Microsoft IAG Multiple Authentication

Contents

PINsafe and IAG/UAG Integration using multiple repositories

This article explains how to use PINsafe with Microsoft IAG/UAG so that different applications are available to users depending on how they authenticated.

These notes are based on IAG Version 3.7 and PINsafe Version 3.6

This article shows the approach required to add this functionality to a standard IAG/UAG and PINsafe integration. Standard integration notes are available from the Microsoft IAG Integration guide and should also be referred to.

Approach

The approach is to create two different repositories on the IAG. One repository will use Agent-XML for authentication the other will use RADIUS.

One repository will be associated with single channel authentication, the other with dual channel authentication.

The login page will determine which repository the user is authenticating based on whether the user has requested a single channel (TURing) image or not.

The IAG will be configured to allow access to specific applications based on the repository a user has authenticated to.\

On the PINsafe server the NAS or Agent associated with the IAG Dual channel repository will be set to accept dual channel authentication only.

Implementation

The names used for repositories etc are just examples, but sometimes names are important, eg the repository of type "other" needs to have the same name as the associated .inc file and needs to reflected in the checkradio() function in PinsafeLogin.asp

PINsafe Configuration

In this example radius will be used for dual channel authentications only so on the PINsafe server

Enable RADIUS server

Create a NAS entry for the IAG

Set ip address and shared secret as required

Set mode to dual channel only for the NAS

Create an Agent entry for the IAG

Set ip address and shared secret as required

IAG Repository Configuration

Copy images.asp to von\IntnernalSite\Images\CustomUpdate

Ensure that it is the version that can also handle index images and ensure that the IP addresses etc match the PINsafe server

```
if request.querystring("index") <> "" then
  Set objWinHttp = Server.CreateObject("WinHttp.WinHttpRequest.5")
  objWinHttp.Open "GET", "http://127.0.0.1:8080/pinsafe/DCIndexImage?username="&request.querystring("username"), false
else
  Set objWinHttp = Server.CreateObject("WinHttp.WinHttpRequest.5")
  objWinHttp.Open "GET", "http://127.0.0.1:8080/pinsafe/SCImage?username="&request.querystring("username"), false
end if
```

Create a new Repository called pinsafe of type other.

Copy the pinsafe.inc file to von\InternalSite\inc\CustomUpdate

Edit pinsafe.inc so that the secret (m_secret), ip address and port matches that of the PINsafe server

```
function checkswivelpwd (userName, password)
LIGHT_TRACE "checkswivelpwd entered for " & userName
LIGHT_TRACE "SWIVEL - lets check if the password is right"
Dim strHTML
m_secret = "secret"
Dim objWinHttp
m_request = "<?xml version=""1.0"" ?><SASRequest><Version>1.0</Version><Action>login</Action><Username>" & username & "</Username><OTC>" & pa
& m_secret & "</Secret></SASRequest>"
Set objWinHttp = Server.CreateObject("WinHttp.WinHttpRequest.5")
```

```
objWinHttp.Open "GET", "http://<ipaddress>:8080/pinsafe/AgentXML?xml=" & m_request, false
```

Create a new Repository called pinsaferadius or type RADIUS.

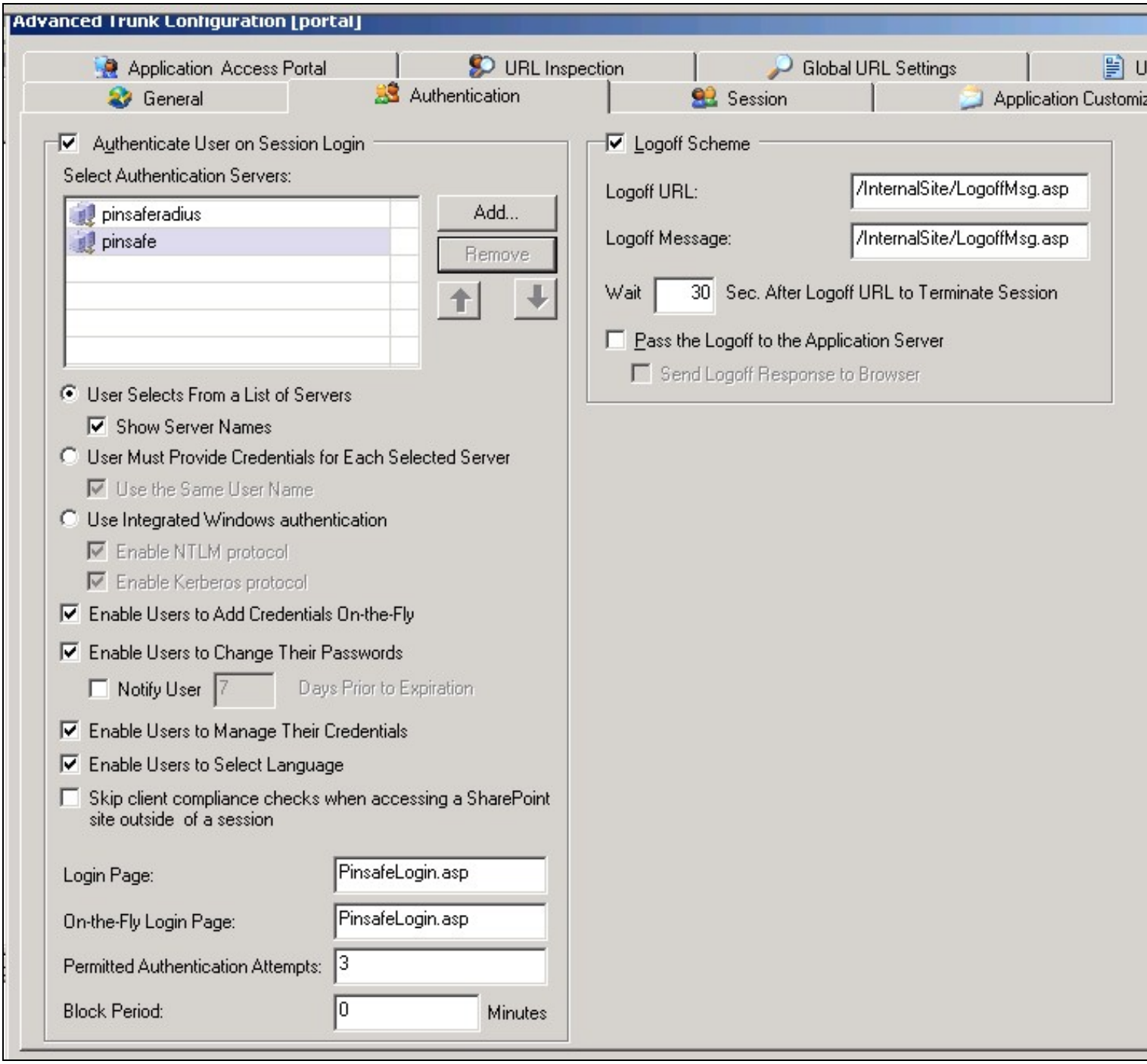
Enter the details of the PINsafe RADIUS server on the config screen.

Trunk Configuration

For the trunk you are using eg portal, ensure that both pinsafe and pinsaferadius repositories are associatd with the page

Also ensure that the option User Selects from A List of Servers is set

Set the login pages to be PINsafeLogin.jsp



Now copy the PINsafeLogin.jsp to von\InternalSite

Edit the PINsafeLogin.jsp to ensure that the repository names match those that you are using and that the dual channel and single channel authentication are matched to the correct repository.

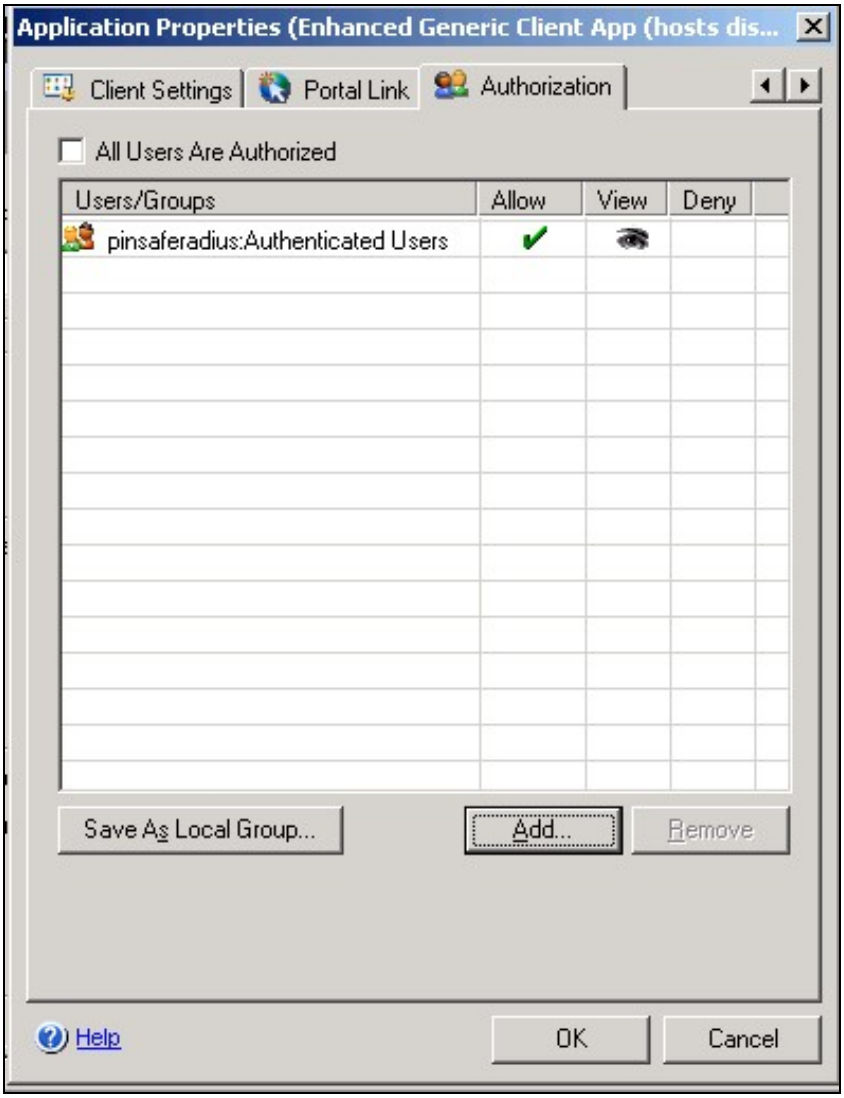
```
function checkradio()
{
  var radiovalue = eval(document.form1.swivel[1].checked);
  var r = document.getElementById("repository");
  if (radiovalue == true)
  {
  //alert("turing");
```

```
  //TURing selected, therefore refresh TURing image
  updateotp();
  //repository for TURing is pinsafe
  r.value = "pinsafe"
  } else{
 //alert("sms");
 updateindex(); //if we are using multi-sms update index will display required index
 r.value = "pinsaferadius"
 //repository for TURing is pinsaferadius
 }
}
```

Application Authorization

With different repositories aligned to different authentication methods, it is possible then to make some applications only accessible when a user has authenticated using the dual channel method.

To do this restrict access to applications to the pinsaferadius group on the Trunk->Applications-.Authorization tab

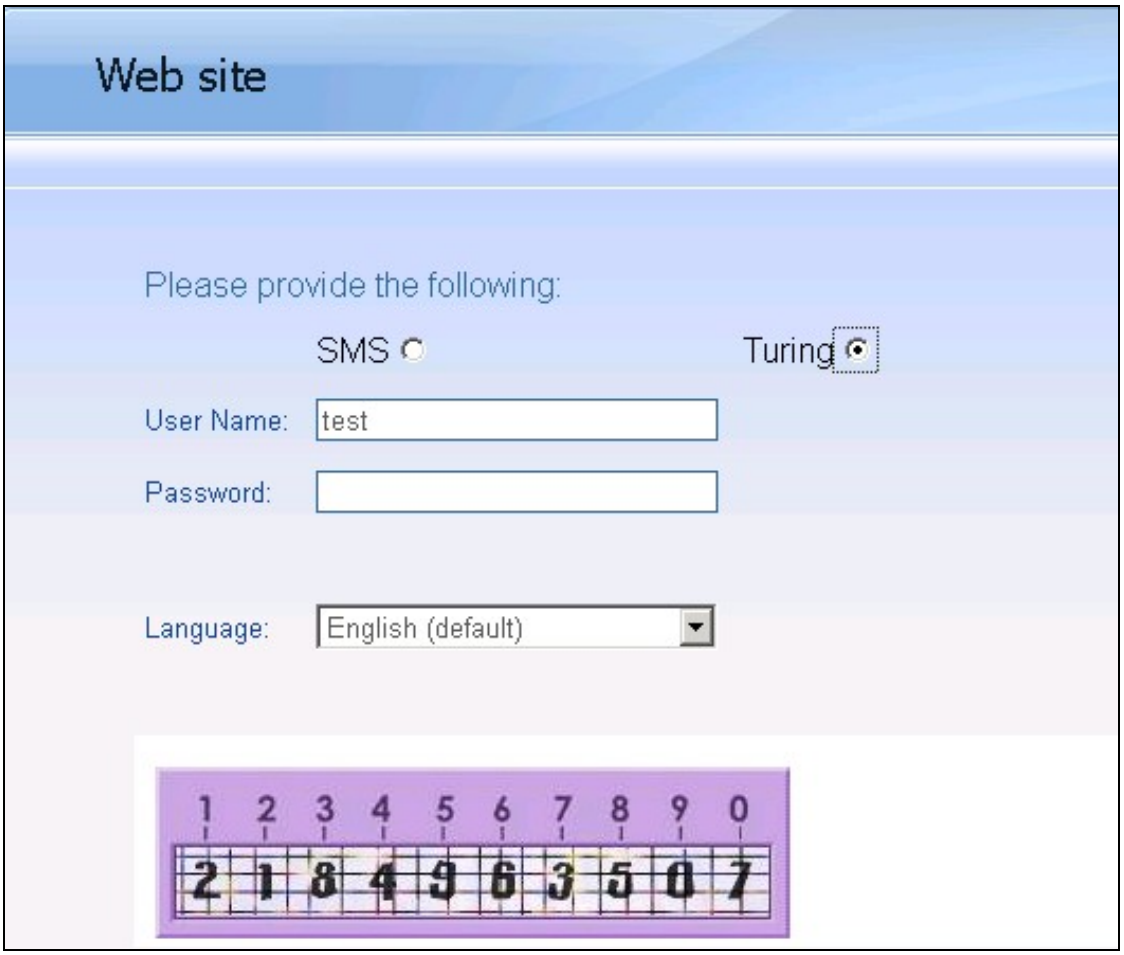


User Experience

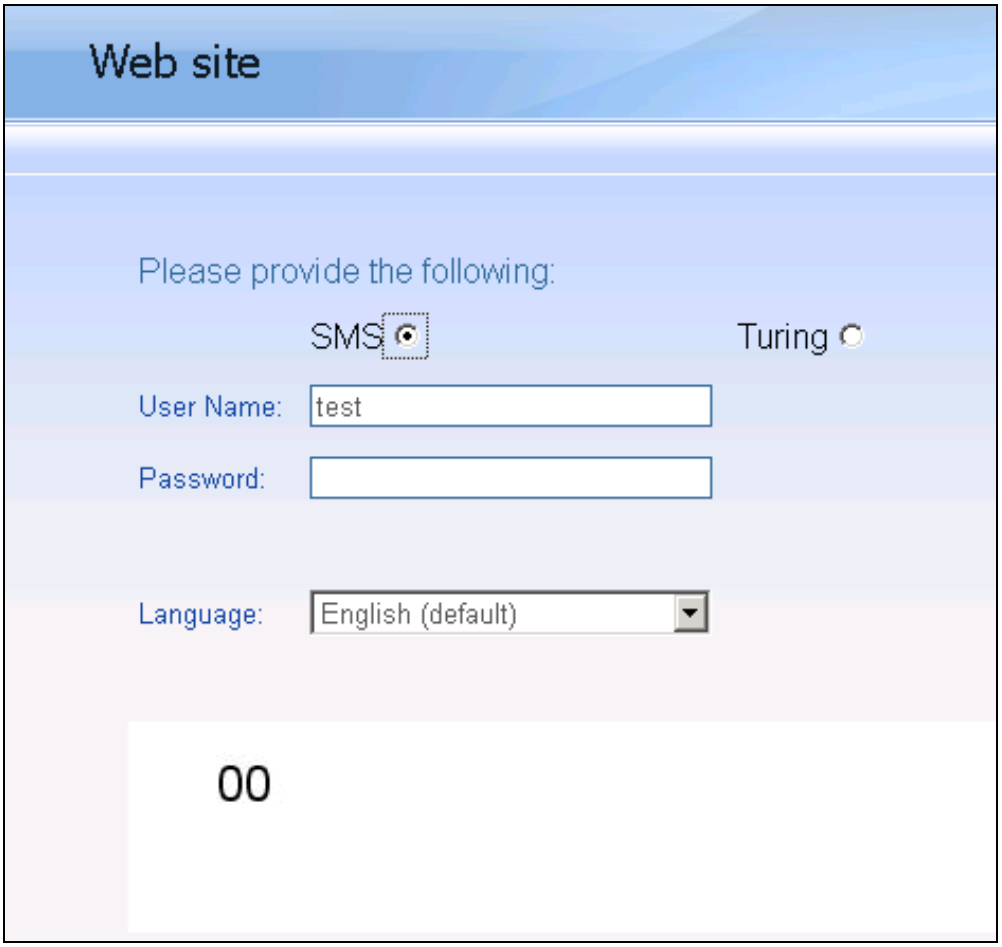
The user is presented with the option of authenticating via SMS or TURing.

To authenticate the user enters their username and then clicks on the authentication method they wish to use.

If they select TURing and TURing image is displayed.



If they select SMS (and multi-SMS is being used) the index of the security string that they need to use is displayed.



(If they have no valid SMS strings, -1 is shown)

When they make their selection the login page automatically associates them with the correct repository.

After authentication they will only have access to applications appropriate to their method of authentication.