# Integrating Swivel Secure's Pinsafe into Fortinet SSL VPN login

## Overview

PinSafe from Swive Secure is an enhance authentication system that utilizes both single and dual factor authentication. The essence of Pinsafe is an ever changing one time password generated from an end user known PIN and a randomly generated string. The one time code is calculated by entering the letters or numbers of the random string according to the position they occur relative to the users PIN code. For example if the user's PIN code is 2468 and the random string is 0987654321 then the one time code will be the 2nd, 4th, 6th, then 8th character from the random string (9753 in this example).

The Pinsafe system can be integrated into the Fortigate login screens to display the random string as a "Turing" string. Whilst not achieving any dual factor authentication in this manner, the system helps to alleviate the problem of keyloggers capturing passwords as they are unique every time.

The Pinsafe system has a dual channel option. In this mode the random string is sent to the user by a different channel. These channels include a request from a different web page, SMS message, email and others. The system can be setup to send a new random string after every login attempt, either successful or not, or the user may request a random string on demand which will have a validity period of 2 minutes.

The Swivelsecure Pinsafe server acts as a radius server to the Fortigate and will provide authentication and accounting.

This document discusses the integration requirements, it does not go into detail regarding how to setup either Pinsafe, or The Fortigate SSL VPN as this information is available in other documents.

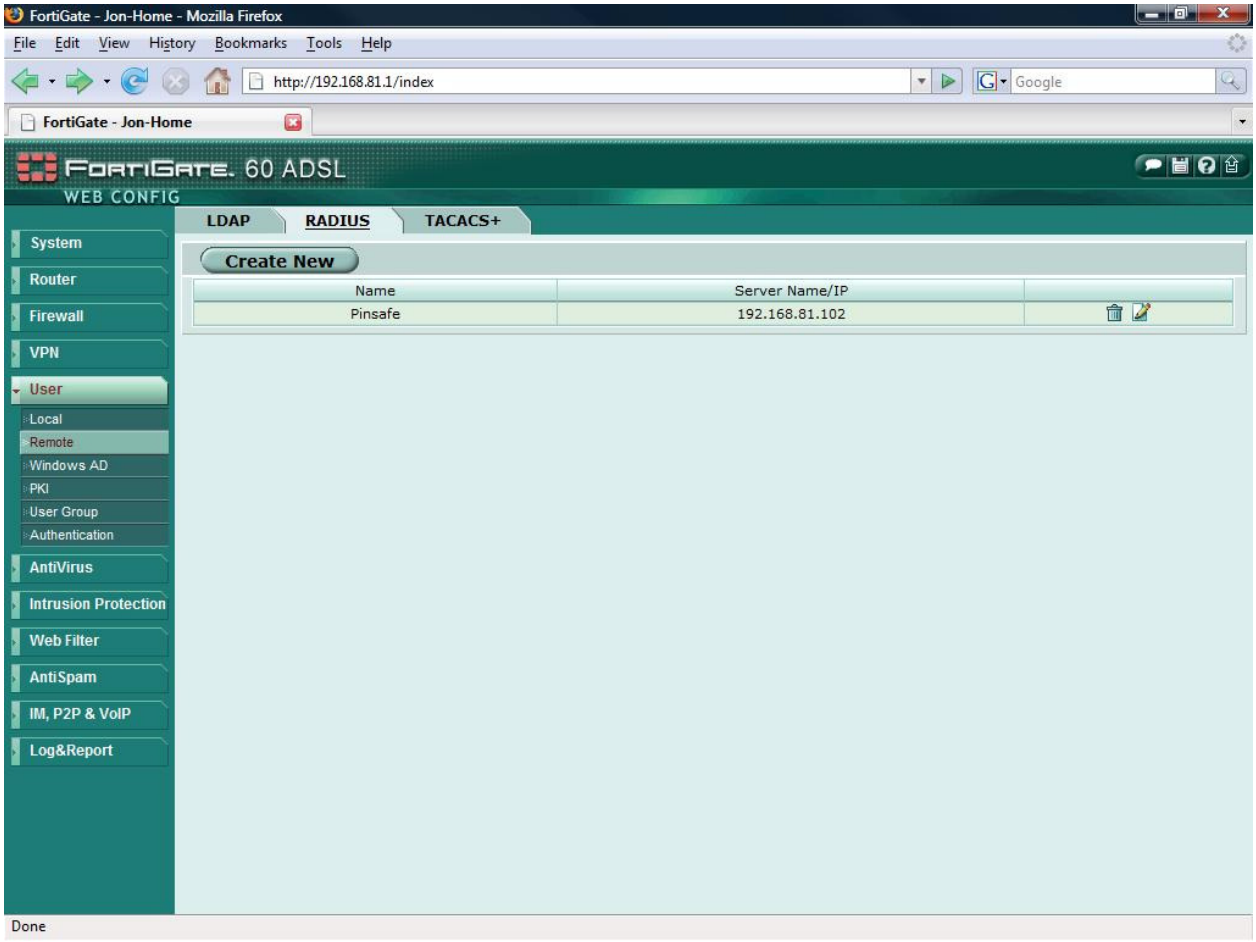
Pinsafe admin guide:

http://www.swivelsecure.com/UserFiles/File/documentmanuals/Administration%20Document.zip

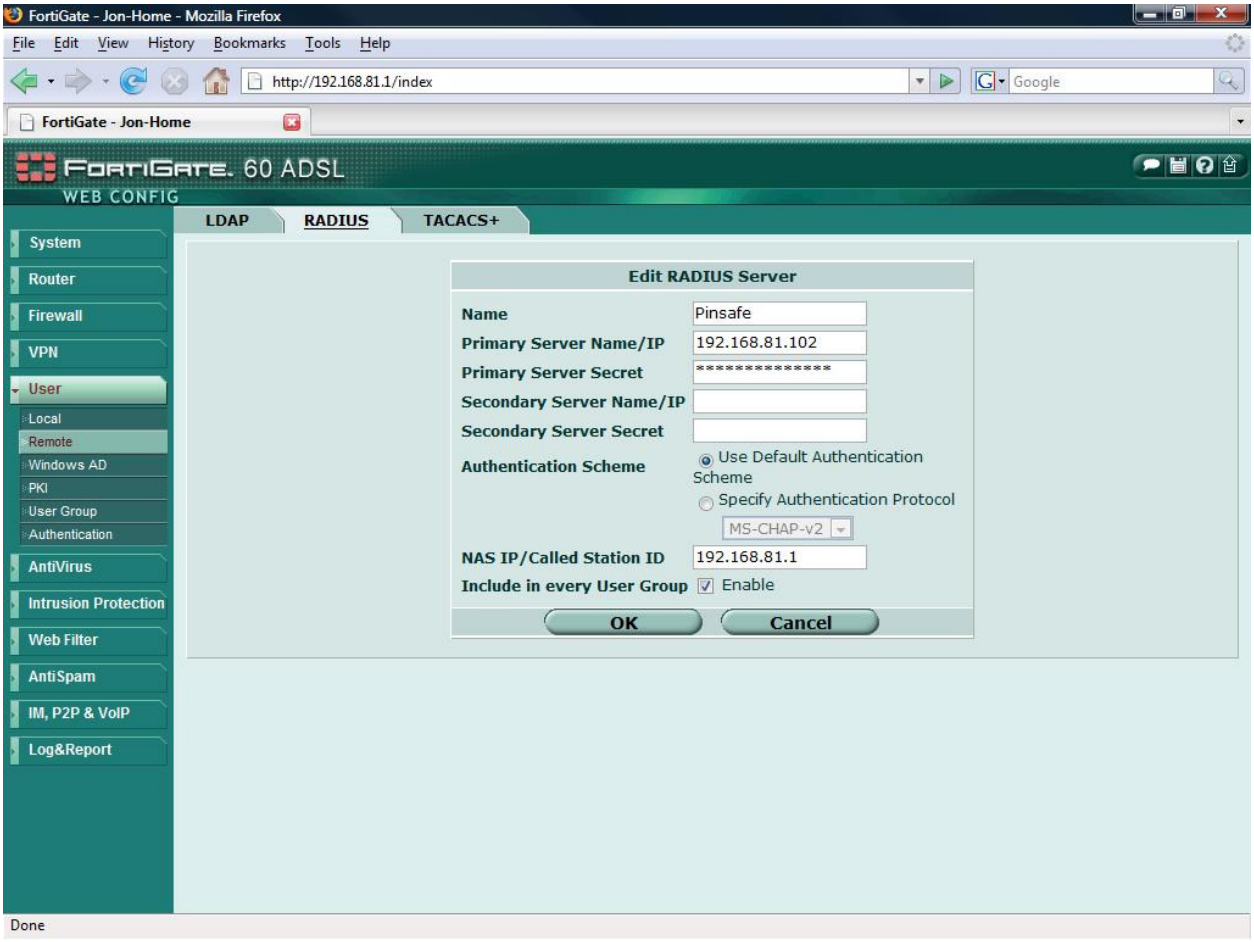
Fortinet SSLVPN guide

http://docs.forticare.com/fgt/techdocs/FortiGate_SSL_VPN_User_Guide_01-30005-0348-20070911.pdf

## Setting up the Pinsafe as a radius server.




Select User=>Remote in the left hand navigation pane, the select the Radius tab. Press create new to bring up the new Radius server option.




Enter a name for the Radius server

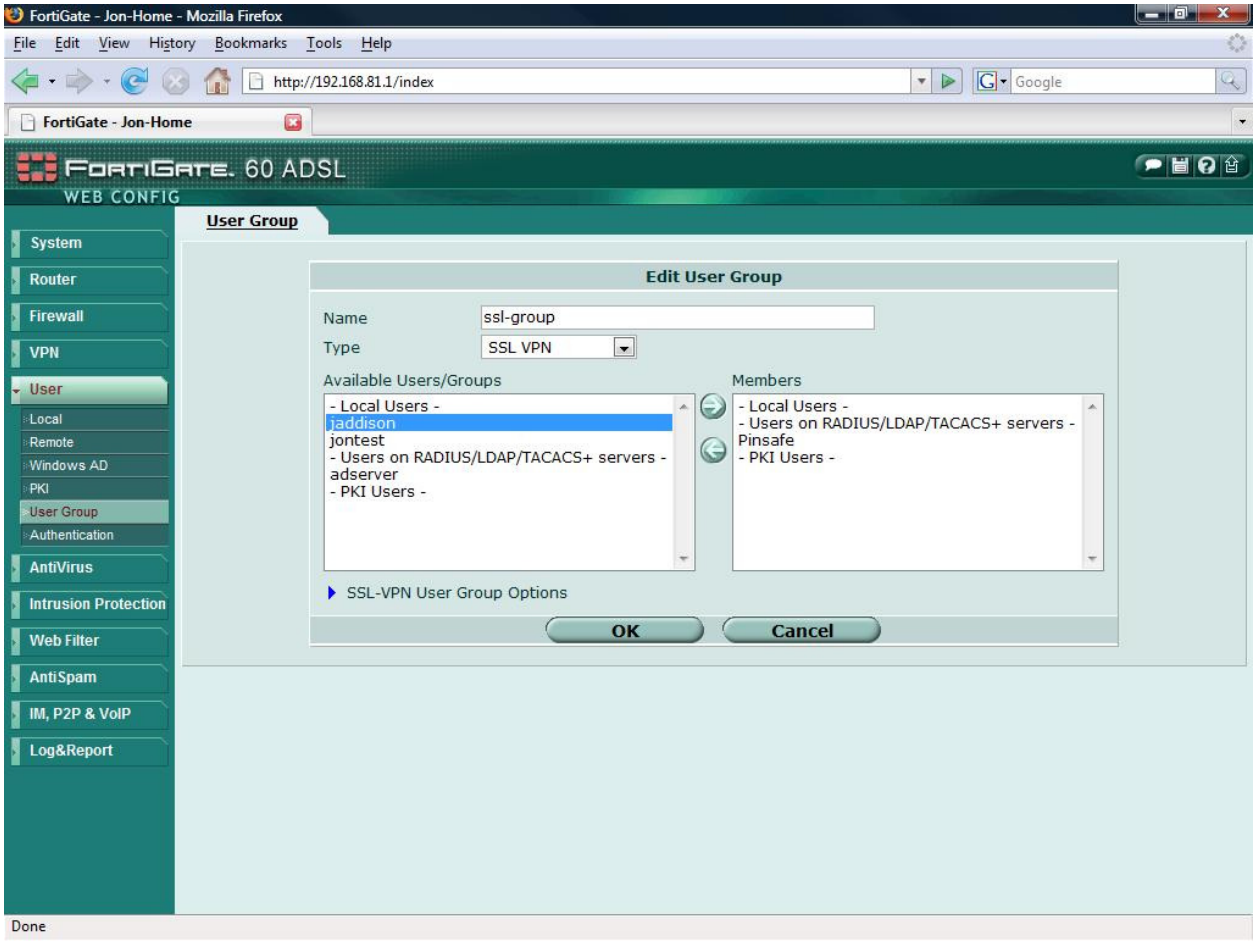
Enter the Radius Server primary IP address

Enter the shared secret chosen between the Fortigate and the Pinsafe server

Enter the IP address of the Interface that will be used to send information to the PINsafe Server

Check the Include in every User Group check box

## Adding the Radius server to the SSL VPN Group




Under User=>User Group slect an existing, or create a new SSL VPN User group

Add the newly created Radius server to the Member list by selecting it from the left hand panel and pressingthe right facing arrow.

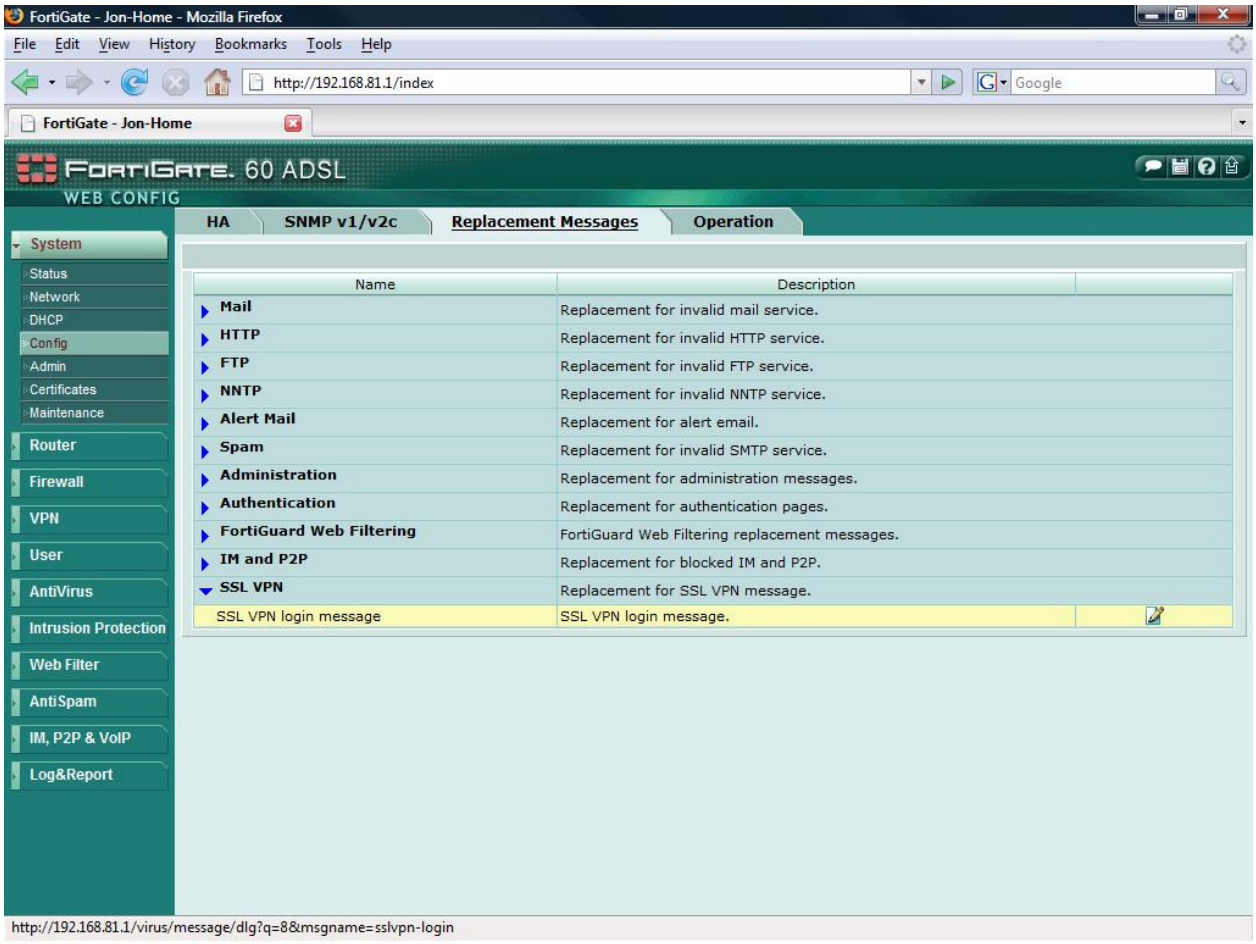
## Integrating Pinsafe into login screens

If you are using a standard dual channel authentication then no further action is required on the Fortigate. The Pinsafe server will send the random strings and pin information to the user on first setup of the account and after every subsequent login attempt.

The following section discusses how to modify the Fortigate authentication screens to integrate with PInsafe Turing numbers and on demand features. For this example we discuss only the SSL VPN login screen, but the theory can be applied to any of the Fortigate's Web Authentication methods.

## Modifying the SSL login screen to integrate with the Pinsafe Server.

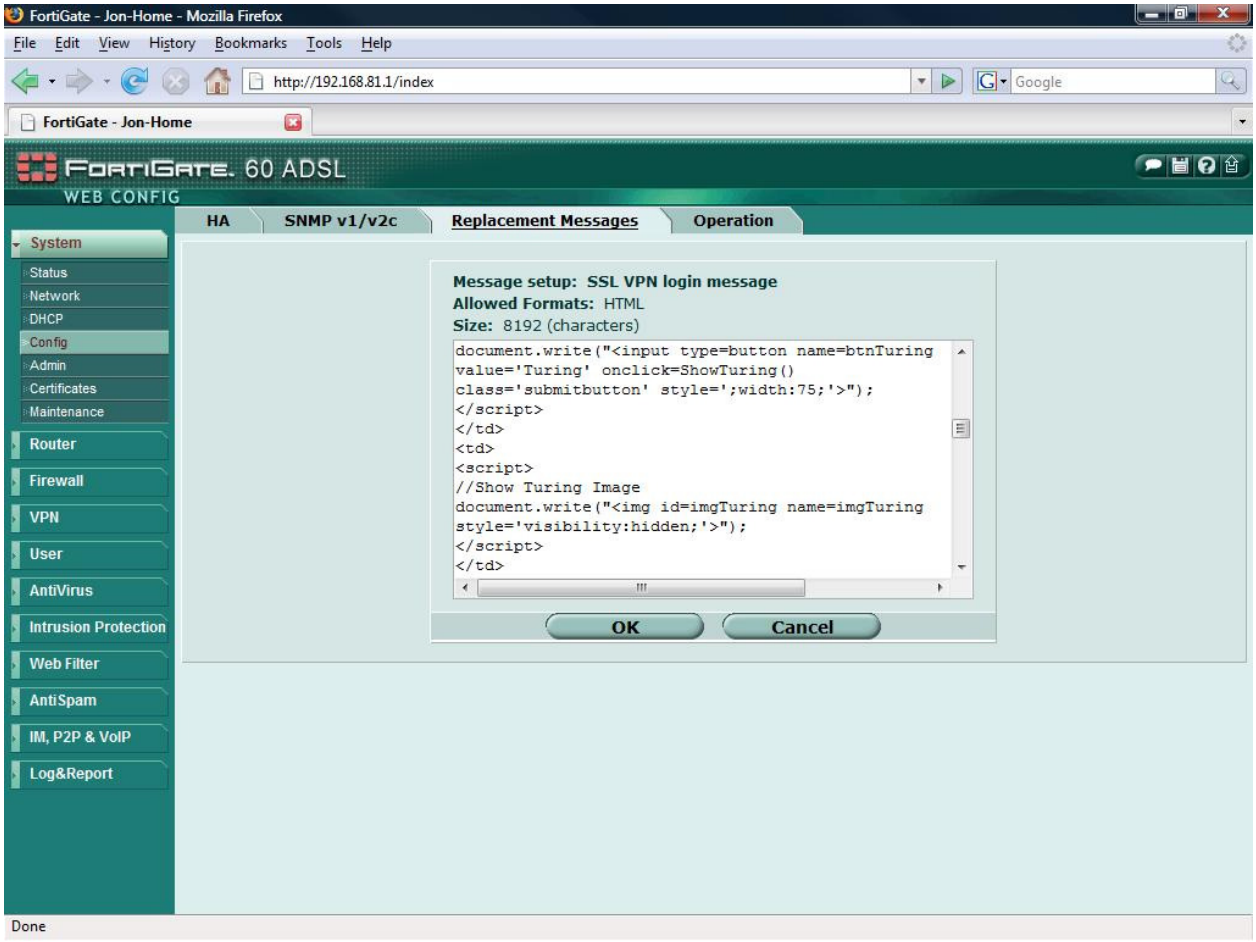
In this scenario we will add some simple client side java script to the default SSL login page, to allow the user to directly request information from the Pinsafe server.




From System=>Config in the left hand navigation pane select the replacement messages tab.

Open the SSL VPN section by pressing on the blue arrow to the left of it.

Press the edit icon to the right of the SSL VPN login message



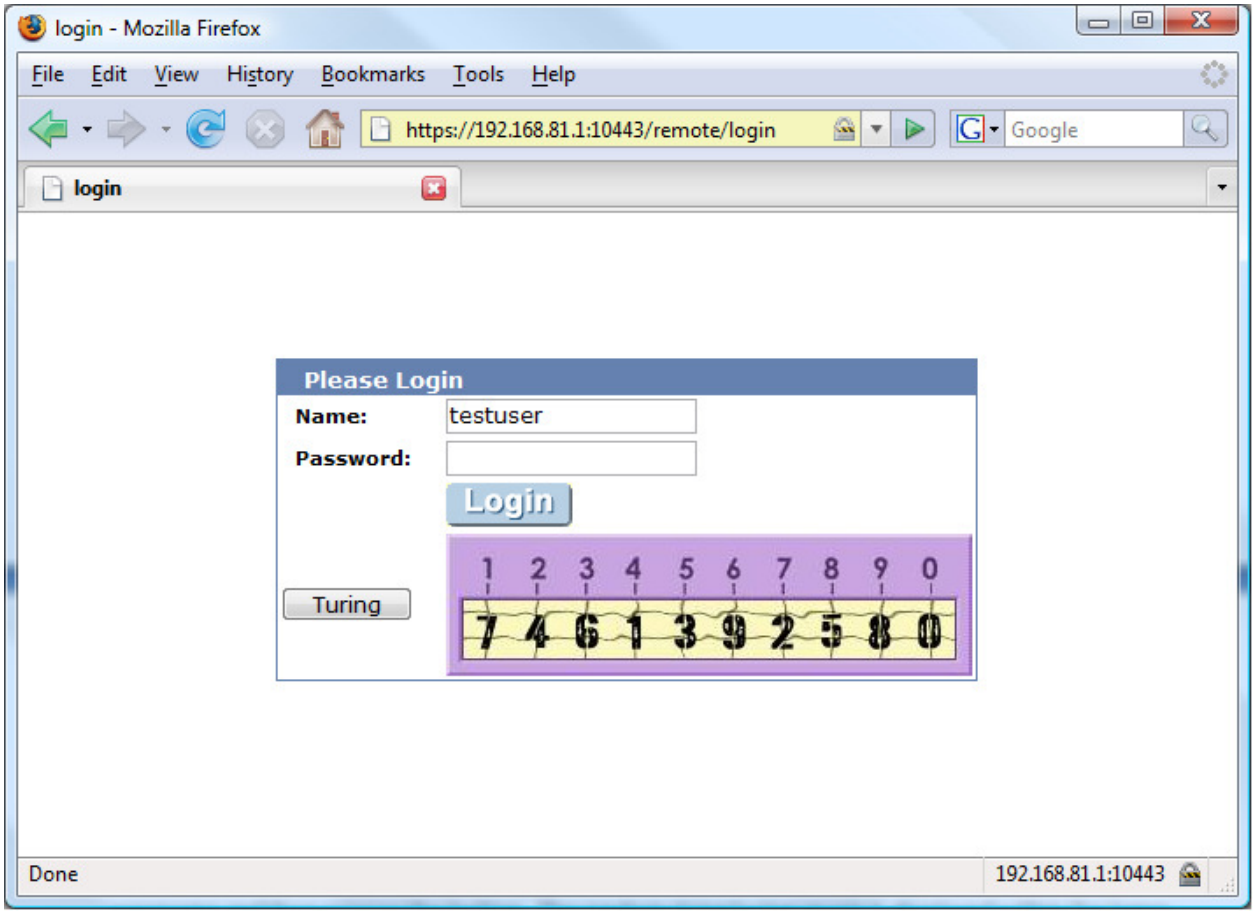

You can directly add to the default page, or simply copy and paste from a HTML/text editor a complete new login page

The example above shows a modified login page already applied.

## Example SSL VPN login pages.

### Display Turing request button and Turing image

In this page a script is included that will display a button called “Turing”. When a user enters his username and then presses the button. The random string Turing image is displayed within the logon box.




With the image displayed the user then enters the one time code according to his PIN number and logs in as usual. The login request is sent to the Pinsafe Radius server for authentication. NB. Pinsafe also allows for the one time code to be appended to a static password for increased security.

## Turing Display Script

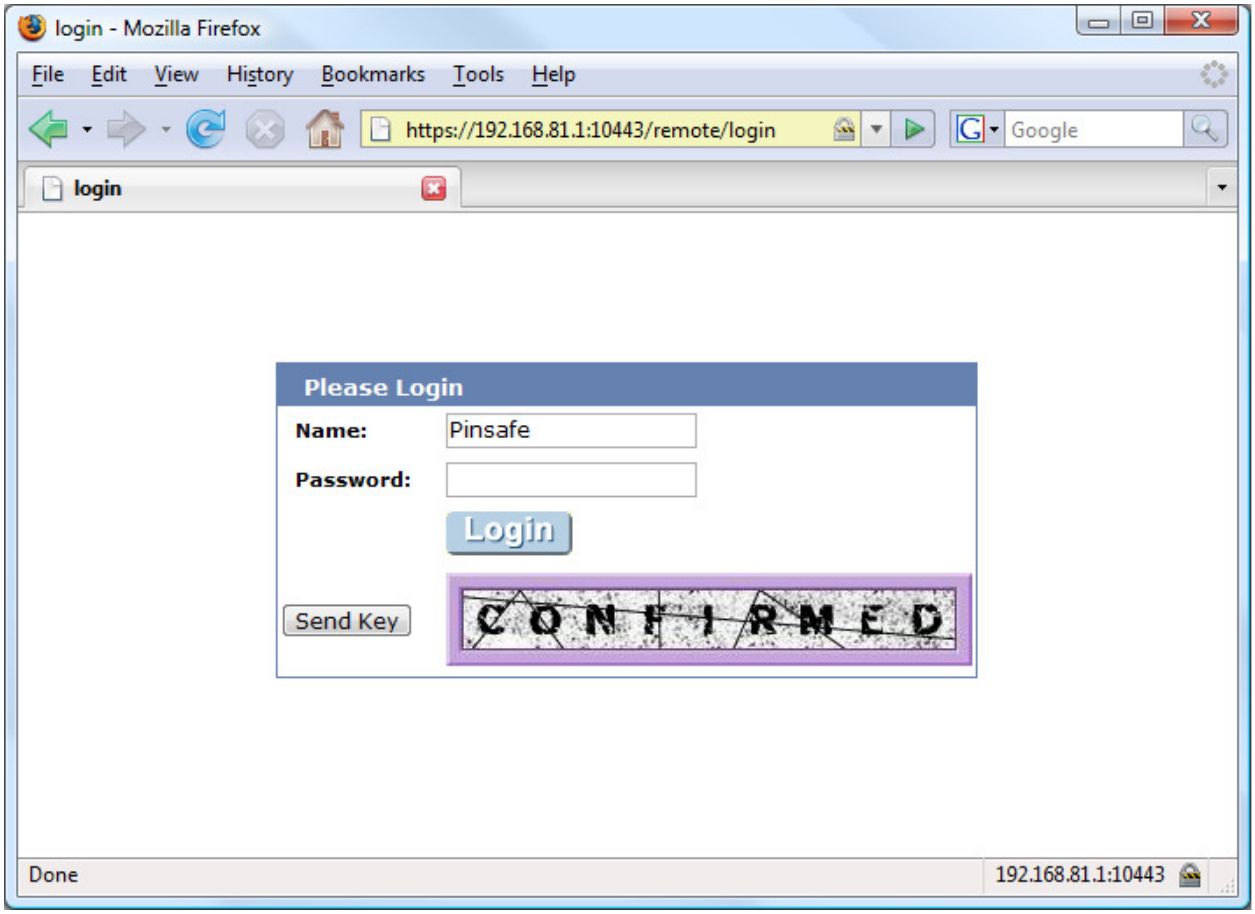
The bold sections indicate additions to the default page

```
<html><head><title>login</title>
<meta http-equiv="Pragma" content="no-cache">
<meta http-equiv="cache-control" content="no-cache">
<meta http-equiv="cache-control" content="must-revalidate">
<link href="/ssl_style.css" rel="stylesheet" type="text/css">
<script language="JavaScript"><!--if (top && top.location != window.location) top.location =
top.location;if (window.opener && window.opener.top) { window.opener.top.location =
window.opener.top.location; self.close(); }//--></script>
</head>
<body class="main">
<center><table width="100%" height="100%" align="center" class="container" valign="middle"
cellpadding="0" cellspacing="0">
<tr valign=middle>
<td>
<form action="%%SSL_ACT%%" method="%%SSL_METHOD%%" name="f">
<table class="list" cellpadding=10 cellspacing=0 align=center width=400 height=180>%%SSL_LOGIN%%
<td>
<script>
//Print Turing Buttom
document.write("<input type=button name=btnTuring value='Turing' onclick=ShowTuring()
class='submitbutton' style=';width:75;'>");
</script>
</td>
<td>
<script>
//Show Turing Image
document.write("<img id=imgTuring name=imgTuring style='visibility:hidden;'>");
</script>
</td>
</table>
%%SSL_HIDDEN%%
</td>
</tr>
</table>
</form>
</center>
</body>
<script>document.forms[0].username.focus();
</script>
```

```
<script>
{
//~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
//
//Configuration section.....

//URL of radiusTuring page on the PINsafe server....
var sUrl="http://pinsafe.server.com:8080/pinsafe/SCImage?username=";

//Names of the username and password texboxes in the page that's calling this script...
//(On Fortinet these are username and credential)
var sNameOfUsernameText = "username";
var sNameOfPasswordText = "credential";

//End configuration section.....
//
//~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

function ShowTuring() {
sUser=document.getElementsByName(sNameOfUsernameText)[0].value;
    if (sUser=="") {
        alert ("Please enter your username first!");
document.getElementsByName(sNameOfUsernameText)[0].focus()
    }else{
        //Find the image using Mozilla compatible pproach...
        varImg = document.getElementById("imgTuring");

        //Set the image SRC and make it visible
        varImg.src = sUrl + sUser;
        varImg.style.visibility = "visible";

        //Set focus to the OTC input
        document.getElementsByName(sNameOfPasswordText)[0].focus()
    }
}
}
</script>
</html>
```

### On Demand Request for one time pad script

The on demand script causes the PInsafe server to simply send the one time pad to the user via the chosen transport (SMTP, SMS etc.)  The script is almost identical, with just the button name and URL requested changed. (change s are highlighted in red in the following script)




In this configuration, the server simply displays confirmed to tell the user that the one time pad has been sent via his/her preferred method. The user then has two minutes to login before this one time pad expires.

NB: If this “on demand” mode is chosen, then the automatic sending of a one time pad after failed or successful login is disabled

## On demand script

```
<html><head><title>login</title>
<meta http-equiv="Pragma" content="no-cache">
<meta http-equiv="cache-control" content="no-cache">
<meta http-equiv="cache-control" content="must-revalidate">
<link href="/ssl_style.css" rel="stylesheet" type="text/css">
<script language="JavaScript"><!--if (top && top.location != window.location) top.location =
top.location;if (window.opener && window.opener.top) { window.opener.top.location =
window.opener.top.location; self.close(); }//--></script>
</head>
<body class="main">
<center><table width="100%" height="100%" align="center" class="container" valign="middle"
cellpadding="0" cellspacing="0">
<tr valign=middle>
<td>
<form action="%%SSL_ACT%%" method="%%SSL_METHOD%%" name="f">
<table class="list" cellpadding=10 cellspacing=0 align=center width=400 height=180>%%SSL_LOGIN%%
<td>
<script>
//Print Turing Buttom
document.write("<input type=button name=btnTuring value='Send Key' onclick=ShowTuring()
class='submitbutton' style=';width:75;'>");
</script>
</td>
<td>
<script>
//Show Turing Image
document.write("<img id=imgTuring name=imgTuring style='visibility:hidden;'>");
</script>
</td>
</table>
%%SSL_HIDDEN%%
</td>
</tr>
</table>
</form>
</center>
</body>
<script>document.forms[0].username.focus();
</script>
<script>
{
```

```
//~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
//
//Configuration section.....

//URL of confirmation page on the PINsafe server....
var sUrl="http://pinsafe.bojondas.com:8080/pinsafe/DCMessage?username=";

//Names of the username and password texboxes in the page that's calling this script...
//(On Fortinet these are username and credential; on Netscreen they are username and password)
var sNameOfUsernameText = "username";
var sNameOfPasswordText = "credential";

//End configuration section.....
//
//~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

function ShowTuring() {
sUser=document.getElementsByName(sNameOfUsernameText)[0].value;
    if (sUser=="") {
        alert ("Please enter your username first!");
document.getElementsByName(sNameOfUsernameText)[0].focus()
    }else{
        //Find the image using Mozilla compatible pproach...
        varImg = document.getElementById("imgTuring");

        //Set the image SRC and make it visible
        varImg.src = sUrl + sUser;
        varImg.style.visibility = "visible";

        //Set focus to the OTC input
        document.getElementsByName(sNameOfPasswordText)[0].focus()
    }
}
}
</script>
</html>
```