HP IceWall SSO

HP IceWall技術レポート:ここが知りたい！8.0 の新機能特集1

今回の技術レポートでは「ここが知りたい！8.0 の新機能特集」の第一弾としまして、「オリジナル URL 対応（Virtual Host 対応）」について紹介いたします。

「オリジナル URL 対応（Virtual Host 対応）」とは、「 Apache HTTP サーバーの Virtual Host 機能との組み合わせにより、既存の Web アプリケーションに対して前段にフォワーダーを追加しても、ユーザがアクセスする URL を変更させないことを可能とする新機能」です。

本トピックスでは、「オリジナル URL 対応の仕組みの基本から設定」、「応用編としてSSL通信をしている場合」を順を追って、詳細に説明してまいります。

複雑な Web アプリ環境への新たなソリューション

HP IceWall SSO は、誕生当初から「リバースプロキシ型（以降、RP 型）」の SSO 製品として、その機能充実とサポートの拡大に励んでまいりました。

ご存知の通り、 RP 型のモジュールには
「使用するプラットホーム（Web サーバー）が限定されない。」
「バックエンド Web サーバーはクライアントから直接アクセスできないために セキュリティ度とメンテナンス性が増す。」
「バックエンド Web サーバーに手を加える必要がない。」
といったメリットがあります。

これらのメリットは、クライアントと既存の Web アプリケーションの間に、 IceWall サーバーが入り、クライアント – Web アプリケーション間の通信を IceWall のフォワーダーが中継することにより実現されているため、 Version 7.0 までの IceWall では、RP 型特有の機能を使用するためには、クライアントは、必ず IceWall サーバー経由の URL でバックエンドサーバーにアクセスしなければいけませんでした。

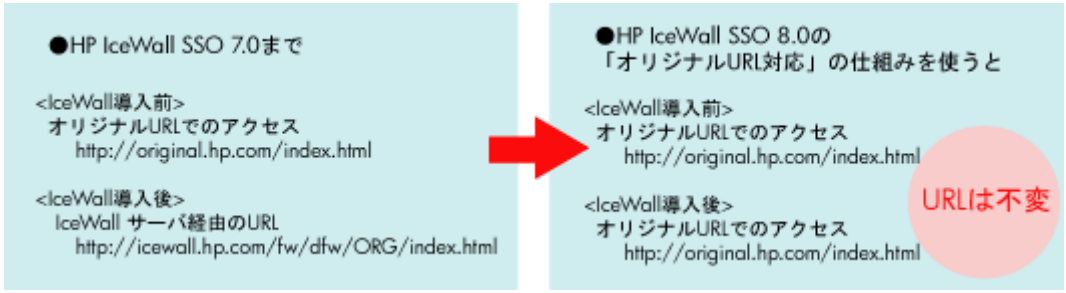
しかし、複雑なトランザクションを使用する独自開発のアプリケーションや、 Java Applet などを多用するアプリケーション・パッケージ製品を使用して構築された環境では、IceWall サーバーのような Reverse Proxy サーバーをクライアントとバックエンド Web サーバーの間に導入する際に、通信の中継を行うことが難しい場合があります。

例えば、IceWall では通常の HTML のようなテキストコンテンツ内部の URL を書き換える機能はありますが、バイナリ化されてしまった URL に対しては URL の書き換えを行う事はできません。このため Java Applet で作成された、独自クライアントを使用して Web アプリケーションにアクセスするタイプのアクセスモデルを使用する環境では、 Web アプリケーションへのアクセス URL が、 Applet プログラム内にハードコーディングされていたりすると、そのコード自体を書き換えない限り、クライアントからのアクセスを IceWall 経由の URL に変更することはできません。

これまで、そういった環境に対しては、Version 7.0 から製品ラインナップに加わった エージェント型モジュールの適用や、使用可能な範囲にのみ限定的に RP 型を使用するといった対応策をご案内してまいりましたが、今回ご紹介する新機能をご使用いただくと、これまで RP 型では対応不可とされて来た環境への新たなソリューションをご提供することが可能になります。

オリジナル URL 対応の仕組み – 基本編 –

従来の RP 型の IceWall を導入する場合、 IceWall 導入前と導入後では以下のように URL を変更する必要がありました。今回ご紹介する、「オリジナルURL 対応」の仕組みを使用すると、IceWall 導入後も、＜ 導入前>と同じオリジナル URL でアクセスを開始することができます。



■「オリジナルURL対応機能」を使用しない場合

上記の＜導入後＞の URL を指定することにより、IceWall サーバー上に配置されたフォワーダーが起動され、エイリアス「 ORG 」で指定されたバックエンド Web サーバーへのリクエストが中継されます。
（このうち、/fw/dfw は IceWall プログラム実体へのパス、ORG は http://original.hp.com をあらわす識別子となります。）

＜オリジナルURL設定前の処理フロー ＞

1. ユーザが http://icewall.hp.com/fw/dfw/ORG/index.html にアクセスする。
2. フォワーダーがバックエンドサーバーの /index.html にアクセスする。
3. バックエンドサーバーがコンテンツを送信する。この時点では href="/index.html" のように記述されている。
4. フォワーダーは バックエンドサーバーから受け取ったコンテンツの ボディ部、ヘッダー部の URL 変換を実行する。
5. このとき、href="/fw/dfw/ORG/index.html" のように変換されることになる。
6. フォワーダーは URL 変換を行ったコンテンツをブラウザに返す。
7. ブラウザは、IceWall 経由の URL でアクセスを続ける。

■「オリジナルURL対応機能」を使用した場合

上記の処理が従来の RP 型のものになりますが、今回ご紹介する、「オリジナル URL 対応」の仕組みを使用すると、冒頭で述べた通り、IceWall 導入後も、＜導入前＞と同じオリジナル URL でアクセスを開始することができます。

＜ 実際の設定方法 ＞

ここからは処理を追いながら、どのような設定が必要となるのか解説していきます。少し細かいですが、お付き合いください。

「オリジナル URL 対応」の仕組みを使用すると、IceWall 導入後も、＜導入前＞と同じオリジナル URL でアクセスを開始することができるため、処理は、

1' ユーザが http://original.hp.com/index.html にアクセスする。

から開始されます。

ここで、「 1' 」で出されたリクエストを、IceWall サーバー上のフォワーダーへ送信するためには、ブラウザに URL が入力された後、このリクエストを出した際に、実際には「 http://icewall.hp.com/fw/dfw/ORG/index.html 」にアクセスさせる必要があります。

まず、リクエスト URL のホスト名部分「 original.hp.com 」にご注目ください。通常、このリクエストは、DNS やローカルの hosts ファイルの設定などにより、名前解決され、クライアントからバックエンド Web サーバーへ送信されますが、DNS やローカルの hosts ファイルの設定を書き換えることにより、リクエストを IceWall サーバーに送信することができます。

DNSへの設定例

```
icewall.hp.com.   IN    A       192.168.0.1
original.hp.com.  IN    CNAME   Icewall.hp.com.
original2.hp.com. IN    CNAME   Icewall.hp.com.
```

DNS に上記のように設定を行えば、「 1' 」のリクエストは、バックエンド Web サーバーではなく、IceWall サーバー上の Apache に送信されます。

ですが、IceWall サーバー上には、指定されたバックエンドサーバーのコンテンツ「 index.html 」は存在しません。このリクエストをバックエンドサーバーへ中継するためには、フォワーダーのパス部分とともにバックエンド Web サーバーのエイリアス「 /fw/dfw/ORG/ 」を「 1' 」の URL へ挿入しなければなりません。

ここで、「 1' 」の URL へ「 /fw/dfw/ORG/ 」部分を挿入するためには Apache HTTPサーバーの組み込みモジュール『mod_rewrite』を利用します。

この mod_rewrite は、RedHat Linux 付属の Apache 等に標準で組み込まれるもので、Apache へのリクエスト URL のパスを、Apache が内部処理を行う前に文字列 変換等をする モジュールとなります 。

例えば・・・
Apache の設定ファイル httpd.conf にて以下のような設定を行うと

httpd.conf 設定例

```
RewriteEngine on

RewriteCond %{HTTP_HOST} ^original¥.hp¥.com
RewriteCond %{REQUEST_URI} !^/fw/dfw.*
RewriteCond %{REQUEST_URI} !^/img/.*
RewriteRule ^(.*) /fw/dfw/ORG/$1 [E=IW_PATH:/fw/dfw/ORG,PT,NS,L]
```

バックエンドWebサーバーが複数ある場合は、ホストごとに設定

条件1： ホストヘッダが「original.hp.com」と一致したら
条件2： リクエストのURLが「/fw/dfwXX」と一致しなかったら
条件3： リクエストのURLが「/img/XX」と一致しなかったら
変換ルール： リクエスト内の「/XX」を「/fw/dfw/ORG/XX」に変換し、環境変数IW_PATHにフォワーダ動作パスとエイリアスを設定

リクエスト URL が条件 1-3 で指定したものと一致すれば、変換ルールにより、リクエスト「 1' 」のパス部分は、「 /fw/dfw/ORG/index.html 」に変換され、環境変数にフォワーダー動作パスとエイリアスを設定した上で、フォワーダーに処理を引き継ぎます。

※ ここでご注意いただきたいのが、「条件 1 」の設定です。この条件をご覧いただくとわかりますが、クライアントから出されたリクエストは、ブラウザが送信したホストヘッダーにより、適切なエイリアスに紐付けられます。（現在、一般的に利用されているブラウザはホストヘッダーに対応していますが、ホストヘッダーに対応していないブラウザでは、このソリューションを複数のバックエンドWebサーバーに使用することはできませんのでご注意ください。）

次に、処理を引き継ぐフォワーダーがApacheで設定された環境変数IW_PATHの情報を取得できるように以下の設定を行っておきます。

dfw.confへの設定例

```
REQUEST_URI=1
VIRTUALPATH_ENV=IW_PATH
```

さて、「 DNS 設定」「 mod_rewrite 」「フォワーダーdfw.conf」の設定により、「 1' 」のリクエストに続き以下のように処理が進みます。

> 2' mod_rewrite が /fw/dfw/ORG/index.html に変換し、環境変数にフォワーダー動作パスとエイリアスを設定する。
> 3' フォワーダーが環境変数より情報を取得し、バックエンドサーバーの /index.html にアクセスする。

これで、「 3' 」のように、従来の RP 型の処理と同様にフォワーダーから、バックエンド Web サーバーへのコンテンツ取得のリクエストが出されます。

その際、完全にブラウザからのアクセスをシミュレーションする必要がある場合は、IceWall の情報継承機能を使用し、クライアントが出すリクエストと同様のヘッダーが送信されるよう、バックエンド Web サーバーごとのホスト設定ファイルの「 HEADER 」および「 HEADER_FILTER 」「 COOKIE_FILTER 」を調整していただく必要があります。

また、ここで必要になるのが、IceWall が発行する認証 Cookie の domain 属性および path 属性の設定です。

今、ご紹介しているケースですと、IceWall にて、何も設定していない状態では、Cookie の仕様により、IceWall が発行する認証 Cookie は、「 domain=original.hp.com; path=/fw 」という属性が設定されたのと同等の状態となります。(path 属性を指定しない場合の Cookie の付加位置に関しては、ブラウザにより若干動作が異なります。)

Cookie の仕様により、domain 属性については後方一致、path 属性については前方一致の場合にのみ、ブラウザにセットされた Cookie はサーバーへ送信されます。この場合、「 1' 」の時点で、IceWall で認証済みであるとしても、認証済みであるということを証明する認証 Cookie は、リクエスト URL 「 http://original.hp.com/XXX 」(XXはfw以外)とともに IceWall サーバーには送信されません。この場合、IceWall は、ユーザが未認証であると判断し、再度、ログイン要求を行います。

そこで、必要になるのが、IceWall の発行する認証 Cookie の属性設定です。IceWall では、フォワーダー設定ファイルの「 COOKIEATTR 」パラメータにて、認証 Cookie の属性を設定することができます。

「1'」のリクエストURLのパス部分が何であっても、認証後のリクエストにて認証Cookieを送信するためには、以下の例のように、path属性「/」を設定する必要があります。

dfw.confへの設定例

```
COOKIEATTR=path=/
```

このように設定すれば、ブラウザは、リクエスト URL 「 https://original.hp.com/XXX 」に対しても Cookie を送付することができます。

また、IceWallにて複数のバックエンドWebサーバーを管理する場合、すべてのサーバーへIceWallの認証Cookieを送信するためには、各サーバーのFQDNの共通部分を認証Cookieのdomain属性として設定する必要があります。例えば、IceWallにて、「original.hp.com」「original2.hp.com」の2つのバックエンドWebサーバーを管理する場合には、以下のように設定します。

dfw.confへの設定例

```
COOKIEATTR=domain=hp.com; path=/
```

※この設定を使用するためには、IceWallの管理対象となる複数のバックエンド Web サーバーは、同一ドメイン「 hp.com 」に所属する必要があります。バックエンドWebサーバーが複数あり、それぞれが別ドメインに所属する場合はこのソリューションは使用できませんのでご注意ください。尚、今回ご紹介した方式を応用してバックエンド Web サーバーと IceWall サーバーが別ドメインに所属する場合に対応したソリューションもありますが、ここでは割愛します。

次に、「 3' 」によるリクエストに対するレスポンスが、バックエンド Web サーバーから返ってきます。

4' バックエンドサーバーがコンテンツを送信する。

通常の RP 型の場合、コンテンツに含まれるリンクなどは、IceWall 経由の URL へ書き換えられますが、オリジナル URL によるアクセスを行う場合、コンテンツに含まれる URL などは、IceWall 経由に書き換えずにブラウザまで返す必要があります。

このため、IceWall のホスト設定ファイルにて以下の様に URL 変換が行われないよう設定し、オリジナルの URL が保たれるようにする必要があります。

ホスト設定ファイル設定例

```
#URLKEY=A,HREF
#URLKEY=BASE,HREF
        (省略)
#URLKEY=INPUT,SRC
#URLKEY=LINK,HREF
```

このように設定することで、コンテンツはオリジナルのままとなり、処理は以下のように進みます。

5' dfw はボディ部、ヘッダー部の URL 変換を実行しない。
6' dfw はオリジナルのままのコンテンツを、ブラウザに送信する。
7' ブラウザは、その後もオリジナルの URL でアクセスを続ける。

さらに、バックエンド Web サーバーからのレスポンスヘッダーについてもオリジナルの URL に保たれるよう、以下の設定が必要になります。

ホスト設定ファイル設定例

UNCONV_HEADER= LOCATION,SET-COOKIE

この設定項目「 UNCONV_HEADER 」は、 version 8.0 より新たに追加されたパラメータです。この設定をバックエンド Web サーバーごとの設定ファイル、ホスト設定ファイルで指定することにより、特定のバックエンド Web サーバーから出された Location ヘッダーおよび Set-Cookie ヘッダーの値は、フォワーダーにて IceWall 経由に変換されません。

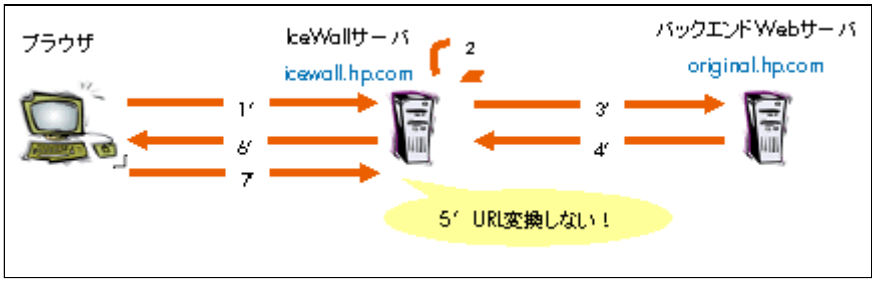
また、ログイン直後のリダイレクト時など、バックエンド Web サーバーではなく IceWall 自身が出す Location ヘッダーを変換するためには、上記の設定に加え、フォワーダーの設定ファイルに以下の設定が必要になります。

dfw.conf 設定例

REPKEY=Location: http://original.hp.com/fw/dfw/ORG/,Location: http://original.hp.com/

以上、これまでご説明した設定を実施することで、オリジナル URL が保証されるようになり、処理は以下のように変更されます。

＜「オリジナルURL」設定後の処理フロー ＞



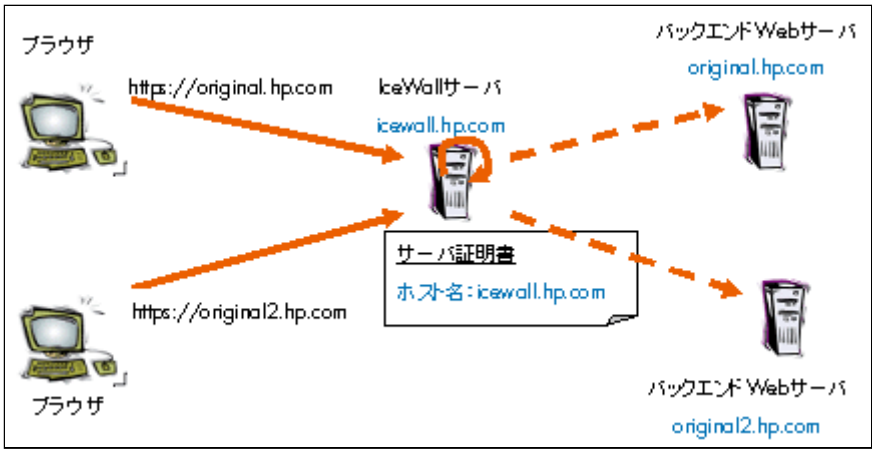
1' ユーザが http://original.hp.com/index.html にアクセスする。この際、DNS 登録上 original.hp.com は IceWall サーバー icewall.hp.com と同じサーバーを指す。
2' mod_rewrite が リクエストパスを /fw/dfw/ORG/index.html に書き換える。
3' フォワーダーがバックエンドサーバーの /index.html にアクセスする。
4' バックエンドサーバーがコンテンツを送信する。
5' dfw はボディ部、ヘッダー部の URL 変換を実行しない。
6' dfw はオリジナルのままのコンテンツを、ブラウザに送信する。
7' ブラウザは、その後もオリジナルの URL でアクセスを続ける。

オリジナル URL 対応の仕組み － 応用編　SSL 通信の場合 －

■クライアント－ IceWall サーバー間でSSL通信を行う場合の問題とは

基本編でご説明した通り、複数台のバックエンド Web サーバーをオリジナル URL 対応で管理する必要がある場合、 IceWall サーバーは、クライアントからは複数のオリジナルの URL によるアクセスを受け付けることになります。

クライアント- IceWall サーバー間で SSL 通信を行う際、クライアントへは、 SSL 通信を行うサーバーの IP アドレス（ホスト名）とサービスを提供するポートの情報が記された「サーバー証明書」が送信されます。



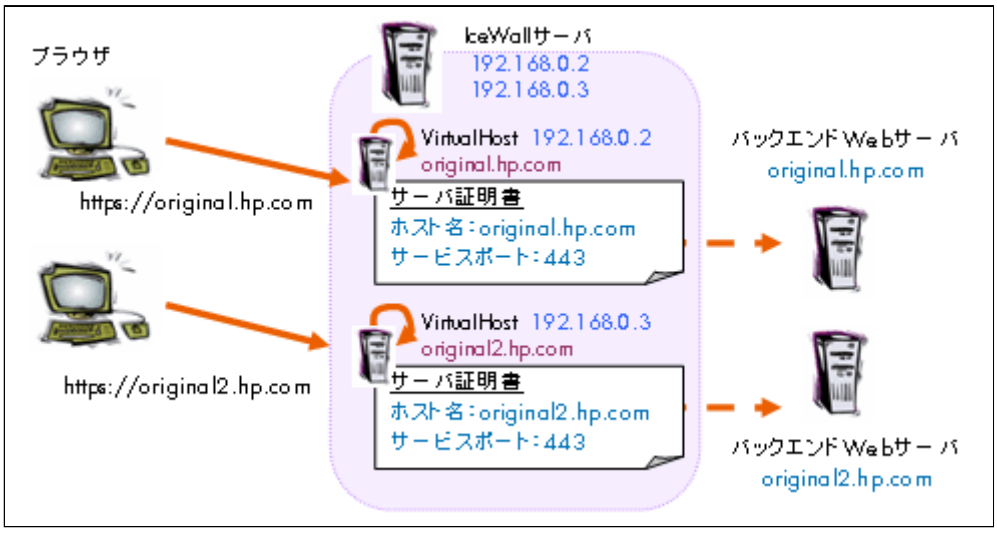
図のように、ブラウザから、それぞれ「 original1.hp.com 」「 original2.hp.com 」へアクセスする場合、それぞれのリクエストURL は基本編でご紹介した DNS の設定により、IceWall サーバーへ送信されます。しかし、サーバーから送られてきた証明書に「 icewall.hp.com 」のサインがあり、ブラウザでは、リクエストした URL とサーバー証明書のホスト名が一致しないため、警告画面が表示されてしまいます。

応用編では、この問題を解決するための設定方法のひとつをご紹介します。

■Apache（Virtual Host）機能による解決方法

この問題を解決するためには、それぞれのオリジナル URL に一致した署名がされたサーバー証明書をクライアントに返す必要があります。それを実現するのが、Apache の『 VirtualHost 』機能です。

VirtualHost とは、一つの HTTP サーバー上で複数の Web サイトを扱うための機能です。この機能を使用し、IceWall サーバー上でそれぞれのオリジナル URL に応じた複数の Web サイトを運営し、それぞれに SSL の設定等を行えば、サーバー証明書の署名の不一致による問題は解決されます。



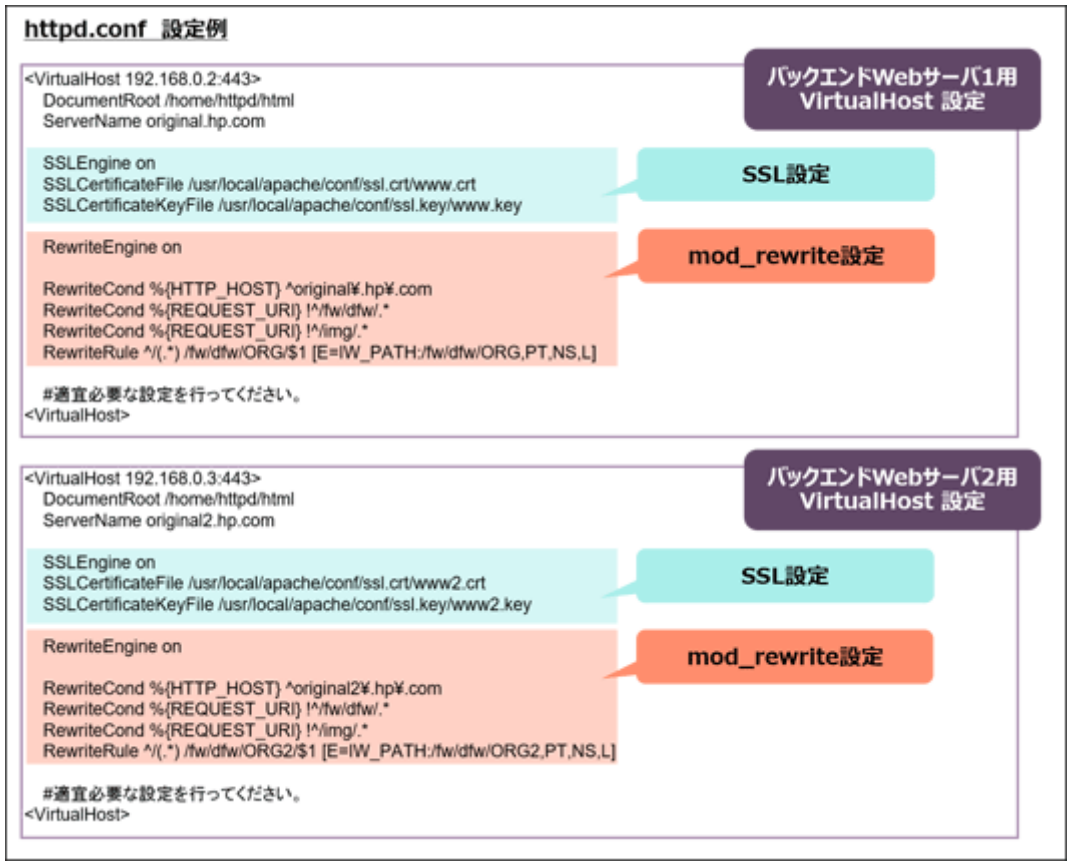
Apache の Virtual Host 機能には「名前ベース」と「 IP ベース」の 2 種類がありますが、ひとつの IP アドレスで複数のサイトを運営する名前ベースでは、サーバーはひとつの IP アドレスしか持たないため、複数の証明書を使用することはできません。

したがって、今回ご紹介する様な例では、IP アドレスベースの Virtual Host 機能を使用する必要があります。IP アドレスベースの場合、IceWall サーバーは、バックエンドサーバーと同じ数の IP アドレスを持つ必要があり、それぞれの IP アドレスは DNS に、以下のように登録されます。

DNSへの設定例

```
icewall.hp.com.   IN   A   192.168.0.1
original.hp.com.  IN   A   192.168.0.2
original2.hp.com. IN   A   192.168.0.3
```

そして、それぞれの VirtualHost ごとにサーバー証明書とキーペアを準備し、以下の例のように、各 VirtualHost ディレクティブの中に適切な設定を行います。



以上が SSL 通信を使用する場合の設定方法の一例になります。

今回ご紹介した設定例を参考にしていただければ、オリジナル URL によるアクセスのままで、RP 型のメリットを生かした IceWall 環境を構築することができます。

2005.1.26　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 土居 恭子
2013.9.19　「オリジナル URL 対応の仕組み － 基本編 － 」内の図を一部修正
2018.2.12　内容を一部更新