Building the NST ISO

Ronald Henderson

rhenderson@unifiedholdings.com

Paul Blankenbaker

paul@mekwin.com

Building the NST ISO

by Ronald Henderson

by Paul Blankenbaker

Information on building the Network Security Toolkit from the ground up.

Table of Contents

Chapter 1. System Requirements

Development System

The requirements for a development system are much more involved. At a minimum, you should have:

- A system with a FULL `Red Hat Linux 9`[1] Linux distribution installed (hopefully you've kept up with the latest security patches).
- At least 2GB of free disk space.
- The ability to burn a CD.
- Unfortunately, you will need `root` access in order to build the Network Security Toolkit (Paul wishes there were a way to build the ISO without `root` access, but we haven't come up with a solution).

Kernel Considerations

You need to be careful about the kernel which is used on your development system. When you install `Red Hat Linux 9`[2], you'll discover that the installation chooses the kernel which is best for the system you are installing it on. For example, if you install `Red Hat Linux 9`[3] onto a Anthlon system, it will choose a version of the kernel built with support for the Anthlon CPU. If you then build the Network Security Toolkit on this system, you'll end up with a ISO image that only works on Anthlon systems.

The following check list will prove helpful when updating the kernel on your development system

- Don't allow the `up2date` utility to automatically update your kernel. Instead, choose to do it by hand using commands like:

  ```
  [root@quesadilla root]# rpm -ivh /lan/download/linux/RedHat/kernel-2.4.20-20.9.i586.rpm
  ```

- If you have access to the `Red Hat Network`[4], you can search the RPMs using the keyword "kernel" to locate the most recent binary versions of the kernel. For portability, it is recommended to choose a *`i386`* or *`i586`* version.
- If you make the mistake of installing the wrong version, it probably won't boot on all systems. Paul wrote this section after burning a Network Security Toolkit with a kernel compiled for Anthlon processors, and then discovering that it wouldn't boot on a laptop with a Celeron CPU.

  Its a bit tricky to remedy this situation. I would not recommend forcing a kernel install over an existing kernel, instead, follow these steps:

  - Install an older version of the kernel (you should be able to use the original on disk 1 of your `Red Hat Linux 9`[5] CD).
  - Reboot your system using the older kernel which you just installed.
  - Remove the kernel that you want to replace.
  - You should now be able to install the version of the kernel which you originally wanted.

- If you are building your own custom kernels, then you won't need to worry about this issue (but you will have to worry about rebuilding the kernel each time a new security patch is released).

Acquiring Source Code

The entire set of source code which we the developers use in producing the Network Security Toolkit is freely available at SourceForge[6]. The URL for the project source code is http://sourceforge.net/projects/nst/.

Regardless of how one acquires the source code, the **configure** command must be invoked before one can use the **make** command.

Via Tarball

Periodically the developers will decide that a good stable point in development has been reached and release a official version of the Network Security Toolkit. You should be able to find the files which have been released at:

http://sourceforge.net/project/showfiles.php?group_id=85467

To download the source files for a paritcular release, you simply download the source tarball having a file name similiar to `nst-1.0.5.tar.gz` (the `1.0.5` indicates the release). Once downloaded, you can extract the tarball in the directory of your choosing.

Via Anonymous CVS

The following commands can be used to anonymously check out the lastest version (assuming you want them in your `$HOME/nst` directory):

```
[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 .
```

CVS Checkout

The anonymous checkout above will not permit you to commit any changes you may have happen to made. If you are a developer associated with the project, you should refer to the CVS Notes[9] and Setting Up ssh[10] notes available at the project web site. Once you've followed those instructions, you can use the following command to checkout the current source code:

```
[root@salsa root]# mkdir $HOME/nst
[root@salsa root]# cd $HOME/nst
[root@salsa nst]# nstcvs checkout .
```

During active development, other developers may commit changes to the source tree. At which point, the source code you are working with will become out of date. You can bring your area back up to date at any time using the commands shown below (assuming you specified the `--sf-user USER` option to the **configure** command):

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

Configuring Your Build Environment

Once you have a copy of the source code, you will need to configure your build environment. The easiest way is typically to use the top level **configure** command to guess at the defaults for your system.

```
[root@salsa root]# cd $HOME/nst
[root@salsa nst]# ./configure
```

We are constantly updating the configuration process. We have designed the default behavior of the **configure** to set things up such that one can then invoke the **make** command and produce a ISO image. For those actively involved in the development of the Network Security Toolkit, it is recommended that you use the following command to see what options are currently configurable:

```
[root@salsa root]# cd $HOME/nst
[root@salsa nst]# ./configure --help | less
                    Network Security Toolkit (NST)
                http://www.networksecuritytoolkit.org/

This top level directory allows one to configure and build the entire
set of source packages making up the NST. If you have a full RedHat
9.0 installation, you can do the following to build the documentation,
web user interface and create a ISO image:

  ./configure
  make

The resulting ISO image can be found using:

  ls src/*.iso.gz

If you have not installed all of the optional packages which the NST
supports (as many additional packages than found under RedHat 9.0 are
supported), you will most likely get a bootable CD with a subset of
the full NST capabilities. You should really refer to the Technical
documentation set at the NST home page for additional notes on the
build process.

The top level configure script must be run without error before you
will be able to build the NST.

Once configured, you can use:

  make help

To see what targets are available.


OPTIONS:

  -c DIR, --config-dir DIR
     The user "customization" directory. We look for optional files which
     control the building of the NST in this directory. It defaults to
     $HOME/.nst if not specified. We look for the optional customization
     files: configure.sh, disable.txt and post_install.sh in this directory.

  -d DIR, --build-dir DIR
     Set the location to be used for building the NST with.
     Default: ${DSTDIR}

  -p PASSWD, --passwd PASSWD
     Set the master password for the NST. WARNING: This will be stored
     as clear text in the config files (the files will be set read only
     for the user, but this is still not recommended). If you omit this
     option, you will be prompted each time your build the ISO to enter
```

```
   the password for the ISO image your are building.

--ct-passwd PASSWD
   Set the "clear text" password used for applications where we
   haven’t found a means to avoid it (setting up the SQL database, and
   the WUI for phpMyAdmin and nessus). This will be stored in one
   file (/etc/nst.conf) with the permissions set such that only the root
   user (or members for the root group) are able to access it.

--ct-snort-passwd PASSWD

   Set the "clear text" password used to access the Snort
   database. This will be stored in one file (/etc/nst.conf) with
   the permissions set such that only the root user (or members for the
   root group) are able to access it.

--prompt-phrase
   If this option is specified, then you will be prompted for a pass
   phrase during the build process. If not specified, the pass phrase use
   for the ssh key will be the same as the root password.

--hostname TEXT
   This option controls the default host name that the probe will
   use at boot time (default is "probe").

--domainname TEXT
   This option controls the default domain name that the probe will
   use at boot time (default is "localdomain").

--hw-clock-utc
   This option tells NST at boot that the system’s hardware clock is set to
   UTC time. If NST is used on system’s with the hardware clock set to
   the local time, do not use this option.

--timezone TZ
   Use this option to set the timezone (TZ) value for the NST ISO. Timezone
   values can be found in the "/usr/share/zoneinfo" directory. Do not prepend
   the "/usr/share/zoneinfo" header to the TZ value.
   (default is "America/New_York")

--boot LABEL
   Use this option to change the default boot configuration. If not
   specified, it defaults to "desktop". For example, Paul uses "--boot
   laptop" to enable PCMCIA services on a default boot. LABEL needs to
   be one of the valid boot configurations shown below:

     base serial desktop laptop server ide utils pcmcia usb noapm

-k KVER, --kernel-ver KVER
   Specify the kernel you want to use (the modules must exist
   under /usr/lib/modules/KVER). Default: ${KERVER_DEF}

--kernel-type KTYPE
   Specify the kernel type (i386, i586, anthlon). If not specified,
   we make a best guess by looking at /boot/kernel.h

--html-dir DIR
   Set the top level directory to build HTML documents for.
   Default: ${HTMLDIR}

--java JVM
   Specify the location of the java virtual machine to use.
   Default: ${JAVA}

--at-macro-options OPTIONS
   Specify any custom options to pass to invocation of com.ccg.macros.at.All.
```

```
     Default: ${MACROOPTIONS}

  --ftp-host HOST
     Specify the FTP host used to upload web pages to web server.
     Default: ${FTPHOST}

  --ftp-dir DIR
     Specify the directory at the FTP host that you will want upload the
     web pages to. Default: ${FTPDIR}

  --ftp-user USER
     Specify the user to login as at the FTP host that you will want to upload
     the web pages to. Default: ${FTPUSER}

  --sf-user USER
     Specify your user ID at sourceforge.net (this is not required, but
     enables those associated with the project to make "releases")

  --sf-mirror URL
     Specify the sourceforge.net mirror site to use as the root URL when
     fetching some of the optional packages. It defaults to:
     ${SOURCE_FORGE_MIRROR}.

  --build-root DIR
     Specify the top level directory where extra packages required for
     a full NST install are stored and built at. Default: /usr/local/src

  --show
     Show configurable variables and what they will be set to. NOTE:
     If you copy this output to $HOME/.nst/configure.sh, you can "tweak"
     the values for future invocations of the configure script.

  --save
     Save current configuration to $HOME/.nst/configure.sh for future
     invocations.

  --on-line-docs
     This option will increase the size of the ISO image
     created (because we will include the larger documents - instead of
     using external links). Developers wishing to update the official
     web site will need to this option (you might be able to get away
     with "make docs upload" - but I wouldn’t bet the farm on it).

  --help
     This is the only option available at the top level configure.


RECOMMENDATION:

If you are a developer which likes to customize their environment, you
can use the following as a starting point for creating your own custom
"configure" of the NST:

#!/bin/bash

#
# Start with defaults and set sourceforge.net user ID to sfuser
#
NSTHOME=$HOME/nst

if [ -f "$NSTHOME/Makefile" ]; then (cd $NSTHOME; make clear); fi

(cd $NSTHOME; ./configure --sf-user sfuser --html-dir $HOME/public_html )


DISABLING:
```

```
It is now possible to disable the addition of specific categories
and/or packages. To do this, create the file $HOME/.nst/disable.txt
and add entries (single entry per line) in the form of CATEGORY or
CATEGORY/PACKAGE. For example, the following two lines would prevent
ALL of the packages in the "application" category from being added as
well as the "wlan-ng" package from the "networking" category:

applications
networking/wlan-ng
```

Building The Network Security Toolkit

Once you have a copy of the source code configured, you should be able to build all of the documentation and the ISO image of the Network Security Toolkit with the following command:

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

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://rhn.redhat.com/
5. https://www.redhat.com/support/resources/howto/rhl9.html
6. http://sourceforge.net/
7. http://sourceforge.net/projects/nst/
8. http://sourceforge.net/project/showfiles.php?group_id=85467
9. http://sourceforge.net/docman/display_doc.php?docid=18122&group_id=85467
10. http://sourceforge.net/docman/display_doc.php?docid=18121&group_id=85467

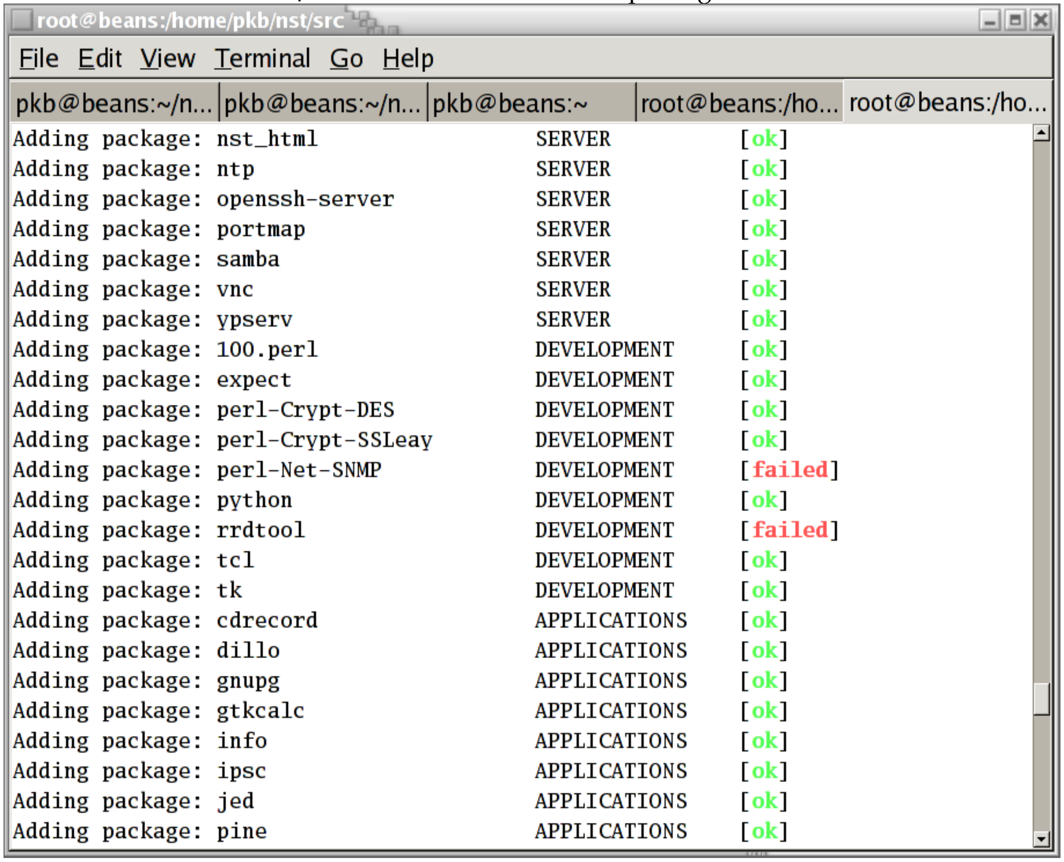
Chapter 2. Optional Packages

While `Red Hat Linux 9`[1] provides a lot of useful features and tools, there are many additional security and diagnostic tools which were desired in the Network Security Toolkit.

Overview

We took the following philosophy towards the additional security and diagnostic tools which are included with the Network Security Toolkit:

- The optional packages are truly optional. This means your development system doesn't need to have the optional packages installed. You will get error and/or warning messages during the build, but you will still end up with a bootable image. The following shows what a build might look like on a development system which does not have the `perl-Net-SNMP` or `rrdtool` package installed:



```
Adding package: nst_html                 SERVER          [ok]
Adding package: ntp                      SERVER          [ok]
Adding package: openssh-server           SERVER          [ok]
Adding package: portmap                  SERVER          [ok]
Adding package: samba                    SERVER          [ok]
Adding package: vnc                      SERVER          [ok]
Adding package: ypserv                   SERVER          [ok]
Adding package: 100.perl                 DEVELOPMENT     [ok]
Adding package: expect                   DEVELOPMENT     [ok]
Adding package: perl-Crypt-DES           DEVELOPMENT     [ok]
Adding package: perl-Crypt-SSLeay        DEVELOPMENT     [ok]
Adding package: perl-Net-SNMP            DEVELOPMENT     [failed]
Adding package: python                   DEVELOPMENT     [ok]
Adding package: rrdtool                  DEVELOPMENT     [failed]
Adding package: tcl                      DEVELOPMENT     [ok]
Adding package: tk                       DEVELOPMENT     [ok]
Adding package: cdrecord                 APPLICATIONS    [ok]
Adding package: dillo                    APPLICATIONS    [ok]
Adding package: gnupg                    APPLICATIONS    [ok]
Adding package: gtkcalc                  APPLICATIONS    [ok]
Adding package: info                     APPLICATIONS    [ok]
Adding package: ipsc                     APPLICATIONS    [ok]
Adding package: jed                      APPLICATIONS    [ok]
Adding package: pine                     APPLICATIONS    [ok]
```

- A top level make target has been provided to aid one in determining which optional packages a development system has and/or needs. The following demonstrates a sample invocation, and a portion of the output generated:

```
[pkb@salsa nst]$ make package-check
kernel-ntfs (version 2.4.20-20.9) was found
mapscsi (version 0.0.11) was found
vtwm (version 5.4.6a) was found
libnet (version 1.1.0) ***WARNING*** was NOT found! Used test:
   strings /usr/lib/libnet.a 2>&1 | grep 1.1.0 > /dev/null 2>&1
arpd (version 0.2) was found
wlan-ng (version 0.2.0-7) ***WARNING*** was NOT found! Used test:
   /bin/rpm -q wlan-ng | /bin/grep ^wlan-ng-0.2.0-7
dillo (version 0.7.3-6) was found
```

- Scripts are provided as an aid to the setting up a development system with the optional packages. As developers ourselves, we wanted a means to keep track of both the location on the Internet of the optional packages as well as how we went about compiling and installing the optional packages. This makes the process of setting up a development system a bit less painful. The following command can be used to show what packages can be fecthed, built and/or installed:

  ```
  [pkb@salsa pkb]$ ls $HOME/nst/src/bin
  /home/pkb/nst/src/bin/arpd_fetch_install
  /home/pkb/nst/src/bin/dillo_fetch_build
  /home/pkb/nst/src/bin/gtkcalc_fetch_install
  /home/pkb/nst/src/bin/hammerhead_fetch_build
  /home/pkb/nst/src/bin/honeyd_fetch_install
  /home/pkb/nst/src/bin/ipsc_fetch_install
  /home/pkb/nst/src/bin/libnet_fetch_build
  /home/pkb/nst/src/bin/mapscsi_fetch_build
  /home/pkb/nst/src/bin/nikto_fetch_build
  /home/pkb/nst/src/bin/ntfs_fetch_install
  /home/pkb/nst/src/bin/ntop_fetch_install
  /home/pkb/nst/src/bin/perl-crypt-des_fetch_install
  /home/pkb/nst/src/bin/perl-net-snmp_fetch_install
  /home/pkb/nst/src/bin/rrdtool_fetch_install
  /home/pkb/nst/src/bin/snort_fetch_build
  /home/pkb/nst/src/bin/tcpreplay_fetch_build
  /home/pkb/nst/src/bin/vtwm_fetch_build
  ```

  These scripts need to be run as `root` as they install software onto the development system.

- Optional packages are downloaded (and sometimes built) under `/usr/local/src`. As a result, once you've installed the optional packages on your system, you may want to go back and clean up the temporary files which were downloaded, extracted and built under `/usr/local/src`.

Optional Package Database

The `Red Hat Network`[2] makes it easy to keep the `Red Hat Linux 9`[3] packages up to date. However, it becomes quite a nightmare in keeping a system up to date with all of the optional packages support by Network Security Toolkit. The following steps were taken in order to help keep a system up to date:

- A simple tab delimited ASCII database is maintained in the file `include/packages/packages.tsv`.
- Several **awk** scripts translate the data at the time of configuration, creating many useful support scripts (and other files) under the `config` directory.
- The scripts which are generated are then used in various makefiles, installation scripts, download scripts, etc.
- Having a single source table makes it much easier to maintain consistency throughout the code.

The following columns are defined within the `include/data/packages.tsv` table:

PKG

  The name associated with the package.

VER

The current version of the optional package which we want to support within Network Security Toolkit.

TYPE

The type of distribution that the package is installed from. Currently this is limited to *`rpm`* or *`tgz`*.

URL

The URL where the software can be retrieved from.

TINSTALL

Blank for the *`rpm`* packages. Otherwise a minimal **bash** shell command that will tell us if the package is installed.

RPM Download Script

The following demonstrates how the helper functions defined as `config/PACKAGE.sh` (like `config/arpd.sh`) can be used when creating scripts to aid the developer in updating an optional package on the system.

```
#!/bin/sh

# $Id$
#
# Load supporting script functions and check to update RPM

. $(dirname $0)/../../config/arpd.sh

arpd_check_rpm_update
```

The above script can be found at `src/bin/arpd_fetch_install`. The script works by making use of the `arpd_check_rpm_update` function in the `config/arpd.sh` which was automatically created at the time of configuration.

Installation Script Example

The following demonstrates how the configuration file `config/arpd.sh` is used by the `src/packages/networking/arpd.sh` installation script.

```
# $Id: arpd.sh,v 1.1 2003/08/29 01:01:02 rwhalb Exp $

# Description:  This script installs the virtual arp
#               daemon into the NST distribution.

# Load package info and log a warning if version change
# ----------------------------------------------------
. "${SRCDIR}/config/arpd.sh"
arpd_install_check 0.2


# Install the package:
# --------------------
check_rpm ${PKG} "${NSTLOCALDIR}" "/usr" \
  "^/usr/sbin/" "^/usr/share/doc/${PKG}" "^/usr/share/man/"
```

The above installation script works using the following philosophy:

- It loads the `config/arpd.sh` support script which was automatically generated from `include/data/packages.tsv` at the time of configuration.
- It uses the `arpd_install_check 0.2` function defined in `config/arpd.sh`. This sets the environment variables: PKG, VER and URL, and more importantly, logs a warning message if the current version of the **arpd** program defined in the database has changed since the installation script was last modified.
- It finally makes use of the `check_rpm` function to install the necessary files needed by the Network Security Toolkit ISO.

Current Database

The following lists the optional packages which are currently defined in the `include/data/packages.tsv` table. Please note the order shown is not alphabetical, but arranged such that if one were to install the entries shown in the order shown, they should not run into any dependency issues. Also, the list may change depending upon your configuration (as a developer, its best to view the technical document as generated for your development system than the one made public on the Internet).

`kernel-ntfs`

Version: *2.4.20-31.9* Type: *rpm* Home URL: http://linux-ntfs.sourceforge.net/ Download URL: http://unc.dl.sourceforge.net/linux-ntfs/kernel-ntfs-2.4.20-31.9.i686.rpm

`mapscsi`

Version: *0.0.11* Type: *tgz* Home URL: http://gort.metaparadigm.com/mapscsi/ Download URL: http://freshmeat.net/redir/mapscsi/20369/url_tgz/mapscsi-0.0.11.tar.gz

`vtwm`

Version: *5.4.6a* Type: *tgz* Home URL: http://www.visi.com/~hawkeyd/vtwm.html Download URL: http://www.visi.com/cgi-bin/cgiwrap/~hawkeyd/dnldcount.cgi?/ftp/users/hawkeyd/X/vtwm-5.4.6a.tar.gz

`libnet`

Version: *1.1.2.1* Type: *tgz* Home URL: http://www.packetfactory.net/projects/libnet/ Download URL: http://www.packetfactory.net/libnet/dist/libnet.tar.gz

`arpd`

Version: *0.2* Type: *rpm* Home URL: http://www.citi.umich.edu/u/provos/honeyd/ Download URL: http://www.spenneberg.com/redirect.php?url=public/Honeypot/arpd/arpd-0.2-rh8_2.i386.rpm

`ettercap`

Version: *NG-0.7.0* Type: *tgz* Home URL: http://ettercap.sourceforge.net/ Download URL: http://unc.dl.sourceforge.net/sourceforge/ettercap/ettercap-NG-0.7.0.tar.gz

`gpsd`

Version: *1.07* Type: *tgz* Home URL: http://www.pygps.org/gpsd/gpsd.html Download URL: http://www.pygps.org/gpsd/downloads/gpsd-1.07.tar.gz `smtpclient`

Version: *1.0.0* Type: *tgz* Home URL: http://www.engelschall.com/sw/smtpclient/ Download URL: http://www.engelschall.com/sw/smtpclient/distrib/smtpclient-1.0.0.tar.gz

`gtkcalc`

Version: *1.0-1* Type: *rpm* Home URL: http://linuxberg.matrix.com.br/home/preview/165246.html
Download URL: http://linuxberg.matrix.com.br/files/gtkcalc-1.0-1.i386.rpm

`perl-Crypt-DES`

Version: *2.03-1* Type: *rpm* Home URL: http://search.cpan.org/dist/Crypt-DES/ Download URL: http://redhat.ifsic.univ-rennes1.fr/contrib/libc6/i386/perl-Crypt-DES-2.03-1.i386.rpm

`perl-Net-SNMP`

Version: *4.1.0-0* Type: *rpm* Home URL: http://search.cpan.org/dist/Net-SNMP/ Download URL: http://dag.wieers.com/packages/perl-Net-SNMP/perl-Net-SNMP-4.1.0-0.dag.rh90.noarch.rpm

`perl-XML-Simple`

Version: *2.08-0* Type: *rpm* Home URL: http://search.cpan.org/dist/XML-Simple/ Download URL: http://dag.wieers.com/packages/perl-XML-Simple/perl-XML-Simple-2.08-0.dag.rh90.noarch.rpm

`ntop`

Version: *3.0-0* Type: *rpm* Home URL: http://www.ntop.org/ Download URL: http://unc.dl.sourceforge.net/sourceforge/ntop/ntop-3.0-0.i386.rpm

`xprobe2`

Version: *0.2* Type: *tgz* Home URL: http://www.sys-security.com/html/projects/X.html Download URL: http://www.sys-security.com/archive/tools/xprobe2/xprobe2-0.2.tar.gz

`ethereal`

Version: *0.10.6* Type: *bz2* Home URL: http://www.ethereal.com/ Download URL: http://www.ethereal.com/distribution/ethereal-0.10.6.tar.bz2

`libpcre`

Version: *4.3* Type: *bz2* Home URL: http://www.pcre.org/ Download URL: http://unc.dl.sourceforge.net/sourceforge/pcre/pcre-4.3.tar.bz2

`adns`

Version: *1.1* Type: *tgz* Home URL: http://www.chiark.greenend.org.uk/~ian/adns/ Download URL: http://www.chiark.greenend.org.uk/~ian/adns/ftp/adns-1.1.tar.gz

`pwgen`

Version: *2.03* Type: *tgz* Home URL: http://sourceforge.net/projects/pwgen Download URL: http://unc.dl.sourceforge.net/sourceforge/pwgen/pwgen-2.03.tar.gz

`rrdtool`

Version: *1.0.41-1.8.0* Type: *rpm* Home URL: http://people.ee.ethz.ch/~oetiker/webtools/rrdtool/ Download URL:

http://people.ee.ethz.ch/~oetiker/webtools/rrdtool/pub/rrdtool-1.0.41-1.8.0.i386.rpm

`snort`

Version: *2.2.0-1.0* Type: *rpm* Home URL: http://www.snort.org/ Download URL: http://dag.wieers.com/packages/snort/snort-2.2.0-1.0.rh9.dag.i386.rpm

`snort-rules`

Version: *2_2* Type: *tgz* Home URL: http://www.snort.org/snort-db/ Download URL: http://www.snort.org/dl/rules/snortrules-snapshot-2_2.tar.gz

`snort-mysql`

Version: *2.2.0-1.0* Type: *rpm* Home URL: http://www.snort.org/ Download URL: http://dag.wieers.com/packages/snort/snort-mysql-2.2.0-1.0.rh9.dag.i386.rpm

`snort-utils-acid`

Version: *0.9.6b23* Type: *tgz* Home URL: http://www.andrew.cmu.edu/~rdanyliw/snort/ Download URL: http://www.andrew.cmu.edu/~rdanyliw/snort/acid-0.9.6b23.tar.gz

`snort-utils-adodb`

Version: *410* Type: *tgz* Home URL: http://adodb.sourceforge.net/ Download URL: http://phplens.com/lens/dl/adodb410.tgz

`snort-utils-jpgraph`

Version: *1.13* Type: *tgz* Home URL: http://www.aditus.nu/jpgraph/ Download URL: http://www.aditus.nu/jpgraph/downloads/jpgraph-1.13.tar.gz

`hping`

Version: *2.0.0-0.rc3* Type: *rpm* Home URL: http://www.hping.org/ Download URL: http://apt.sw.be/packages/hping/hping-2.0.0-0.rc3.rh90.dag.i386.rpm

`tcpreplay`

Version: *2.2.2* Type: *tgz* Home URL: http://tcpreplay.sourceforge.net/ Download URL: http://umn.dl.sourceforge.net/sourceforge/tcpreplay/tcpreplay-2.2.2.tar.gz

`kismet`

Version: *2004.04.R1* Type: *tgz* Home URL: http://www.kismetwireless.net/ Download URL: http://www.kismetwireless.net/code/kismet-2004-04-R1.tar.gz

`mbrowse`

Version: *0.3.1-0* Type: *rpm* Home URL: http://www.kill-9.org/mbrowse/ Download URL: http://apt.sw.be/packages/mbrowse/mbrowse-0.3.1-0.dag.rh90.i386.rpm

`nessus`

Version: *2.0.12* Type: *tgz* Home URL: http://www.nessus.org/ Download URL: http://ftp.nessus.org/nessus/nessus-2.0.12/src/nessus-libraries-2.0.12.tar.gz `nemesis`

Version: *1.4beta3* Type: *tgz* Home URL: http://www.packetfactory.net/projects/nemesis/ Download URL: http://www.packetfactory.net/projects/nemesis/nemesis-1.4beta3.tar.gz

`ipsc`

Version: *0.4.3-1* Type: *rpm* Home URL: http://dag.wieers.com/packages/ipsc/ Download URL: http://apt.sw.be/packages/ipsc/ipsc-0.4.3-1.dag.rh90.i386.rpm

`john`

Version: *1.6-1* Type: *rpm* Home URL: http://www.openwall.com/john/ Download URL: http://www.megaloman.com/~hany/_data/RPM/doors4.0/john-1.6-1.i686.rpm

`nikto`

Version: *1.32* Type: *tgz* Home URL: http://www.cirt.net/code/nikto.shtml Download URL: http://www.cirt.net/nikto/nikto-current.tar.gz

`airsnort`

Version: *0.2.4-0.a* Type: *rpm* Home URL: http://airsnort.shmoo.com/ Download URL: http://apt.sw.be/packages/airsnort/airsnort-0.2.4-0.a.rh90.dag.i386.rpm

`ifgraph`

Version: *0.4.10rc1* Type: *tgz* Home URL: http://ifgraph.sourceforge.net/ Download URL: http://ifgraph.sourceforge.net/candidate/ifgraph-0.4.10rc1.tar.gz

`ngrep`

Version: *1.42-1* Type: *rpm* Home URL: http://ngrep.sourceforge.net/ Download URL: http://apt.sw.be/packages/ngrep/ngrep-1.42-1.rh90.dag.i386.rpm

`tsclient`

Version: *0.132-1* Type: *rpm* Home URL: http://www.gnomepro.com/tsclient/ Download URL: http://www.gnomepro.com/tsclient/tsclient-0.132-1.i386.rpm

`hammerhead`

Version: *2.1.3-0* Type: *rpm* Home URL: http://hammerhead.sourceforge.net/ Download URL: http://apt.sw.be/packages/hammerhead/hammerhead-2.1.3-0.dag.rh90.i386.rpm

`linux-wlan-ng`

Version: *0.2.1-pre21* Type: *tgz* Home URL: http://www.linux-wlan.org/ Download URL: ftp://ftp.linux-wlan.org/pub/linux-wlan-ng/linux-wlan-ng-0.2.1pre21.tar.gz

`phpMyAdmin`

Version: *2.5.7* Type: *tgz* Home URL: http://www.phpmyadmin.net/home_page/ Download URL: http://umn.dl.sourceforge.net/sourceforge/phpmyadmin/phpMyAdmin-2.5.7.tar.gz

`nmap`

Version: *3.55-1* Type: *rpm* Home URL: http://www.insecure.org/nmap/ Download URL: http://download.insecure.org/nmap/dist/nmap-3.55-1.i386.rpm `nmap-frontend`

Version: *3.55-1* Type: *rpm* Home URL: http://www.insecure.org/nmap/ Download URL: http://download.insecure.org/nmap/dist/nmap-frontend-3.55-1.i386.rpm

`tcpreen`

Version: *1.2.4-1* Type: *rpm* Home URL: http://www.simphalempin.com/dev/tcpreen/ Download URL: http://aleron.dl.sourceforge.net/sourceforge/tcpreen/tcpreen-1.2.4-1.i586.rpm

`firefox`

Version: *0.9* Type: *tgz* Home URL: http://www.mozilla.org/products/firefox/ Download URL: http://ftp.mozilla.org/pub/mozilla.org/firefox/releases/0.9/firefox-0.9-i686-linux-gtk2+xft.tar.gz

`wipe`

Version: *2.1.0* Type: *bz2* Home URL: http://wipe.sourceforge.net/ Download URL: http://umn.dl.sourceforge.net/sourceforge/wipe/wipe-2.1.0.tar.bz2

`syslinux`

Version: *2.10-1* Type: *rpm* Home URL: http://syslinux.zytor.com/ Download URL: ftp://ftp.kernel.org/pub/linux/utils/boot/syslinux/RPMS/syslinux-2.10-1.i386.rpm

`docbook-xsl`

Version: *1.64.1* Type: *tgz* Home URL: http://docbook.sourceforge.net/ Download URL: http://umn.dl.sourceforge.net/sourceforge/docbook/docbook-xsl-1.64.1.tar.gz

`tcpstat`

Version: *1.5* Type: *tgz* Home URL: http://www.frenchfries.net/paul/tcpstat/ Download URL: http://www.frenchfries.net/paul/tcpstat/tcpstat-1.5.tar.gz

`gdome2`

Version: *0.8.1* Type: *rpm* Home URL: http://gdome2.cs.unibo.it/index.html Download URL: http://gdome2.cs.unibo.it/rpm/gdome2-0.8.1-1.i386.rpm

`gkismet`

Version: *0.0.8* Type: *tgz* Home URL: http://gkismet.sourceforge.net/ Download URL: http://gkismet.sourceforge.net/gkismet-0.0.8.tar.gz

`etherape`

Version: *0.9.0* Type: *tgz* Home URL: http://etherape.sourceforge.net Download URL: http://unc.dl.sourceforge.net/sourceforge/etherape/etherape-0.9.0.tar.gz

`dvd+rw-tools`

Version: *5.3.4.2.4* Type: *rpm* Home URL: http://fy.chalmers.se/~appro/linux/DVD+RW/ Download URL: http://download.fedora.us/fedora/redhat/9/i386/RPMS.testing/dvd+rw-tools-5.3.4.2.4-0.fdr.3.rh90.i386.rpm `elinks`

Version: *0.9.1* Type: *bz2* Home URL: http://elinks.or.cz/ Download URL: http://elinks.or.cz/download/elinks-0.9.1.tar.bz2

`rdesktop`

Version: *1.3.1* Type: *tgz* Home URL: http://www.rdesktop.org/ Download URL: http://unc.dl.sourceforge.net/sourceforge/rdesktop/rdesktop-1.3.1.tar.gz

`libdnet`

Version: *1.7-0* Type: *rpm* Home URL: http://libdnet.sourceforge.net/ Download URL: http://ftp.lug.ro/mirror/apt.sw.be/dag/packages/libdnet/libdnet-1.7-0.dag.rh90.i386.rpm

`libevent`

Version: *0.8* Type: *tgz* Home URL: http://monkey.org/~provos/libevent/ Download URL: http://monkey.org/~provos/libevent-0.8.tar.gz

`honeyd`

Version: *0.8b* Type: *tgz* Home URL: http://www.honeyd.org/ Download URL: http://www.citi.umich.edu/u/provos/honeyd/honeyd-0.8b.tar.gz

`firewalk`

Version: *5.0-1* Type: *rpm* Home URL: http://www.packetfactory.net/projects/firewalk/ Download URL: http://apt.sw.be/packages/firewalk/firewalk-5.0-1.rh90.dag.i386.rpm

`firestarter`

Version: *0.9.2-3* Type: *rpm* Home URL: http://firestarter.sourceforge.net Download URL: http://unc.dl.sourceforge.net/sourceforge/firestarter/firestarter-0.9.2-3.i386.rpm

`nttcp`

Version: *1.47-0* Type: *rpm* Home URL: http://home.leo.org/~elmar/nttcp/ Download URL: http://apt.sw.be/packages/nttcp/nttcp-1.47-0.dag.rh90.i386.rpm

`tightvnc`

Version: *1.3dev5* Type: *rpm* Home URL: http://www.tightvnc.com/ Download URL: http://unc.dl.sourceforge.net/sourceforge/vnc-tight/tightvnc-1.3dev5-1.i386.rpm

`tightvnc-server`

Version: *1.3dev5* Type: *rpm* Home URL: http://www.tightvnc.com/ Download URL: http://unc.dl.sourceforge.net/sourceforge/vnc-tight/tightvnc-server-1.3dev5-1.i386.rpm

`bandwidthd`

Version: *1.2.1b* Type: *tgz* Home URL: http://bandwidthd.sourceforge.net/ Download URL: http://unc.dl.sourceforge.net/sourceforge/bandwidthd/bandwidthd-1.2.1b.tgz

`iperf`

Version: *1.7.0* Type: *tgz* Home URL: http://dast.nlanr.net/Projects/Iperf/ Download URL: http://dast.nlanr.net/Projects/Iperf/iperf-1.7.0-source.tar.gz `packETH`

Version: *1.2* Type: *tgz* Home URL: http://packeth.sourceforge.net/ Download URL: http://unc.dl.sourceforge.net/sourceforge/packeth/packETH-1.2.tar.gz

`nbtscan`

Version: *1.5.1* Type: *tgz* Home URL: http://www.inetcat.org/software/nbtscan.html Download URL: http://www.inetcat.org/software/nbtscan-1.5.1.tar.gz

`tcptrack`

Version: *1.1.2-1* Type: *rpm* Home URL: http://www.rhythm.cx/~steve/devel/tcptrack/ Download URL: http://www.rhythm.cx/~steve/devel/tcptrack/release/1.1.2/binaries/rpm/rh9/tcptrack-1.1.2-1.i386.rpm

`bing`

Version: *1.3.5* Type: *tgz* Home URL: http://fgouget.free.fr/bing/bing_src-readme-1st.shtml Download URL: http://fgouget.free.fr/bing/bing_src-1.3.5.tar.gz

`etherwake`

Version: *1.08* Type: *tgz* Home URL: http://packages.debian.org/stable/net/etherwake Download URL: http://http.us.debian.org/debian/pool/main/e/etherwake/etherwake_1.08.orig.ta

`gtkspell`

Version: *2.0.4-0* Type: *rpm* Home URL: http://gtkspell.sourceforge.net/ Download URL: http://apt.sw.be/packages/gtkspell/gtkspell-2.0.4-0.dag.rh90.i386.rpm

`gaim`

Version: *0.79-0rh9* Type: *rpm* Home URL: http://gaim.sourceforge.net/ Download URL: http://unc.dl.sourceforge.net/sourceforge/gaim/gaim-0.79-0rh9.i386.rpm

`nload`

Version: *0.6.0* Type: *tgz* Home URL: http://www.roland-riegel.de/nload/index.html?lang=en Download URL: http://unc.dl.sourceforge.net/sourceforge/nload/nload-0.6.0.tar.gz

`libpcap`

Version: *0.8.3* Type: *tgz* Home URL: http://www.tcpdump.org/ Download URL: http://www.tcpdump.org/release/libpcap-0.8.3.tar.gz

`tcpdump`

Version: *3.8.3* Type: *tgz* Home URL: http://www.tcpdump.org/ Download URL: http://www.tcpdump.org/release/tcpdump-3.8.3.tar.gz

`fping`

Version: *2.4-0.b2* Type: *rpm* Home URL: http://www.fping.com/ Download URL: http://apt.sw.be/packages/fping/fping-2.4-0.b2.dag.rh90.i386.rpm

`argus`

Version: *3.3* Type: *tgz* Home URL: http://argus.tcp4me.com/ Download URL: http://www.tcp4me.com/code/argus-archive/argus-3.3.tgz `chntpw`

Version: *0.0.20040116* Type: *rpm* Home URL: http://home.eunet.no/~pnordahl/ntpasswd Download URL: http://dag.wieers.com/packages/chntpw/chntpw-0.0.20040116-2.rh90.dag.i386.rpm

NTFS Support

Adding read support for the NTFS file system is a rather non-trivial step on a `Red Hat Linux 9`[160] system. You basically have two choices:

- Build your own custom kernel and enable NTFS file system support. This means you'll constantly need to build your entire kernel on your development system each time `Red Hat Linux 9`[161] releases a new version with security patches.
- Download the appropriate RPM from http://linux-ntfs.sourceforge.net/ for your kernel and install it.

We typically use option two as it is less time consuming. If you feel particularily lucky, you can use the **bin/ntfs_fetch_install** script to download a good guess at the appropriate RPM for your system. At the time of this writing, this script only downloads the RPM and tells you what you need to do to actually install it (Paul was hesitant to automate the installation of any modules into the active kernel on your development system).

Once installed, your Network Security Toolkit will then be capable of mounting NTFS file systems in read only mode. This can be particularily handy in the recovery of files from Windows machines which are in trouble.

Notes

1. https://www.redhat.com/support/resources/howto/rhl9.html
2. http://rhn.redhat.com/
3. https://www.redhat.com/support/resources/howto/rhl9.html
4. http://linux-ntfs.sourceforge.net/
5. http://unc.dl.sourceforge.net/linux-ntfs/kernel-ntfs-2.4.20-31.9.i686.rpm
6. http://gort.metaparadigm.com/mapscsi/
7. http://freshmeat.net/redir/mapscsi/20369/url_tgz/mapscsi-0.0.11.tar.gz
8. http://www.visi.com/~hawkeyd/vtwm.html
9. http://www.visi.com/cgi-bin/cgiwrap/~hawkeyd/dnldcount.cgi?/ftp/users/hawkeyd/X/vtwm-5.4.6a.tar.gz
10. http://www.packetfactory.net/projects/libnet/
11. http://www.packetfactory.net/libnet/dist/libnet.tar.gz
12. http://www.citi.umich.edu/u/provos/honeyd/
13. http://www.spenneberg.com/redirect.php?url=public/Honeypot/arpd/arpd-0.2-rh8_2.i386.rpm
14. http://ettercap.sourceforge.net/
15. http://unc.dl.sourceforge.net/sourceforge/ettercap/ettercap-NG-0.7.0.tar.gz
16. http://www.pygps.org/gpsd/gpsd.html
17. http://www.pygps.org/gpsd/downloads/gpsd-1.07.tar.gz

18. http://www.engelschall.com/sw/smtpclient/
19. http://www.engelschall.com/sw/smtpclient/distrib/smtpclient-1.0.0.tar.gz
20. http://linuxberg.matrix.com.br/home/preview/165246.html
21. http://linuxberg.matrix.com.br/files/gtkcalc-1.0-1.i386.rpm
22. http://search.cpan.org/dist/Crypt-DES/
23. http://redhat.ifsic.univ-rennes1.fr/contrib/libc6/i386/perl-Crypt-DES-2.03-1.i386.rpm
24. http://search.cpan.org/dist/Net-SNMP/
25. http://dag.wieers.com/packages/perl-Net-SNMP/perl-Net-SNMP-4.1.0-0.dag.rh90.noarch.rpm
26. http://search.cpan.org/dist/XML-Simple/
27. http://dag.wieers.com/packages/perl-XML-Simple/perl-XML-Simple-2.08-0.dag.rh90.noarch.rpm
28. http://www.ntop.org/
29. http://unc.dl.sourceforge.net/sourceforge/ntop/ntop-3.0-0.i386.rpm
30. http://www.sys-security.com/html/projects/X.html
31. http://www.sys-security.com/archive/tools/xprobe2/xprobe2-0.2.tar.gz
32. http://www.ethereal.com/
33. http://www.ethereal.com/distribution/ethereal-0.10.6.tar.bz2
34. http://www.pcre.org/
35. http://unc.dl.sourceforge.net/sourceforge/pcre/pcre-4.3.tar.bz2
36. http://www.chiark.greenend.org.uk/~ian/adns/
37. http://www.chiark.greenend.org.uk/~ian/adns/ftp/adns-1.1.tar.gz
38. http://sourceforge.net/projects/pwgen
39. http://unc.dl.sourceforge.net/sourceforge/pwgen/pwgen-2.03.tar.gz
40. http://people.ee.ethz.ch/~oetiker/webtools/rrdtool/
41. http://people.ee.ethz.ch/~oetiker/webtools/rrdtool/pub/rrdtool-1.0.41-1.8.0.i386.rpm
42. http://www.snort.org/
43. http://dag.wieers.com/packages/snort/snort-2.2.0-1.0.rh9.dag.i386.rpm
44. http://www.snort.org/snort-db/
45. http://www.snort.org/dl/rules/snortrules-snapshot-2_2.tar.gz
46. http://www.snort.org/
47. http://dag.wieers.com/packages/snort/snort-mysql-2.2.0-1.0.rh9.dag.i386.rpm
48. http://www.andrew.cmu.edu/~rdanyliw/snort/
49. http://www.andrew.cmu.edu/~rdanyliw/snort/acid-0.9.6b23.tar.gz
50. http://adodb.sourceforge.net/
51. http://phplens.com/lens/dl/adodb410.tgz
52. http://www.aditus.nu/jpgraph/
53. http://www.aditus.nu/jpgraph/downloads/jpgraph-1.13.tar.gz
54. http://www.hping.org/
55. http://apt.sw.be/packages/hping/hping-2.0.0-0.rc3.rh90.dag.i386.rpm

56. http://tcpreplay.sourceforge.net/
57. http://umn.dl.sourceforge.net/sourceforge/tcpreplay/tcpreplay-2.2.2.tar.gz
58. http://www.kismetwireless.net/
59. http://www.kismetwireless.net/code/kismet-2004-04-R1.tar.gz
60. http://www.kill-9.org/mbrowse/
61. http://apt.sw.be/packages/mbrowse/mbrowse-0.3.1-0.dag.rh90.i386.rpm
62. http://www.nessus.org/
63. http://ftp.nessus.org/nessus/nessus-2.0.12/src/nessus-libraries-2.0.12.tar.gz
64. http://www.packetfactory.net/projects/nemesis/
65. http://www.packetfactory.net/projects/nemesis/nemesis-1.4beta3.tar.gz
66. http://dag.wieers.com/packages/ipsc/
67. http://apt.sw.be/packages/ipsc/ipsc-0.4.3-1.dag.rh90.i386.rpm
68. http://www.openwall.com/john/
69. http://www.megaloman.com/~hany/_data/RPM/doors4.0/john-1.6-1.i686.rpm
70. http://www.cirt.net/code/nikto.shtml
71. http://www.cirt.net/nikto/nikto-current.tar.gz
72. http://airsnort.shmoo.com/
73. http://apt.sw.be/packages/airsnort/airsnort-0.2.4-0.a.rh90.dag.i386.rpm
74. http://ifgraph.sourceforge.net/
75. http://ifgraph.sourceforge.net/candidate/ifgraph-0.4.10rc1.tar.gz
76. http://ngrep.sourceforge.net/
77. http://apt.sw.be/packages/ngrep/ngrep-1.42-1.rh90.dag.i386.rpm
78. http://www.gnomepro.com/tsclient/
79. http://www.gnomepro.com/tsclient/tsclient-0.132-1.i386.rpm
80. http://hammerhead.sourceforge.net/
81. http://apt.sw.be/packages/hammerhead/hammerhead-2.1.3-0.dag.rh90.i386.rpm
82. http://www.linux-wlan.org/
83. ftp://ftp.linux-wlan.org/pub/linux-wlan-ng/linux-wlan-ng-0.2.1pre21.tar.gz
84. http://www.phpmyadmin.net/home_page/
85. http://umn.dl.sourceforge.net/sourceforge/phpmyadmin/phpMyAdmin-2.5.7.tar.gz
86. http://www.insecure.org/nmap/
87. http://download.insecure.org/nmap/dist/nmap-3.55-1.i386.rpm
88. http://www.insecure.org/nmap/
89. http://download.insecure.org/nmap/dist/nmap-frontend-3.55-1.i386.rpm
90. http://www.simphalempin.com/dev/tcpreen/
91. http://aleron.dl.sourceforge.net/sourceforge/tcpreen/tcpreen-1.2.4-1.i586.rpm
92. http://www.mozilla.org/products/firefox/
93. http://ftp.mozilla.org/pub/mozilla.org/firefox/releases/0.9/firefox-0.9-i686-linux-gtk2+xft.tar.gz

94. http://wipe.sourceforge.net/
95. http://umn.dl.sourceforge.net/sourceforge/wipe/wipe-2.1.0.tar.bz2
96. http://syslinux.zytor.com/
97. ftp://ftp.kernel.org/pub/linux/utils/boot/syslinux/RPMS/syslinux-2.10-1.i386.rpm
98. http://docbook.sourceforge.net/
99. http://umn.dl.sourceforge.net/sourceforge/docbook/docbook-xsl-1.64.1.tar.gz
100. http://www.frenchfries.net/paul/tcpstat/
101. http://www.frenchfries.net/paul/tcpstat/tcpstat-1.5.tar.gz
102. http://gdome2.cs.unibo.it/index.html
103. http://gdome2.cs.unibo.it/rpm/gdome2-0.8.1-1.i386.rpm
104. http://gkismet.sourceforge.net/
105. http://gkismet.sourceforge.net/gkismet-0.0.8.tar.gz
106. http://etherape.sourceforge.net
107. http://unc.dl.sourceforge.net/sourceforge/etherape/etherape-0.9.0.tar.gz
108. http://fy.chalmers.se/~appro/linux/DVD+RW/
109. http://download.fedora.us/fedora/redhat/9/i386/RPMS.testing/dvd+rw-tools-5.3.4.2.4-0.fdr.3.rh90.i386.rpm
110. http://elinks.or.cz/
111. http://elinks.or.cz/download/elinks-0.9.1.tar.bz2
112. http://www.rdesktop.org/
113. http://unc.dl.sourceforge.net/sourceforge/rdesktop/rdesktop-1.3.1.tar.gz
114. http://libdnet.sourceforge.net/
115. http://ftp.lug.ro/mirror/apt.sw.be/dag/packages/libdnet/libdnet-1.7-0.dag.rh90.i386.rpm
116. http://monkey.org/~provos/libevent/
117. http://monkey.org/~provos/libevent-0.8.tar.gz
118. http://www.honeyd.org/
119. http://www.citi.umich.edu/u/provos/honeyd/honeyd-0.8b.tar.gz
120. http://www.packetfactory.net/projects/firewalk/
121. http://apt.sw.be/packages/firewalk/firewalk-5.0-1.rh90.dag.i386.rpm
122. http://firestarter.sourceforge.net
123. http://unc.dl.sourceforge.net/sourceforge/firestarter/firestarter-0.9.2-3.i386.rpm
124. http://home.leo.org/~elmar/nttcp/
125. http://apt.sw.be/packages/nttcp/nttcp-1.47-0.dag.rh90.i386.rpm
126. http://www.tightvnc.com/
127. http://unc.dl.sourceforge.net/sourceforge/vnc-tight/tightvnc-1.3dev5-1.i386.rpm
128. http://www.tightvnc.com/
129. http://unc.dl.sourceforge.net/sourceforge/vnc-tight/tightvnc-server-1.3dev5-1.i386.rpm
130. http://bandwidthd.sourceforge.net/

131. http://unc.dl.sourceforge.net/sourceforge/bandwidthd/bandwidthd-1.2.1b.tgz
132. http://dast.nlanr.net/Projects/Iperf/
133. http://dast.nlanr.net/Projects/Iperf/iperf-1.7.0-source.tar.gz
134. http://packeth.sourceforge.net/
135. http://unc.dl.sourceforge.net/sourceforge/packeth/packETH-1.2.tar.gz
136. http://www.inetcat.org/software/nbtscan.html
137. http://www.inetcat.org/software/nbtscan-1.5.1.tar.gz
138. http://www.rhythm.cx/~steve/devel/tcptrack/
139. http://www.rhythm.cx/~steve/devel/tcptrack/release/1.1.2/binaries/rpm/rh9/tcptrack-1.1.2-1.i386.rpm
140. http://fgouget.free.fr/bing/bing_src-readme-1st.shtml
141. http://fgouget.free.fr/bing/bing_src-1.3.5.tar.gz
142. http://packages.debian.org/stable/net/etherwake
143. http://http.us.debian.org/debian/pool/main/e/etherwake/etherwake_1.08.orig.tar.gz
144. http://gtkspell.sourceforge.net/
145. http://apt.sw.be/packages/gtkspell/gtkspell-2.0.4-0.dag.rh90.i386.rpm
146. http://gaim.sourceforge.net/
147. http://unc.dl.sourceforge.net/sourceforge/gaim/gaim-0.79-0rh9.i386.rpm
148. http://www.roland-riegel.de/nload/index.html?lang=en
149. http://unc.dl.sourceforge.net/sourceforge/nload/nload-0.6.0.tar.gz
150. http://www.tcpdump.org/
151. http://www.tcpdump.org/release/libpcap-0.8.3.tar.gz
152. http://www.tcpdump.org/
153. http://www.tcpdump.org/release/tcpdump-3.8.3.tar.gz
154. http://www.fping.com/
155. http://apt.sw.be/packages/fping/fping-2.4-0.b2.dag.rh90.i386.rpm
156. http://argus.tcp4me.com/
157. http://www.tcp4me.com/code/argus-archive/argus-3.3.tgz
158. http://home.eunet.no/~pnordahl/ntpasswd
159. http://dag.wieers.com/packages/chntpw/chntpw-0.0.20040116-2.rh90.dag.i386.rpm
160. https://www.redhat.com/support/resources/howto/rhl9.html
161. https://www.redhat.com/support/resources/howto/rhl9.html
162. http://linux-ntfs.sourceforge.net/

Chapter 3. Documentation

Background

Currently there are two classes of documentation associated with this project. The HTML documentation makes use of a Java preprocessor. The "book" type of documentation is written in DocBook[1] following the guidelines provided by the *FreeBSD Documentation Project Primer for New Contributors*[2]. Both of these forms of documentation attempt to separate style from content. There is a longer learning curve in using them, but result in a very nice way to maintain documents.

Requirements

In order to build the documentation, you will need the following set of tools:

- A Java run time environment the `/usr/bin/gij` which comes with a full `Red Hat Linux 9`[3] installation is sufficient.
- The *ccg.jar* file should be listed in your CLASSPATH environment variable. This should automatically be downloaded for you if it is needed when you run the **configure** command. If your build machine is not connected to the Internet, you can get the necessary JAR file at http://www.mekwin.com/jar/ccg.jar by using a system that is.
- The *docbook-utils* and *docbook-utils-pdf* packages (these are bundled with a `Red Hat Linux 9`[5] distribution).
- You will need to install the latest `docbook-xsl`[6] style sheets in order to get the proper HTML output. Use the command: **make package-check** to verify that you have the proper version. If your system doesn't have the proper version installed, you will want to run the **src/bin/docbook-xsl_fetch_install** to download and install them in the proper location for your Network Security Toolkit build.

Building The Documentation

By default, documents are placed under the directory `tmp/public_html/nst` and `tmp/public_html/nstwui` under the source directory. You can change this by passing the `--html-dir DIR` option to the **configure** script.

Personally, I like to specify `--html-dir $HOME/public_html`. This allows me to view the documentation on my development system (which has a web server running), by pointing a browser at `http://localhost/${USER}/nst/index.html`.

The following assumes you have a copy of the Network Security Toolkit source code under the `$HOME/nst` directory.

If you have not yet run the **configure**, you will need to do so now.

```
[pkb@salsa pkb]$ cd $HOME/nst
[pkb@salsa nst]$ ./configure
```

To build all of the documentation at once:

```
[pkb@salsa pkb]$ cd $HOME/nst
[pkb@salsa nst]$ make docs
```

You can check to see what files were created using the following commands (assuming you used the defaults when you ran **configure**):

```
[pkb@salsa nst]$ find tmp/public_html
```

If you are a developer working on the documentation, you can build portions of the documentation individually.

To build JUST the DocBook documents:

```
[pkb@salsa pkb]$ cd $HOME/nst/docs
[pkb@salsa docs]$ make
```

As the DocBook documents can take a long time to build, you may find yourself in the situation where you're saying "Dang, I wish there was a way to just build this one document!" Well, your lucky day has come. You can invoke **make help** under the `docs` directory (be prepared to pipe this into **less**) to see all of the individual documents that can be built. To use this feature, you should invoke the **make setup** at least once (this will install the shared files for you). For example, to build JUST the HTML version of this technical manual:

```
[pkb@salsa pkb]$ cd $HOME/nst/docs
[pkb@salsa docs]$ make setup
[pkb@salsa docs]$ make tech.html
```

As it turns out, large DocBook[7] files can take a long time to build. If you are working on a particular section of a document, it is possible to build just that section. This will further reduce your compile time. However, it should be noted, that the sub document compiled will take the place of the entire parent document. For example, the following commands would compile the `docs/user/vpn.xml` to its HTML form and and install it as if it were the entire document known as: Using the Network Security Toolkit[8].

```
[pkb@salsa pkb]$ cd $HOME/nst/docs
[pkb@salsa docs]$ make setup
[pkb@salsa docs]$ make user.vpn
```

To build JUST the web site HTML:

```
[pkb@salsa pkb]$ cd $HOME/nst/html
[pkb@salsa html]$ make
```

To build the JUST the web base user interface:

```
[pkb@salsa pkb]$ cd $HOME/nst/wui
[pkb@salsa wui]$ make
```

Tweaking the HTML Output

It took a long time to figure this out, but we finally managed to adjust the output of the HTML pages generated when compiling the documentation. In order to do this, we needed to:

- Create a custom `chunk.xsl` style sheet.
- Create a custom `docbook.css` file - which controls how the various parts of a document are rendered.
- Create and install auxillary files (bitmaps) referenced by the above two files.

The `docs/chunk.xsl` File

In order to tweak the HTML produced by our DocBook[9] source files, we needed to create a custom XSL style sheet. This custom file includes the normal DocBook[10] formatting rules, and then adjusts a few settings to our liking (we were particularily interested in the inclusion of a custom CSS file). The `docs/chunk.xsl` is built from code within the `docs/configure` shell script. If you want to make permanent changes, you will need to modify `docs/configure`. The following shows the contents of `docs/chunk.xsl` at the time this document was created:

```
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version=’1.0’>

<!--

  WARNING: This file was automatically genereted at the time of
  configuration. Do not edit directly! Edit the file docs/configure
  instead!

  This style sheet is used to tweak the docbook output. More
  information can be found at:

    http://docbook.sourceforge.net/release/xsl/current/doc/html/index.html

  Also, try searching google.com with the following in double quotes:

    Customizing DocBook XSL stylesheets

  To tweak colors, brush up on CSS, and have a crack at the file: docbook.css

  -->

  <xsl:import href="file:///usr/share/sgml/docbook/xsl-stylesheets-1.65.1-1/html/chunk.

  <!-- Indicate we will be including our own custom style sheet -->

  <xsl:param name="html.stylesheet">../docbook.css</xsl:param>

  <!-- Turn off rules below/above header/footer - we’ll use CSS borders -->

  <xsl:param name="header.rule" select="0"></xsl:param>
  <xsl:param name="footer.rule" select="0"></xsl:param>

  <!-- Turn off image scaling when generating HTML output -->

  <xsl:param name="make.graphic.viewport" select="1"></xsl:param>
  <xsl:param name="ignore.image.scaling" select="1"></xsl:param>

  <!-- Make external links stay within window (don’t let them take
  over entire frame) -->

  <xsl:param name="ulink.target" select="’_self’"></xsl:param>

  <!-- Allow the use of role attribute to override class in <entry>,
       <para>, <emphasis> and <phrase> objects This allows us to
       customize the CSS attributes in HTML output -->

  <xsl:param name="entry.propogates.style" select="1" />
  <xsl:param name="emphasis.propogates.style" select="1" />
  <xsl:param name="para.propogates.style" select="1" />
  <xsl:param name="phrase.propogates.style" select="1" />

  <!-- Spend extra CPU time to try and produce valid and pretty HTML -->

  <xsl:param name="html.cleanup" select="1" />
  <xsl:param name="make.valid.html" select="1" />

  <!-- callout customizations -->

  <xsl:param name="callout.graphics" select="’1’" />
  <xsl:param name="callouts.extension" select="’1’" />
  <xsl:param name="callout.graphics.path" select="’../images/callouts/’" />
```

```
</xsl:stylesheet>
```

There are many "dials" that can be adjusted in the `docs/chunk.xsl` file. To learn more about these various settings, read the DocBook HTML Parameter Reference[11]. Also, you may want to try searching Google for Customizing DocBook XSL stylesheets[12].

The `docs/docbook.css` File

If you examine the `docs/chunk.xsl`, you'll notice that it specifies a custom `html.stylesheet`. This CSS file is found at `docs/docbook.css` and controls the colors, fonts and formatting of the HTML pages. It currently has the form:

```
/* $Id: docbook.css,v 1.9 2004/06/17 23:05:36 pblankenbaker Exp $

   CSS formats for docbook generation
*/

/* Adjust docbook <table></table> areas */

div.table th {
  background: #000030;
  color: white;
}

div.table td {
  background: #ddddff;
  color: black;
}

div.table td.even {
  color: black;
  background-color : white;
}

div.table td.odd {
  color: black;
  background-color : #ddddff;
}

div.table td.total {
  color: black;
  font-weight: bold;
  font-size: 12pt;
  background-color : #FFFF66;
}

/* <note></note> <warning></warning> and <error></error> sections */

div.note {
  background-color : #eeffff;
}

div.warning {
  background-color : #ffffaa;
}

div.error {
  background-color : #ffeeee;
}

/* Adjustments to docbook <screen></screen> areas */
```

```
pre.screen {
  color: #33FF33;
  background-color : black;
  border-style: solid;
  border-width: 4px;
  border-color: darkgray;
  margin: 24pt;
}

pre.screen b {
  color: yellow;
}

/* Adjust docbook <programlisting></programlisting> areas */

pre.programlisting {
  border-style: solid;
  border-width: 4px;
  border-color: green;
  background: url(../images/computer_paper.png);
  margin: 24pt;
}

/* Adjust docbook <command></command> areas */
.command {
  font-weight : bold;
}

/* Adjust the header/footer regions */

.navheader {
  color: #f7f0d7;
  background-color : #000030;
  font-size: 8pt;
}

.navfooter {
  color: #f7f0d7;
  background-color : #000030;
  font-size: 8pt;
}
```

There are several things that should be noted in the `docs/docbook.css` file:

- Colors can be applied based upon the original DocBook type. For example, the `pre.screen` adjusts the color settings for <pre> sections of the resulting HTML documents, but only if they are given the class of `screen`. To find the class of DocBook output, one needs to review the raw HTML produced by DocBook. Hint: Try viewing the source of this page and search for the string `programlisting`.
- The `pre.programlisting` defines a bitmap to be used for the background. The reference to the bitmap is relative to the location of the `docbook.css` file. Hence, the developer needs to be careful when adding bitmaps. It involves not only creating the actual bitmap, but also updating the make rules such that it will be installed in the proper location.

Adding New Documents

If you are a developer interested in adding new documents to the Network Security Toolkit project, there are a couple of things to consider.

If you add any new files to the document structure (this is especially true for the DocBook XML files), you will need to re-run the **configure** command so that the `Makefiles` are re-generated to pick up the newly added files.

Notes

1. http://www.docbook.org/
2. http://www.freebsd.org/doc/en_US.ISO8859-1/books/fdp-primer/
3. https://www.redhat.com/support/resources/howto/rhl9.html
4. http://www.mekwin.com/jar/ccg.jar
5. https://www.redhat.com/support/resources/howto/rhl9.html
6. http://docbook.sourceforge.net/
7. http://www.docbook.org/
8. ../user/index.html
9. http://www.docbook.org/
10. http://www.docbook.org/
11. http://docbook.sourceforge.net/release/xsl/current/doc/html/index.html
12. http://www.google.com/search?q=%2B%22Customizing+DocBook+XSL+stylesheets%22&sourceid=r
search&start=0&start=0&ie=utf-8&oe=utf-8

Chapter 4. Creating A Release

Periodically, developers associated with the Network Security Toolkit project will create a public release of the Network Security Toolkit. The following provides a very brief outline of the steps involved in creating a release:

Review `include/dist/configure.sh`

Review and update the `include/dist/configure.sh` file. Make sure it is consistent with the environment variables produced by the current top level **configure** command. Pay particular attention to the KERVER setting as it is apt to change between releases.

Check `RELEASE` Values

Check the values of the `RELEASE`, `RELEASE_LAST` and `RELEASE_NEXT` variables found near the top of the root `configure` script. Just to be safe, invoke **configure** to make sure that the release value is set properly in the `config/config.*` files.

```
[root@quesadilla nst]# emacs -nw configure
[root@quesadilla nst]# ./configure
[root@quesadilla nst]#
```

Update `html/log/changelog.at`

Update the `html/log/changelog.at` file to indicate that the release has been closed (no more changes for the release). This involves changing the first line of the file from:

```
@changeLogBegin("@release()")
```

To a hard coded release number followed by the date of the release. For example, we changed it to the following for the `1.0.4` release:

```
@changeLogBegin("1.0.4","2003-04-06")
```

Run nstcvs commit

Make sure that you invoke **nstcvs commit** to verify that all modifications are checked in.

```
[root@quesadilla nst]# nstcvs commit
cvs commit: Examining .

... lots of CVS "Examining" messages
    if a editor window asking for a comment appears
    then you still have files that are not checked in ...

cvs commit: Examining wui/include/at
cvs commit: Examining wui/include/make
[root@quesadilla nst]#
```

Run make release-tag

Use the **make release-tag** command to tag (or re-tag) all of the files for the release.

```
[root@quesadilla nst]# make release-tag

... lots of CVS messages - may pause several times ...

[root@quesadilla nst]#
```

Run make release

Use the **make release** to produce a source tarball of the files making up the release and the ISO image. These are the two files which will be released to the world at large and producing them takes a LONG time.

```
[root@quesadilla nst]# make release
Attempting to check out current source code...

... lots of CVS messages as it checks out a fresh copy and compiles ...

[root@quesadilla nst]#
```

Generate Release Information

Each time a release is created, we will want to generate a release information XML file which provides information about the release. The `release-info` target is used to create this information. HOWEVER, you should only invoke this command if the **make release** you invoked in the prior step completed successfully.

This command takes a long time as it creates the `html/log/release-1.0.6.xml` file.

```
[root@quesadilla cvs_build]# make release-info
Searching for: 'NST_BOOT_PASSWD="' in:
  /opt/nst/cvs_build/src/nst-1.0.6.iso.gz
This takes a while...
Creating: /opt/nst/cvs_build/config/release-1.0.6.at
Creating: /opt/nst/cvs_build/html/log/release-1.0.6.xml
Creating: /opt/nst/cvs_build/html/log/manifest-1.0.6.html
Creating: /opt/nst/cvs_build/html/log/nstisopasswd-1.0.6.bash
Creating: /opt/nst/cvs_build/html/log/manifest-1.0.7.html
Checking CVS status of release-1.0.6.xml...ERROR - failed to add to CVS!
Checking CVS status of manifest-1.0.6.html...ERROR - failed to add to CVS!
Checking CVS status of nstisopasswd-1.0.6.bash...already in CVS
Checking CVS status of manifest-1.0.7.html...ERROR - failed to add to CVS!
[root@quesadilla cvs_build]# (cd html/log; nstcvs update -Pd *1.0.6*)
M manifest-1.0.6.html
cvs update: use ‘cvs add’ to create an entry for nstisopasswd-1.0.6.bash
cvs update: use ‘cvs add’ to create an entry for release-1.0.6.xml
cvs update: use ‘cvs add’ to create an entry for manifest-1.0.7.html
[root@quesadilla cvs_build]#
```

Figure 4-1. Running make release-info

> **Note:** Please notice the errors in the above output. After running the **make release-info** command, the files `html/log/release-1.0.6.xml`, `html/log/manifest-1.0.6.html`, `html/log/manifest-1.0.7.html` and `html/log/nstisopasswd-1.0.6.bash` should be created and added to the CVS repository if they were not already there. Unfortunately, automating the CVS **add** doesn't always work as planned. If you see output similar to the above, you will need to manually add these files to the CVS repository.

> **Note:** You should probably take the time to verify the newly created **nstisopasswd-1.0.6.bash** script, burn, boot and test the ISO image before releasing the ISO image to the world at large.

Transfer to SourceForge[1]

If everything looks good, you can start the process of transferring the source tarball and ISO image to the `FILES` section of the Network Security Toolkit project at SourceForge[2].

Update `RELEASE`

You should immediately change the value of the `RELEASE`, `RELEASE_LAST` and `RELEASE_NEXT` variables at the top of the `configure` script to their next logical values. For example if the last value of `RELEASE` had been `1.0.6` then you should set `RELEASE_LAST` to `1.0.6` and set `RELEASE` to `1.0.7`. Don't worry about making a bad guess for the `RELEASE_NEXT` value - it can be changed at any time in the future. The `configure` file should be committed and all developers should be notified that they need to update their work areas and run the `configure` command. This is important! If you fail to follow this last step and a developer invokes the `make release-tag` then the tags in the CVS repository will no longer agree with the files in the source tarball which you went to so much trouble to release (its not the end of the world, just a pain in the butt).

Start Next Release Section in Changelog

You can go ahead and start a new section in the `html/log/changelog.at` file for the next release you will be working on. It involves adding lines resembling the following to the top of `html/log/changelog.at`:

```
@changeLogBegin("@release()")

@changeLogEntry("2004-04-07"„"Updated necessary HTML files so we can
start working on the next release.")

@changeLogEnd()
```

Figure 4-2. Updating `html/log/changelog.at` For Next Release

Update/Add Manifest Links

As we've created a new release, you will want to add links to the new manifest in the files: `html/links.html`, `html/side.html` and `html/log/manifest.html`. You can find the locations by searching for the string `1.0.5`.

> **Note:** Currently, we haven't removed old manifest information. We will eventually need to change this philosophy and start trimming the old manifest information. If someone really needs it they can always pull it out of the CVS repository.

Commit and Publish HTML Updates

Now that we've finalized the release information, we need to publish this information to the general public. In order to accomplish this, you will want to commit any modified files and then run: **make docs upload**.

Grab a Beer

If you've made it this far, go grab yourself a beer - you've earned it.

Notes

1. http://sourceforge.net/
2. http://sourceforge.net/

Chapter 5. Web User Interface

The Network Security Toolkit system provides such a set of powerful tools, it often becomes difficult to keep track of all of them. It was decided that a Web User Interface (WUI) could provide the following:

- A simple interface for common tasks.
- A way for the developers to write common scripts and comments describing what they wanted to accomplish (its very easy to add features to the Network Security Toolkit today and forget about them and/or how to use them in six months).

The general steps for adding a WUI front end page include:

- Pick a directory (or create a new one) under `wui/cgi-bin` to create your CGI script under. For example, the front end to the **top** command is found at `wui/cgi-bin/system/top.cgi`.
- Copy an existing CGI script to use as a starting point to your new interface. The `wui/cgi-bin/system/top.cgi` works well as a template for simple interfaces.
- Update the top level `wui/index.html` file and add a link to your newly created script.

Here are a few tips to keep in mind when creating your own CGI scripts:

- Use the `Environment` link in the `System` section of the WUI to get an idea of what environment variables your CGI script will have at its disposal.
- Make sure that the first line of your script starts with the `@bashCgiBegin()` macro and that the last line of your script (or prior to any exit point) contains the `@bashCgiEnd()` macro.
- Make sure that you pair each invocation of the `@bashCgiOutBegin()` macro with a corresponding `@bashCgiOutEnd()` invocation (alternatively, you can use `/bin/cat <<EOF` and `EOF` lines to mark sections of text to put out).
- Be careful about indentation. Unfortunately, the `/bin/cat <<EOF` requires the matching `EOF` to be at the start of the line (which means that the `@bashCgiOutEnd()` macro must also be at the start of a line).

A Simple Script

The simplest type of WUI interface you can create is one that runs a standard command line program and captures the output. We provide several macros to help facilitate the process.

The following shows the entire script which makes up the front end to the **top** interface:

```
@bashCgiBegin("$Id: top.cgi,v 1.3 2003/10/28 21:40:26 pblankenbaker Exp $","2003-09-26"

. ../include/form.sh

@bashCgiOutBegin()

@p("The following shows the processes currently running on the system.")

@runCommand("/usr/bin/top -n 1 -b")

@bashCgiOutEnd()

@bashCgiEnd()
```

A State Examination

You will often want to make a front end that inspects the current state of the machine and generates different HTML output based upon the state. The following (`wui/cgi-bin/server/http_logs.cgi`) script provides an example of this. It defines a `show_log()` function which checks for the existance of log files prior to displaying the contents. The `show_log()` function is then invoked several times for the various log files associated with the **httpd** daemon.

```
@bashCgiBegin("$Id: http_logs.cgi,v 1.4 2004/07/01 18:38:15 pblankenbaker Exp $","2003-

. ../include/form.sh

set_query_options

LOG_DIR=/var/log/httpd

@bashCgiOutBegin()

@p("This page shows the log files associated with the @apache() web
server. Use the button below if you want to clear the log files:")

<center>
<table><tr>

<td>
<form action="http_logs.cgi">
<input type="hidden" name="action" value="show">
<input type="submit" value="Refresh">
</form>
</td>

<td>
<form action="http_logs.cgi">
<input type="hidden" name="action" value="clear">
<input type="submit" value="Clear Log Files">
</form>
</td>

</tr></table>
</center>

@bashCgiOutEnd()

# clear_log FILE
#
#    Resets the contents of the log files

clear_log() {
  local LFILE="${LOG_DIR}/${1}";

  if [ -f "${LFILE}" ]; then
    /usr/bin/sudo /bin/rm -f "${LFILE}"
  fi

  /usr/bin/sudo /bin/touch "${LFILE}"
  /usr/bin/sudo /bin/chmod 644 "${LFILE}"
}

# show_log FILE
#
#    Generates output showing contents of FILE

show_log() {

@bashCgiOutBegin()
@heading("${1}")
@bashCgiOutEnd()

  if [ -f "${LOG_DIR}/${1}" ]; then
@bashCgiOutBegin()
<a name="#${1}"></a>
@p("The following shows the contents of @bold("${LOG_DIR}/${1}"):")

@runCommand("/bin/cat ${LOG_DIR}/${1}")

@bashCgiOutEnd()
  else

@bashCgiOutBegin()
@p("@bold("SORRY!") We could not find the file @bold("${LOG_DIR}/${1}").")
@bashCgiOutEnd()

  fi
}

#
# List of log files maintained by apache
#

LOG_FILES=(ssl_error_log ssl_access_log ssl_request_log error_log access_log)

#
# See if user wants us to clear the log files access_log
#

if [ "${QUERY_action:-show}" = "clear" ]; then

  printf "<h2>Clearing Log Files</h2>\n"

  for l in ${LOG_FILES[*]}; do
    clear_log "${l}"
  done

@bashCgiOutBegin()

@p("We have cleared the contents of the log files
(@bold("${LOG_FILES[*]}")) located under the @bold("${LOG_DIR}")
directory. We will now @bold("reload") the web server (you may notice
a slight pause).")

@bashCgiOutEnd()

@bashCgiEnd()

  /usr/bin/sudo /etc/rc.d/init.d/httpd reload > /dev/null 2>&1
  exit 0;
fi

#
# Show index to log files
#

printf "<center><table border=\"1\"><tr><th>Logs</th>\n"
for l in ${LOG_FILES[*]}; do
  printf "<td><a href=\"#${l}\">${l}</a></td>\n"
done
printf "</tr></table></center>\n\n"

#
# Show the access and error log files
#
```

```
for l in ${LOG_FILES[*]}; do
  show_log "${l}"
done

@bashCgiEnd()
```

Forms Processing

Some of the more complex interfaces will provide input fields and buttons for the user to configure and control the Network Security Toolkit. This is the highest level of scripting available. In order to accomplish this, your script will need to do the following:

- Output the necessary HTML to provide the input fields for the end user.
- Determine if the user has already entered values (are we processing a user request). This is optional and only needed if you decide to keep the processing of the form submission within the same script that is used to generate the input form itself.
- Provide output regarding the state of the form processing (or state of the machine). In general, there is typically a lot of similar output generated both when presenting the form to the user and processing the user's request.

The `wui/cgi-bin/networking/ntp_query.cgi` script can be used as a simple example of this. It does the following:

- It uses the standard `@bashCgiBegin()` macro to generate the standard start of a HTML document (keeps the pages generated consistent).
- Includes the file `../include/form.sh` which contains a set of helper functions for processing forms. It then invokes the `set_query_options` function to parse out any form parameters which might be present in the QUERY_STRING. This greatly simplifies life as we can then simply check to see if the QUERY_host value has been set instead of trying to parse the contents of the QUERY_STRING ourselves.
- We then put out information about the page.
- If the QUERY_host is set (which happens once the user presses the submit button to make a request), we will go ahead and run the command and show the results. If the QUERY_host variable is not set, then we omit this output.
- Finally, we generate a HTML form to allow the user to specify a host to make a NTP query against (so the QUERY_host will be set if the user presses the submit button).

The following shows the actual script which does the above:

```
@bashCgiBegin("$Id: ntp_query.cgi,v 1.1 2003/10/23 01:45:50 pblankenbaker Exp $","2003-

. ../include/form.sh

set_query_options

@bashCgiOutBegin()

@p("This page provides allows one check the time at other systems
running the Network Time Protocol (NTP) service.")

@bashCgiOutEnd()

#
# See if we need to make the query
#
```

```
if [ "$QUERY_host" != "" ]; then

@bashCgiOutBegin()

<h2>NTP Query Results</h2>

<p>The following shows the results of our NTP query on
<b>${QUERY_host}</b>:</p>

@runCommand("/usr/local/sbin/ntpdate -q ${QUERY_host}")

@bashCgiOutEnd()

else
  QUERY_host="${SERVER_ADDR}"
fi

#
# Show area to make next query from
#

@bashCgiOutBegin()

<h2>Check Server</h2>

<p>You can specify the IP address or host name of the server you would
like to check below and then press the button to retrieve time
information from the server.</p>

<center>
<form action="ntp_query.cgi">
<input type="text" name="host" value="${QUERY_host}">
<input type="submit" value="Query Time">
</form>
</center>

@bashCgiOutEnd()


@bashCgiEnd()
```

Available/Macros

The following macros can be used when constructing **bash** scripts:

@bashCgiBegin("CVSID","DATE","TITLE",["MIME_TYPE"])

> This macro needs to be at the start of the CGI script. It generates the necessary HTML output for the start of the document produced. The `CVSID` corresponds to the CVS keyword `Id` (enclose with dollar signs). The `DATE` should be the date the script was initially created in the form of `YYYY-MM-DD`. The `TITLE` should be a short title to go with the page. The `MIME_TYPE` parameter is optional and if omitted defaults to `text/html` which is what you'll want 99% of the time.

@bashCgiEnd()

> This macro needs to be at the end of the CGI script. It generates the standard HTML output to end the page.

@bashCgiOutBegin()

This macro can be inserted at any point within the script when you want to insert a block of HTML text to be put out to the user (it inserts `cat <<EOF` into the script). When you are within a `@bashCgiOutBegin()/@bashCgiOutEnd()` block, you are free to include HTML output, the output of commands (by enclosing in `$()` like `$(date)`) or any of the standard `@macro()` commands - there are way to many to list here. You will need to be careful if you need to output either the dollar sign ("$") or at symbol ("@") - trial and error may be needed (try `"\$"` and `@quote("@")`).

@bashCgiOutEnd()

This macro marks the end of a section of HTML output that was inserted into the body of the script. It inserts the text `EOF` into the script, which means that it needs to be left aligned (unfortunately, I don't think you can indent this).

@runCommand("CMD")

This macro inserts the necessary code into your script to show both the invocation of `CMD` and the output (normal and error output). The information will be shown in what appears to be a separate window (it looks good - trust me). It should be noted that `2>&1` is appended to the end of the invocation of `CMD` (so, don't try to redirect standard error yourself). This macro is only useful for commands that don't require user interaction and dump their results to the standard output and error devices.

Helper Scripts

There are many shell script functions to aid in the production of output. These helper functions are squirreled away in the `wui/cgi-bin/include` directory. Typically they are included within a script in a form similar to:

```
. ../include/form.sh
. ../include/service.sh
```

The following sections describe the functions that can be found in each collection.

../include/form.sh

This file contains a set of shell functions that can aid a developer in generating HTML output and process arguments from forms. It includes the following set of functions:

`create_html_ram_disk`

Some of the CGI scripts will produce output files that need to be accessible later on. This function makes sure that a temporary work area for these files is available. After invoking this script, you should be able to use the ramHtmlDir environtment variable to determine the location (at the time of this writing it was set to `/mnt/ram4/html`). It does not hurt to invoke this function more than once (the setup is only done one time). This function returns 0 if the space is available.

`filesize FNAME`

This macro prints out the size of file `FNAME` in kilobytes (like "40K") and returns 0. It prints out "0K" and returns 1 if the file isnt' found.

`html_escape`

This function escapes HTML characters using **sed**. It acts as a filter. For example:

```
echo "<p>" |
    html_escape
```

Would result in:

```
&lt;p&gt;
```

.

`set_query_options`

This is an extremely powerful function for parsing key/value pairs from the QUERY_STRING environment variable and unescaping the special percent characters which might appear. For example:

```
QUERY_STRING='key=Text+To+Submit&host=127.0.0.1'

set_query_options
```

Would find the `key` and `host` values and define QUERY_key to "`Text To Submit`" and QUERY_host to "`127.0.0.1`".

`unpercent ORIG`

This function translates an escaped form parameter to the ASCII string you want (it converts '+' to ' ' and '%XX' to single character represented by hex code). For example `unpercent "P+%42"` would yield `"P B"`.

../include/service.sh

This file contains a set of shell functions that can aid a developer in determining if a certain process or service is running on the system. It includes the following set of functions:

`control_service NAME start|stop`

This function attempts to either `start` or `stop` the service `NAME` (for example `sshd`). It also generates HTML output showing the results of the attempt.

`service_running NAME`

This function returns 0 if the service `NAME` (like `sshd`) appears to be running, or 1 if the service appears to be stopped. It only works for services which support the `status` option.

`service_running_or_show NAME LABEL`

This function returns 0 if the service `NAME` is already running, otherwise, it outputs the necessary HTML so that the user can start the service and returns 1. The `LABEL` is used in the HTML.