Network Security Toolkit FAQ

Ronald Henderson

rhenderson@unifiedholdings.com

Paul Blankenbaker

paul@mekwin.com

Network Security Toolkit FAQ

by Ronald Henderson

by Paul Blankenbaker

Where to find answers to frequently asks questions related to the use of the Network Security Toolkit.

Table of Contents

Chapter 1. General Information

What is the Network Security Toolkit?

The Network Security Toolkit is a bootable CDROM based on `Red Hat Linux 9`[1] with additional network security tools installed and ready to run. Before each release, the developers strive to make certain that the released ISO image contains the latest security patches and fixes for `Red Hat Linux 9`[2] and the additional network security tools.

What are the System Requirements?

The system requirements for running the Network Security Toolkit include following:

- A Pentium class CPU (Pentium II 266MHz or above).
- At least 128MB of RAM.
- A CDROM drive and BIOS capable of booting from a CD. It should be noted that the CDROM drive is the bottle neck in the system. If your system has an old `4X` CDROM drive, you'll find spending $20 USD on a new `52X` drive will vastly improve your experience.
- A `Red Hat Linux 9`[3] supported Ethernet Adapter

It is important to note that a hard disk is not required. And though the Network Security Toolkit has the capability to access hard disks, it does not do so by default.

In addition to the minimum requirements, the following equipment is supported:

- A video adapter and monitor.
- A keyboard.
- A standard serial port.
- Multiple Ethernet Adapters.
- A mouse.
- Disk drives.
- USB devices.

What is the Recommended System?

If you are like Ron (heavily involved in setting up and Enterprise Network Security Architectures in the real world), you'll probably find yourself wanting to build custom Network Security Toolkit systems. Most likely, you'll be able to use most components that you come across. However, if you'd rather be safe, you may want to make use of Ron's current entry level parts list. If nothing else it serves as a good starting point for a suggested component list to get NST up and going. The following table is a breakdown of Ron's entry level parts preference and approximate prices as of 2004-Mar-21 (all amounts are in USD). A great resource for these components can be purchased at InternetiShop[4]. This particular system configuration includes 3 10/100 network interfaces. It is important to stress that a CDROM with a fast read speed (52x) is beneficial to experience the best results from NST.

Table 1-1. Ron's Entry Level NST Component List

Component	Specification	Price
Case	Tower Configuration	$20.00
Motherboard	Intel Socket 478 (Includes USB and a 10/100 NIC)	$55.00
CPU	2.0GHz Celeron	$75.00
Memory	DDR 266 512MB	$85.00
CDROM	52x Speed	$20.00
10/100 NICs	2 Additional 10/100 NICs	$18.00
Total:		$273.00

Where do I go for information?

Information about the Network Security Toolkit can be found at the project's main web site: http://www.networksecuritytoolkit.org/.

Where can I get the Network Security Toolkit?

You can find the latest ISO images of the Network Security Toolkit at the project's SourceForge[6] site: http://sourceforge.net/projects/nst.

What is the password?

For those of you that simply download the ISO, burn it and boot it, the quick answer to this question is: "You get to set/choose the password for the `root` user each time you boot the Network Security Toolkit CD." After doing so you will be able to log in as `root` with the password that you specified.

> **Note:** For release `1.0.4` and earlier, you log in as `root` with the password of `nst@2003` initially. You should immediately run the **nstpasswd** command to change this to a different value.

Addressing the issue of the default password for the `root` user has been time consuming. This sounds like such a simple problem, but has caused Ron and I headaches in coming up with a proper balance between security and convenience. Here are some of the issues we need to deal with:

- There are people who use the Network Security Toolkit to learn about both network security and the Linux operating system. These people may inadvertantly boot up a Network Security Toolkit probe connected to the Internet without running the **nstpasswd** command to reset the default password. While this may be convenient, it leads to an insecure system which others could easily gain access to.
- Some Network Security Toolkit probe systems do not have keyboards or displays attached (this is actually our targeted platform for a running Network Security Toolkit probe). This makes running the **nstpasswd** command inconvenient as one will either need to connect a serial terminal or make a network **ssh** connection in order to change the passwords.
- Those who build their own ISO image from the source can set a custom password so that their Network Security Toolkit probe will be secure at boot time. This will

be inconvenient (as building a Network Security Toolkit ISO image from scratch takes a considerable amount of time and effort.

- Sometimes, one might choose one of the boot options (like `base`) which doesn't load in the utilities of the CD. In this situation, we can't run the **nstpasswd** command as all of the files that need to be updated have not yet been loaded into RAM. However, in this situation, the network functionality of the Network Security Toolkit system is not running, so the system is still secure at this point (as long as someone doesn't have physical access to the system, they can't make use of it).

We want to provide a ISO image that is easy for everyone to burn and use, but at the same time we don't like the idea of thousands of Network Security Toolkit probes being connected to the Internet with open access for anyone who knows to log in as `root` with a password of `nst@2003`.

So, starting with release `1.0.5` of the Network Security Toolkit, we've decided that we will force the running of the **nstpasswd** command for everyone who simply downloads the ISO image, burns it and boots it. This will add some inconvenience, but will enforce a better form of security than simply "hoping" that everyone remembers to run **nstpasswd**.

> **Note:** We only force you to set the password if you select one of the boot options that loads the utilities off of the CD. If you select the `base` option at the boot screen, we will not force you to set a new password and fall back to the default of `nst@2003` for the `root` user. A system booted in this fashion is secure as network functions are not enabled.

This will be inconvenient for those that want to use a system which doesn't have a keyboard or display (its awfully tough to type in a new password without a keyboard).

We have found that it is possible for us to modify the contents of a ISO image prior to burning a CD. So, with each new release, we will provide a **nstisopasswd-1.0.6.bash** script (we may provide a Windows or Java utility to do this as well at some point in the future) which you can use to set the password in the ISO image.

If you use the **nstisopasswd-1.0.6.bash** script on the `nst-1.0.6.iso` file and then burn it to a CD, you won't be forced to set the password each time you boot the CD and your Network Security Toolkit probe will be secure at boot time.

How can I Avoid Setting the Password Each Time It Boots?

If you find it tedious to specify a new password each time you boot the Network Security Toolkit, there are two possible solutions.

- Use the **nstisopasswd-1.0.5.bash** script (you can find a link to this script on the manifest[8] page associated with the Network Security Toolkit distribution you downloaded (make sure you use the script which matches the version number you downloaded). The following demonstrates how one can change the password to `NEWPASS`:

```
[pkb@localhost pkb]$ gunzip nst-1.0.5.iso.gz
[pkb@localhost pkb]$ ./nstisopasswd-1.0.5.bash NEWPASS nst-1.0.5.iso
[pkb@localhost pkb]$
```

Figure 1-1. Changing The Password In The ISO

> **Note:** This feature started with release `1.0.5` of the Network Security Toolkit. If you have an earlier version of the Network Security Toolkit ISO, you will NOT be able to do this.

- If you build your own custom Network Security Toolkit ISO image from the source files, you can configure your own custom password by including `--passwd NEWPASS` when you invoke the **configure** command.

What is the IP Address?

By default, the Network Security Toolkit uses DHCP to determine its IP address. If your machine has a keyboard and monitor attached or you are able to connect to it via the serial port, you can use the following to determine your IP address:

```
[root@probe root]# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:01:02:68:27:12
          inet addr:192.168.0.11  Bcast:192.168.0.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:119 errors:0 dropped:0 overruns:1 frame:0
          TX packets:76 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          RX bytes:19117 (18.6 Kb)  TX bytes:13105 (12.7 Kb)
          Interrupt:3 Base address:0x9000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)

[root@probe root]#
```

If you do not have a keyboard and mouse attached or serial access, the determination of the IP address assigned becomes a more difficult task. You will need to locate the logs of your DHCP server to determine what address was assigned.

Alternatively, if you have access to port scanning software (like `nmap`[9]), you can scan your network for ports `22`, `80` and `443` as shown in the following:

```
[pkb@salsa pkb]$ nmap -p 22,80,443 192.168.12.0/24
Starting nmap V. 3.00 ( www.insecure.org/nmap/ )
Interesting ports on nacho.linux.bogus (192.168.12.1):
Port       State       Service
22/tcp     open        ssh
80/tcp     open        http
443/tcp    open        https

Interesting ports on rice.linux.bogus (192.168.12.2):
Port       State       Service
22/tcp     open        ssh
80/tcp     open        http
443/tcp    open        https

All 3 scanned ports on tamale.linux.bogus (192.168.12.5) are: closed

Interesting ports on salsa.linux.bogus (192.168.12.7):
Port       State       Service
22/tcp     open        ssh
80/tcp     open        http
443/tcp    open        https
```

```
All 3 scanned ports on flan.linux.bogus (192.168.12.8) are: closed

Interesting ports on mole.linux.bogus (192.168.12.9):
(The 1 port scanned but not shown below is in state: closed)
Port       State       Service
22/tcp     open        ssh
443/tcp    open        https

Nmap run completed -- 256 IP addresses (6 hosts up) scanned in 4 seconds

[pkb@salsa pkb]$
```

Figure 1-2. Using nmap To Locate NST

Since the Network Security Toolkit probe only provides services on ports `22` and `443`, I can look at the above output and see that `192.168.12.9` is the only system meeting these restrictions (the key being the absence of port `80`).

If you want to be really crafty, you can use the above information to do an even better job of locating the Network Security Toolkit probes on your network. The following command sequence, looks for the string `networksecuritytoolkit` in the `index.html` page for each system found by **nmap** that looks like it might be a Network Security Toolkit probe (has port `443` open, but port `80` is closed).

```
[pkb@salsa pkb]$ nmap -oX - -p 80,443 192.168.1.0/24 |\
awk -F\" -- '{ if ( $1 == "<address addr=" ) { ip=$2; } \
if ( $4 == "80" ) ip=""; if ( ip != "" && $4 == "443" && \
system(sprintf("wget -O - https://%s/index.html 2>/dev/null | \
grep networksecuritytoolkit > /dev/null",ip)) == 0) \
{ printf("%s\n",ip); } }'
192.168.1.3
[pkb@salsa pkb]$
```

Figure 1-3. Advanced nmap NST Location

Admittedly, the above isn't very much fun to type. Paul needs to create a **nstfind-probes** script to encapsulate the above behavior.

Notes

1. https://www.redhat.com/support/resources/howto/rhl9.html
2. https://www.redhat.com/support/resources/howto/rhl9.html
3. https://www.redhat.com/support/resources/howto/rhl9.html
4. http://www.internetishop.com/
5. http://www.networksecuritytoolkit.org/
6. http://sourceforge.net/
7. http://sourceforge.net/projects/nst
8. ../../log/manifest-1.0.5.html
9. http://www.insecure.org/nmap/

Chapter 2. Trouble Shooting

Why Does My NST Boot So Slow?

The Network Security Toolkit is designed to provide you with a full complement of Linux tools directly from the CD. Due to the heavy use of the CD at boot time, one will come to find that the speed of the CD drive is the bottle neck.

For example, Paul finds that the Network Security Toolkit runs fine on his old Sager Laptop with a 466MHZ, however it takes a bit long to start due to the speed of the CD drive.

If you have a slow CD drive in your system, you'll find that a $20 (USD) investment in a new 52X CD drive will greatly improve your experience with the Network Security Toolkit.

Why can't I boot the Network Security Toolkit CD?

There are several conditions which might prevent your test system from booting the Network Security Toolkit CD.

BIOS Set To Boot Hard Drive

Once I have a system setup, I typically configure the BIOS to boot directly from the hard disk. This minimizes the boot time as the BIOS doesn't need to waste time checking for a bootable floppy or CD each time the system starts. However, this also means that I won't be able to boot from my Network Security Toolkit CD. So, after cursing at myself for forgetting about this once again, I make sure that the BIOS is configured to check for a CD prior to booting from the hard disk.

BIOS Halts On Errors

A typical BIOS will stop the boot process if vital system components are missing (think keyboard here). If you are like me, you'll find it is very easy to convert an existing computer to a Network Security Toolkit probe by removing the hard drive, video card, keyboard, mouse, etc. While this is fine as far as the Network Security Toolkit is concerned, the BIOS will complain that its missing some vital component (like the keyboard) and fail to boot from the Network Security Toolkit CD. To remedy this problem, one typically needs to temporarily connect a keyboard and monitor to the system, enter into the BIOS configuration and configure the BIOS specific option to prevent the BIOS from halting the boot process when it detects errors. If you're smarter than me, you'll remember to do this *BEFORE* you strip the components from the system you are converting into a Network Security Toolkit probe.

Finicky CD-ROM Drive

Ron and I have both run into a situation where a certain combination of a CD-ROM drive and BIOS resulted in the inability to boot the Network Security Toolkit. On some systems, I've found that if I avoid the *ide-scsi* driver (you include the *NST_CDROM_IDE* option at the boot prompt) that it will sometimes fix the problem.

On other systems, we've found that if the ISO image is of a particular size, the system will fail to boot. However, if we simply add some filler files to increase the size of the ISO image and burn a new CD, we've found that the same system will boot the Network Security Toolkit. Fortunately, this problem occurs rarely. Its a frustrating problem as we have not been able to track down the underlying cause yet.

Managed NIC

Paul has a 3COM 3C905C-TX-M managed Ethernet card. This Ethernet card has its own BIOS and attempts to boot from the local area network before falling back to booting from the local hard disk. When this feature was enabled, Paul could not get the system to boot from the Network Security Toolkit CD. The BIOS configuration utility built into the Ethernet card did not appear to have a means to disable this feature. So, Paul gave up and replaced the card.

Kernel/CPU Mismatch

If you have a custom built Network Security Toolkit, it is possible that it was done a particular type of CPU. If you then attempt to boot the CD on a system without this type of CPU it might not work. Paul discovered this when he built a Network Security Toolkit distribution on a Anthlon based system and ended up with a CD that would make it part way through the boot process on a Celeron based laptop and then reboot the system as soon at it attempted to uncompress and load the kernel.

Why did my keyboard stop working?

Paul has seen the case where a USB keyboard works to configure BIOS settings and it works at the `ISOLINUX` boot screen, but then stops working once the Network Security Toolkit has booted. This happens if the *`base`* configuration is chosen as the boot method (or any other configuration which does not include USB support). Paul typically avoids using the *`base`* configuration in this situation as it doesn't load enough drivers to support a USB keyboard.

So, if you have a USB keyboard, you will not be able to boot the *`base`* configuration. You must choose one of the other configurations which include the *`NST_USB`* option in order to have the necessary drivers loaded to support a USB keyboard. The *`desktop`* (which is the default boot), *`laptop`*, *`server`*, and *`usb`* configurations all include support for USB keyboards.

If you really need to boot the *`base`* configuration of the Network Security Toolkit, you may need to find a PS/2 style keyboard (or USB to PS/2 adapter for your keyboard).

If you are feeling adventurous, you could check your BIOS settings to see if there is a parameter which allows your BIOS to emulate a PS/2 keyboard.

What's Wrong with My Console Keyboard and Fonts?

When you boot the Network Security Toolkit CDROM, it assumes that you will be using a `us` style keyboard and uses the default font from your computer's BIOS (I admit that I'm guessing a bit on the font statement).

If you don't have a "`us`" keyboard, you will find this quite aggravating as some keys that you press will not yield the desired results. For example, the '@' and '"' characters appear to be swapped on a "`uk`" keyboard.

To fix the keyboard problem, you will want to run the **loadkeys** utility.

```
[root@probe root]# loadkeys uk
Loading //lib/kbd/keymaps/i386/qwerty/uk.map.gz
[root@probe root]#
```

Figure 2-1. Using loadkeys For A "uk" Keyboard

Note: We have found that a least one of the keyboard maps is broken in the Red Hat Linux 9 distribution. In particular, the `dvorak` keyboard map does not load properly, but the `ANSI-dvorak` map does appear to work.

Knowing that the **loadkeys** and **setfont** utilities are available on the system is useful. However, knowing what choices are available can be a bit tricky. The following command can be used to list the available keyboard maps:

```
[root@probe root]# (cd /lib/kbd/keymaps/i386; find . -name "*.map.gz") | less
./azerty/azerty.map.gz
./azerty/be-latin1.map.gz
./azerty/fr-latin1.map.gz
./azerty/fr-latin9.map.gz
./azerty/fr.map.gz
./azerty/fr-pc.map.gz
./azerty/wangbe2.map.gz
./azerty/wangbe.map.gz
./dvorak/ANSI-dvorak.map.gz
./dvorak/dvorak-l.map.gz
./dvorak/dvorak.map.gz
./dvorak/dvorak-r.map.gz
./fgGIod/trf.map.gz
./fgGIod/tr_f-latin5.map.gz
./include/applkey.map.gz
./include/backspace.map.gz
./include/ctrl.map.gz
./include/euro1.map.gz
./include/euro2.map.gz
./include/euro.map.gz
./include/keypad.map.gz
./include/unicode.map.gz
./include/windowkeys.map.gz
:q
[root@probe root]#
```

Figure 2-2. Finding The Available Keyboard Maps

You should be able to specify key map files that are NOT in the `include` directory. For example, "**loadkeys wangbe**" should work (and load the keyboard map file `./azerty/wangbe.map.gz` shown above). However, "**loadkeys ctrl**" will not work as the `ctrl.map.gz` file resides in the `include` directory.

Chapter 3. Tricks On Using The Tools

This section of the FAQ describes some useful tricks when using the tools bundled with the Network Security Toolkit.

Where Are the ettercap Configuration Files?

The main `ettercap`[1] configuration file can be found at `/etc/etter.conf`.

The initial `ettercap`[2] plug-in configuration files are stored in a compressed file `ettercap_plugin_configs.tar.bz2` under the directory `/usr/local/ettercap-NG-0.7.0/share`. The first time you run `ettercap`[3] (`/usr/local/bin/ettercap`), these files are extracted to the `/var/ettercap` directory.

> **Note:** To customize the plug-in configuration files, you must run `ettercap`[4] once, quit it, and then edit the files found under the `/var/ettercap` directory.

How Can I Use My Own ettercap Configuration Files?

If you spend much time customizing your `ettercap`[5] configuration files, you won't want to lose your changes. The easiest way to avoid this, is to make a copy of your files to a local hard disk or thumb drive, and then install them (typically via a symbolic link) prior to running the `/usr/local/bin/ettercap` command.

Here are the steps to create Network Security Toolkit customizations such that the `ettercap`[6] command would use the configurations on a thumb drive (or other non-volatile media) instead of the defaults on the CDROM.

- First, we will mount our media (a thumb drive in this example), and create a directory structure suitable for the **lnstcustom** alias as well as our `ettercap`[7] configuration (NOTE: long file names are required - there may be issues if your file system doesn't support long filenames):

```
[root@probe root]# mount -t vfat /dev/sda1 /mnt/memstick
[root@probe root]# mkdir -p /mnt/memstick/nst/etc /mnt/memstick/nst/share
[root@probe root]#
```

- Now that we've prepared our file structure on our thumb drive, we'll install the default config files that come bundled with the Network Security Toolkit with the following commands:

```
[root@probe root]# cp /etc/etter.conf /mnt/memstick/nst/etc/
[root@probe root]# tar xjf \
/usr/local/ettercap-NG-0.7.0/share/ettercap_plugin_configs.tar.bz2 \
-C /mnt/memstick/nst/share
[root@probe root]#
```

- At this point, we now have the necessary configuration files which we may wish to edit available on our thumb drive. We will create (or update) the file `/mnt/memstick/nst/setup.sh` with the following lines to install our custom ettercap configuration.

```
# Not necessary if used via lnstcustom
NSTHOME="${NSTHOME:-.}"

# Install our ettercap config file if present
```

```
if [ -f "${NSTHOME}/etc/etter.conf" ]; then
  if [ -f "/etc/etter.conf" ]; then
    /bin/rm -f "/etc/etter.conf"
  fi
  /bin/ln -s "${NSTHOME}/etc/etter.conf" /etc
fi

# Install our plug-in customizations if present
if [ -d "${NSTHOME}/share/ettercap" ]; then
  if [ -L "/var/ettercap" ]; then
    /bin/rm -f "/var/ettercap"
  elif [ -d "/var/ettercap" ]; then
    /bin/rm -fr "/var/ettercap"
  fi
  /bin/ln -s "${NSTHOME}/share/ettercap" /var
fi
```

The above may seem like overkill, but allow us to seemlessly integrate this into the **lnstcustom** process.

- We've now completed the steps required for customizing our `ettercap`[8] experience. We'll use the **umount** the thumb drive (so that all of our work is actually written to the disk), and have it ready for use via the **lnstcustom** command.

```
[root@probe root]# umount /mnt/memstick
[root@probe root]#
```

OK, now that we've taken the time to prepare a customized environment for `ettercap`[9], lets try it out. To make use of the customized environment, we'll use the following steps:

- First, we'll plug in our prepared thumb drive.
- Now, we'll use the following **lnstcustom** command to prepare the system to use the `ettercap`[10] configuration files on our thumb drive instead of the ones that come on the Network Security Toolkit CDROM:

```
[root@probe root]# lnstcustum nst sda1 vfat
[root@probe root]#
```

- That's all there is to it. From this point on, each time we run the **ettercap** command, it will use the configuration files found on our thumb drive. In addition, any changes we make to our `ettercap`[11] configuration files will persist between Network Security Toolkit sessions.

How Do I Start/Stop/Customize Argus?

You can use the Network Security Toolkit web based user interface to start, stop and access the **argusd**. This is handy if you just want to get a quick idea of what `argus`[12] can be used for. However, using the Network Security Toolkit web based user interface to start up `argus`[13] just brings up a simple default configuration used to monitor a couple of web servers involved in the Network Security Toolkit project. Once you decide on using `argus`[14] for a real purpose, you'll want to customize its configuration.

Starting and stopping `argus`[15] is done via a standard service script `/etc/rc.d/init.d/argusd`. The following is all that is required to start it up:

```
[root@probe root]# /etc/rc.d/init.d/argusd start
Starting argusd:                                           [  OK  ]
[root@probe root]#
```

The Network Security Toolkit preconfigures argus[16] to use /var/argus as the location for its data and configuration files. If this directory is not found when you first start argus[17], it will be created and initialized with the Network Security Toolkit default configuration from /usr/local/argus/data.tar.bz2. You will most likely want to customize the /var/argus/config file for the systems you want argus[18] to monitor (refer to the documentation at the argus[19] site for details on configuration).

Once you've customized your argus[20] config file, you'll want to signal the **argusd** to reload its config. You can either restart the **argusd** service, or you can use the /usr/local/argus/sbin/argusctl command in the following manner:

```
[root@probe root]# /usr/local/argus/sbin/argusctl hup
ARGUS/2.0 200 OK
[root@probe root]#
```

There are many things you can do with the **/usr/local/argus/sbin/argusctl** command, try invoking it the arugment help for additional details.

What is the Argus URL?

After configuring and starting argus[21], you'll probably want to make use of its web based user interface. If you used the Network Security Toolkit WUI to start argus[22], you can just click on the link provided. Alternatively, you can point your browser at https://HOST/argus/argus.cgi.

Why Do I Have to Login to Argus?

Since you can do so much through the Network Security Toolkit web based user interface, you must always authorize yourself prior to gaining access. The argus[23] package has its own web based user interface and also requires authorization prior to allowing one to access the service. Unfortunately, Paul's Perl skills are lacking, and he could not quickly determine what was required in order to disable the argus[24] login screen. He was able to figure out how to set the default configuration such that if you login with the user ID set to root you should be able to gain access to argus[25] regardless of the password you specify.

If you are a Perl developer and can offer Paul a suggestion on what needs to be done to the /usr/local/argus/html/argus.cgi script (Paul thinks its somewhere in the web_login subroutine), then please drop a note in the NST Forum[26].

Why Aren't There Graphs In Argus?

While argus[27] supports nice graphs, this feature didn't make it into this release of the Network Security Toolkit. There were conflicts in upgrading to the version of the gd graphics required by argus[28] (argus[29] wanted a newer version than several other packages - such as ntop[30]). This will be addressed in future releases of the Network Security Toolkit.

How Do I Get Argus To Send Email Notifications?

The argus[31] service is capable of sending out email notifications when systems that it has been configured to monitor go down or up (have a state transition). In order to accomplish this, the following things need to be done:

- You must have the **sendmail** service running on your Network Security Toolkit probe. This is accomplished via the **setup_sendmail** script.

- You must specify `yes` to one or more `sendnotify` parameters in your `/var/argus/config` file.
- You must specify a valid email address in one or more `notify` parameters in your `/var/argus/config` file.

Take a look the file `/var/argus/config`. It has comments around the lines that need to be changed to enable email.

How Do I Simply My Argus Setup?

> **Note:** This tip is intended for those who have already read through the Using the Network Security Toolkit[32] document (in particular, the `Getting Started` and `File Systems` section).

You can extend your **lnstcustom** `setup.sh` script to automate the configuration and starting of the `argus`[33] service. There are several ways to accomplish this, the following outlines a method to make a permanent setup. It assumes the following:

- You already understand how to use the **lnstcustom** command AND have a pre-existing setup you wish to extend.
- The file system mounted under `$NSTHOME` is writable and fully supports the concept of permissions and ownership. If you are using a FAT file system on a thumb drive - you will need to adjust these steps as a FAT file system does not allow one to specify ownership of files.

Initializing the `$NSTHOME/var/argus` Directory.

First we will need to initialize our customized `argus`[34] area. We will use the following set of commands:

```
[root@probe root]# lnstcustom nst hda5 ext3 (1)
[root@probe root]# mkdir -p $NSTHOME/var/argus (2)
[root@probe root]# (cd $NSTHOME/var/argus; tar xjf /usr/local/argus/data.tar.bz2) (3)
[root@probe root]# chown -R apache.apache $NSTHOME/var/argus (4)
[root@probe root]#
```

(1) This loads an existing Network Security Toolkit customization setup assuming that its located in the directory `nst` under a `ext3` file system found on the 5th partition of the first IDE hard drive (`hda5`). The parameters you supply to this command will depend upon your setup.

(2) Creates a directory for our permanent `argus`[35] configuration and statistics.

(3) Initializes our permanent `argus`[36] directory with the default setup for the Network Security Toolkit probe. You will want to replace or edit the `$NSTHOME/var/argus/config` file for the systems you want to monitor.

(4) This sets the ownership of the `argus`[37] files to `apache.apache` which is necessary in order to make use of the `argus`[38] web based user interface. This is also the reason a FAT based file system can't be directly used for this setup.

Updating `$NSTHOME/setup.sh`.

We now need to add the following to our existing `$NSTHOME/setup.sh` script:

```
# Startup sendmail (assuming we configured argus for email notifications)
```

```
/usr/local/bin/setup_sendmail (1)

# Only setup argus if it isn't yet running
if ! /etc/rc.d/init.d/argusd status > /dev/null; then

  # If /var/argus hasn't been setup yet, use our area
  if [ ! -d /var/argus ]; then

    # Create symbolic link under /var so NST will use our argus config
    /bin/ln -s $NSTHOME/var/argus /var (2)
  fi

  # Start up the argus service
  if [ -d /var/argus ]; then
    /etc/rc.d/init.d/argusd start (3)
  fi
fi
```

(1) This starts up the **sendmail** service using the default settings so that `argus`[39] will be able to send out email notifications. You may need to specify arguments to this command depending upon your situation (use **setup_sendmail --help** for details - or read the Using the Network Security Toolkit[40] document).

(2) This symbolic link will prevent the Network Security Toolkit from installing its default configuration and will cause `argus`[41] to use the configuration we prepared under `$NSTHOME/var/argus`.

(3) Finally, this command starts up the `argus`[42] service with our customized environment.

> **Note:** The above script assumes that the `argus`[43] service has not been previously started on the Network Security Toolkit. If the `/var/argus` directory already exists, it will fail (as the creation of the symbolic link will fail).

Notes

1. http://ettercap.sourceforge.net/
2. http://ettercap.sourceforge.net/
3. http://ettercap.sourceforge.net/
4. http://ettercap.sourceforge.net/
5. http://ettercap.sourceforge.net/
6. http://ettercap.sourceforge.net/
7. http://ettercap.sourceforge.net/
8. http://ettercap.sourceforge.net/
9. http://ettercap.sourceforge.net/
10. http://ettercap.sourceforge.net/
11. http://ettercap.sourceforge.net/
12. http://argus.tcp4me.com/
13. http://argus.tcp4me.com/
14. http://argus.tcp4me.com/
15. http://argus.tcp4me.com/

16. http://argus.tcp4me.com/
17. http://argus.tcp4me.com/
18. http://argus.tcp4me.com/
19. http://argus.tcp4me.com/
20. http://argus.tcp4me.com/
21. http://argus.tcp4me.com/
22. http://argus.tcp4me.com/
23. http://argus.tcp4me.com/
24. http://argus.tcp4me.com/
25. http://argus.tcp4me.com/
26. http://sourceforge.net/forum/?group_id=85467
27. http://argus.tcp4me.com/
28. http://argus.tcp4me.com/
29. http://argus.tcp4me.com/
30. http://www.ntop.org/
31. http://argus.tcp4me.com/
32. ../user/
33. http://argus.tcp4me.com/
34. http://argus.tcp4me.com/
35. http://argus.tcp4me.com/
36. http://argus.tcp4me.com/
37. http://argus.tcp4me.com/
38. http://argus.tcp4me.com/
39. http://argus.tcp4me.com/
40. ../user/
41. http://argus.tcp4me.com/
42. http://argus.tcp4me.com/
43. http://argus.tcp4me.com/

Chapter 4. Source Code

How Can I Get The Source?

Each time a Network Security Toolkit ISO is released, a .tar.gz file containing the source code used to build it is also released. You can find both the ISO image and the source code .tar.gz file in the `Files` section at http://sourceforge.net/projects/nst.

For details on setting up a development system capable of creating the Network Security Toolkit ISO from the source, see: Building the NST ISO[2].

How Can I Get The Current Source?

The Network Security Toolkit source code is held in a CVS repository at SourceForge[3]. The URL for the Network Security Toolkit project at SourceForge[4] is http://sourceforge.net/projects/nstYou may browse the source code online at http://cvs.sourceforge.net/cgi-bin/viewcvs.cgi/nst/.

You can checkout the current version of the source code with the following set of CVS commands (just press the **Enter** key when prompted for a password):

```
[root@salsa root]# mkdir $HOME/nst
[root@salsa root]# cd $HOME/nst
[root@salsa nst]# cvs -d:pserver:anonymous@cvs.sourceforge.net:/cvsroot/nst login
Logging in to :pserver:anonymous@cvs.sourceforge.net:2401/cvsroot/nst
CVS password:
[root@salsa nst]# cvs -z3 -d:pserver:anonymous@cvs.sourceforge.net:/cvsroot/nst co .
```

Please note, if SourceForge[7] is busy, its possible the login or update will fail and you will need to try again later. Also, its possible you will not get the current days worth of updates (SourceForge[8] doesn't always provide up to the minute updates for anonymous users).

How Do I Get The Red Hat 9 Branch?

When the Network Security Toolkit project moved from `Red Hat 9` to `Fedora Core 2` as its base distribution, we created a branch under CVS called `rh9`. This allows us to continue patching/updating the `Red Hat 9` version of the Network Security Toolkit if desired.

The following figures demonstrate how one can access the source code under the `rh9` CVS branch:

```
[root]# mkdir -p /usr/local/src/nst/rh9
[root]# cd /usr/local/src/nst/rh9
[rh9]# cvs -d:pserver:anonymous@cvs.sourceforge.net:/cvsroot/nst login
Logging in to :pserver:anonymous@cvs.sourceforge.net:2401/cvsroot/nst
CVS password:PRESS ENTER KEY
[rh9]# cvs -d:pserver:anonymous@cvs.sourceforge.net:/cvsroot/nst checkout -r rh9 .

... lots of output ...

[rh9]#
```

Figure 4-1. Anonymous Check Out of Red Hat 9 Branch

```
[root]# mkdir -p /usr/local/src/nst/rh9
[root]# cd /usr/local/src/nst/rh9
[rh9]# export CVS_RSH=ssh
[rh9]# export CVSROOT=:ext:USERID@cvs.sourceforge.net:/cvsroot/nst
[rh9]# cvs checkout -r rh9 .

... lots of output ...

[rh9]#
```

Figure 4-2. Developer Check Out of Red Hat 9 Branch

As a developer, any work you do in the `rh9` branch will be completely separate from any work you do in the main development branch. You will need to read up on the branch "merging" capabilities of CVS if you want to merge changes from one branch into another.

> **Note:** You can use the CVS `status` command on any CVS file to see what branch you are working on. It reports branch information on the `Sticky Tag:` line, but only if the file is part of a branch.

How Do I Merge Changes Across CVS Branches?

If you are a developer which makes modifications to one branch of the CVS repository that you would like to see reflected in another branch of the CVS repository, you will want to use the merging capabilities of CVS (use **info cvs** for detailed CVS documentation).

The following figures demonstrate how one merges between the `HEAD` branch and the `rh9` branch:

```
[root]# cd /opt/nst/rh9/src/docs
[docs]# cvs update -j HEAD faq/source.xml
RCS file: /cvsroot/nst/docs/faq/source.xml,v
retrieving revision 1.7
retrieving revision 1.9
Merging differences between 1.7 and 1.9 into source.xml
[docs]# cvs commit faq/source.xml
[docs]#
```

Figure 4-3. Merging From `HEAD` Branch to `rh9` Branch

```
[root]# cd /opt/nst/fc2/src/docs
[docs]# cvs update -j rh9 user/scripts.xml
RCS file: /cvsroot/nst/docs/faq/scripts.xml,v
retrieving revision 1.7
retrieving revision 1.9.2
Merging differences between 1.7 and 1.9.2 into scripts.xml
[docs]# cvs commit user/scripts.xml
[docs]#
```

Figure 4-4. Merging From `rh9` Branch to `HEAD` Branch

> **Warning**
>
> It is possible to recursively merge entire directories (include the `-Pd` option as you normally would). However, one needs to be EXTREMELY careful prior to doing so as its very easy to merge in incompatible code.

How Do I View Differences Across CVS Branches?

Before merging changes in one CVS branch with another, it is often desirable to see what the differences are in the two versions of the file.

The following demonstrates how one views the differences between the file `.xinitrc` in the current directory as compared to the current version checked into the `rh9` branch:

```
[root]# cvs diff -r rh9 .xinitrc
Index: .xinitrc
===================================================================
RCS file: /cvsroot/nst/src/packages/system/user-root/root/.xinitrc,v
retrieving revision 1.2
retrieving revision 1.3
diff -r1.2 -r1.3
6c6,8
< if [ -x /usr/X11R6/bin/vtwm ]; then
---
> if [ -x /usr/X11R6/bin/fluxbox ]; then
>   exec /usr/X11R6/bin/fluxbox
> elif [ -x /usr/X11R6/bin/vtwm ]; then
[root]#
```

Figure 4-5. Comparing Source File To `rh9` Branch

The following demonstrates how one views the differences between the file `.xinitrc` in the current directory as compared to the current version checked into the `HEAD` (current development) branch:

```
[root]# cvs diff -r HEAD .xinitrc
Index: .xinitrc
===================================================================
RCS file: /cvsroot/nst/src/packages/system/user-root/root/.xinitrc,v
retrieving revision 1.3
retrieving revision 1.2
diff -r1.3 -r1.2
6,8c6
< if [ -x /usr/X11R6/bin/fluxbox ]; then
<   exec /usr/X11R6/bin/fluxbox
< elif [ -x /usr/X11R6/bin/vtwm ]; then
---
> if [ -x /usr/X11R6/bin/vtwm ]; then
[root]#
```

Figure 4-6. Comparing Source File To `HEAD` (Current Development) Branch

How Do I Build The Network Security Toolkit ISO?

Assuming you've checked out the NST source code under the directory `$HOME/nst` and you are logged in as `root`, the following set of commands are used to create the ISO image. It should be noted that by default, we will be using A LOT of space under `$HOME/nst/tmp` during the build (think GB).

```
[root@salsa root]# cd $HOME/nst
[root@salsa nst]# ./configure
Configuration is complete. Configured as:

... lots of output ...

Use "make" to build src/nst-1.0.6.iso.gz, or "make help" to see
additional targets.
```

```
[root@salsa nst]# make

... Lots of output - takes awhile ...

[root@salsa nst]# ls -l src/*.iso.gz
```

Please note. The output will change over time, but will resemble that which is shown above.

What can I make?

Assuming you've checked out the NST source code under the `$HOME/nst` and you've already run **configure**, you can use the following to see what can be made:

```
[root@salsa root]# cd $HOME/nst
[root@salsa nst]# make help

The following make targets are available

  make
     Default build - same as 'make all'
  make all
     Default build. Invokes 'make all' on docs html wui src
     Produces burnable image: src/nst-1.0.1.iso.gz
  make help
     Displays this help screen
  make docs
     Builds documentation packages
  make clean
     Removes temporary files and all of
  make clear
     Removes temporary files and all of
  make package-check
     Checks to see if your system is up to date for the optional packages
  make upload
     Uploads the NST documentation (use 'make docs' first).
  make release-tag
     Tags current CVS source files for release 1.0.1
  make ['REV=-r 1.0.1'] release
     Builds the full source distribution based on CVS repository
     Add 'REV=-r 1.0.1' if you want a particular release (not tested yet)
     Produces: nst-1.0.1.tar.gz
  make update
     Updates your CVS files (except files you've modified)
     to the current head of the CVS source tree.
  make commit
     Commits all updates you've made to CVS.
  make cd
     Burns bootable NST CD (use 'make all' first)
  make cdrw
     Just like 'make cd', but includes 'blank=fast' to reuse cdrw
[root@salsa nst]#
```

The available **make** targets will change over time (the above output may not be the same as what you get from the current source tree).

Why do I get so many warnings and errors during the build?

Building the Network Security Toolkit ISO from scratch is a non-trivial task. If you are not willing to read the technical documentation, it is not recommended that you try to do it (just download the precompiled binary ISO at SourceForge[9]).

We designed the building of to be flexible. There are several things that lead to error and warning messages. Most of the warning messages you'll see are due to optional security packages which are not bundled with `Red Hat Linux 9`[10]. Here are the general types of errors and warnings that one is likely to encounter:

- A partial `Red Hat Linux 9`[11] installation. If you do not install all of the packages making up `Red Hat Linux 9`[12] on your system, its possible that you will be missing something required to build the Network Security Toolkit.
- Permission errors are possible if you attempt to build the Network Security Toolkit ISO as a non `root` user. In order to build the ISO image, there are many commands which need to be run and files which need to be accessed that require `root` level access.
- Warnings for other packages are common if the package is not part of the standard `Red Hat Linux 9`[13] installation. The Network Security Toolkit attempts to install many security programs which are not bundled with the `Red Hat Linux 9`[14]. These warnings will not prevent the ISO from working, it just means that the optional package will not be available.

How Do I Find Broken Links?

During the addition of new packages to the Network Security Toolkit, it is often necessary to create symbolic links. It is quite easy to create broken symbolic links.

The **find** command can be useful for finding symbolic links (when you specify `-type l`) from a starting directory. You can then test each of the symbolic links found to see if they refer to a non-existant file. The following demonstrates how this was used on the `/usr/local/bin` directory as the **sendmail** package was being added.

```
[root@probe root]# for f in $(find /usr/local/bin -type l); do \
if [ ! -e "$f" ]; then echo $f; fi; done
/usr/local/bin/hoststat
/usr/local/bin/kbdrate
/usr/local/bin/mailq.sendmail
/usr/local/bin/makemap
/usr/local/bin/newaliases.sendmail
/usr/local/bin/purgestat

[root@probe root]#
```

Figure 4-7. Finding Broken Links

Notes

1. http://sourceforge.net/projects/nst
2. ../tech/
3. http://sourceforge.net/
4. http://sourceforge.net/
5. http://sourceforge.net/projects/nst
6. http://cvs.sourceforge.net/cgi-bin/viewcvs.cgi/nst/
7. http://sourceforge.net/

8. http://sourceforge.net/
9. http://sourceforge.net/
10. https://www.redhat.com/support/resources/howto/rhl9.html
11. https://www.redhat.com/support/resources/howto/rhl9.html
12. https://www.redhat.com/support/resources/howto/rhl9.html
13. https://www.redhat.com/support/resources/howto/rhl9.html
14. https://www.redhat.com/support/resources/howto/rhl9.html

Chapter 5. Serial Port

How does the Network Security Toolkit use the serial port?

The Network Security Toolkit makes use of serial port */dev/ttyS0* (COM1) at boot up.

What settings should I use to connect to the serial port?

You should set a baud of *19200*, parity to *none*, data bits to *8*, stop bits to *1*. *VT220* terminal emulation is also desirable.