TARANTELLA

Set up a Home Secure Global Desktop Enterprise Edition Remote Access Server

Purpose

This document provides a step-by-step walk-through of how I set up my home network and Tarantella Secure Global Desktop Enterprise Edition (EE) to allow me to access my home network and computers while I'm traveling. This walk-through is for illustrative purposes only. Any of the steps, policies, or procedures I used may not be appropriate to your environment – common sense should always prevail.

Introduction

Tarantella's Secure Global Desktop Enterprise Edition allows remote users to access and run applications remotely using only a Java-enabled web-browser. That is, EE lets me access the computers in my home network and run applications on them, including X11 and Windows applications. I can run these applications from almost anywhere over almost any type of connection, securely. When I'm away from home and need to be able to access applications and data on my computers at home, EE allows me to do so.

Of course, there are other utilities that provide varying degrees of remote access capabilities, but EE provides advantages over many such utilities, such as:

Firewall Traversal – This features tunnels all network traffic (https + Tarantella's AIP protocol) over a single port, TCP port 443. This means not only do I not have to open any ports on my home router, but more importantly, I don't have to open ports on any intervening firewalls (because, for the most part, I can't). If I'm at work or at a client's site, it's virtually certain that the SSH port, for example, is closed, while the 443 port is generally open –because it's usually considered a safe and well-known protocol. In addition, EE can negotiate client-side proxy servers, while most other utilities cannot.

Performance – The Adaptive Internet Protocol (AIP) is a proprietary protocol optimized to run graphical applications over low-speed network links. AIP automatically determines network conditions such as bandwidth and latency, and tunes itself accordingly for best performance.

SSL Security – connections to my home machines are encrypted via SSL to prevent sniffing of my private data and passwords.

No Fat Client installation – all I need is a Java-enabled web browser. This means I can use almost any PC to access my remote network.

File transfer / Printing – EE allows me to retrieve remote files and print remote documents on local printers.

Multiple Authentication Providers – EE allows you to use your existing authentication provider(s). By default, UNIX/Linux passwd is used, but LDAP, Active Directory, NT Domain, RSA SecurID and others can be used to act as EE authenticators, giving you greater control and flexibility.

Single Sign-on – EE can be used as a centralized Single Sign-on (SSO) system, so you need not keep track of your different user-id's and passwords across your various systems.

Web API – Enterprise Edition 4 uses SOAP/XML calls to interact with the server and uses Java Server Pages (JSP's) to provide the user interface. This gives me a platform for learning to code JSP's.

If you'd like to get a feel for what Secure Global Desktop does, log in to the public demonstration site at:

https://eedemo.tarantella.com

Click "Log in"

And at the login box, click the Log In button. On the left side of the resultant page, called the webtop, is a list of demo applications you can interact with.

My Environment

To give a bit of background, my home is serviced by a 768K/128K ADSL line and I use a Linksys WRT54G wireless router as a NAT device to allow my computers to share the ADSL connection. The router is assigned an IP address via DHCP by my service provider, so my WAN IP address changes periodically.

I have several computers in my home network running several flavors of Linux, as well as Windows XP and Windows Server 2003. These computers are connected via both wired and wireless connection. I run a local DNS server for local name services and most of my computers are wired to my local hub. I use the built-in DHCP service on my router to assign IP addresses to wireless clients.

Because my router is assigned a dynamic IP address, accessing my home network via IP address won't work. I'd have no way, when away from home, to know the IP address of my router if it changed since the last time I made note of it. In addition, since I'll be using SSL, a variable IP address requires I use a fixed DNS name to access my network. The Linksys router I have supports Dynamic DNS, which correlates a fixed DNS name with the IP addresses your provider assigns you even though it may periodically change. This way, you can access your home network with a consistent DNS name without having to keep up with IP address changes.

To install EE, I needed to choose one of my Linux servers to run the EE software and act as the gateway into my network. EE runs on this machine and proxies all application connections to my other systems. This means all of my internal systems must be reachable by this host. The system I chose for this function in my network is of a modest configuration: an

older PC running SUSE LINUX Standard Server 8, with a 450MHz Pentium II, 256MB RAM, 4GB hard disk, and a network card.

In this example, I have a small internal network (mydomain.com) with name resolution services setup for resolving local addresses.

AN IMPORTANT NOTE REGARDING HOSTNAMES

The internal hostname of your EE gateway server is called your Peer DNS Name – this is the internal hostname other servers in your network will know you by. In my example, this name is "myhost.mydomain.com". The DNS name your EE gateway host is known by from outside your network is your External DNS Name, and in the example is "myname.homelinux.net". Substitute your own hostnames for these in the following examples.

Getting Started:

Before you begin, you should read the installation documentation for EE. The EE 4 documentation can be found here:

Installation Guide:
http://www.tarantella.com/support/documentation/sgd/ee/4.0/install.html

Getting Started:
http://www.tarantella.com/support/documentation/sgd/ee/4.0/start.html

Other resources include:

Web Services Documentation:
http://www.tarantella.com/support/documentation/sgd/ee/4.0/webservices/

Support Newsgroup:
news://pubnews.tarantella.com

1

Obtain a DNS Address for Your Network

As mentioned above, I wanted to assign a permanent DNS name that I could use to access my network. Since my router supports DynDNS, I elected to use their service and went through the following steps. The Linksys router also supports the TZO service, http://www.tzo.com, and would be setup similarly.

From my home network I went to http://ww.dyndns.org/services/dyndns to register a hostname that I could associate with my home network. I first created an account for myself and then went through the Add Host procedure. This creates a hostname like "myname.homelinux.net" (DynDNS has 43 domains under which you can register your hostname, including the homelinux.net domain used in the example). This page detects your currently-assigned IP address Confirm the address is correct and add the hostname you've chosen.

2

Configure Router to Update DDNS Service

The way Dynamic DNS works is that when the WAN IP address of my router changes the Dynamic DNS service provider is notified of the change The Linksys router has a built-in DDNS client to perform this notification. Alternatively, if your router doesn't support this function, you can obtain a DDNS software client to install on a machine on your network to perform these updates. See:

http://www.dyndns.org/services/dyndns/clients.html

Log in to Linksys router as admin and set DDNS client information.

(1) Older versions of the Linksys firmware may be "embargoed" by DynDNS, due to a misbehavior in the client code – you may need to update the firmware on your router for DynDNS to accept updates from your router.

Go to Setup tab
Go to DDNS tab
DDNS Service – on the pull-down select "DynDNS.org" (or TZO)
User Name – enter username of your DDNS account created in step *(1)*
Password – enter password of your DDNS account created in step *(1)*
Host Name – enter your DDNS URL as created in step *(1)*, e.g. "myname.homelinux.net".

Click Save Settings.

3

Configure Router to Port Forward 'https' Traffic

Go to Applications and Gaming Tab
Go to Port Range Forward Tab
Application – Enter "https"
Start – 443
End – 443
Protocol – TCP
IP Address – enter IP address of your EE server

Click Save Settings.

4

Turn off /remove Apache Server on Linux Box

If you have a web server installed on your EE server system turn off or remove it. EE includes a pre-configured Apache web server and Tomcat server. To keep things simple, use the web server that comes with the EE distribution. You can use your own web server if you prefer and the EE documentation discusses how to do this, but to keep things simple, I just use the one that comes with EE.

5

Create "Service" Accounts

Before installing EE you need to create a user & group to own the Apache web server files. The name of this account is fixed as "ttaserv" with a group name of "ttaserv". The login-id and group-id are not important. For example:

```
# groupadd ttaserv
# useradd -g ttaserv -d /opt/tarantella ttaserv
```

For EE 4 and above you need create a second ID called “ttasys”. This id is a non-privileged user that the majority of EE processes will run as. This must be a member of the same “ttaserv” group created above. For example:

```
#  useradd -g ttaserv -d /opt/tarantella ttasys
```

6

Install EE Base Pack on Linux Server

Run the EE installation script:

```
# sh <path>/ttai3li.shx
```

The installation script will verify the archive, and then ask you to accept the terms of the license agreement. It will then display something like the following:

```
------------------------------------------------------
Setting up Tarantella Secure Global Desktop Enterprise
Edition
------------------------------------------------------
Secure Global Desktop Setup recommends you use the
following settings:

Installation type = install 4.00.903
Installation directory = /opt/tarantella
Peer DNS name = myhost.mydomain.com
License mode = Evaluation (30-day limit)
HTTP port = 80 [not currently in use]
Archive logs every week? = yes (Sunday 03:00 hours)

Are these settings OK?
  Y - Yes, install using these settings
  N - No, tell me more about the options and let me
change the settings
  Q - Quit now

OK to use these settings? [Y]
```

The most important setting here is the Peer DNS name. This should be the fully-qualified domain name of your server and should be resolvable in the local network. Check to be sure this is correct.

Also, it would be a good time to mention that some Linux distributions will assign the hostname (myhost.mydomain.com) as an alias to the loopback address (127.0.0.1). This will cause problems and should be corrected before you begin.

Once you’ve accepted the installation settings the installation process will run to completion. It’s probably a good time to get some coffee.

7 Install EE Security Pack

The EE security pack provides SSL encryption of the AIP data connections between the client and the EE server. Install in the usual way (note that on the CD distribution, this is found in the "tsp" subdirectory).

```
# sh <path>/tspi3li.shx
```

No interaction is required for the installation of this component.

8 Enable SSL

Enabling SSL connections will encrypt the traffic between your roaming location and your home network and is mandatory for EE Firewall Traversal to function. Recall that EE uses two connections between a client browser and an EE server – the http connection and the AIP connection. To SSL-encrypt these two protocols, two X.509 certificates are required – one for the web server (https) and one for the EE Security Pack (AIP). In practice, these certificates are usually shared between the web server and the EE Security Pack. The Apache web server supplied with EE is pre-configured to use the same server certificate as the EE Security Pack.

Server certificates are generally purchased from a Certificate Authority (CA) such as Verisign or Thawte. However, for most casual home installations, this is probably unduly expensive. It's possible to use self-signed X.509 server certificates to enable https connections. Such certificates are typically not considered truly secure as they are not issued by a trusted CA, but many would consider them to be secure enough for this type of project. I'll leave this for you to decide. The following section describes two ways to use self-signed server certificates in Tarantella.

```
# tarantella security certrequest –country US –state
state –orgname home
```

This command displays some output, then:

```
The hostname to be used in the certificate request is
myhost.mydomain.com.

Do you want to use this hostname? [yes] n
Enter the fully qualified hostname [ myhost.mydomain.
com] myname.homelinux.net

[ a number of informational lines will be shown,
then ]

Create CSR Now?  [yes] <Enter>
```

The generated Certificate Signing Request is displayed between the two lines marked:

```
------CUT HERE------
```

If you wish to have this signed by a CA, take this CSR and send it to the CA in whatever manner they specify. If you wish to use self-signed certificates, you can use the EE built-in command to generate a self-signed certificate. Notice that this is an unsupported and undocumented feature, please use

at your own risk.

```
# tarantella security selfsign
```

which results in the output:

```
A self-signed certificate has been generated and in-
stalled.
To enable SSL connections, use 'tarantella security
start'.

Users will be prompted to trust this certificate.  To
stop the prompts, install the certificate as the custom
Certificate Authority:
tarantella security customca –rootfile /opt/tarantella/
var/tsp/cert.pem

IMPORTANT:  Self-signed certificates should be used for
TEST PURPOSES only.
```

When using self-signed certificates, you might wish to install your CA root certificate to prevent being prompted to trust the server certificate, as mentioned above. The command to do this is:

```
# tarantella security customca –rootfile /opt/tarantel-
la/var/tsp/cert.pem
```

Now that the certificate is installed, start SSL connections with the following command:

```
# tarantella security start
```

to which the system responds:

```
- myhost.mydomain:  Enabled secure connections
```

9

Enable EE Firewall Forwarding

In the following section we're going to make several changes to the Apache and EE configuration to support the Firewall Traversal feature. These steps include:

• Modify the Apache ServerName associated with the 127.0.0.1 Virtual Host to match your external DNS name.

• Modify EE so that it communicates its secure AIP connections on port 443.

• Modify EE so that it knows to setup an SSL listener on port 443. This listener process, "ttassld", binds to the network interface on port 443, receiving all https packets from the network. If inbound packets are destined for the web server, the packets will be forwarded to the 127.0.0.1:443 interface (where Apache is bound) and if the packets are destined for EE they will be handled appropriately.

• Modify the EE External DNS name to the resolvable DNS name. When clients connect to a Tarantella server this is the name they are instructed to connect to. By default, this is the Peer DNS Name. However, for external clients, this name is unresolvable.

9.1 Edit the Apache Configuration File

Locate the Apache configuration file, located in this release in:

```
/opt/tarantella/webserver/apache/1.3.297_mod_ssl-
2.8.16_openssl-0.9.7d_jk1.2.5/conf/httpd.conf
```

and make the following changes (note that this directory name may vary):

9.1.a Change Apache Listen Port

Locate the line that reads:

```
Listen 443
```

and change this to read:

```
Listen 127.0.0.1:443
```

9.1.b Change Virtual Host Container

Locate the line that reads:

```
<VirtualHost _default_:443>
```

and change this to read:

```
<VirtualHost 127.0.0.1:443>
```

9.1.c Change Virtual Host Container

Several lines below, in this same "VirtualHost" section, locate the line

```
ServerName myhost.mydomain.com
```

and change to read:

```
ServerName myname.homelinux.net
```

Save changes and exit.

9.2 Change EE Settings

In this section, we make several changes to Enterprise Edition default settings, using the Tarantella Arraymanager. The Arraymanager is a visual tool used to modify system settings in EE. The Objectmanager is used to edit applications, application servers, and so on.

From an XSession on your server, run the following command (note the tarantella command is located in `/opt/tarantella/bin` – you may wish to add this to your path, or create a shell alias):

```
# tarantella arraymanager &
```

An authentication dialog box will display; enter “administrator” for the user-id, and the root password as the password.

Select “Array” and click the “Properties” button.
Change “Port Numbers, Encrypted Connections” from 5307 to 443.
Select (left-click) <Servername> object, where <Servername> is the Peer DNS name (“myhost.mydomain.com” in this example).

- Edit the field in the right-hand pane so that it reads:

```
*:myname.homelinux.net
```

Expand the <Servername> object, so that a hierarchy of objects is displayed.
- select “Security”, and click “Properties

- enter Firewall Forwarding URL as:

```
https://127.0.0.1:443
```

Click “Apply” then “Exit”.

10 Restart Services

```
# tarantella webserver restart -ssl
# tarantella restart
```

11 Test to Ensure You can Log On

From a supported web browser, type the following URL:

https://myname.homelinux.net

If everything is configured correctly, you should see an EE landing page with various options. If you do not see this page check to ensure your DNS resolver is returning the proper IP address for your network name. If it is, check to ensure that you’ve established port forwarding properly on your router so that TCP packets on port 443 are forwarded to the proper IP address.

From the landing page, click “Log In”.

Log in as “administrator”, using the root password for that system – note that the “Administrator” account is an alias for “root” provided by a “Person Object” in the EE data store.
If all was successful, you should see a list of default applications listed in the left-hand pane of this page. Click on any application to launch.

Note: If telnet is not installed on your system, EE will configure the default applications to use ssh as the connection mechanism. To run X-Windows applications, the ssh configuration parameter “ForwardX11” must be set to “yes”. Some distributions have a default setting of “no”. Usually this is set in file “`/etc/ssh/ssh_config`”.

12

Customize Your Installation

Next, you'll want to add applications, servers, optional services such as Client Drive Mapping, and so on. The best source for this type of information is found here:

http://www.tarantella.com/support/documentation/sgd/ee/4.0/start.html

This will guide you through publishing applications, adding users, etc.

Another useful resource is the Tarantella public support newsgroups, which can be found at:

news://pubnews.tarantella.com

EE is designed to support Windows applications, most commonly through the use of the Terminal Services component of Windows 2000 Advanced Server or Windows Server 2003 (though other mechanisms exist). This service provides the RDP network protocol so that remote users can connect to such a server and run remote Windows sessions.

To enable the built-in RDP service in Windows, do the following:

Windows Application Notes

Windows 2000

Install Terminal Services in Remote Administration mode, by running:
Control Panel > Add/Remove Programs, "Windows Components", select "Terminal Services", and check to ensure that "Remote Administration Mode" is selected.

Windows XP

Enable Remote Assistance Connections, by running:
Control Panel > System, then selecting the "Remote" tab, and ensuring that "Allow users to connect remotely to this computer" is selected.

Windows Server 2003

Enable Remote Assistance Connections, by running:
Control Panel > System, then selecting the "Remote" tab, and ensuring that "Allow users to connect remotely to this computer" is selected.

Other Windows Versions

For accessing computers that are running a version of Windows that doesn't support RDP, one simple solution is installing a VNC server on the desktop and a VNC client on your EE server. There's a significant loss of functionality, with no printer or drive mapping, and performance is, well, limited. But it works.

Create an X-Windows application that launches the VNC client. For example:

Application name: VNC Snoopy
Application Pathname: /usr/local/bin/tightvnc
Command Line Parameters: snoopy.mydomain.com
Display Using: Client Window Management