Sentry Firewall CD HOWTO

Table of Contents

Table of Contents

Sentry Firewall CD HOWTO

Stephen A. Zarkos, Obsid@Sentry.net

v1.0, 2002−03−20

This document is designed as an introduction on how the Sentry Firewall CDROM works and how to get started using the system.

---

6. Building a Custom Sentry CD

7. More Information

---

1. Introduction

This is the long−overdue Sentry Firewall CDROM howto. I hope this document helps get you started using the Sentry Firewall CD and answers any questions you might have regarding how the system works. The most current version of this howto can be obtained at the following URL: http://www.SentryFirewall.com/files/howto/.

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 What is the Sentry Firewall CD?

The Sentry Firewall CD is a Linux−based bootable CDROM suitable for use in a variety of different operating environments. The system is designed to be configured dynamically via a floppy disk or over a network. This allows one to configure the system dynamically, eventho much of the actual system is on read−only(CDROM) media.

1.2 Why would I use a CD−based firewall or server?

There are several advantages of using a CDROM based system in various security related environments. The main system is centered around the ramdisk; a compressed file system image which is loaded into RAM at boot time. Any changes to the ramdisk image are temporary, and will be undone upon the next reboot. Furthermore, the ramdisk, kernel, binaries, etc, related to the operating system are kept on read−only media(CDROM). This means that if the security of a box running a CDROM based system is ever compromised the attacker can at best own the box until the next reboot. So there is no real threat of having to go through the tedious task of rebuilding and hardening the system after a successful attack is discovered.

1.3 I'm a Linux newbie, will Sentry Firewall CD be a good choice for me?

At the moment, the Sentry Firewall CD is based on a pretty generic Slackware Linux system. You should probably be somewhat familiar with Linux and how to configure the system in order to get the most use out of the CD. But, even if you are a Linux newbie, I encourage you to give it a shot anyway – it's free, after all.

But, basically, there are no GUIs, no scripts to do it for you. The idea behind the configuration of the CD is that you are able to reconfigure the system by replacing the startup scripts and the various system and configuration files present on the system at boot time. Most of these are simply text files and shell scripts that you need to edit by hand in order to configure properly. There are, however, usually plenty of resources available to assist you in configuring a specific service or daemon(HOWTOs on linux.org, for example).

1.4 Minumum Requirements

- x86 computer with CD–ROM
- BIOS that supports the eltorito standard(booting from the cdrom).
- 32MB RAM(64MB or more recommended)
- Easy access to coffee/tea/soda or equivalent stimulant.
- Floppy disk drive(optional)

1.5 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. How the CD Works (Overview)

This section is just an overview to explain how the Sentry Firewall CD works, that is, from the process of loading the kernel to running the Sentry Firewall CD configuration scripts located on the RAMDisk.

2.1 The boot process

Booting from the CDROM is a fairly familiar process. The BIOS execs the bootloader(Syslinux) − which then displays a bootprompt and loads the kernel and ramdisk into memory. Once the kernel is running, the ramdisk is then mounted as root(/).

An obvious necessity for deploying CDROM based systems is the ability to dynamically configure the system for various environments with different configurations, which is what a good majority of this project is dedicated to building. A simple way to do this is to give the user the ability to customize the startup scripts located in /etc/rc.d before they are actually used, as well as the ability to customize other important system configuration files.

At boot time, the /etc and /etc/rc.d directories are nearly empty. On a Slackware system the first rc file to run is /etc/rc.d/rc.S − and it is from this file where we run the configuration scripts that look for a configuration file(sentry.conf), and place the proper configuration and system files in /etc and various subdirectories under /etc. If there is not a configuration directive for a specific file, or if a configuration file cannot be found, then the default system files are used − which are located in /etc/default/* on the ramdisk.

2.2 ISOLINUX

Early versions of the Sentry Firewall CD utilized the 2.88MB floppy emulation method, along with either lilo or syslinux to boot the kernel and load the ramdisk. This method proved very limiting for two reasons; A) the total size of the compressed ramdisk AND kernel was limited to 2.88MB, and B) it was quite slow compared to the current method.

The Sentry Firewall CD is currently utilizing the isolinux.bin boot record with no emulation in order to properly boot the CDs. This allows us to use a much larger ramdisk and offer a choice of several kernels to boot at boot time.

More information about syslinux can be found at syslinux.zytor.com.

2.3 The CD Configuration scripts

As previously mentioned, the first rc script to run on a Slackware system is called /etc/rc.d/rc.S. It is from this file where we run our configuration scripts, which reside in /etc/rc.d/SENTRY/ on the ramdisk. The first script to run is called 'cd−config.pl', which is essentially the mainline for the entire program. The other scripts that are used are called 'get_config.pl', 'process_conf.pl', and 'networking.pl'. These scripts were written specifically for this project, and are essentially the mainstay of the entire configuration process.

In depth review of these scripts is a little beyond the scope of this document, but is covered a bit in the file called 'DOCUMENTATION' available on the website (http://www.SentryFirewall.com/). The files are written in perl, and do several important things; read in and parse the configuration file(sentry.conf), locate and retrieve the important files detailed in the sentry.conf file, and replace the system default files with the ones the user has defined in the configuration file.

3. Obtaining the CDROM

3.1 Downloading

The CDROM is distributed as a gzip or bzip2 compressed iso image, and is generally between 95−105MB in size. Available download mirrors are listed on the websites, http://www.SentryFirewall.com/ or http://Sentry.Sourceforge.net/.

3.2 Purchasing

The Sentry Firewall CD is also available for purchase over the web. Although the iso image is free to use and distribute, purchasing the CD will help support the project and help ensure continued development. For more information please visit http://www.SentryFirewall.com/purchase/

Custom versions of the CD tailored to a specific network configuration are also available upon request. Please email me for more information.

3.3 Burning the CDROM

This section will attempt a general overview on how to burn the CD iso image once you have obtained it from one of the mirrors. All the commands presume you're working in Linux, if not, then I'm afraid you're on your own.

First, let's decompress the iso image:
NOTE: Make sure you have enough disk space, the decompressed iso image can be somewhere between 250MB and 300MB.

```
blah@wherever:~$ gzip −d sentrycd.iso.gz

or

blah@wherever:~$ bzip2 −d sentrycd.iso.bz2
```

Verify the integrity of the iso image,

```
blah@wherever:~$ md5sum −b sentrycd.iso
```

Now, let's try to burn the CD. You'll need the 'cdrecord' utility available, it can be obtained here. You will want to run 'cdrecord −scanbus' in order to find the 'dev' value required for the following command. You will also need to know the write speed of your CDRW. Details on how to set this all up are beyond the scope of this document, please refer to the CD Writing HOWTO for more details.

```
blah@wherever:~$ DEV="DEV_LINE_HERE" SPEED="SPEED"
```

```
blah@wherever:~$ cdrecord -v -data speed=$SPEED dev=$DEV sentrycd.iso
```

That's it, you now have a Sentry Firewall CDROM. By the way, you may have to be root to do all this.

Keep in mind, if you simply want to look at the ISO image without actually burning the CD, you can mount the image on a loopback device;

```
blah@wherever:~$ mount -o loop ./sentrycd.iso /MOUNT_POINT
```

Where "MOUNT_POINT" is where you would like the CD mounted. You may then 'cd' to the MOUNT_POINT directory and poke around – don't forget to 'umount' the image once you're finished. This assumes you have support in your kernel for the loopback device. You probably do, but once again, recompiling kernels is beyond the scope of this document.

---

4. Using the Sentry Firewall CDROM

4.1 Introduction

The configuration scripts which are run from /etc/rc.d/rc.S first look for a configuration file called 'sentry.conf' on a floppy disk which, if present, will be mounted on /floppy. In order to configure the Linux system for use in any particular environment the user must have the ability to replace the system default files with his/her own copies. The 'sentry.conf' file basically tells the configuration scripts which files it should replace and where those files are.

A good example of a sentry.conf file can be found on the Sentry Firewall CD in the directory /SENTRY/scripts/cd–config/. Configuration floppy disk images(1.44M) can also be found in /SENTRY/images/ on the CD.

4.2 The sentry.conf file

The main configuration file for the system is called 'sentry.conf'. It will first be looked for on a floppy disk(/dev/fd0). The file accepts several configuration directives, many of which will be discussed below.

Example

A basic configuration file looks like the following (everything after a '#' sign is interpreted as a comment):

```
----snip----
## Basic Sentry Firewall CD config file(sentry.conf)

rc.M = /floppy/config1/rc.M
rc.inet1 = /floppy/config1/rc.inet1
```

```
passwd = /floppy/config1/passwd
shadow = /floppy/config1/shadow

# EOF #
----snip----
```

The syntax is pretty simple, the default 'rc.M' file will be replaced with the user defined 'rc.M' file located in the '/floppy/config1/' directory. Same goes for 'rc.inet1', 'passwd', and the 'shadow' file. But it is imortant to remember, the first place the sentry.conf file will be looked for is on /dev/fd0, which if found, will be mounted on /floppy. This is why all these files appear to be located in the /floppy directory, it is simply the mount point for the floppy disk.

Unfortunately, you cannot arbitrarily replace files, for example the following will likely not be parsed correctly:

```
foo.conf = /floppy/config1/foo.conf
```

The configuration scripts only recognize a certain number of configuration files. There are other very easy ways to replace files that are not supported by default, however. These will be discussed below.

4.3 Network Configuration

As of version 1.0.5, a new syntax for the configuration directives are recognized; those with an "http://" or "ftp://" prefix. This basically means that the following syntax is now supported:

```
inetd.conf = ftp://user:pass@123.123.123.123/config1/inetd.conf
```

In order to accomplish this the configuration scripts need to have the ability to set up an ethernet interface, as well as obtain nameserver information from the sentry.conf file. The syntax to accomplish this is the following:

```
device{1..10} = <device>:<driver>:<IP address>[|Gateway_IP]

or..

device{1..10} = <device>:<driver>:dhcp[|Hostname]
```

And to set up a nameserver:

```
nameserver = <IP_ADDRESS>
```

So, for example to set up an interface called "eth0", which uses the "tulip" driver and can obtain its ip address from a DHCP server, we can use the following line:

```
device1 = eth0:tulip:dhcp
```

As you can see, a total of 10 devices are allowed. Let's say we now want to set up an interface "eth1" that uses an "rtl8139" chip, and has a static IP(192.168.1.2) and a default gateway(192.168.1.1):

```
device2 = eth1:8139too:192.168.1.2|192.168.1.1
```

NOTE: It is important to keep in mind that whatever devices you set up during the configuration process will be promptly taken down after the configuration is complete. This setup is only used so you can retrieve configuration files over the network, via http and ftp. For more permanant network configuration, please use the rc.inet1 file.

Example

```
----snip----
## Basic Sentry Firewall CD config file to retrieve files via http or ftp.

device1 = eth0:tulip:192.168.1.2|192.168.1.1
nameserver = <MY_DNS_IP>

rc.M = ftp://user:pass@config.sentry.net/node1/rc.M
rc.inet1 = http://user:pass@config.sentry.net/all_nodes/rc.inet1

passwd = http://user:pass@config.sentry.net/all_nodes/passwd
shadow = ftp://user:pass@config.sentry.net/node1/shadow

# EOF #
----snip----
```

4.4 Other Useful Configuration Directives

Copy file /floppy/someconfig.conf to /etc/someconfig.conf –

```
/floppy/someconfig.conf |= /etc/someconfig.conf

OR, this does the same thing.

/etc/someconfig.conf = /floppy/someconfig.conf
```

Make a symlink called /etc/someconfig.conf that points to /etc/otherconfig.conf –

```
/etc/someconfig.conf => /etc/otherconfig.conf
```

The include directive. Grabs another sentry.conf file either from another location –

```
include = ftp://user:pass@config.sentry.net/node1/sentry.conf
```

Keep in mind, however, that the include directive is one of the first directives to be parsed. Any configuration directives parsed from the included sentry.conf file that conflict with directives in the previously parsed sentry.conf files will clobber the old ones.

4.5 Putting it all together, managing multiple nodes from a single location.

In order to manage multiple nodes at a single location, you can use a bare sentry.conf file located on a floppy disk, and then grab files from your ftp or http servers.

```
----snip----
## Basic Sentry Firewall CD config file.

device1 = eth0:tulip:dhcp
nameserver = <DNS_IP>
include = ftp://user:pass@config.sentry.net/node1/sentry.conf

----snip----
```

The included sentry.conf file will then be parsed, and files replaced via http or ftp if you like. You can now edit your sentry.conf and configuration files at a central location.

4.6 Example sentry.conf and disk images

An example configuration disk image is available on the CDROM. The disk is an ext2 formatted disk, and is located in the '/SENTRY/images/' directory on the CD. There is also a very complete sentry.conf file on the disk which may help clarify alot of these directives. Use a command like the following to create the configuration disk:

```
blah@wherever:~$ dd if=/cdrom/SENTRY/images/ext2-144.img of=/dev/fd0
2880+0 records in
2880+0 records out
```

---

5. Overview of Available Configuration Directives

5.1 Replacing rc/config files

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

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

Where the location of the file is often '/floppy/filename'

The following rc/config files are currently supported

```
rc.M
rc.netdevice
rc.inet1
rc.inet2
rc.local
```

```
rc.modules
rc.firewall
rc.firewall.nat
fstab
passwd
shadow
group
shells
profile
resolv.conf
hosts
ftpusers
hostname
newsyslog.conf
openssl.cnf
syslog.conf
syslog-ng.conf
inetd.conf
proftpd.conf
squid.conf
httpd.conf
smb.conf
snort.conf
pptpd.conf
pppoe.conf
gated.conf
zebra.conf
hosts.equiv
shosts.equiv
ssh_config
sshd_config
ssh_host_key
ssh_host_key.pub
ssh_host_dsa_key
ssh_host_dsa_key.pub
ssh_host_rsa_key
ssh_host_rsa_key.pub
ssh_known_hosts
ssh_known_hosts2
```

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

5.2 'device' directive support

Set up an ethernet device to use during configuration.

```
     device[#] = [device_name]:[driver_name]:[IP_Address]<|gateway>
     device[#] = [device_name]:[driver_name]:dhcp<|hostname>

     NOTE: 1) <hostname> and <gateway> are optional, but sometimes required.
           2) Most ethernet devices are supported.  If you find one that isn't
              and you think it should be, please let me know.
           3) "device1" to "device10" are supported.

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

5.3 'nameserver' directive

Set up a nameserver to use during configuration.

```
nameserver = <DNS_IP>
```

5.4 'include' directive

Retrieve and parse another 'sentry.conf' file.

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

Or, with network support -

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

5.5 Copying files (|=)

Copy file from one location to the other.

```
     Syntax: source_file |= dest_file

Example:
     Copy file /floppy/daemon.conf to /etc/daemon.conf
     /floppy/daemon.conf |= /etc/daemon.conf
```

5.6 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
```

5.7 'cdrom' directive

Defines which device the CDROM is. Most of the time the CDROM is detected and mounted using the /etc/rc.d/rc.cdrom script. But this makes the process less error−prone.

```
     Syntax: cdrom = <DEVICE>

Example:
     cdrom = /dev/hdc
```

5.8 'cron' directive

Replace a user's crontab file(located in /var/spool/cron/crontabs/).

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

5.9 hostname

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 define the hostname itself.

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

---

6. Building a Custom Sentry CD

6.1 Introduction

This section will attempt to describe how to create a custom Sentry Firewall CDROM. Unfortunately, I do not have to time to go into every detail. But at the very least I will try and provide for you an overview of the CD creation process.

6.2 The development system(How I do it)

My development system consists of 2 separate Slackware installations. First, I have a very complete Slackware installation on my main hard drive(/dev/hda). I then have /dev/hdb1, upon which I have another, bare bones, Slackware installation. This installation generally has no compiling tools or X stuff. I usually have /dev/hdb1 mounted on /mnt, that's not a critical element, but I thought I'd mention it since I will refer to /mnt alot from now on. I then have a folder called /CD−FW on the /dev/hdb1 installation. Throughout this entire process, the Slackware installation on /dev/hda is the live running system, and it is from here that I compile the needed tools, kernels, etc and basically run everything.

I begin by copying nearly all the files from /mnt to /mnt/CD−FW in order to produce what will be the next Sentry Firewall CD. For example, I have a script that does something like the following:

```
## This usually spits out alot of errors, usually crap about
## hard links and such, but they are all ignorable.

cp −Rdp /mnt/bin /mnt/CD−FW/
cp −Rdp /mnt/sbin /mnt/CD−FW/
cp −Rdp /mnt/lib /mnt/CD−FW/
cp −Rdp /mnt/usr/bin /mnt/CD−FW/usr/
cp −Rdp /mnt/usr/sbin /mnt/CD−FW/usr/
cp −Rdp /mnt/usr/local/bin /mnt/CD−FW/usr/local/
cp −Rdp /mnt/usr/lib /mnt/CD−FW/usr/
cp −Rdp /mnt/usr/libexec /mnt/CD−FW/usr/
cp −Rdp /mnt/usr/share /mnt/CD−FW/usr/
cp −Rdp /mnt/usr/man /mnt/CD−FW/usr/
```

I then need to create alot of other folders and files in /mnt/CD−FW in order to get a fairly complete system. Things in /mnt/CD−FW/var for instance will often have to be created by hand in order to avoid copying alot of crap I don't need. Notice, however, that I don't copy any /dev files, since these files will be on the ramdisk(when I get around to creating it).

Ok, so now we have /mnt/CD−FW. To make this easy for you, this is essentially the exact same thing that's present on the Sentry Firewall CDROM. All I did was use the 'mkisofs' utility on /mnt/CD−FW. So the stuff on the CD is actually a copy of what's in the /mnt/CD−FW directory on my hard drive.

Having a separate, but unused Slackware system on /dev/hdb1 makes it easy for me to install and upgrade packages as I need them. For example, if I want to use the upgradepkg utility to upgrade a package I get from ftp.slackware.com, I can simply do something like the following:

```
root@mybox:~# cp /tmp/zlib.tar.gz /mnt
root@mybox:~# cd /mnt; chroot /mnt
root@mybox:/# upgradepkg zlib.tar.gz; exit
.....
```

Then, all I need to do is re−run the script mentioned above, the one that copies all those files, to update the /mnt/CD−FW directory.

6.3 The RAMdisk Image

That's all nifty, but now comes the hard part... making the ramdisk. If you take a look at the /isolinux directory on the CDROM, you will see a bunch of files, one of them is called 'initrd.img' − there are several others as well, such as isolinux.cfg, message.txt, and isolinux.bin. These files are required by isolinux in order to work properly. Take a look at those files and the documentation that comes with syslinux to get a better idea of what all that does. In any case, the 'initrd.img' file is, in fact, the compressed ramdisk image.

To take a look at the 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
```

In a nutshell, I use the file '/SENTRY/scripts/MK−CD/mkrootdsk.sh' to create the rootdisk. Please read that file and the disclaimer before you decide to use it. It runs perfectly on my system, but may not run well at all on yours. It basically attempts to create a rootdisk image to use with the Sentry CD, but it is very long and may be somewhat difficult to comprehend at times. This is what happens when I start a project and fail to utilize proper child safety restraints.

6.4 Making the ISO Image

The next file I use is called 'mkiso.sh'. The script generally just declares a few variables and runs the 'mkisofs' utility. The command I normally run looks like the following:

```
root@mybox:~# cd /mnt/CD-FW
root@mybox:/mnt/CD-FW# mkisofs -o sentrycd.iso -R -V "Sentry Firewall CD [v1.2.0]" -v \
   -T -d -D -N \
   -b isolinux/isolinux.bin \
   -c isolinux/eltorito.cat \
   -no-emul-boot -boot-load-size 4 -boot-info-table \
   -A "Sentry Firewall v1.2.0 (Slackware 8.0)"
........
```

And that's it, I burn the CD and test it. For reference, the following files are available on the CDROM:

- /SENTRY/scripts/MK−CD/mkrootdsk.sh (builds the rootdisk)
- /SENTRY/scripts/MK−CD/mkiso.sh (builds final ISO image)
- /SENTRY/scripts/MK−CD/record−cd.sh (burns the ISO to a CD)

---

7. More Information

7.1 Mailing List

Thanks to SourceForge.net, there is a mailing list available for the Sentry CD.

- Subscribe
- Archives

7.2 Frequently Asked Questions

A FAQ is currently being maintain on the Sentry Firewall website, it can be accessed via the following URL: http://Sentry.SourceForge.net/files/FAQ.

7.3 About Sentry Network Security

Sentry Network Security is an itty bitty company that specializes in building and maintaining Linux based firewalls, as well as various other network related services. We also offer custom Sentry Firewall CD solutions, to help configure and build any number of systems utilizing the Sentry Firewall CD. For more information, or if you'd like to send me any bad jokes or poetry, please email me.
