Sentry Firewall CD(tm) Reference and Usage Guide

Stephen A. Zarkos, Obsid@Sentry.net

v0.7.0, 2004-08-30

This document is designed to be a comprehensive reference and usage guide for the Sentry Firewall CD. This document in some cases covers some of the same material covered in the Sentry Firewall CD HOWTO and FAQ, but its primary purpose is as a reference and supplement to the other documentation available for the Sentry Firewall CD.

---

1. Introduction

This document is designed to be a more comprehensive reference guide to the Sentry Firewall CD. This is not designed to be a HOWTO, but instead to be a supplement to the HOWTO and the other available documentation. The following are some of the topics covered in this document:

- Available configuration directives and their usage.
- Miscellaneous setup information for specific daemons or services.
- Unique features of each branch.
- Technical outline of the configuration scripts.

If you would like to add anything to this document, or if you have any questions or comments please feel free to email me, Obsid@Sentry.net.

1.1 Further Reading

- Sentry Firewall CD HOWTO
- Sentry Firewall CD Reference Guide
- Sentry Firewall CD FAQ
- Mailing Lists

1.2 Copyrights and Disclaimer

The current copyright and disclaimer can be found on the website; http://www.SentryFirewall.com/files/COPYRIGHT. It applies to the Sentry Firewall CD, and all the scripts and documentation associated with it.

2. Overview of Available Configuration Directives

This section provides a general overview of the directives that are supported by all the Sentry Firewall CD branches. Since the "host" distribution varies with each branch, the configuration and initialization files used by each branch will vary. The branch-specific directives will be covered in later sections.

2.1 Replacing Files

To replace a file that is supported by the configuration scripts, you may use the following syntax:

```
filename_directive = /location/of/filename
```

Where "filename_directive" is one of the directives supported by the configuration scripts. The file location can reference a file on a floppy disk(ie. /floppy/config/filename) or can also be in a URI format. The supported prefixed include "http://", "https://", "ftp://", "sftp://",and "scp://". For example:

```
snort.conf = /floppy/snort.conf
<or>
snort.conf = scp://user:pass@<server>/config/snort.conf
```

The current ’sentry.conf’ file for each branch is available in the "/SENTRY/scripts/cd-config/" directory on the ISO, or on the website. The default sentry.conf file for each branch should contain all the directives currently supported.

2.2 Copying Files

To replace files not supported by the configuration scripts, use the ’|=’ file copy directive.

```
Syntax: source_file |= dest_file, OR
        dest_file = source_file

Example:  Copy file /floppy/daemon.conf to /etc/daemon.conf

     /floppy/daemon.conf |= /etc/daemon.conf
     <or>
     /etc/daemon.conf = /floppy/daemon.conf
     <or>
     /etc/daemon.conf = scp://<user>:<pass>@<server>/config/daemon.conf
     <or>
     scp://<user>:<pass>@<server>/config/daemon.conf |= /etc/daemon.conf
```

2.3 Making Symlinks

Create a symlink

```
Syntax: dest_file => source_file(where the symlink points to)

Example:
     ## Make symlink called /etc/somefile.conf that points to /etc/otherfile.conf
     /etc/somefile.conf => /etc/otherfile.conf
```

2.4 Make a Directory.

Note: The 'mkdir' directive is only available with the Sentry Firewall CD versions 1.5.0-rc14 or newer.

```
Syntax:
     mkdir <PATH/DIRECTORY>[:MASK]
```

Make a directory in the specified location with the specified permissions(MASK). MASK is optional and defaults to 0755(rwxr-xr-x). This directive can be useful if you want to copy files at boot-time to a directory that does not exist on the ramdisk by default.

2.5 'device' Directive

Set up an ethernet device to use during configuration. This device will only be used during configuration to grab files via http(s)/ftp/sftp/scp and is taken down after configuration is complete.

```
device[1..10] = [device_name]:[driver_name]:[IP_Address]<|gateway>
device[1..10] = [device_name]:[driver_name]:dhcp<|hostname>

NOTES: 1) <hostname> and <gateway> are optional, but sometimes required.

       2) Only one <gateway> can be declared, that is, you cannot set
          up more than one default gateway.

      *3) Devices set up with the 'device{1..10}' directive are TEMPORARY
          and are taken down after the configuration process is complete.
          See rc.inet1{.conf} for more permanent network setup.

       4) Please see file: /SENTRY/scripts/cd-config/networking.pl for
          list of supported devices.  Most 10/100BaseT ethernet devices
          should be supported.

Examples:
     device1 = eth0:tulip:192.168.1.50|192.168.1.1
     device2 = eth1:via-rhine:dhcp
```

2.6 'nameserver' Directive

Set up a nameserver to use during configuration.

```
Syntax: nameserver = <DNS_IP>
```

2.7 Proxy Support Directives

Set up a proxy for pulling files via http(s), or ftp.

```
Syntax:
     http_proxy = http://<hostname>/
     ftp_proxy = http://<hostname>/
     proxy-user = <PROXY_USER>
     proxy-passwd = <PROXY_PASSWORD>
```

2.8 Passive FTP Support

Use passive ftp instead of active ftp to retrieve files via ftp.

```
Syntax:
     passive-ftp = <on|off>  ## Default == off
```

2.9 ’include’ Directive

Retrieve and parse another ’sentry.conf’ file.

```
Syntax:
     include = </location/of/sentry.conf>

     Or, with network support -

     include=<ftp|http>://[<user>:<pass>@]<SERVER_IP></path/to/sentry.conf>
```

2.10 The ’path<#>’ Directive.

Note: The ’path<#>’ directive is only available with the Sentry Firewall CD versions 1.5.0-rc13 or newer.

Path statements tell the configuration scripts where to look for files. These can specify a path on a local or remote system. The variables "path1" to "path10" are allowed.

```
Syntax:
     path<#> = <PATH>
     path<#> = <URI>
```

NOTE: <URI> should point to a directory on a remote system, NOT just a file.

```
Examples:
     path1 = /floppy/node1/config/
     path2 = scp://user:pass@someserver/node123/config/
     path3 = http://user:pass@someserver/node123-backup/config/
     etc etc...
```

You may then use the following syntax when declaring a file within your sentry.conf:

```
Examples:
     squid.conf = squid.conf
     or
     /etc/someconf.conf = someconf.conf
```

The configuration scripts will first look for "squid.conf" or "someconf.conf" in $m_point, which is usually /floppy. If it isn’t found, then the system will try path1..path10 in order until "squid.conf" or "someconf.conf" is found. This not only makes for less typing when creating your sentry.conf, but it also allows you to add some redundancy to the configuration process.

2.11 ’cdrom’ Directive

Defines which device the CDROM is. If not declared the configuration scripts will still try to probe for and mount the CD. But declaring this is much easier/faster/safer.

```
Syntax:
     cdrom = <DEVICE>

Example:
     cdrom = /dev/hdc
```

2.12 ’cron’ Directive

Replace a user’s crontab file.

```
Syntax:
     cron:<USERNAME> = </LOCATION/OF/CRONTAB_FILE>
```

2.13 ’hostname’ Directive

Defines the hostname of the local machine. This directive can be used to either point to a file containing the hostname of the local machine, or to simply define the hostname itself.

```
Syntax:
     hostname = </path/to/file>
     <or>
     hostname = MYHOSTNAME
```

2.14 The ’add_swap’ directive

Note: The ’add_swap’ directive is only available with the Sentry Firewall CD versions 1.5.0-rc11 or newer.

The ’add_swap’ directive tells the configuration scripts to add a swap partition at configuration time. If the ":format" option is appended to the variable, then the configuration scripts will also format the partition before activating it.

Warning: An improper setting of this variable could cause serious damage to data.

```
Usage:
     add_swap = /dev/hda1
     add_swap = /dev/hda1:format
```

2.15 The ’root_size’ directive

Note: The ’root_size’ directive is only available with the Sentry Firewall CD versions 1.5.0-rc11 or newer.

The ’root_size’ directive allows one to change size of root(/) at configuration time(before any other files are copied). By default the root filesystem is around 18MB in size. This option allows you to change the size of the root filesystem if you need more/less space. Also - since root is mounted on a tmpfs filesystem - this area can be swapped out as needed. The suffix g, m, or k is accepted for binary

kilo, mega and giga. If no suffix is added, a size in megabytes is presumed.

```
Usage:
    root_size = "18M"
```

The size of the root file system can also be changed after configuration by simply remounting it, ie "mount -oremount,size=24M /"

3. Directive Reference

The following is a list of file directives currently supported by the various branches of the Sentry Firewall CD. Please note that this information is highly subject to change. Currently supported configuration directives can be found in the sample sentry.conf file available on the ISO image in the "SENTRY/scripts/cd-config" directory, or online at http://www.SentryFirewall.com/.

3.1 SENTRYCD(-DEVEL) Branches

The following is a list of file directives currently supported in the SENTRYCD and SENTRYCD-DEVEL branches(slackware-based).

Initialization scripts:

```
rc.M                    Multiuser init script(runlevel 3).
rc.6                    Halt or reboot
rc.dhcpd                Startup script for ISC DHCP daemon.
rc.netdevice            Load modules for network devices(before rc.inet1 is run).
rc.inet1                Set up ethernet interfaces.
rc.inet1.conf           Configuration file for rc.inet1.
rc.inet2                Start network daemons.
rc.inet2.conf           Configuration file for rc.inet2.
rc.keymap               Load keyboard map.
rc.local                Local system init script.
rc.modules              Load any needed modules.
rc.firewall             Firewall script.
rc.firewall.nat         Nat-specific firewall script(not always used).
rc.firewall.save        Use if firewall script was generated using iptables-save.
rc.ntpd                 ntpd initialization script.
rc.sendmail             Sendmail initialization script.
rc.snort                Snort initialization script.
```

Important system configuration files:

```
fstab                   fstab(5)
ftpusers                ftpusers(5)
group                   group(5)
hosts                   hosts(5)
hosts.equiv             hosts.equiv(5)
hosts.allow             hosts_access(5)
hosts.deny              hosts_access(5)
inittab                 inittab
modules.conf            modules.conf(5)
openssl.cnf             OpenSSL configuration file.
passwd                  passwd(5)
profile                 bash(1)
resolv.conf             resolv.conf(5)
shadow                  shadow(5)
shells                  shells(5)
```

Daemon configuration files:

```
bgpd.conf                   Configuration file for bgpd(http://www.zebra.org/).
dhcpd.conf                  Configuration file for dhcpd.
dnsmasq.conf                Configuration file for dnsmasq.
httpd.conf                  Configuration file for Apache HTTP Daemon(http://www.apache.org/).
inetd.conf                  Configuration file for inetd(8).
ipsec.conf                  Configuration file for ipsec(http://www.freeswan.org/).
ipsec.secrets               IPSec secrets file for IKE/IPsec authentication.
named.conf                  Configuration file for named(8).
l2tpd.conf                  Configuration file for l2tpd, Layer 2 Tunnelling Protocol Daemon(http://www.l2tpd.org/)
newsyslog.conf              Configuration file for newsyslog.
ospfd.conf                  Configuration file for ospfd(http://www.zebra.org/).
portsentry.conf             Configuration file for portsentry.
pppoe.conf                  Configuration file for RP-PPPOE(http://www.roaringpenguin.com/pppoe/).
pptpd.conf                  Configuration file for pptpd(http://poptop.lineo.com/).
proftpd.conf                Configuration file for proftpd(http://www.proftpd.net/).
ntp.conf                    Configuration file for ntpd.
rinetd.conf                 Configuration file for rinetd.
ripd.conf                   Configuration file for ripd(http://www.zebra.org/).
rndc.conf                   Configuration file for named control utility, rndc(8).
sendmail.cf                 Configuration file for sendmail(http://www.sendmail.org/).
smb.conf                    Configuration file for Samba(http://www.samba.org/).
snort.conf                  Configuration file for snort(http://www.snort.org).
squid.conf                  Configuration file for squid(http://www.squid-cache.org/).
ss5.conf                    Configuration file for Socks Server 5(http://digilander.libero.it/matteo.ricchetti/)
stunnel.conf                Configuration file for stunnel(http://stunnel.mirt.net/).
stunnel.pem                 Certificate chain PEM file for stunnel.
syslog.conf                 syslogd(8) configuration file.
syslog-ng.conf              Configuration file for syslog-ng(http://www.balabit.com/products/syslog_ng/).
vsftpd.conf                 Configuration file for vsftp daemon(http://vsftpd.beasts.org/).
wlan.conf                   Configuration for prism based wireless cards(http://www.linux-wlan.com).
gated.conf                  Configuration for GateD.
ulogd.conf                  Configuration file for ulogd.
zebra.conf                  Configuration file for zebra(http://www.zebra.org/).
```

OpenSSH configuration files:

```
ssh_config                  Configuration file for ssh(1).
sshd_config                 Configuration file for sshd(8).
shosts.equiv                Like hosts.equiv(5) but for ssh.
ssh_host_key                Private rsa1 host key file.
ssh_host_key.pub            Private rsa1 host key file.
ssh_host_dsa_key            Private rsa1 host key file.
ssh_host_dsa_key.pub        Private rsa1 host key file.
ssh_host_rsa_key            Private rsa1 host key file.
ssh_host_rsa_key.pub        Private rsa1 host key file.
ssh_known_hosts             Public host keys of known ssh servers.
ssh_known_hosts2            Public host keys of known ssh servers.
```

Configuration directives for Webmin(http://www.Webmin.com/).

```
start_webmin                enable|disable Webmin.  Default is "disable".
webmin_config               Main Webmin configuration file(/etc/webmin/config).
miniserv.conf               Config file for Webmin http(s) daemon.
miniserv.pem                SSL cert for Webmin http(s) daemon.
                            An SSL cert will be created by rc.webmin if
                            one is not specified.
miniserv.users              Password file used for Webmin.
                            Default user:pass is sentry:SENTRY.
                            NOTE: If this file is not replaced webmin
                            will NOT start.
```

Misc. configuration directives.

```
cron:<user>                      Replace <user>’s crontab with specified file.
add_swap                    Initialize a swap partition at configuration time.
root_size                   Change the size of root(/) at configuration time.
```

3.2 SENTRYCD-DEB(-DEVEL) Branch Directives

The following is a list of file directives currently supported in the SENTRYCD-DEB and SENTRYCD-DEB-DEVEL branches(debian-based).

UNDER CONSTRUCTION

4. Configuration Reference

This section contains general information about how to configure the Sentry Firewall CD, including information on which files to edit and how to setup specific daemons and services on a running system.

More information about configuring specific daemons or services can be found in the HOWTO.

4.1 Init script configuration

This section covers specifically the configuration of the system via the various init scripts kept in /etc/rc.d/(Slackware) or /etc/init.d/(Debian).

rc.inet1 and rc.inet1.conf

NOTE: This section applies to the SENTRYCD and SENTRYCD-DEVEL(Slackware-based) branches.

Versions of the Sentry Firewall CD before 1.5.0-rc7 utilized a perl-based rc.inet1 script to add and configure interfaces. The syntax of this file was similiar to the syntax used in the sentry.conf file for network configuration support -

```
    $interface(1..10) = "<IF>:<IP ADDRESS[/NETMASK]>|<DHCP>"

## Examples:
    $interface1 = "eth0:192.168.1.1/24";        ## Set up eth0 with ip 192.168.1.1 and
                                                ## netmask 255.255.255.0.
    $interface2 = "eth0:192.168.1.2/24";        ## Bind second IP to eth0.
    $interface3 = "eth1:dhcp";                  ## Use DHCP to set up eth1.
```

As you can see, the syntax is fairly simple. You can configure an interface or add an IP address by adding a "$interface#" variable.

As of version 1.5.0-rc7, the Sentry Firewall CD utilizes a modified version of the rc.inet1 and rc.inet1.conf init files that appeared in Slackware 9.1. Once again, you alter the network setup by altering a bunch of variables, except this time you edit /etc/rc.d/rc.inet1.conf instead of rc.inet1 itself. By default, this file supports the setup of up to four ethernet devices. A basic entry to set up eth0 looks something like the following:

```
IPADDR[0]="192.168.1.10"
NETMASK[0]="255.255.255.0"
USE_DHCP[0]=""
DHCP_HOSTNAME[0]=""
ETH0_ALIAS[0]=""        ## For multiple IPs on interface.
ETH0_ALIAS[1]=""        ## Ditto.

GATEWAY="192.168.1.1"
```

To use DHCP instead of static IP addresses, set the "USE_DHCP[0]" directive to "yes". You may also add any number of additional "alias" IP addresses to each interface with the "ETHx_ALIAS[#]" variable.

NOTE: Keep in mind that if neither of these rc files suit your needs you may simply replace the rc.inet1 file at boot time with your own script by using the 'rc.inet1' directive in your sentry.conf file.

4.2 Further Reading

- HOWTOs at Linux.org
- FreeS/WAN Documentation
- Bridging Documentation
- General Firewall Setup (Sentry Firewall CD HOWTO)
- Snort Setup (Sentry Firewall CD HOWTO)
- BIND Setup (Sentry Firewall CD HOWTO) | DNS HOWTO

If there is something you wish to add to this list or to the documentation in general, please feel free to email Obsid@Sentry.net.

5. The sentrycd(-devel) Branches

This section contains information about features or directives that are unique to the SENTRYCD and SENTRYCD-DEVEL(slackware-based)branches.

This section is still under construction.

6. The sentrycd-deb(-devel) Branches

This section contains information about features or directives that are unique to the SENTRYCD and SENTRYCD-DEVEL(slackware-based)branches.

This section is still under construction.

6.1 Start/Stop a Service or Daemon

This directive gives you the ability to start or stop a service at bootup. The syntax looks like the following:

```
service:[start|stop] = <path/to/service_init_file>
```

For example:

```
httpd:stop
or
httpd:start = /floppy/config/httpd
```

In the above example, we are telling the Sentry Firewall CD to either start or stop the http daemon at bootup. The optional argument "<path/to/service_init_file>" is usually not necessary, but is used to actually replace the startup script located in /etc/rc.d/init.d/, in case you ever wanted to do so.

7. The Rootdisk

7.1 Overview

A rootdisk is a gzip-compressed RAMdisk image that is mounted as root(/) during the boot process. Currently, the rootdisk for the Sentry Firewall CD is around 16-18 MB in size, uncompressed.

It is, of course, possible to increase the size of the rootdisk to accommodate your needs. When you increase the size of this image you also need to increase the "ramdisk_size" parameter passed to the kernel at boot time by syslinux. This parameter can be adjusted either manually at the initial boot prompt or in the "isolinux.cfg" file kept in the isolinux directory on the CD. If the uncompressed ramdisk size is larger than this parameter, the boot process will not continue passed the loading of the kernel. That is, root(/) will never be mounted and you will get a kernel panic.

As of version 1.5.0-rc11 the Sentry Firewall CD utilizes a tmpfs file system for its root partition. The tmpfs file system, also known as "virtual memory file system" or "shm fs", provides two major advantages:

- Tmpfs file systems reside in virtual memory, which means they can be swapped out.
- Tmpfs file systems can be resized.

With this new development it is now possible to resize root(/) at configuration time(or any time afterward) without rebuilding the rootdisk or ISO. Please see the "root_size" and "add_swap" configuration directives for more details.

7.2 Creating the rootdisk

I use a script called ’mkrootdsk.sh’. This is a bash shell script that formats/mounts the disk image, and then creates or copies the files to the disk image as needed.

If you would like to attempt to use the mkrootdsk.sh script please be sure to read through it first, as it tends to be a bit hacky at times. It runs perfectly on my development system, but may not run well at all on yours. The output from the script should look something like the following:

```
Sentry Firewall CD-ROM: mkrootdsk.sh
Copyright (C) Stephen A. Zarkos, Obsid@Sentry.net
Ok, let’s get to it...

[+] Creating /root/rootdisk/root... Done.
[+] Ok, starting to copy stuff to the rootdisk...
[+] Making directories: root dev proc etc sbin bin lib mnt mnt1 mnt2 mnt3 mnt4 opt cdrom floppy tmp
tmp/drivers var initrd... Done.
[+] Copying /dev files... Done.
[+] Working in /var... Done.
[+] Working in /home... Done.
```

```
[+] Working in /bin... Done.
[+] Working in /sbin... Done.
[+] Working in /lib... Done.
[+] Working in /etc... Done.
[+] Building drivers-2.4.tar.gz(network config support).
    [+] Using /cdrom/lib/modules/2.4.25GENERIC.
[+] Tar/Gzipping /root/rootdisk/root... Done.
[+] Zeroing out file: /root/rootdisk/initrd.img... Done.
[+] Creating ext2 file system on /root/rootdisk/initrd.img... Done.
[+] Mounting initrd.img on /root/rootdisk/mnt... Done.
[+] Copying files to rootdisk... Done.

[+] /root/rootdisk/initrd.img is still mounted, do you want me
    to unmount it? (y/n) y
    [+] Unmounting /root/rootdisk/mnt... Done.
    [+] Gzipping /root/rootdisk/initrd.img... Done.

Location of new rootdisk --> /root/rootdisk/initrd.img
```

The finished ramdisk image is then copied to the isolinux directory before creating the actual ISO image.

7.3 Editing the rootdisk

To look at and modify the initrd.img image, do something like the following:

```
blah@wherever:~$ cp /cdrom/isolinux/initrd.img /tmp/initrd.img.gz
blah@wherever:~$ gzip -d /tmp/initrd.img.gz
blah@wherever:~$ mount -o loop /tmp/initrd.img /MOUNT_POINT
```

You may then cd to /MOUNT_POINT and edit the files on the rootdisk. Once you are finished you can then unmount and gzip the initrd.img file and place it back in the isolinux directory.

7.4 Details of the rootdisk layout

The following are some notes about the layout and design of the rootdisk.

- /usr is a symlink to /cdrom/usr.
- Many of the standard system binaries on the system are symlinked to busybox when possible. Few binaries actually reside on the rootdisk.
- All the rest of the binaries in /bin /sbin are just symlinks to the /cdrom/{bin,sbin}.
- Several libraries are present on the rootdisk. The exact libraries and versions may vary, please take a look at the mkrootdsk.sh script for those details.
- /etc and /etc/rc.d are nearly empty at boot time. All the standard(default) system files are kept in /etc/default/*. The configuration scripts(perl) that are run at boot time will either take the files from the location specified in the configuration file(sentry.conf) or make symlinks to these default files from /etc/*.
- The perl scripts are located on the rootdisk in either /etc/rc.d/SENTRY/ or /etc/init.d/SENTRY/.

8. The Sentry Firewall Configuration Scripts

This section is designed as a technical outline of the configuration scripts built for the Sentry Firewall CD. These configuration are responsible for finding and parsing the sentry.conf file and the directives contained therein.

As of version 1.5.0-rc14 there are six configuration scripts overall on the system. These are kept in the "/etc/rc.d/SENTRY/" or the "/etc/init.d/SENTRY/" directory on the rootdisk. They are also available on the CD in the "<CDROM>/SENTRY/scripts/cd-config/" directory or online. The configuration scripts are called ‘cd-config.pl’, ‘do_config.pl’, ‘file_functions.pl’, ‘get_config.pl’, ‘networking.pl’, and ‘process_conf.pl’. The details of these files are outlined below.

8.1 cd-config.pl

This is the first perl script to run. It is usually called from the rc.S or rcS file, depending on the branch. This file contains the following:

- **Mainline**
  Starts by calling ’get_config()’, then ’process_conf()’ to process the config file and place the directives into the global %prefs hash. Then calls ’do_config()’ to start the configuration process.
  - **Input:** STDIN (yes|no), not always used.
  - **Returns:** Nothing.
- **do_log()**
  Takes as input a single string that will be written to a logfile($logfile). $logfile is ’/var/log/SENTRY_LOG’ by default.
  - **Input:** $_[0] = string to print to log file.
  - **Returns:** 0 == fail, 1 == success.

8.2 get_config.pl

This file contains the following:

- **get_config()**
  This function attempts to mount /dev/fd0 on /floppy, and if that fails it tries mounting /dev/hda1 on /mnt. If one of these succeeds it then looks for a configuration file(sentry.conf) and puts that in a global array called @conf.
  - **Input:** $_[0] = name of configuration file.
  - **Returns:** 0 == fail, 1 == success.
- **do_command()**
  This function accepts a string that will be passed to the system to run as a command and a timeout(seconds) as input. It then attempts to run the command and timeout the operation once the "$timeout" has expired to avoid blocking. If a timeout value is not passed to the function "$timeout" is set to 10 seconds. Because of some of the oddities of this function, it may mangle "$_". If calling do_command() from a loop, try and use temporary variables instead of relying on $_.
  - **Input:** $_[0] = Command to run, $_[1] = timeout(seconds).
  - **Returns:** 0 == fail, 1 == success, 2 == attempt timed out.

8.3 process_conf.pl

This file contains the following:

- **process_conf()**
  This function parses the global array "@conf" and places it into global hash %prefs. It will also implement the "=>" and "|=" directives.

- **Input:** Nothing.
- **Returns:** 0 == fail, 1 == success

- **do_include()**
  This function will attempt to retrieve file specified in the "include" configuration directive and shove it into "@conf". It will then call ’process_conf()’ to take and parse the file. This process will continue until there are no more ’include’ directives to parse.
  - **Input:** Nothing.
  - **Returns:** 0 == fail, 1 == success

8.4 do_config.pl

In general, this is the only configuration file you would need to modify if you would like to create a Sentry Firewall CD for any Linux distribution. This file contains the following:

- **do_config()**
  Parses configuration contained in @conf. Copies files specified in the configuration file and makes necessary symlinks in /etc. There are also a number of special configuration directives handled in this function such as ‘nameserver’, ‘cdrom’, ‘start_webmin’, ‘add_swap’, ‘root_size’, etc.
  %etc_vars replaces @etc_vars, @ssh_vars, and %specdir used in earlier versions of the configuration scripts. This hash contains all the files that are supported within sentry.conf, as well as their locations. For example, configuration directive ‘squid.conf’ has a value of ‘/etc/squid’. Adding a new element to this list effectively adds a new directive that can be used within sentry.conf.
  - **Input:** Nothing.
  - **Returns:** 0 == fail, 1 == success
- **mount_cdrom()**
  This function will attempt to mount the Sentry Firewall CD on /cdrom. Uses either what was defined with the "cdrom" directive in sentry.conf, or the information from ‘dmesg‘. This was dealt with in rc.cdrom in older versions of the Sentry Firewall CD.
  - **Input:** Nothing.
  - **Returns:** 0 == fail, 1 == success
- **fix_modules()**
  This function copies or symlinks the contents of "/cdrom/lib/modules/$kversion" to "/lib/modules/$kversion" unless the filename is ’TRANS.TBL’. ’TRANS.TBL’ files are created when the ISO image was made and generally causes annoying errors with depmod.
  - **Input:** Nothing.
  - **Returns:** 0 == fail, 1 == success
- **recurse_dirs()**
  This function recurses through /etc/default and fills @dirs with directory names. This can be an expensive operation, but it is useful when we start making symlinks from /etc/{dir} to /etc/default/{dir}.
  - **Input:** $_[0] = Base path to start recursion. $_[1] = Base directory to start recursion.
  - **Returns:** Nothing.
- **add_swap()**
  This function initializes and adds a swap partition based on the ’add_swap’ directive in the sentry.conf file.
  - **Input:** Nothing
  - **Returns:** 0 == fail, 1 == success

- **root_size()**
  This function remounts root to change its size based on the ’root_size’ directive in the sentry.conf file.
  - **Input:** Nothing
  - **Returns:** 0 == fail, 1 == success
- **nameserver()**
  This function contains a block of code that was previously contained within do_config(), and was moved to its own function to make do_config() a bit easier to read. $prefs{’nameserver’} should contain an IP address for a DNS server which is written to /etc/resolv.conf. This directive is used primarily for network configuration support.
  - **Input:** Nothing
  - **Returns:** 0 == fail, 1 == success
- **no_shadow()**
  no_shadow() contains a block of code that was previously contained within do_config(), and was moved to its own function to make do_config() a bit easier to read. This function deals specifically with $prefs{’shadow’}. If the ‘shadow’ directive is not declared within sentry.conf, or if the file is not readable, then ‘/etc/ssh/sshd_config’ and ‘/etc/inetd.conf’ is replaced with a more strict default - essentially making it impossible to log in remotely with a default password.
  - **Input:** Nothing
  - **Returns:** 0 == fail, 1 == success
- **start_webmin()**
  This function reads $prefs{’start_webmin’} to determine if it should allow webmin to start at boot time via rc.webmin. start_webmin() also checks to make sure the ‘miniserv.users’ file was declared and is readable. If it is not readable, webmin will not be allowed to start, regardless of the value of $prefs{’start_webmin’}. Webmin can, of course, still be started manually later on.
  - **Input:** Nothing
  - **Returns:** 0 == fail, 1 == success
- **merge_fstab()**
  This function contains the block of code that was previously contained within do_config(), and was moved to its own function to make do_config() a bit easier to read. Actually, it was originally called fix_fstab(), and then renamed and mostly rewritten as merge_fstab(). This function basically just checks the user-defined fstab file and makes sure it has at least the entries that are contained in ‘/etc/default/fstab’.
  - **Input:** Nothing
  - **Returns:** 0 == fail, 1 == success
- **merge_passwd()**
  This function deals with the ‘shadow’, ‘passwd’, and ‘group’ directives to make sure the user-defined files contain at least those that are contained within the system default.
  - **Input:** $_[0] = ‘shadow’, ‘passwd’, or ‘group’.
  - **Returns:** 0 == fail, 1 == success

8.5 file_functions.pl

This script contains a number of general functions that are used throughout the other configuration scripts. Most of the functions are related to working with files and directories. Some of these were contained in do_config.pl in earlier versions of the CD.

- **create_dir()**
  This function creates a specified directory. create_dir() is mainly utilized with the ‘mkdir’ directive.
  - **Input:** $_[0] = <PATH/DIRECTORY>[:MASK].
  - **Returns:** 0 == fail, 1 == success.
- **vrfy_file()**
  vrfy_file() takes in one variable; a hash key name($var). This function verifies the existence of a file. If the path/filename exists as stated in the sentry.conf file, then the function returns 1. Otherwise vrfy_file() tries to find the file using the $m_point path and then passes $var onto locate_file() to try and locate the file using the $path{1..10} variables.
  - **Input:** $_[0] = hash key name($var).
  - **Returns:** 0 == fail, 1 == success, 2 == syntax or misc. error.
- **locate_file()**
  locate_file() takes in one variable; a hash key name($var). This function attempts to locate the file using the $path{1..10} variables declared in sentry.conf.
  - **Input:** $_[0] = hash key name($var).
  - **Returns:** 0 == fail, 1 == success.
- **retr_file_net()**
  Attempts to retrieve file if destination looks like a http(s)/(s)ftp/scp URI. Currently, only the ‘NOERR’ option is supported which prevents retr_file_net() from logging errors.
  - **Input:** $_[0] = URI/FILENAME to retrieve. $_[1] = Options.
  - **Returns:** 0 == fail, 1 == success.
- **vrfy_path()**
  Ensure the directory to the passed file exists. This function is often called before attempting to copy a file.
  - **Input:** $_[0] = PATH/FILENAME
  - **Returns:** 0 == success.
- **pcopy()**
  Copies $src to $trg and creates target directory if needed. Options that can be passed to this function include ‘PERM’, ‘LINK’, ‘RECURS’ which simply translate to the ‘-p’, ‘-dp’, and ‘-Rdp’ options that are passed to ‘/bin/cp’. The default behavior is to pass no options to ‘/bin/cp’.
  - **Input:** $_[0] = source file. $_[1] = destination file. $_[2] = options.
  - **Returns:** 0 == fail, 1 == success.
- **do_log()**
  Takes as input a single string that will be written to a logfile($logfile). $logfile is ’/var/log/SENTRY_LOG’ by default.
  - **Input:** $_[0] = string to print to log file.
  - **Returns:** 0 == fail, 1 == success.

8.6 networking.pl

This file contains the following:

- **networking()**
  This function utilizes the "device{1..10}" configuration directive to set up or take down an ethernet interface and set up a default gateway if necessary.
  - **Input:** $_[0] = 1|NET_UP or 2|NET_DOWN, set up or take down networking.
  - **Returns:** 0 == fail, 1 == success, 2 == failed to initialize device or assign an IP address.

- **retr_file()**
  This function attempts to retrieve a file via http(s)/ftp/scp/sftp using wget or scp/sftp.
  - **Input:** $_[0] = Filename to retrieve, $_[1] = Location of file(URI).
  - **Returns:** Returns 0 on failure, otherwise returns the local path/filename of the retrieved file.
- **mk_batch()**
  Function to create a batch file for use with sftp. Since sftp doesn't have a recursive -r option, we need to create a batch file to tell sftp to manually grab the stuff in the subdirectories under /etc/sysconfig. This is only used when dealing with the sysconf_dir directive, which is not valid in the "sentrycd" branch, but useful in other branches such as "sentrycd-rh" and may be useful in other Sentry Firewall based Linux distributions.
  - **Input:** $_[0] = Remote path from where files should be retrieved.
  - **Returns:** 0 == fail, 1 == success
- **remove_mods()**
  This function uses 'rmmod' to remove any previously loaded ethernet driver modules for network configuration. Use the hash "%depend" to remove the module and its dependencies in the proper order.
  - **Input:** $_[0] = Name of ethernet driver to remove.
  - **Returns:** 0 == fail, 1 == success
- **load_mods()**
  This function uses 'modprobe' to load ethernet driver modules for network configuration support. Uses the hash "%depend" to load the module and its dependencies in the proper order.
  - **Input:** $_[0] = Name of ethernet driver to load.
  - **Returns:** 0 == fail, 1 == success