



# Snowy Technology Preview - Citrix Secure Gateway

**By Citrix Consulting Services**

**Citrix Systems, Inc.**



## Notice

The information in this publication is subject to change without notice.

THIS PUBLICATION IS PROVIDED "AS IS" WITHOUT WARRANTIES OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. CITRIX SYSTEMS, INC. ("CITRIX"), SHALL NOT BE LIABLE FOR TECHNICAL OR EDITORIAL ERRORS OR OMISSIONS CONTAINED HEREIN, NOR FOR DIRECT, INCIDENTAL, CONSEQUENTIAL OR ANY OTHER DAMAGES RESULTING FROM THE FURNISHING, PERFORMANCE, OR USE OF THIS PUBLICATION, EVEN IF CITRIX HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES IN ADVANCE.

This publication contains information protected by copyright. Except for internal distribution, no part of this publication may be photocopied or reproduced in any form without prior written consent from Citrix.

The exclusive warranty for Citrix products, if any, is stated in the product documentation accompanying such products. Citrix does not warrant products other than its own.

Product names mentioned herein may be trademarks and/or registered trademarks of their respective companies.

**Copyright © 2001 Citrix Systems, Inc., 6400 NW 6th Way, Ft. Lauderdale, Florida 33309 U.S.A. All rights reserved.**

Version History		
October 12, 2001	Citrix Consulting Services	Version 1.0

# Table of Contents

<b>DOCUMENT OVERVIEW</b> .....	<b>2</b>
<b>SNOWY ARCHITECTURE</b> .....	<b>3</b>
NFUSE ARCHITECTURE WITHOUT SNOWY .....	3
SNOWY COMPONENTS .....	4
<i>How the Components Interact</i> .....	4
<b>CONDUCTING PROOF OF CONCEPT</b> .....	<b>6</b>
HARDWARE.....	6
SNOWY TECHNOLOGY PREVIEW INSTALLATION AND CONFIGURATION.....	6
<i>Windows 2000 Assumptions</i> .....	6
<i>MetaFrame XP Installation</i> .....	7
<i>Installing SSL Certificates</i> .....	7
<i>Installing Test Certificates</i> .....	7
<i>NFuse Web Site</i> .....	9
<i>Snowy Component Installation</i> .....	9
NOTABLES .....	9
ADDITIONAL INFORMATION.....	11
<b>APPENDIX: HOW TO CONFIGURE NFUSE TO SHARE LOGIN FORM WITH RSA SECURID</b> .....	<b>12</b>

## Document Overview

This document is intended to:

- Explain the value that Snowy provides
- Give a quick overview of the components and their interaction
- Point out installation anomalies
- Provide a basic understanding of functionality that will be provided in the future

*It does not intend to replace documentation available at the Snowy Technology Preview Page in the CDN website <http://www.citrix.com/cdn>. This information is available by becoming a member of the Citrix Developers Network. Upon logon to the CDN, perform a search for "snowy." The Technology Preview Page is temporarily hidden. Currently, this is the only way to find the information.*

# Snowy Architecture

## NFuse Architecture without Snowy

The diagram below depicts standard access to MetaFrame through an NFuse website. Overall, this configuration is convenient and secure, but still leaves many weaknesses available for an intruder to exploit. The web server can be secured with SSL and further secured with a solution such as RSA SecurID. This still leaves the MetaFrame server farm open to people who can make a direct ICA connection to published applications on that farm. Furthermore, pre-FR1 for MetaFrame XP servers did not provide the functionality to encrypt traffic to and from clients utilizing SSL. Another issue with this configuration is that the addresses of the MetaFrame servers had to be visible from outside the firewall.

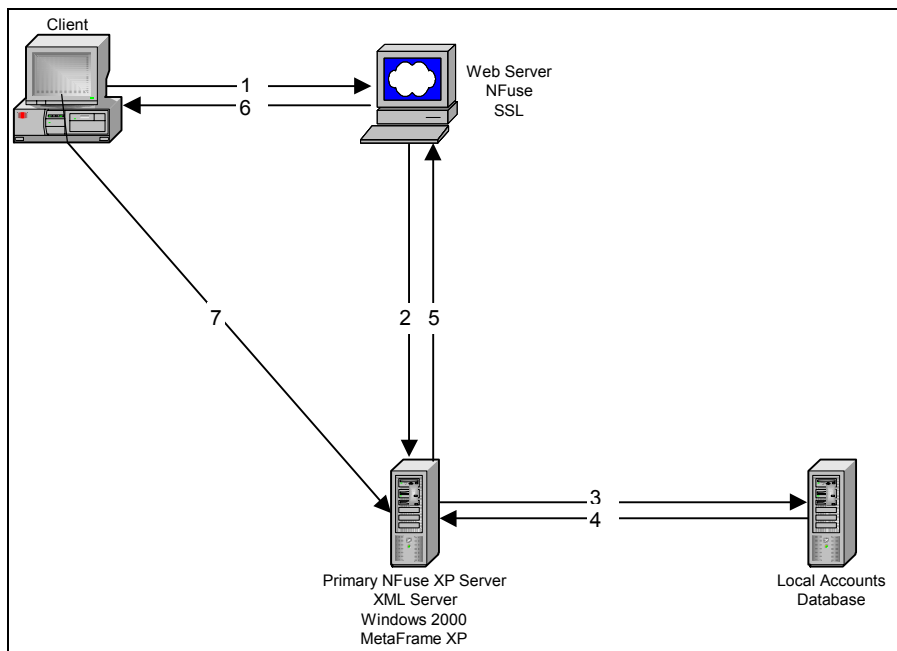


Figure 6.1: User authentication through NFuse

Citrix has come up with a solution to these issues and named it the Citrix Secure Gateway. This solution consists of a server that stands in front of the servers in a MetaFrame farm acting as a secure gateway.

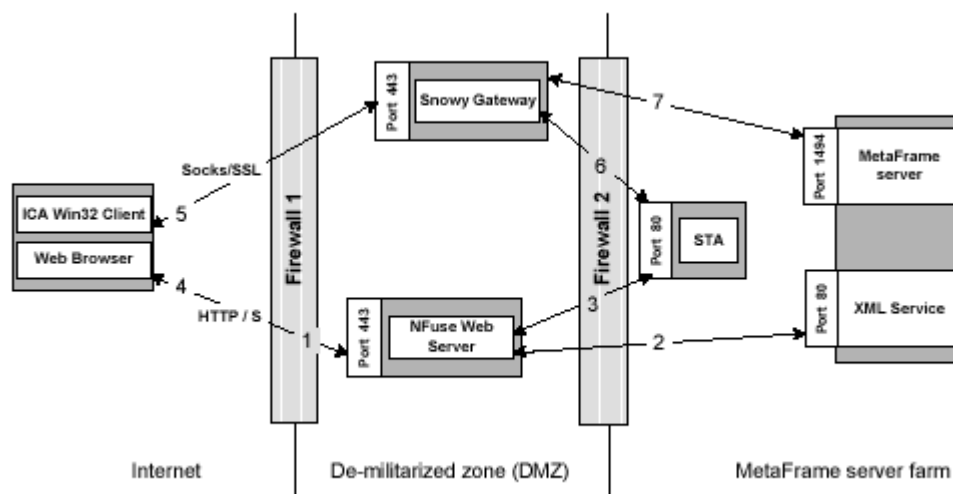
## Snowy Components

In order to secure traffic coming in over the Internet, the Snowy Gateway is installed in the DMZ (de-militarized zone) as a security perimeter that protects MetaFrame resources and applications on the corporate Intranet. The Snowy solution involves the interaction of five network components:

- ✓ A Citrix MetaFrame XP server farm
- ✓ A Snowy Gateway server
- ✓ A Citrix NFuse enabled Microsoft IIS 5.0 server
- ✓ A Snowy Ticket Authority (STA) server
- ✓ A client device with the ICA Win32 Client, Version 6.20 or higher installed

## How the Components Interact

A typical Snowy Gateway configuration is shown below to illustrate how the various components interact to provide security.



As illustrated in the figure above, the following communications take place between Snowy components before a secure connection is established.

1. A remote user launches a browser connection to an NFuse Web server on port 80 (HTTP), or port 443 (HTTPS<sup>1</sup>). You can increase security by deploying a secure NFuse Web server (HTTPS).<sup>2</sup>

The NFuse Web portal requires the user to authenticate using his/her user credentials.

2. NFuse uses the user credentials to contact the XML Service on a MetaFrame server and obtain a list of applications that the user is authorized to access. NFuse populates the Web portal page with the list of published

<sup>1</sup> HTTPS is normal HTTP wrapped in a secure SSL layer.

<sup>2</sup> Information on configuring this can be found in the appendix.

applications that the user is authorized to access. The communications so far are the normal sequence of events that occur when an NFuse Web server is deployed to serve up published applications to ICA Client users.

3. When the user clicks on a published application link, NFuse sends the IP address for the requested MetaFrame server to the STA and requests a Snowy ticket for the user. The STA saves the IP address and issues the requested Snowy ticket to NFuse.
4. NFuse generates an ICA file, containing the ticket issued by the STA and sends it to the client browser.
5. The browser passes the ICA file to the ICA Client, which then launches an SSL connection to the Snowy Gateway. Initial SSL handshaking is performed to establish the identity of the Snowy Gateway.
6. The Snowy Gateway accepts the ticket from the ICA Client and uses information contained in the Snowy ticket, to identify and contact the STA for verification. If the STA is able to validate the ticket, it returns an IP address of the MetaFrame server on which the requested application resides. If the ticket is invalid or expired, the STA informs the Snowy Gateway, and a client error message is displayed.
7. On receipt of the IP address for the MetaFrame server, the Snowy Gateway establishes an ICA connection between the ICA Client and the MetaFrame server. The Snowy Gateway monitors data flowing through the connection, and encrypts/decrypts client-server communications.

## Conducting Proof of Concept

This section details the configuration of the POC available for inspection in the CCS Lab.

### Hardware

**Server:** nfuse.citrix.com

**IP Address:** 10.7.13.101

**Function:** IIS/NFuse server

**Build:** Manual, web server

**Server:** anotheriis.citrix.com

**IP Address:** 10.7.13.136

**Function:** IIS/Snowy Ticket Authority

**Build:** Manual

**Server:** sgmachine.citrix.com

**IP Address:** 10.7.13.233

**Function:** Snowy Gateway Server

**Build:** Manual

**Server:** sma-test.citrix.com

**IP Address:** 10.7.13.250

**Function:** MetaFrame XP Server, Zone Data Collector, Data Store

**Build:** Manual, METAFRAME XP application server

## Snowy Technology Preview Installation and Configuration

### Windows 2000 Assumptions

The following tasks are to be completed prior to the installation of other components.

1. Windows 2000 Server installed on all machines with Service Pack 2.
2. Server Fully Qualified Domain Names registered in DNS.
3. Terminal Services installed in Application Server mode.

## MetaFrame XP Installation

This installation of this server uses most of the defaults in the XP installation process. Before installing Snowy components, certificates must be installed on the Citrix Secure Gateway and the NFuse web server. The following instructions detail installation of the certificates. Follow these instructions for installing certificates on both components. In this case, IIS was on both servers. IIS can be removed from the CSG after certificate installation. In the case that the CSG box does not have IIS, the certificate can be processed on another server with IIS. The certificate is then exported to the CSG. If problems are experienced with this approach, the first approach should be considered.

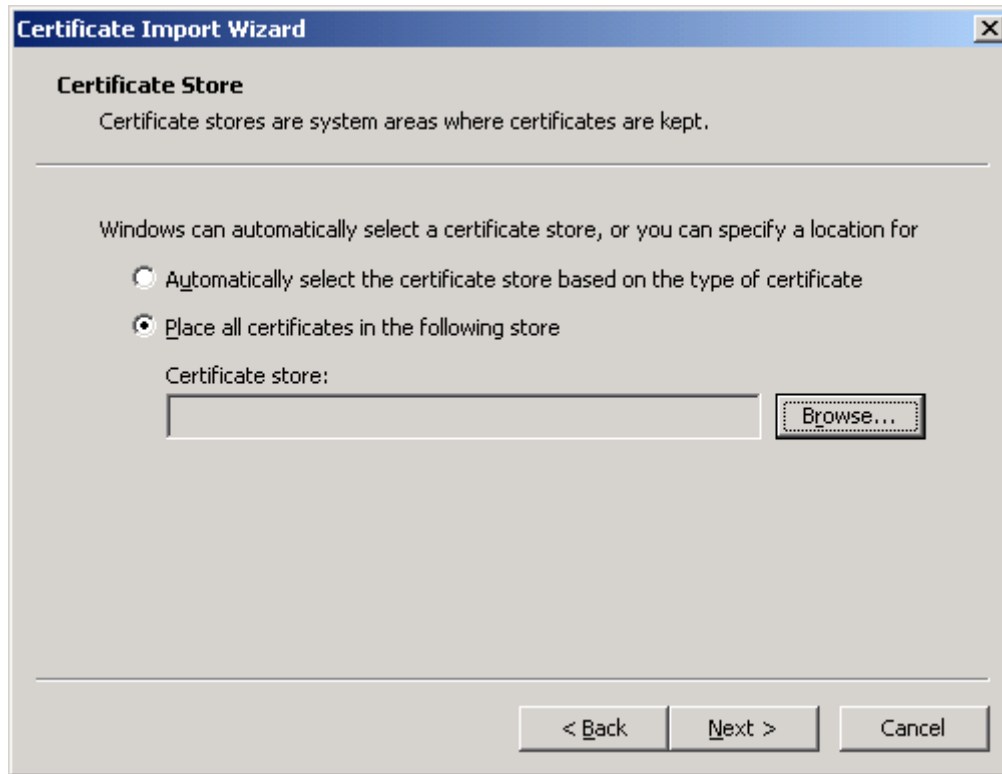
### Installing SSL Certificates

1. Open up Internet Services Manager and right click on the default web site.
2. Select Properties and go to the Directory Security tab. Click on the button **Server Certificate** in the Secure Communications section.
3. The IIS Certificate Wizard will run. Select **Create a new certificate** and click the Next button.
4. Keep the default selection, **Prepare the request now, but send it later**.
5. Use "MetaFrame" for the certificate name and select the 1024 bit length.
6. Enter "Citrix" for the organization name and "Citrix" for the organizational unit.
7. When asked for the **Common Name**, enter "www.citrix.com" in the case of the NFuse web server and "sgmachine.citrix.com" in the case of the CSG.
8. Fill out the rest of the information about location. When asked to do so, save the certificate and continue through the wizard until it finishes.
9. Send this certificate request to Verisign to obtain a test certificate. Verisign will send back a signed certificate.
10. Go to Properties, Directory Security tab again in Internet Services Manager and select the **Server Certificate** button.
11. Select **Process the pending request and install the certificate**.
12. When prompted, enter the signed certificate information and finish running through the wizard.
13. Free trial certificates from Verisign were used in this POC. They require a root certificate to be installed on the servers and on every client accessing the system. Copy the public key that was obtained from Verisign on to the server. Follow the steps in the [Installing test certificate](#) section of this document to install the certificate root on the server.
14. Copy the Public key certificate file to all client devices that will MetaFrame through the secure NFuse site and follow the steps in the [Installing test certificate](#) section to install the certificate.

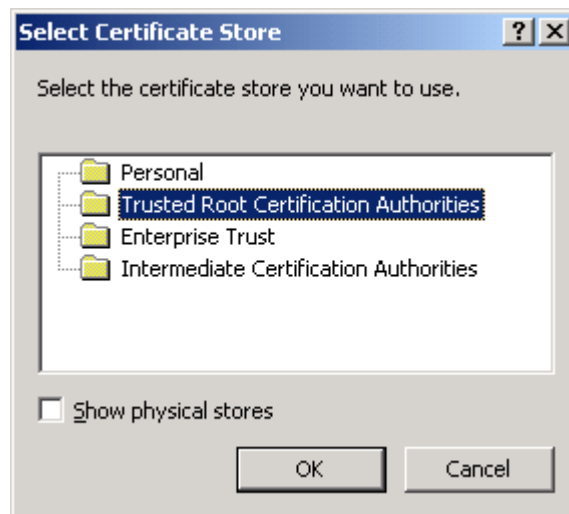
### Installing Test Certificates

1. Select "getcacert.cer" and double click open
2. Click **Install Certificate....** Then select **Next**

3. Select the **Place all certificates in the following store** option as shown below.



4. Select **Browse** and then in the following screen, select **Trusted Root Certification Authorities** and click **OK** as shown below.



5. Select **Next** and then click **Finish**

6. To verify the installation of the test certificate root, go into Internet Explorer then go to Tools/Internet Options/ Content/ Certificates (trusted root authorities tab) it should read as "For VeriSign Authorized Testing Only".

## NFuse Web Site

The following sections detail the NFuse setup steps for the POC environment.

### NFuse POC Assumptions

- ✓ NFuse 1.5 will be used.
- ✓ Microsoft IIS will be used as the web server.
- ✓ The NFuse web pages for the POC will consist of those provided with the Snowy Technology Preview. No customizations will be made.
- ✓ SSL will be used to secure the POC web site.

### Installing NFuse Web Extensions on the web server

The installation will also ask about installing the ICA clients onto the web server. Select "Yes" to install the ICA clients to the NFuseClients directory. Provide the path to the ICA Client's CD. These clients are required to take advantage of the functionality provided with the Snowy Technology Preview. After the NFuse installation is completed, the updated Win32 ICA Web Client that was downloaded from the Citrix website needs to be placed on the web server in the Inetpub\wwwroot\Citrix\ICAWEB\en\Win32 directory. This client consists of the ica32t.exe and wficat.cab files. These files need to replace the ones that were installed from the MetaFrame ICA Clients CD. If the updated client is not used, clients will not be able to connect through Snowy to MetaFrame servers.

The NFuse web site was configured to use SSL by generating a certificate request from Verisign and installing the certificate as described above. After installation of the certificate that was named <https://www.citrix.com/snowy>, the web site was configured to use SSL by following the steps below:

- ✓ Open up Internet Services Manager and right click on the default web site.
- ✓ Select Properties and go to the Directory Security tab. Click on the button **Edit** in the Secure Communications section.
- ✓ Select the check box to require a secure channel.

## Snowy Component Installation

After installation of certificates on the Citrix Secure Gateway and the NFuse web server, installation of Snowy components can begin. Snowy Components must be installed in three places:

- ✓ Snowy Gateway server
- ✓ Citrix NFuse Microsoft IIS 5.0 server
- ✓ Snowy Ticket Authority (STA) server

Follow the instructions in the Snowy Administrator's Guide downloaded from the Citrix Developer's Network.

## Notables

In the case of the POC in the CCS Lab, the website will require HTTPS. It can be accessed here:

<https://www.citrix.com/snowy>

If you would like to check the connection status (SSL or regular) and you don't use Connection Center, you can use the Connection Status dialog box (accessible via system menu of the ICA client window), or you can simply invoke **netstat** from the command line to see the connection port number. If netstat is reporting **443** or **https** connection type for your ICA connection, it will indicate that you have SSL enabled.

Certificates were installed on the web server to secure it. To further secure the web site, instructions are provided in [Appendix 1](#) on securing it with RSA SecurID.

As seen in the diagram, all traffic crossing Firewall 1 is secured with SSL. Traffic in the diagram is crossing Firewall 2 through ports 80 and 1494 (the standard MetaFrame port). While traffic between NFuse and the XML service can use ports other than 80, and SSL, for this release, ICA traffic between CSG and MetaFrame server can only be over 1494. Secure ICA can be used for greater security in this area of traffic; future releases may include SSL functionality here. Another feature that did not make the preview was the configuration capability in the Ticket Authority (STA) to allow it to secure traffic with SSL and function over a port other than 80. This functionality may be included in the final release.

Other notables include:

- ✓ CSG v1.0 supports multiple, redundant STAs out-of-the-box. It is important to mention that CSG v1.0 supports multiple STAs for **redundancy** purposes only. It will use the first STA in the list if it works, and will switch to backup STA(s) if the primary STA fails. This feature was not available in the Technology Preview. CSG v1.0 documentation will have necessary information on how to configure and use multiple STAs. You may use an additional external LB system to load-balance STAs, however this is not expected to be a common case because the STA is not a performance-critical component. Two STAs (one primary and one backup) will probably be sufficient even for a large system. The CSG on the other hand does support load-balanced arrays.
- ✓ CSG v1.0 will support IIS only, for NFuse and STA hosting. Necessary changes are being rolled into standard NFuse so that the next NFuse release will be able to support CSG out of the box on any platform that is supported by NFuse.
- ✓ There are currently no plans to have STA available on other (non-Windows) platforms, but this is still under discussion.
- ✓ There is no additional configuration necessary on the MetaFrame servers themselves such as publishing applications in a special way. For example, if an application is published without any encryption, the CSG box will provide SSL. From the MetaFrame point of view the connection will not be seen as SSL.
- ✓ If you would like to encrypt the connection between the CSG and the MetaFrame server, use SecureICA by enabling SecureICA settings on the MetaFrame server. SecureICA is not as strong as SSL, but may be adequate for the CSG to MetaFrame link. It also has much less overhead and does not require additional certificates to be installed.
- ✓ All communication between Snowy components is done using XML over HTTP. STA traffic is not as sensitive as NFuse XML traffic - STA protocol never sends passwords or anything like that. A future release will secure the STA traffic, but it is basically XML over HTTP Communication. Snowy v1.0 does not support SSL for inter-component communication. STA protocol is very simple. NFuse is requesting a ticket from STA asking STA to save the MetaFrame server address, and then Snowy is presenting the ticket and wants to get METAFRAME server address. There are a few other capabilities, but tickets processing is the basic operation.
- ✓ It is actually possible to use HTTPS (SSL) to secure STA traffic between NFuse and STA, however Snowy<->STA communication cannot use SSL now. If the customer would like to secure traffic between Snowy components, IPSec or a similar mechanism is recommended. Because Snowy v1.0 runs on Win2K, IPSec is available out of the box.
- ✓ The STA can be placed in the DMZ, with no problems. If you would like to get the maximum possible security, you might want to run STA in a separate VLAN, so it will not be directly accessible from any network (access will be filtered by the firewall), but it is really up to the customer.

## Additional Information

Visit the following links to find documents relating to Citrix Secure Gateway (CSG):

- Placeware training [http://au-ctxeng/projects\\_data/securityserver/techpreview\\_train/snowy\\_train.htm](http://au-ctxeng/projects_data/securityserver/techpreview_train/snowy_train.htm)
- FAQ [http://au-ctxeng/Projects\\_data/securityserver/pm\\_files/Snowy Internal FAQ 1\\_0.doc](http://au-ctxeng/Projects_data/securityserver/pm_files/Snowy Internal FAQ 1_0.doc)
- Internal presentation [http://au-ctxeng/Projects\\_data/securityserver/pm\\_files/Internal Presentation Aug 2001.ppt](http://au-ctxeng/Projects_data/securityserver/pm_files/Internal Presentation Aug 2001.ppt)
- Tech Preview software [http://au-ctxeng/projects\\_data/securityserver/tech\\_preview/snowy.htm](http://au-ctxeng/projects_data/securityserver/tech_preview/snowy.htm)
- Technical training PPT [http://au-ctxeng/Projects\\_data/securityserver/pm\\_files/STP Technical Training.ppt](http://au-ctxeng/Projects_data/securityserver/pm_files/STP Technical Training.ppt)

## Appendix: How to configure NFuse to share login form with RSA SecurID

### Anatoliy Panasyuk who configured this in-house provided the following text:

One of the features provided by Citrix Secure Gateway (CSG) is the ability to enforce NFuse logins – the user is required to authenticate to NFuse before he/she would be able to open MetaFrame connection.

This feature can be used to enforce strong authentication before allowing users to connect to MetaFrame. RSA SecurID is one of the commonly used two-factor authentication solutions, and many CSG users might want to use it with RSA SecurID with it.

This description provides some technical details on the possible configuration and on the login page configuration in particular.

### How to secure NFuse with RSA ACE client (to support SecurID tokens).

RSA ACE client can secure NFuse. The web master needs to install appropriate RSA ACE client or equivalent software on the web server, and then configure it to secure NFuse site.

If IIS web server is used, RSA ACE client will add extra tabs in the “Properties” dialog box in Information Services Manager. Security can be set on an individual site, as well as on a particular required section of the web site.

### How to create single RSA/NFuse logon page.

Being used in standard configuration, RSA ACE client will add initial RSA login page that will be displayed on the first attempt to use NFuse WEB site. This login page will be displayed *before* NFuse login page, and from the user point of view, it will look like double-login – the user will be asked to login twice:

- RSA login page will be displayed first, asking for the user name and SecurID code.
- NFuse login page will be displayed after successful RSA login, asking for the user name, domain (optional), and password.

This double-login is undesirable because it affects user experience. Double-login problem can be resolved by creating custom login page that will ask for all required information for both RSA and NFuse authentication.

RSA login page is displayed first, and this combined login page should be used instead of the default RSA login page.

In order to have single login page we have to add NFuse credentials fields to RSA login page. Because some of the fields (User Name field) are typically the same for both RSA and NFuse login, it is possible to use single input field for this type of information.

If NFuse login with pre-configured NT domain is used, modified RSA login page will have three fields:

- User Name
- RSA SecurID code
- NT Domain password

If NFuse login is configured to allow user to specify NT domain, modified RSA login page will have four fields:

- User Name
- RSA SecurID code
- NT Domain
- NT Domain password

Because of the RSA login page processing, only User Name and RSA SecurID code should be submitted for RSA login processing. The rest of the fields should be saved somehow for NFuse login processing that will take place right after successful RSA login.

One of the possible ways to achieve this is to make RSA login page (common login page, in fact) to save entered NFuse credentials in a temporary cookie on the client. Right after RSA login processing NFuse login page will be invoked. NFuse login page can read saved credentials from the cookie to skip the default NFuse login page. At the same time, the cookie with NFuse login credentials can be destroyed.

If RSA login failed, RSA login page will be re-displayed, and this page can immediately destroy the cookie. Even if the connection terminates right after RSA login, and NFuse pages did not destroy cookie, the browser will automatically destroy the cookie as soon as the browser closes, i.e. the cookie will never be stored in the client persistent storage.

There are some other possible technical ways to achieve the same result. The above description outlines just one of the possible ways to achieve this. The main idea is to use single login page that asks for both types of credentials (RSA and NFuse) simultaneously and then deliver NFuse credentials to NFuse page somehow.

### **Modifications of the RSA login page.**

RSA login page is just an HTML page with input fields for the user name and RSA SecurID code. It is typically stored in passcode.htm file in the System32\ACECLNT directory. This file can be modified using any editor - HTML or just a plain text editor such as Notepad. However, ACE client processing of the passcode.htm file imposes certain limitations on the supported syntax and content of this file. For example, quotes should be avoided (use **&quot;**; syntax instead). For full information on the passcode.htm file changes please refer to RSA documentation.

Sample version of the modified passcode.htm file is available in Appendix A. Please note that this sample is provided "as is", with no implicit or explicit warranty of any form, and is not supported by Citrix or RSA. It is just a sample to illustrate the proposed solution. Use it at your own risk.

### **Modifications of the NFuse login page.**

In order to retrieve and use NFuse credentials left in the cookie (or by other means) by the modified RSA login page, NFuse pages should be modified. Exact code changes depend on the version of NFuse site that is used. However, these changes are rather trivial.

Basically NFuse ASP code (if IIS is used) should detect credentials left in the cookie by RSA login page (by retrieving the cookie), verify these credentials and if credentials are valid, skip NFuse login page going directly to the main NFuse applications page. If NFuse credentials left by RSA page are not valid or are missing, standard NFuse login page can be displayed.

If NFuse detected credentials, left by RSA login page, it can destroy the cookie after credentials retrieval.

If another mechanism (based on some other technique instead of cookies) is used to transfer credentials from RSA login page to NFuse, NFuse ASP scripts (if IIS is used) should be modified accordingly.

## Sample Passcode.htm file

Please note, this sample is provided "as is", with no explicit or implicit warranty. It is just a sample to illustrate the proposed solution, use it at your own risk.

```
<HTML><HEAD>
<TITLE>Citrix Systems Asia Pacific</TITLE>
</HEAD>
<body bgcolor="#FFFFFF" LEFTMARGIN="0" TOPMARGIN="0" MARGINWIDTH="0" MARGINHEIGHT="0"
background="/images/bkgrd_main.gif" >
<table width="100%" border="0" cellpadding="0" cellspacing="0">
<tr>
<td height="94" border="0" background="/images/homebanner.jpg">
<table width="100%" border="0" cellspacing="0" cellpadding="0">
<tr>
<td align="center" width="18%">

</td>
<td align="center" width="20%"></td>
</tr>
</table>
</td>
</tr>
</table>
<table width="100%" border="0" cellspacing="0" cellpadding="0">
<tr>
<td width="28"></td>
<td colspan="3" valign="top">
<p align="center"><font size="2"><strong><br>
<br>
</strong></font>
</td>
</tr>
</table>
<H1 ALIGN=CENTER>Employee Login</H1>
<P>
<HR>The page you are attempting to access requires you to login.</P>
```

<P>Please enter your Username, MIS Password and SecurID PASSCODE in the following fields, and then click &quot;Send.&quot; <br>If you make a mistake, use &quot;Reset&quot; to clear the fields.

<HR>%s </P>

<P><FORM NAME="Form2"></P>

<CENTER><TABLE>

<TR>

<TD><B>Username:</B></TD>

<TD><INPUT TYPE=TEXT NAME="username" VALUE="" MAXLENGTH=32></TD>

</TR>

<TR>

<TD><B>Domain Password:</B></TD>

<TD><INPUT TYPE=PASSWORD NAME="domainPassword" VALUE="" MAXLENGTH=32></TD>

</TR>

<TR>

<TD><B>SecurID PASSCODE:</B></TD>

<TD><INPUT TYPE=PASSWORD NAME="passcode" VALUE="" MAXLENGTH=16></TD>

</TR>

</TABLE></CENTER>

<CENTER><P>

<INPUT TYPE="BUTTON" NAME="SendButton" VALUE="Send" OnClick="processSend2()" LANGUAGE="JavaScript" >

<INPUT TYPE=RESET VALUE="Reset">

</P></CENTER>

</FORM>

<HR>

<span style="visibility:hidden">

<P><FORM method=POST NAME="Form1" action="%s"></P>

<CENTER><TABLE>

<TR>

<TD><B>Username:</B></TD>

<TD><INPUT TYPE=TEXT NAME="username" VALUE="%s" MAXLENGTH=32></TD>

</TR>

<TR>

<TD><B>PASSCODE:</B></TD>

```

<TD><INPUT TYPE=PASSWORD NAME="passcode" VALUE="" MAXLENGTH=16></TD>
</TR>
</TABLE></CENTER>
<HR>
<CENTER><P>
<INPUT TYPE="BUTTON" NAME="SendButton" VALUE="Send" OnClick="processSend()" LANGUAGE="JavaScript" >
<INPUT TYPE=RESET VALUE="Reset">
</P></CENTER>
<INPUT TYPE=HIDDEN NAME="referrer" VALUE="%s">
</FORM>
</span>
<script language=JavaScript>
<!--
var myForm = document.forms.Form1;
var myForm2 = document.forms.Form2;
function processSend() {
myForm.submit();
}
function processSend2() {
var tmps;
myForm.username.value = myForm2.username.value;
myForm.passcode.value = myForm2.passcode.value;
SetCookie ("NFuseData_NFuse_User", myForm2.username.value, null, "/" );
SetCookie ("NFuseData_NFuse_Password", myForm2.domainPassword.value, null, "/" );
myForm.submit();
}
// make enter work like submit
function lemmein(keypressed) {
    var key;
    if (document.all)
        key = window.event.keyCode;
    else
        key=keypressed.which;
    if (key==13) myForm.submit();
}

```

```

}
if (navigator.appName == "Netscape")
    myForm.passcode.onkeypress = lemmein;
var popup_auth = null;
// when the popup closes attempt to reload this page
function check_popup()
{
    if (popup_auth && popup_auth.open && !popup_auth.closed)
        setTimeout("check_popup()", 500);
    else
        self.document.location.reload(true);
}
// open a popup window for authentication
if (self != top) {
    var h = screen.height - 96;
    var w = screen.width - 48;
    if (navigator.appName.indexOf("Netscape") != -1) {
        popup_auth = window.open(location, "SecurIDPopup", "screenx=16,screeny=16,left=16,top=16,height=" + h +
",width=" + w);
    } else {
        popup_auth = window.open("", "SecurIDPopup", "screenx=16,screeny=16,left=16,top=16,height=" + h + ",width="
+ w);
        if (popup_auth.document.location.protocol != "http:" && popup_auth.document.location.protocol != "https:")
            popup_auth = window.open(location, "SecurIDPopup");
    }
    setTimeout("check_popup()", 500);
} else {
    if (myForm2.username.value == "") myForm2.username.focus();
    else myForm2.passcode.focus();
}
//
function getCookieVal (offset) {
    var endstr = document.cookie.indexOf(";", offset);
    if (endstr == -1)
        endstr = document.cookie.length;

```

```

return unescape(document.cookie.substring(offset, endstr));
}
//
function FixCookieDate (date) {
var base = new Date(0);
var skew = base.getTime(); // dawn of (Unix) time - should be 0
if (skew > 0) // Except on the Mac - ahead of its time
    date.setTime (date.getTime() - skew);
}
//
function GetCookie (name) {
var arg = name + "=";
var alen = arg.length;
var clen = document.cookie.length;
var i = 0;
while (i < clen) {
    var j = i + alen;
    if (document.cookie.substring(i, j) == arg)
        return getCookieVal (j);
    i = document.cookie.indexOf(" ", i) + 1;
    if (i == 0) break;
}
return null;
}
//
function SetCookie (name,value,expires,path,domain,secure) {
document.cookie = name + "=" + escape (value) +
    ((expires) ? "; expires=" + expires.toGMTString() : "") +
    ((path) ? "; path=" + path : "") +
    ((domain) ? "; domain=" + domain : "") +
    ((secure) ? "; secure" : "");
}
//-->
</script>

```



```
</td>  
</tr>  
</table>  
</BODY>  
</HTML>
```



6400 NW 6<sup>th</sup> Way

Fort Lauderdale, FL 33309

954-267-3000



<http://www.citrix.com>

Copyright © 2000 Citrix Systems, Inc. All rights reserved. Citrix, WinFrame and ICA are registered trademarks, and MultiWin and MetaFrame are trademarks of Citrix Systems, Inc. All other products and services are trademarks or service marks of their respective companies. Technical specifications and availability are subject to change without prior notice.