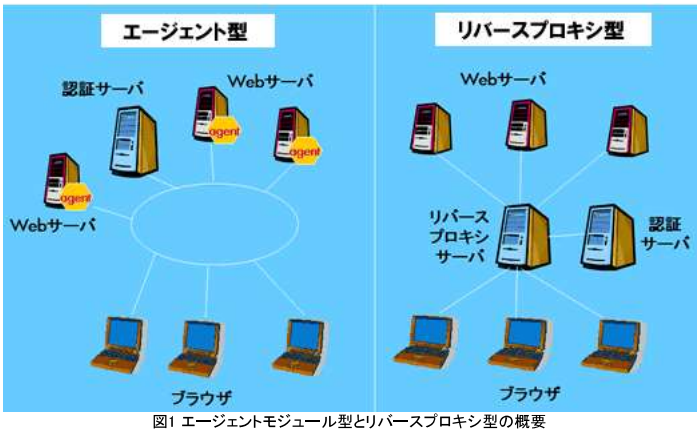


図1 エージェントモジュール型とリバースプロキシ型の概要

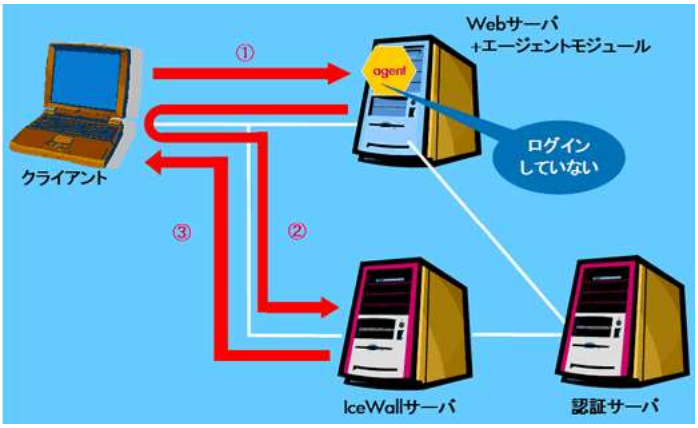
エージェントモジュール型では、認証・認可を行う認証サーバと、WebサーバのPlug-inであるエージェントモジュール（各Webサーバにインストール）の二種類のモジュールがあります。ブラウザから各Webサーバにリクエストがあると、それらのWebサーバのエージェントモジュールは、認証サーバに対して認証・認可の問い合わせを行います。
一方、リバースプロキシ型では、認証・認可を行う認証サーバと、ブラウザからの全リクエストの代行や、認証サーバに対する認証・認可の問い合わせを行うリバースプロキシサーバがあります。

エージェントモジュールの処理の流れ

ここでは、HP IceWall SSOのエージェントモジュールを導入した場合のログイン、アクセス制御の処理フローを解説します。なお、エージェントモジュールを使用する前提条件として、IceWallサーバ（フォワーダ）と認証サーバが必要です。

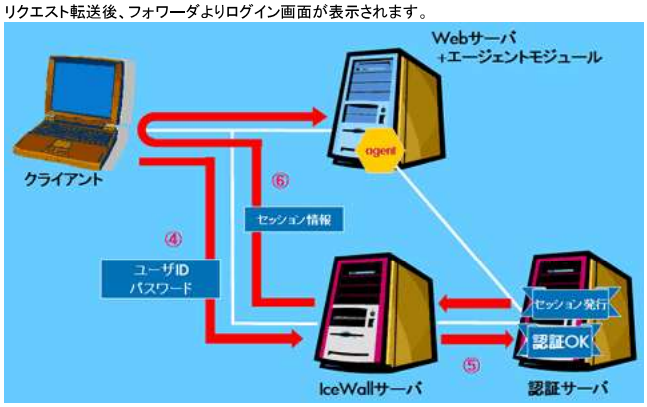
1. ログインの処理フロー

ログイン処理フローについて解説します。なお、クライアントはIceWallにログインしていない状態で、認証Cookieが発行されていないものとします。

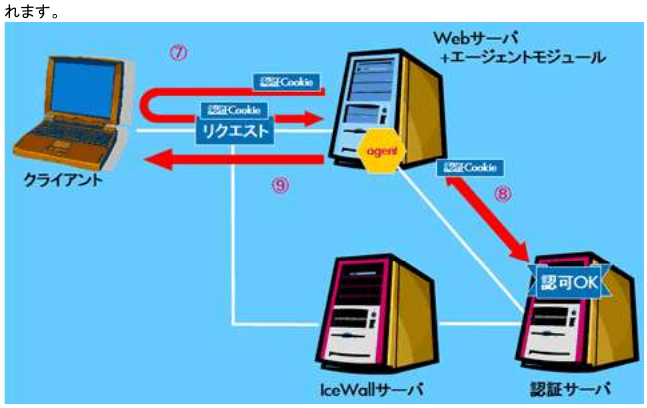


1. クライアントより、エージェントモジュールがインストールされているWebサーバ（以下Agentサーバ）へリクエストを送信します。このリクエストはAgent サーバ上に存在するコンテンツ表示を要求するリクエストとなります。
2. Agent サーバはクライアントからの要求に対してIceWall へのログインが行われているかどうかを確認します。ログインしていないと判断した場合にはIceWall サーバ上のフォワーダへ、クライアント経由でリクエストを転送します。

3. リクエスト転送後、フォワーダよりログイン画面が表示されます。



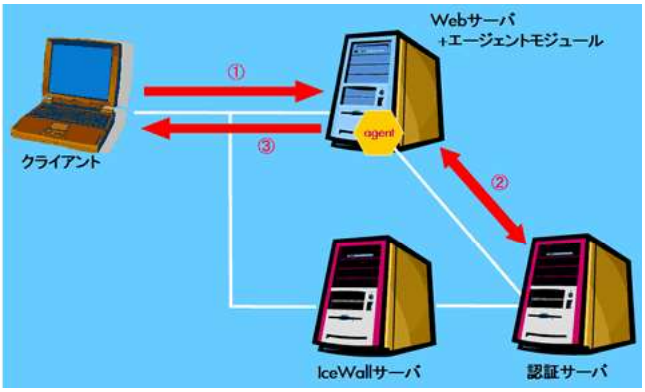
4. クライアントよりフォワーダへログイン要求を送信します。
5. フォワーダは認証サーバに対してユーザ認証を要求します。このユーザ認証の動作は従来のリバースプロキシ型のIceWallの動作と変わりません。
6. ユーザ認証に成功すると、セッション情報がフォワーダからクライアント経由でAgent サーバに転送されます。



7. フォワーダより転送されたセッション情報をクライアントに認証Cookie として通知します。クライアントは認証Cookie を受信後、自動的に①で送信したコンテンツに対するリクエストをAgent サーバに送信します。
8. Agent サーバはリクエスト受信後、リクエストしたユーザのログイン確認を行ない、認証サーバに対してリクエストされたコンテンツへのアクセス制御要求を行います。
9. アクセス制御の結果で参照権限がある場合、Agent サーバはクライアントにリクエストされたコンテンツを表示します。

以上でログイン処理は終了です。以降のAgentサーバへのアクセスの処理は「2.アクセス制御の処理フロー」のようになります。

2. アクセス制御の処理フロー



1. クライアントより、Agentサーバへリクエストを送信します。このリクエストはAgent サーバ上に存在するコンテンツ表示を要求するリクエストとなります。
2. Agent サーバはリクエスト受信後、リクエストしたユーザのログイン確認を行ない、認証サーバに対してリクエストされたコンテンツへのアクセス制御要求を行います。
3. アクセス制御の結果で参照権限がある場合、Agent サーバはクライアントにリクエストされたコンテンツを表示します。

以上がAgentサーバのアクセス制御処理フローになります。ログイン後のアクセスはIceWallサーバを経由しないため、より優れたパフォーマンスが期待できます。

エージェントモジュール型と、リバースプロキシ型のメリット・デメリットは以下のようになります。

	エージェントモジュール型	リバースプロキシ型
メリット	・ブラウザからアクセスする際に、ボトルネックになる箇所が少なく、パフォーマンスに優れている。	・使用するプラットホーム（Webサーバ）が限定されない。 ・バックエンドWebサーバに手を加える必要がない。 ・バックエンドWebサーバはクライアントから直接アクセスできないためにセキュアである。
デメリット	・Webサーバ毎に エージェントモジュール（plug-in）をインストールする必要があり、手間がかかる	・ブラウザからのアクセスは、すべてSSOサーバを経由するため、SSOサーバが高負荷になる。

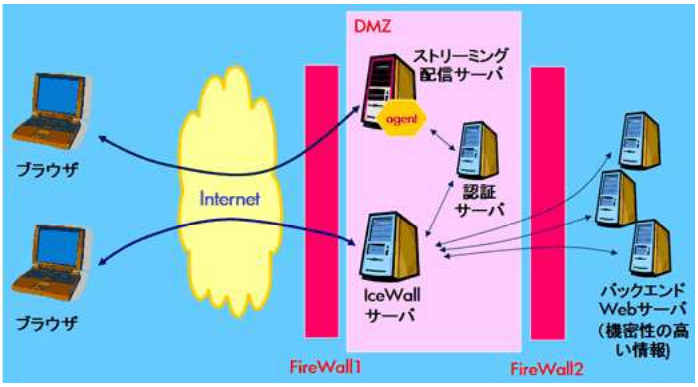
	・エージェントモジュールがWebサーバに対応していない場合がある。	

HP IceWall SSOは、リバースプロキシ型とエージェントモジュール型の両方に対応しておりますので、同一の認証サーバを使用して両方式のSSOを行うことができます。二つの方式を組み合わせて導入することで、片方だけでは多少難ありと思われるようなSSOシステムも実現可能となるのです。次に、そのような、リバースプロキシ型とエージェントモジュール型の両方を取り入れたSSOシステムの構成例をご紹介いたします。

エージェントモジュールを活用したシステム構成例

構成例1: ストリーミングサービスを含めたSSO環境

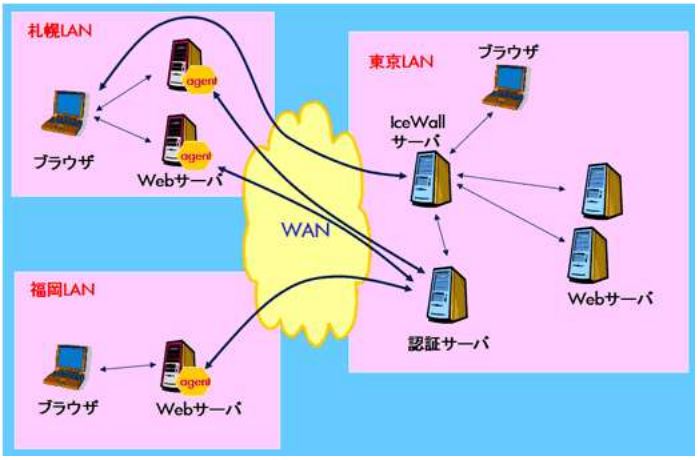
エージェントモジュールをストリーミングサーバに導入することにより、他のWebサーバとのSSOを簡単に実現します。



この例では、エージェントモジュールを導入することにより、従来リバースプロキシでは行われることがあまりなかった、SSO環境でのストリーミングサービスを実現しています。一つのIceWallシステムの中で、非常に大きなトラフィックが発生するストリーミングに関してはブラウザから直接配信サーバにアクセスし、機密性の高い情報に関しては従来通りリバースプロキシサーバ(IceWallサーバ)のバックエンドに配置することで、エージェントモジュール型とリバースプロキシ型の双方の強みが生かせます。

構成例2: 社内拠点間でのSSO

エージェントモジュールは、日本全国に散在する拠点間のSSOにも効果を発揮します。



こちらは、Webサーバが分散している環境でのSSO実現例です。
Webサーバが東京、札幌、福岡と物理的に離れた場所に配置されています。ユーザ情報については東京で一括管理することとします。
ユーザ情報が東京にあるため、認証サーバは同じ東京に配置します。東京のLAN内ではリバースプロキシ型IceWallでシステムを構成します。また、札幌と福岡に配置されたWebサーバにはIceWallのエージェントモジュールをインストールします。東京・札幌・福岡のWebサーバへのセッション情報は全て東京の認証サーバで管理されるため、拠点間でのSSOが実現されます。エージェントモジュールをWebサーバに導入した札幌と福岡にいるユーザは、東京のIceWallサーバへ毎回アクセスしなくても自分の所属するLAN内に配置されたWebサーバにSSOでアクセスすることが可能になります。
この例のように、WANを介したSSOでは、エージェントモジュールを併用した構成が好適であることが多々あります。エージェントモジュールを使用することにより、トラフィックの軽減・処理の分散が実現でき、ユーザにとって快適な環境でSSOが可能となります。

以上のように、HP IceWall SSOではリバースプロキシ型・エージェントモジュール型双方をご提供できるメリットを生かし、お客様の環境に最適なソリューションをご提案いたします。

2004.12.21 日本ヒューレット・パッカード コンサルティング・インテグレーション統括本部 テクニカルコンサルタント
松本 繁叔