VPN OneTouch Integration

Generic One Touch / VPN Integration

This article explains the generic integration of a VPN with Swivel Push (formerly One Touch) authentication.

It focusses on the modification of the VPN Log-in page and integration with the One Touch login page as the other elements are the same (eg RADIUS config) for other forms of authentication.

NOTE: in order to use One Touch with RADIUS, you must have Swivel software version 3.11.5 or later.

One Touch Page

When a user authenticates to a VPN via One Touch, the user is redirected to a One-Touch login page to initiate the OneTouch and then returned to the VPN Login Page once the One Touch has been completed.

The One Touch login page is available from here. This should normally be unzipped and deployed in the wepapps2 folder of a Swivel Appliance. If you have a version 3 appliance, you may need assistance from Swivel Secure to copy the file to this folder.

There is a settings file that may need to be edited. This file will be under webapps\onetouch\WEB-INF\classes\settings.properties

```
pinsafessl=false
pinsafeserver=localhost
pinsafecontext=pinsafe
pinsafesecret=secret
pinsafeport=8181
imagessl=true
imageserver=demo.swivelcloud.com
imagecontext=proxy
imageport=8443
selfsigned=true
```

Most settings will not need to be edited however you will need to change the secret to match that set on the Swivel Core Server and the image server needs to reflect the public hostname of the appliance.

You need to restart tomcat for new settings to work.

Modifying VPN Login Page for One Touch

When a user authenticates to a VPN via One Touch, the user is redirected to a One-Touch login page to initiate the OneTouch and then returned to the VPN Login Page once the One Touch has been completed.

The VPN Login page must therefore perform the redirect and manage the return.

This is done my adding some javascript to the header of the login page as follows.

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>
```

This allows the use of JQUERY without needing to host on the VPN.

```
<Script>
function redirect(){
window.location.replace("https://demo.swivelcloud.com:8443/onetouchdemo/onetouch?returnurl=" + encodeURIComponent(window.location.href ) );
}
```

This is the function that redirects to the One Touch Login page, so the url needs to be changed to match the url for the specific installation. It passes its own url as a parameter to the One Touch page to the One Touch page can redirect back.

```
var QueryString = function () {
  var query_string = {};
  var query = window.location.search.substring(1);
  var vars = query.split("&");
  for (var i=0;i<vars.length;i++) {
    var pair = vars[i].split("=");
       // If first entry with this name
    if (typeof query_string[pair[0]] === "undefined") {
      query_string[pair[0]] = pair[1];
       // If second entry with this name
    } else if (typeof query_string[pair[0]] === "string") {
     var arr = [ query_string[pair[0]], pair[1] ];
     query_string[pair[0]] = arr;
} else {
    query_string[pair[0]].push(pair[1]);
  }
}
  return query_string;
} ();
```

Just java script function for reading parameters passed in.

```
$(document).ready(function(){
       usernamePassedIn = QueryString["username"];
      passwordPassedIn = QueryString["password"];
      sessionIdPassedIn = QueryString["sessionId"];
      if(typeof sessionIdPassedIn == 'undefined') {
          redirect();
      } else {
          $('[name=password]').val(passwordPassedIn);
          $('[name=password2]').val(sessionId);
          $('[name=login]').val(usernamePassedIn);
          document.getElementsByName("vpnForm")[0].submit();
```

```
        }
    });
</Script>
```

This function runs when the page has finished loading. It looks for values passed in. If those values are not present, it redirects the user to the One Touch Login Page. If the user has already instigated the One Touch authentication the One Touch login page will redirect back with their username, password and sessionId passed in as parameters. This function will use these parameters to populate the login form and then submit that login form.

This section of the javascript will need to be edited to match the specifics of the VPN login form and how it is to be used. For example the names of the HTML elements need to match, ie the name of the login form and the names of the input fields. Also you may only be using one password field which may only require the sessionId or require the password and sessionId concatenated.