How to Set Up a Custom Challenge Page for Authentication

https://campus.barracuda.com/doc/11143237/

Setting up a custom challenge page is a three-step process, explained below. For the following example, we assume the following:

- A backend server at 192.168.128.10 needs access restricted to authenticated users. Web application resources residing at "http://192.168.128.10/secure" require users to authenticate before gaining access.
- User authentication uses an SMS Passcode server.
- **Service-1** (10.10.10.2:80) is configured on the Barracuda Web Application Firewall to secure the access to the web application, with 192.168.128.10:80 configured as the server.

Step 1 - Create a Custom Challenge Page

To create a custom challenge page, you should use script languages like PHP, JavaScript, or CGI Perl to read the parameters from the incoming request URL, and use those values in the HTML file. In the examples below we are using PHP and ASP.NET scripts to create the custom challenge page "challenge.php"/"challenge.aspx". It must contain the following:

- Form ID = nclogin
- Name = nclogin
- Action = "/nclogin.submit"
- Method = POST
- Username field should be named - **f_username**
- Password field should be named - **f_passwd**
- An additional hidden parameter named **f_method** should be specified with value "LOGIN"
- Form Fields should include "Challenge User Field" and "Challenge Prompt Field". Note that these field names needs to be configured on the **ACCESS CONTROL > Authentication** page by editing a service. For more information, see Configuring the Barracuda Web Application Firewall to use the custom challenge page .

```
<META HTTP-EQUIV="CACHE-CONTROL" CONTENT="NO-CACHE">
<html>
<head>
<title>Login</title>
</head>
<body bgcolor="#00C0C0">
<font color="black">
<hr width="100%">
<center>Authentication and Access control</center>
<hr width="100%">
```

```
<center><b>Login</b></center>
<form action="/nclogin.submit" method="POST">
<table><tbody>
<tr>
<td align="right"><b><?php echo
$_GET["challenge_prompt"];?></b></td><td><input type="password" size="32"
name="f_passwd"></td>
</tr>
</tbody></table>
<p>
<input type="hidden" name="f_username" value=<?php echo
"\"".$_GET["challenge_user"]."\"";?>>
<input type="hidden" name="f_method" value="LOGIN">
<input type="submit" value="Login">
<input type="reset" value="Reset">
</p>
</form>
</font>
</body>
</html>
```

```
<%@ Page Language="C#" %>
<META HTTP-EQUIV="CACHE-CONTROL" CONTENT="NO-CACHE">
<html>
<head>
<title>Login</title>
</head>
<body bgcolor="#00C0C0">
<font color="black">
<hr width="100%">
<center>Authentication and Access control</center>
<hr width="100%">
<center><b>Login</b></center>
<form action="/nclogin.submit" method="POST">
<table><tbody>
<tr>
<td align="right"><b><%= Request.QueryString["challenge_prompt"]
%></b></td><td><input type="password" size="32" name="f_passwd"></td>
</tr>
</tbody></table>
<p>
<input type="hidden" name="f_username" value=<%=
Request.QueryString["challenge_user"] %>>
<input type="hidden" name="f_method" value="LOGIN">
<input type="submit" value="Login">
```

```
<input type="reset" value="Reset">
</p>
</form>
</font>
</body>
</html>
```

Step 2 - Deploy the Created Custom Page on Your Web Server

To use the "challenge.php" file, deploy it on your web server. For example:

- The IP address of the web server is 192.168.128.10
- And the “challenge.php” is available by accessing "http://192.168.128.10/challenge.php"

Step 3 - Configure the Barracuda Web Application Firewall to Use the Custom Challenge Page

Once the custom challenge page is deployed on your web server, configure the Barracuda Web Application Firewall to use the custom challenge page by doing the following:

1. Configure Authentication Service – Specify the authentication database server (RADIUS) that will be used to authenticate the user's credential. Ensure the IP address of the server is pointing to the SMS Passcode server. See How to Configure Authentication and Access Control (AAA).
2. Configure Authentication Policy – Create an authentication policy for the service you want to secure. To do this, click **Edit** next to the relevant Service (**Service-1** in this example) on the **ACCESS CONTROL > Authentication** page. In the **Edit Authentication Policy** window, configure the following:
    1. Set **Status** to *On*.
    2. Select the **Authentication Service** from the drop-down list. Note that this is the authentication service created in Step 3.1.
    3. Specify values for the following fields:
        - **Auth Challenge URL** – Enter the URL where a user is redirected if the authentication service requires additional credentials such as Passcode or PIN. In this example, it is “/challenge.php”.

          The **Auth Challenge URL** should be added to the allow list in Global ACLs on the **SECURITY POLICIES > Global ACLs** page. See Steps to Configure a Global ACL Rule.
        - **Challenge User Field** – Enter the name of the query string field passed to the challenge URL that contains the challenged user's username. By default, the value is set to *challenge_user*.

- **Challenge Prompt Field** – Enter the name of the query string field passed to the challenge URL that contains the prompt string received in a RADIUS challenge message. By default, the value is set to *challenge_prompt*.

4. Specify appropriate values for other parameters and click **Add**. For more information, click **Help** in the web interface.

3. **Configure Authorization Policy** – Create an authorization policy indicating which resources become accessible after a successful authentication on the **ACCESS CONTROL > Authorization** page using the following steps:
   1. On the **ACCESS CONTROL > Authorization** page, in the **Add Authorization Policy** section, specify values for the following:
      - **Service** – Select the relevant service (*Service-1* in the example) from the drop-down list.
      - **Policy Name** – Enter a name for the authorization policy. Example: *secure.access*
      - **URL Match** – Enter the URL of the secured part of the web application. In this example the URL is: */secure/*
      - **Login Method** – Select *HTML Form*.

        The **Auth Challenge URL** does not support *HTTP Basic Authentication*.
   2. Click **Add**.

Custom challenge URL is supported *only* for two-factor authentication (SMS Passcode and RSA SecurID).

Steps to Configure a Global ACL Rule

1. Go to the **SECURITY POLICIES > Global ACLs** page.
2. Select the policy associated with the service from the **Policy Name** drop-down list.
3. In the **Create Global ACL** section, specify values for the following fields:
   - **URL ACL Name** – Enter a name for the URL ACL.
   - **URL Match** – Enter the auth challenge URL, as per the above example it is */challenge.php*.
   - **Action** – Set to *Allow*.
4. Specify values for other parameters as required and click **Add**.