Application Note

Thin Client Computing Best Practices Guide

Juniper Networks Secure Access SSL VPN Solution for Thin Client Computing

Juniper Networks, Inc.
1194 North Mathilda Avenue
Sunnyvale, California 94089
USA
408.745.2000
1.888 JUNIPER
www.juniper.net

Part Number: 350124-001 Feb 2008

Table of Contents

Introduction

Thin Client Computing is a technology that involves executing an application on a centrally managed server while displaying the application on client systems. Enterprises primarily use Thin Client Computing to cost effectively allow users to remotely control and access applications written for one computing platform from other computing platforms. It also provides another tangible benefit in that the data is housed on the server, which is generally at the enterprise data center in a well-protected environment. This design alleviates the need for users to download data and "work offline" with that data, potentially exposing it to threats, such as viruses or even being accidentally left behind on a public PC.

The most prevalent Thin Client Computing systems include:

- Microsoft's Windows Terminal Services server—Capable of transporting the Windows desktop, as well as Windows-based applications to desktop computing devices, Microsoft's Remote Desktop Protocol (RDP) is the underlining technology to deliver this service to its clients.

  This service is available on Windows NT Server 4.0, Terminal Server Edition (Terminal Server), Windows 2000 Terminal Services, Windows 2003 server, and in a limited fashion, on Windows XP, 2K and Vista PCs. Clients use the Microsoft Remote Desktop program bundled with the Windows operating system to access the Windows Terminal Service Server.
- Citrix (including Presentation, NFuse/Web Interface and Secure gateways) server—This is server software that can host applications like Microsoft Office and Adobe Acrobat. Applications are executed on the centrally managed Citrix server while clients can access those applications through various tools, including program installations on the client or Web interface. At first glance, it looks and feels like Microsoft Remote Desktop (and in fact the underlying technology is indeed RDP). However, Citrix has bundled in several usability features which make it more convenient and user-friendly.

Scope

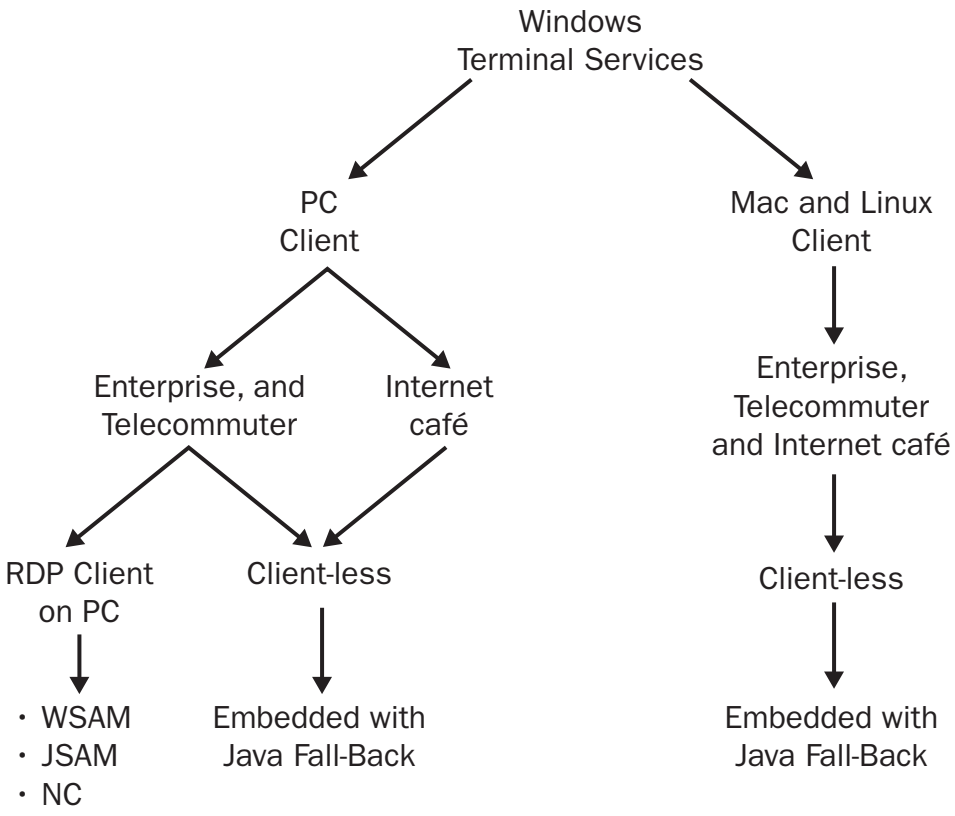
This document serves as a Thin Client Best Practices guide. It covers the most popular deployment scenarios and also outlines best practices for supporting a variety of users in a heterogeneous enterprise environment. This includes support for Windows, MacOS and Linux end users. There are many other Secure Access methods which can be used to deploy Microsoft Terminal Services and Citrix, however, they are not covered here as they are not best practices. Please refer to the SA Administration Guide for more details, as well as all of the configuration options for securing Citrix and Microsoft Terminal Services.

Description and Deployment Scenario

Windows Terminal Services

Juniper Networks Secure Access (SA) SSL VPN products support the native Windows RDP protocol for Microsoft Windows Terminal server deployments. SA allows network administrators to provide a solution for their entire client population—PC, Mac, Linux, Internet kiosk, telecommuter or office environment. SA can improve the Windows Terminal Services experience with embedded security functions such as strong encryption, remote access and cache cleaner.

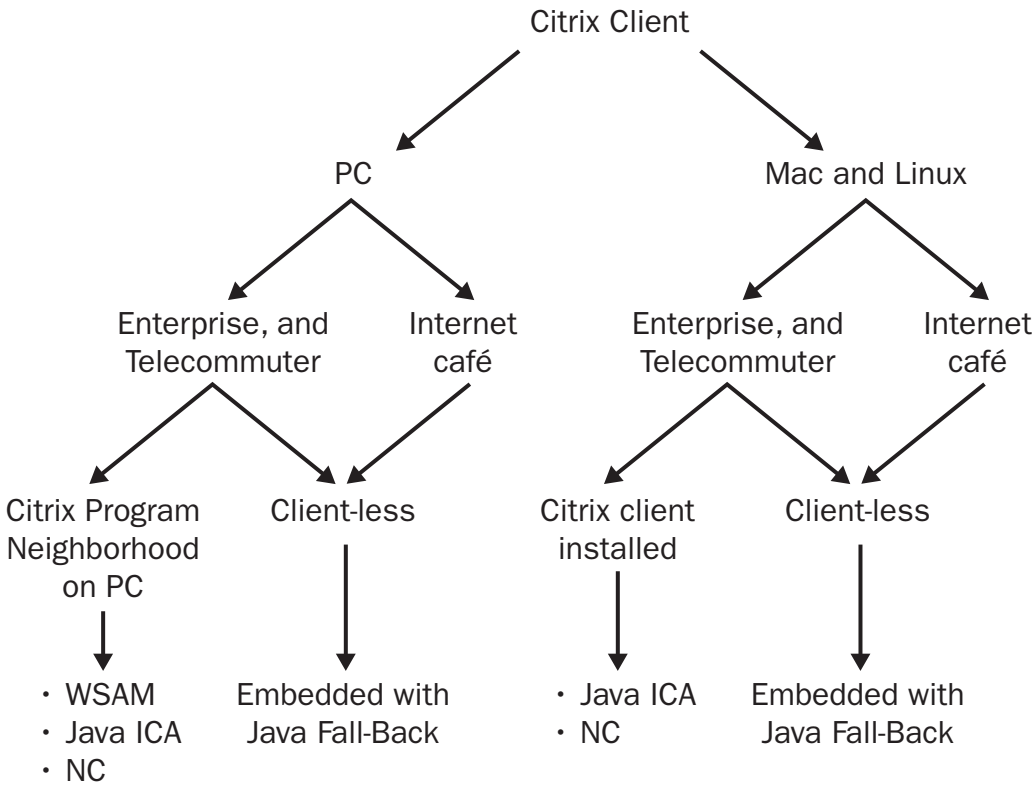
The SA best practices for Windows Terminal server are illustrated below:



Note: Always check the Juniper support site for the latest SA supported client platforms information.

Citrix

Juniper Networks Secure Access SSL VPN provides solutions that complement Citrix deployments. Secure Access encompasses security enhancements that allow Citrix server deployments to reach broader audiences in different scenarios—the enterprise, telecommuting and Internet kiosk access. SA can also generate a "Webified" interface for the native Citrix Presentation server. In addition, SA is able to perform load-balancing among multiple Citrix Application servers starting with release 5.2. These are some of the important features required for enterprises to successfully roll out Citrix services.



Note: Again, please check the Juniper support site for the latest SA supported client platforms information.

SA Access Methods

SA Core Access and Java Applets

SA Core Access provides access to a host of applications and resources from any Windows, Mac or Linux environments. The SA dynamically rewrites HTML and Java (among other content) on the fly, so that the end user's Web browser is able to properly retrieve subsequent and embedded URLs, objects and data.

Java applet support on the SA is one of the example services of the Core functionality. The SA can publish bookmarks which invoke a Java applet with a certain configuration, for example Java ICA. Administrators can now also use this feature to host applications like Java-based Windows RDP client via the Java applet upload framework, and deliver the applet on-the-fly. This removes the need to host Java applets on a backend Web server, thus simplifying the overall process and setup.

Because the SA intermediates the Layer 7 HTTP communication channel, the Core Access functionality is very granular in terms of access control, content manipulation and header management, down to the individual URL and object levels.

SAM

SAM (Secure Access Manager) is an agent for clients to access backend application servers in a client-server fashion. These clients and applications are connected directly to one another using a tunneling mechanism for that applications protocol and socket. SAM must be downloaded to SA subscribers and invoked locally. There are two different flavors of SAM.

JSAM (Java SAM)

This is a Java applet that is executed on the client. It is a universal solution for different types of clients (Windows, Mac and Linux) and requires only a Web browser and Java Virtual Machine (JVM).

SA system administrators can configure JSAM to support RDP or Citrix ICA traffic, based on the configured ports (3389 and 1424/2598, respectively). Once invoked, JSAM creates loopback sockets on the client PC. Application data is then forwarded to these sockets (rather than directly to the known IP of the application server) relying on either external Domain Name System (DNS) resolution to the predefined loopback addresses, or by modifying the Hosts file to resolve the known application server hostnames to the loopback sockets instead. This way JSAM is sure to get the application traffic and can tunnel it accordingly. Since JSAM uses sockets, if multiple backend servers are to be utilized, a socket must be configured for each server and port that needs to receive tunneled application data. Note: Administrative rights are required to modify the Hosts file.

WSAM (Windows SAM)

This is a Windows binary that installs just-in-time on the PC and supports applications on both TCP and UDP ports. It offers a host- or application-based approach for clients to reach Windows Terminal or Citrix Presentation servers, among others. However, administrator level access to the PC is necessary in order to install WSAM. This is because WSAM intercepts traffic at the Windows Kernel layer, using a TDI filter. While this may sound intrusive, TDI is actually a hook designed just for this purpose and in fact, this solution is superior to JSAM because no DNS resolution or Host file mapping needs to be changed. In Application mode, WSAM detects the request for a socket and merely reroutes that socket over to its SSL-encapsulated tunnel. This way, WSAM is sure to catch any and all traffic coming out of an application, even if it is not the standard port the application uses. In Host mode, WSAM detects all requests coming from any application with the matching destination socket, and then reroutes them over the SSL-encapsulated tunnel. On the backend, the SA strips off the SSL-encapsulation and forwards the full header + payload to the server, so that the client and server

can communicate directly as if there were no SA in between. Configuration is also simplified in that the only thing necessary is to configure the application executable name or destination host address. Note: Administrative rights are required to install WSAM, unless using the Juniper Installer Service.NC (Network Connect)

NC provides a Layer 3 (network layer) solution for connectivity. That is, the client PC will actually receive an IP address that resides on the locally-connected corporate network segment. This can be done using Dynamic Host Configuration Protocol (DHCP) or with a configured IP pool. On the client side, NC establishes its own network interface card (NIC) interface (virtual adapter) and modifies the routing table to tunnel traffic over that (SSL-encapsulated) network interface. The SA then terminates the SSL-encapsulated tunnel and forwards the traffic to Windows Terminal Services and Citrix Presentation servers directly. This behavior is exactly like a traditional IPSec VPN, and since the SA performs a "Proxy Address Resolution Protocol (ARP)" on behalf of all connected clients (IPs), it can support server-initiated requests for applications that need to open up connections/sockets to remote clients, such as with voice over IP (VoIP) applications.

The SA can perform comprehensive host checker functions to ensure an endpoint is free from threats like viruses and malware, and it helps minimize the risk of extending Layer 3 connectivity to potentially untrusted or unsafe PCs, such as those of users working from home or connecting from an airport kiosk PC. Note: Administrative rights are required to install NC unless using the Juniper Installer Service.

Note: All tools listed above are provisioned on the basis of a subscriber's role. The SA administrator can activate these tools and assign proper resources. However, the administrator needs to ensure SA subscribers have been properly mapped to a role in order to gain access to the tools.

Juniper Secure Access SSL VPN Configuration Scenarios

MSTS with the SA Embedded TS Client (Highly Recommended)

This method is the most powerful and provides the most seamless user experience. This functionality will auto-launch an existing Microsoft Terminal Services (MSTS) client on the end user's PC, if available, or else it will automatically provision the MSTS client on the fly. It can also optionally fall back to a Java RDP client, also delivered on the fly. Configuration on the SA is as follows:

- Under the Resource Profiles | Terminal Services, a new profile may be created.
- The Hostname/IP and port (usually 3389) must be specified here.
- Java Fall-Back Support is highly recommended as it enables non-Windows browsers to leverage the same bookmark and configuration, simplifying the end user experience and administrative configuration. This will be covered in the next section. Please review the Java configuration details and optionally implement that functionality in order to enable it here.
- An access control list (ACL) should also be set at this time.
- When you click Save and Continue, you will be prompted to select which Roles this resource applies to.
- After saving, a default bookmark will be created for this Resource. This bookmark will now be visible to all users of the selected Roles.
- The bookmark may now also be modified to enable specific MSTS features.
- Most of these feature options are part of MSTS and should be self-explanatory, however, we will explain a few specific ones here:
  - Authentication can be left blank or configured to use Single Sign-On (SSO) as follows:
  - Username: <USER> or DOMAIN\<USER> should be used
  - Variable Password: <PASSWORD>

Note: The "Variable Password" field should be used with SA Variables, and the "Password" field should be used if explicitly hard-coding a static username and password for all users.

- Path to Application (if desired; leave blank for the full desktop)
- Desktop Settings – These options can be disabled in order to conserve bandwidth and reduce MSTS latency.

MSTS Java Applet (and Fall-Back)

This method is the most innovative and supports the widest array of clients. Based in Java (and requiring only a JVM), this method invokes a Web page with an embedded Java RDP Client/Applet, obtained from http://properjavardp.sourceforge.net/. Originally designed to run as a command-line on Linux, MacOS or Windows, Juniper Technical Marketing actually took the original source code and created their own applet, complete with parameters. Over the next year, the Open Source project took on new life and now produces its own applet with all of the parameters needed to utilize all of the advanced MSTS functionality. Configuration is as follows:

- Under the Resource Profiles | Web, a new Profile should be created.
- The first step here is to select Hosted Java Applet from the drop-down menu and then edit the Applet List. These applet files can be obtained from the above URL or by contacting a Juniper Systems Engineer for the .zip
- During upload, the Applet .zip file should be uncompressed, and will then be expanded onto the SA system for use by the Resource Profile.
- In the New Resource Profile, the uploaded applet should now be listed. Select it.
- Proper ACLs should also be configured to explicitly allow this traffic.
- When you Save and Continue, select the Role(s) to which this resource should apply.
- Lastly, you will be prompted to create a bookmark as follows:
  - Enter a name and optional description for the Bookmark
  - Click Generate HTML
  - You will now see some HTML in the text area.
  - In most cases, this will suffice; however, there are some MSTS features you may want to use. Here is our recommended HTML for a standard Full Desktop MSTS session with SSO:

```
<html>
<head>
<title>ProperJavaRDP</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>
<body>
```

```
<h2>You may close this window when you are finished with your RDP
session...</h2>
  <applet code="net.propero.rdp.applet.RdpApplet.class"
          codebase="<< CODEBASE >>"
          archive="properJavaRDP-1.1.jar,log4j-java1.1.jar,properJavaRDP11-
1.1.jar,properJavaRDP12-1.1.jar,properJavaRDP13-1.1.jar,properJavaRDP14-
1.1.jar,java-getopt-1.0.11.jar"
          width="640" height="480"
          name="ProperJavaRDP" align="top">
    <param name="code" value="net.propero.rdp.applet.RdpApplet.class">
    <param name="codebase" value="<< CODEBASE >>">
    <param name="archive" value="properJavaRDP-1.1.jar,log4j-
java1.1.jar,properJavaRDP11-1.1.jar,properJavaRDP12-1.1.jar,properJavaRDP13-
1.1.jar,properJavaRDP14-1.1.jar,java-getopt-1.0.11.jar">
    <param name="cabbase" value="">
    <param name="name" value="ProperJavaRDP">
    <param name="geometry" value="1024x768">
    <param name="align" value="top">
    <param name="server" value="172.20.1.19">
    <param name="port" value="3389">
    <param name="username" value="<<USER>>">
    <param name="password" value="<<PASSWORD>>">
    <param name="domain" value="ACMEGIZMO">
    <param name="bpp" value="24">
  </applet>
</body>
</html>
```

- If Fall-Back is to be utilized, now that there is a working applet hosted on the SA, the MSTS Resource Profile may be configured to enable this applet. You may also customize the HTML further, and set whether this applet should be used all of the time or only as a Fall-Back mechanism, for example, if the SA is unable to download and install the Windows-based MSTS client on the fly.

Note: Using Hosted Java applets also requires a proper SA hostname to be configured and used by end users. This is configured under System | Network | Overview | Hostname. The configured hostname here must exactly match the hostname that end users type in when accessing the SSL VPN. Failure to do so (for most Java applets) will result in the applet not working properly or being unable to connect.

MSTS with WSAM

This method requires a preinstalled MSTS client on the end user PC and supports only Windows OSs. Configuration on the SA is as follows:

- Under the Role configuration, SAM must be enabled. This is done under Roles | <ROLE> | General.
- Under <ROLE> | SAM | Options, WSAM must be selected. Auto-launch may also be enabled at this time.
- Under <ROLE> | SAM | Applications, the settings for the MSTS client must be defined. This can be done in one of two ways, either by tunneling traffic from the application "mstsc.exe" or by tunneling all traffic from any application to the destination terminal server(s) (hostname or IP, port 3389).
- Proper ACLs should also be configured to explicitly allow this traffic through the SA. This is configured under Resource Policies | SAM | Access Control.
- There is no SA SSO (Single Sign-On) with this method.

MSTS with JSAM

This method also requires a preinstalled MSTS client on the end user PC. Although it supports Windows, Mac or Linux, it is also not very flexible because it requires a JVM, specific tunnels to be set up for each and every socket, and relies upon DNS or Host file change to redirect the client application to the loopback listener. Configuration on the SA is as follows:

- Under the Role configuration, SAM must be enabled. This is done under Roles | <ROLE> | General.
- Under <ROLE> | SAM | Options, JSAM must be selected. Auto-launch may also be enabled at this time.
- Under <ROLE> | SAM | Applications, the settings for the MSTS servers must be defined. This is done by adding the server hostname (not IP), port (usually 3389), and then a loopback IP, and port (again usually 3389). The loopback IP may be "*" which means the SA will auto-assign a loopback IP. This method is flexible because it will auto-pick the next available loopback IP if one is already being used; however, it requires the Host file to be changed (by JSAM at run-time). If DNS is to be used, a specific loopback IP should be specified (example, 127.0.0.5), and then the external DNS entry for the MSTS server should be pointed to that loopback IP. In most cases, this DNS change should not affect any internal users accessing the application internally, as they rely on internal corporate DNS to resolve these hostnames which is typically separate from external corporate DNS.
- Proper ACLs should also be configured to explicitly allow this traffic through the SA. This is configured under Resource Policies | SAM | Access Control.
- There is no SA SSO with this method.

MSTS with NC

This method also requires a preinstalled MSTS client but is the easiest method to configure. NC should be set up with a valid IP pool for the corporate network, and ACLs should be in place to allow this. No SA SSO is available for this method. With this method, users log in and launch NC; they then run Citrix just as if they were connected to the LAN.

Citrix

Citrix with the SA Embedded TS Client (Highly Recommended)

This method is the most powerful and provides the most seamless user experience. This functionality will auto-launch an existing Citrix client on the end user's PC, if available, or else it will automatically provision the Citrix client on the fly. It can also optionally fall back to the Java ICA ("jICA") client, also delivered on the fly. Configuration on the SA is as follows:

- Under the Resource Profiles | Terminal Services, a new profile may be created.
- Type: Citrix using default ICA.
  - A custom ICA file may also be uploaded to specify custom Citrix options.
  - Custom ICA files typically also contain the application to launch.
  - If a Custom ICA is not employed, the desktop will be launched.
- The Hostname/IP and port (usually 1494 and/or 2598) must be specified here.
- Java Fall-Back Support is highly recommended as it enables non-Windows browsers to leverage the same bookmark and configuration, simplifying the end user experience and administrative configuration. This will be covered in the next section. Please review the Java configuration details and optionally implement that functionality in order to enable it here.
- An ACL should also be set at this time.
- When you click Save and Continue, you will be prompted to select which Roles this resource applies to.
- After saving, a default bookmark will be created for this Resource. This bookmark will now be visible to all users of the selected Roles.
- The bookmark may now also be modified to enable specific Citrix features.
- Most of these feature options are part of Citrix, and should be self-explanatory, however we will explain a few specific ones here.

Authentication can be left blank or configured to use SSO as follows:

- Username: <USER> or DOMAIN\<USER> should be used.
- Variable Password: <PASSWORD>.

Note: The "Variable Password" field should be used with SA Variables, and the "Password" field should be used if explicitly hard-coding a static username and password for all users.

- Domain Credentials: This Citrix function allows the client to retrieve credentials from the Windows PC's Cached Credentials, and includes support for Smart and Common Access (CAC) Cards to authorize directly to the backend MetaFrame Presentation Server.
  - Path to Application (if not in the ICA file; leave blank for the full desktop).
  - Session Reliability improves the session quality using buffering. Auto-client Reconnect enables clients to seamlessly reconnect to a previous session, if they did not actually log off and end that session

Citrix Java Applet (and Fall-Back)

This method supports the widest array of clients. Based in Java (and requiring only a JVM), this method invokes a Web page with an embedded Java ICA Client/Applet, obtained from http://www.citrix.com/. Configuration is as follows:

- Under the Resource Profiles | Web, a new Profile should be created.
- The first step here is to select Hosted Java applet from the drop-down menu and then edit the Applet List. These applet files can be obtained from Citrix directly.
- During upload, the applet files should be uncompressed, and will then be expanded onto the SA system for use by the Resource Profile.
- In the New Resource Profile, the uploaded applet should now be listed. Select it.
- Proper ACLs should also be configured to explicitly allow this traffic.
- When you Save and Continue, select the Role(s) to which this resource should apply.
- Lastly, you will be prompted to create a bookmark as follows:
  - Enter a name and optional description for the Bookmark.
  - Click Generate HTML.
  - You will now see some HTML in the text area.

In most cases, this will suffice; however, there are some Citrix features you may want to configure. Here is our recommended HTML to launch a Notepad (application configured on MPS) session with SSO:

```
<html>
<head>
<title>JICA94 Applet</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>
<body>
  <applet code="com.citrix.JICA"
          codebase="<< CODEBASE >>"
          archive="cryptojN.jar,JICA-audioN.jar,JICA-browseN.jar,JICA-cdmN.
jar,JICA-clipboardN.jar,JICA-configN.jar,JICA-coreN.jar,JICAEngN.jar,JICA-
printerN.jar,JICA-seamlessN.jar,JICA-sicaN.jar,JICA-zlcN.jar,sslN.jar"
          width="1024" height="768"
          name="JICA94" align="top">
    <param name="code" value="com.citrix.JICA">
    <param name="codebase" value="<< CODEBASE >>">
    <param name="archive" value="cryptojN.jar,JICA-audioN.jar,JICA-
browseN.jar,JICA-cdmN.jar,JICA-clipboardN.jar,JICA-configN.jar,JICA-coreN.
jar,JICAEngN.jar,JICA-printerN.jar,JICA-seamlessN.jar,JICA-sicaN.jar,JICA-
zlcN.jar,sslN.jar">
    <param name="cabbase" value="">
    <param name="name" value="JICA94">
    <param name="width" value="1024">
    <param name="height" value="768">
    <param name="align" value="top">
    <param name="HTTPBrowserAddress" value="172.20.1.19">
    <param name="Address" value="Notepad">
    <param name="InitialProgram" value="#Notepad">
    <param name="Username" value="<< USERNAME >>">
    <param name="Password" value="<< PASSWORD >>">
    <param name="EncryptionLevel" value="1">
```

```
    <param name="BrowserProtocol" value="HTTPonTCP">
    <param name="Start" value="Auto">
    <param name="End" value="https://SA.com">
    <param name="Compress" value="on">
    <param name="MaximumCompression" value="on">
    <param name="TWIMode" value="on">
  </applet>
</body>
</html>
```

- If Fall-Back is to be utilized, now that there is a working applet hosted on the SA, the Citrix Resource Profile may be configured to enable this applet. You may also customize the HTML further, and set whether this applet should be used all of the time or only as a Fall-Back mechanism, for example, if the SA is unable to download and install the Windows-based Citrix client on the fly.

Note: Using Hosted Java applets also requires a proper SA hostname to be configured and used by end users. This is configured under System | Network | Overview | Hostname. The configured hostname here must exactly match the hostname that end users type in when accessing the SSL VPN. Failure to do so (for most Java applets) will result in the applet not working properly or being unable to connect.

Citrix with WSAM

This method requires a preinstalled Citrix client on the end user PC and supports only Windows OSs. Configuration on the SA is as follows:

- Under the Role configuration, SAM must be enabled. This is done under Roles | <ROLE> | General.
- Under <ROLE> | SAM | Options, WSAM must be selected. Auto-launch may also be enabled at this time.
- Under <ROLE> | SAM | Applications, the settings for the Citrix client must be defined. This can be done in one of two ways, either by tunneling traffic from the application "wfica32.exe" (ICA Web Client) or "pn.exe" (Program Neighborhood), or else by tunneling all traffic from any application to the destination terminal server(s) (hostname or IP, port 1494 and most likely also port 2598, depending on your MetaFrame port configuration).
- Proper ACLs should also be configured to explicitly allow this traffic through the SA. This is configured under Resource Policies | SAM | Access Control.
- The best practice for this scenario, if implemented, is to then let users access Citrix Web Interface (formerly "NFuse") through the rewriter. In this way, SSO can be enabled making this solution more seamless to the end user. When accessing Web Interface through the SA rewriter, the cacheable bit must be set on all .ica files. This is configured under Resource Policies | Web | Caching. The policy must enable smart-caching for all .ica files (under the Citrix Web Interface/Nfuse Web server URL).
- SSO (Single Sign-On) to Web Interface/NFuse can be configured as follows (this assumes users access the Web Interface/NFuse Portal through the SA rewriter):
- Go to Resource Policies | Web | SSO.

- Create a New Policy with the following settings (Note: These settings may differ based on Citrix Web Interface/NFuse Portal Authentication settings.):
  - Resource: http://<wi.hostname.or.ip:port/citrix/metaframe
  - ENABLED: Perform the POST defined below
  - POST Details:
  - http://wi.hostname.or.ip:port/citrix/metaframe/default/login.aspx
  - state = state = LOGIN
  - logintype = LoginType = Explicit
  - domain = domain = DOMAIN (use the actual DOMAIN here)
  - Username = user = <USER>
  - Password = password = <PASSWORD>

Note: To determine actual POST URL and form field requirements, browse to the Web Interface/NFuse portal with a Web browser and view the HTML source of the login page. Look for the <form action = > to see what the action URL is, and look for <input name = > to see the name of the fields required.

Citrix Web Interface ("NFuse") with JSAM

This method is very involved and includes configuration of JSAM sockets (one for each Citrix server and port), SSO, JSAM Auto-Launch, and Caching policies. This method is not a recommended Best Practice, although it is supported by the SA.

Citrix Web Interface ("NFuse") with Java ICA

This scenario has a user accessing the Web Interface portal through the SA rewriter, and launching the jICA (Java ICA) client. The SA will dynamically rewrite the client on the fly and secure the jICA communications with no additional software downloads. While this is the least intrusive method, jICA does not have all of the seamless functionality some Citrix users have grown to love. That is why jICA is usually recommended as a fall-back method, as outlined previously. Configuration on the SA is as follows:

- Create a new Profile under Resource Profiles | Web
- Type: "Citrix Web Interface/jICA"
- Web Interface URL: Hostname or IP of the Web Interface/NFuse Portal
  - ENABLED: Java ICA Client with NFuse
  - Web Interface version: Depends on actual Web Interface/NFuse version
  - MetaFrame servers: List the Citrix MetaFrame Presentation Servers here and then click Add
  - Access Control: Be sure the resource is properly listed here
  - [X] SSO (Single Sign-On) to Web Interface/NFuse can be configured as follows:
    - Resource: http://<wi.hostname.or.ip:port/citrix/metaframe
    - Post URL: http://wi.hostname.or.ip:port/citrix/metaframe/default/login.aspx
    - State = state = LOGIN
    - LoginType = LoginType = Explicit
    - Domain = domain = DOMAIN (use the actual DOMAIN here)
    - Username = user = <USER>
    - Password = password = <PASSWORD>
    - [X] Send the following data as request headers
    - Resource: http://wi.hostname.or.ip:port/*
    - Headername: Cookie | WIClientInfo = icaClientAvailable = true

Note: To determine actual POST URL and form field requirements, browse to the Web Interface/NFuse portal with a Web browser and view the HTML source of the login page. Look for the <form action=> to see what the action URL is, and look for <input name=> to see the name of the fields required.

- When you click Save and Continue, you will be prompted to select which Roles this resource applies to.
- After saving, a default bookmark will be created for this Resource. This bookmark will now be visible to all users of the selected Roles.

Citrix with NC

This method also requires a preinstalled Citrix client, but it is easiest to configure. NC should be set up with a valid IP pool for the corporate network, and ACLs should be in place to allow this. No SA SSO is available for this method. With this method, users can launch NC and then run the Citrix Web Client or Program Neighborhood just as if they were connected to the LAN.

Miscellaneous Tips and Tricks

- SSO versus Load Balancing versus Seamless Windows
  - SSO is only available with Web Interface/NFuse, and the Web Client invoked via ActiveX. It can also be configured for use in HTML which launches a jICA applet.
  - Seamless Windows and MPS Load Balancing are only available with the Citrix Program Neighborhood clients, although Web Interface/NFuse does permit Load Balancing based on MPS decisions.
- Launching the CTS client from a Web Portal
  - Many customers like the CTS client method, but do not utilize the SA portal/index page. In this case, to still leverage the CTS client functionality, Web Portal Administrators can simply create a hyperlink based on the following URL example: https://SA/dana/term/winlaunchterm.cgi?host=1.2.3.4&type=Windows&serverPort=3389&screenSize=fullScreen *(Refer to the Admin Guide for more details).*
- Code-Signing Certificates
  - These are required if the SA re-signs a Java applet which was previously signed by a Trusted Root CA, such as the Citrix jICA applet. Administrators should upload their own (Trusted Root CA) code-signing certificate for the SA to use in this process.
- Troubleshooting
  - ActiveX Download and Install are not working.
    - If using an Uploaded ICA file, be sure to enable “Users can add sessions” under <ROLE> | Terminal Services | Options.
    - If ActiveX is disabled, be sure to configure a Java Fall-Back Applet for use.
  - Policy Tracing can be used to nail down SSO issues. Specifically, it will tell the Administrator when an SSO policy is triggered, and when it is not (but perhaps should have been).

Summary

As shown, this best practices document helps users deploy Microsoft Terminal Services and Citrix applications for secure remote access with Juniper’s industry-leading SSL VPN appliances. It demonstrates the secure access methods that can be used with Microsoft Terminal Services and Citrix as well as other guidelines to effectively deploy these applications for secure remote access.

About Juniper Networks

Juniper Networks, Inc. is the leader in high-performance networking. Juniper offers a high-performance network infrastructure that creates a responsive and trusted environment for accelerating the deployment of services and applications over a single network. This fuels high-performance businesses. Additional information can be found at www.juniper.net.

CORPORATE HEADQUARTERS
AND SALES HEADQUARTERS FOR
NORTH AND SOUTH AMERICA
Juniper Networks, Inc.
1194 North Mathilda Avenue
Sunnyvale, CA 94089 USA
Phone: 888.JUNIPER (888.586.4737)
or 408.745.2000
Fax: 408.745.2100
www.juniper.net

EUROPE, MIDDLE EAST, AFRICA
REGIONAL SALES HEADQUARTERS
Juniper Networks (UK) Limited
Building 1
Aviator Park
Station Road
Addlestone
Surrey, KT15 2PG, U.K.
Phone: 44.(0).1372.385500
Fax: 44.(0).1372.385501

EAST COAST OFFICE
Juniper Networks, Inc.
10 Technology Park Drive
Westford, MA 01886-3146 USA
Phone: 978.589.5800
Fax: 978.589.0800

ASIA PACIFIC REGIONAL SALES HEADQUARTERS
Juniper Networks (Hong Kong) Ltd.
26/F, Cityplaza One
1111 King's Road
Taikoo Shing, Hong Kong
Phone: 852.2332.3636
Fax: 852.2574.7803

To purchase Juniper Networks solutions, please contact your Juniper Networks sales representative at 1-866-298-6428 or authorized reseller.