Iptables

LeRoy D. Cressy
leroy@lrcressy.com

v 0.04 May 15, 2004

Editors:

Rita J. Cressy
Sibylla C. Cressy

Contents

1 Introduction

1.1 Other Sources of Information

- http://iptables-tutorial.frozentux.net/iptables-tutorial.html

  This is an excellent tutorial that goes a lot deeper than this hour-long talk.

- http://www.netfilter.org/documentation/

  This is the site where you will find information on every aspect of iptables in various languages.

1.2 Debian Packages

Building a firewall with iptables can either be a very simple matter by using a preconfigured firewall script. Here is the listing that “apt-cache search iptables” produces on a Debian system:

- ipmenu – A cursel iptables/iproute2 GUI
- kernel–patch–ttl – TTL matching and setting
- acidlab – Analysis Console for Intrusion Databases
- arptables – ARP table administration
- ebtables – Ethernet bridge frame table administration
- ferm – maintain and setup complicated firewall rules
- fiaif – An easy to use, yet complex firewall
- filtergen – packet filter generator for various firewall systems
- fireflier–client–gtk – Interactive firewall rule creation tool – GTK client
- fireflier–client–qt – Interactive firewall rule creation tool – QT client
- fireflier–server – Interactive firewall rule creation tool – server
- firehol – An easy to use but powerful iptables stateful firewall
- firewall–easy – Easy to use packet filter firewall (usually zero config)
- fwanalog – iptables log–file report generator (using analog)
- fwbuilder–ipt – Linux iptables policy compiler for Firewall Builder
- fwbuilder–iptables – Linux iptables policy compiler for Firewall Builder
- fwlogwatch – Firewall log analyzer
- gnome–lokkit – basic interactive firewall configuration tool (GNOME interface)
- guarddog – firewall configuration utility for KDE
- guidedog – NAT/masquerading/port-forwarding configuration tool for KDE
- ipac – IP accounting configuration and statistics tool
- ipac–ng – IP Accounting for iptables(kernel ≥2.4)
- iptables – IP packet filter administration tools for 2.4.xx kernels
- iptables–dev – development files for iptable's libipq and libiptc
- iptstate – Top–like state for netfilter/iptables
- kernel–patch–ulog – Netfilter userspace logging patch.
- knetfilter – A GUI for configuring the 2.4 kernel IP Tables
- lokkit – basic interactive firewall configuration tool (console interface)
- netstat–nat – A tool that display NAT connections

- psad – The Port Scan Attack Detector
- reaim – Enable AIM and MSN file transfer on Linux iptables based NAT
- shorewall – Shoreline Firewall (Shorewall)
- shorewall–doc – Shoreline Firewall (Shorewall) Documentation
- uif – Advanced iptables-firewall script
- ulogd – The Netfilter Userspace Logging Daemon
- webmin–firewall – iptables control module for webmin
- wflogs – The modular firewall log analyzer of the WallFire project

As you can see there are a plethora of packages available for those who don't want to or care to write a script themselves. Some of these packages can be used as a guide for writing you own script rules. With all of the packages that are available why would you want to write your own script? My reason is that when I first started using iptables there wasn't a decent script available, and now I am just maintaining the firewall script that is several years old and still blocks most of the stuff that I don't want to come through on my network.

1.3 What Firewalls Will Not Prevent

In a lot of corporate environments, the danger does not always come from the outside. On numerous occasions, a disgruntled employee who quits or is fired could fdisk or wipe clean the hard drive on their computer. Most firewalls will not prevent an emailed virus from infecting the windows machines on the network.

A properly configured firewall will prevent an outside attack, but it will not prevent someone inside the network from doing something stupid, like downloading a windows screen saver that contains a virus or worm, opening up any windows executable email attachment, or not having an up to date windows virus protection program running **on all windows clients.**

A firewall is just one small portion of your total security policy. One key thing to remember is that your total security is only as strong as the weakest link in the total security policy. Weak passwords that can be broken with a password cracker like "john" are just as bad as a sloppy firewall configuration.

The worst thing that can happen is to think that your network is secure. Just look at http://www.cert.org to see the continuous stream of attacks and all of the buffer overflows along with all of the patches for the various software on our systems. A network that is connected to the Internet is never secure. There is no perfect lock! The best that we can do is to make the system as secure as possible for today. Possibly tomorrow someone will figure out how to get through our security code.

!

A security system is only as strong as its weakest link

> Print the above sentence in a very large font and paste it along the top of your monitor.
>
> Look at it every day, and try to understand the implications. The weakest link property is the main reason why security systems are so fiendishly hard to get right.[1]

1.4 What firewalls are capable of doing

- Port forwarding
- Port blocking
- Address blocking
- Matching various criteria
- Allow only specific addresses ssh access from the Internet.
- Allow only certain internal addresses access to port 80 on the Internet. In other words, only allow specific individuals access to the World Wide Web.
- Filter for known viruses
- And many other things

There are many things that you can do with a firewall to protect your network from malicious attacks along with preventing the internal users from doing something that they shouldn't. I cannot stress enough that this is only a small portion of your total network security.

!

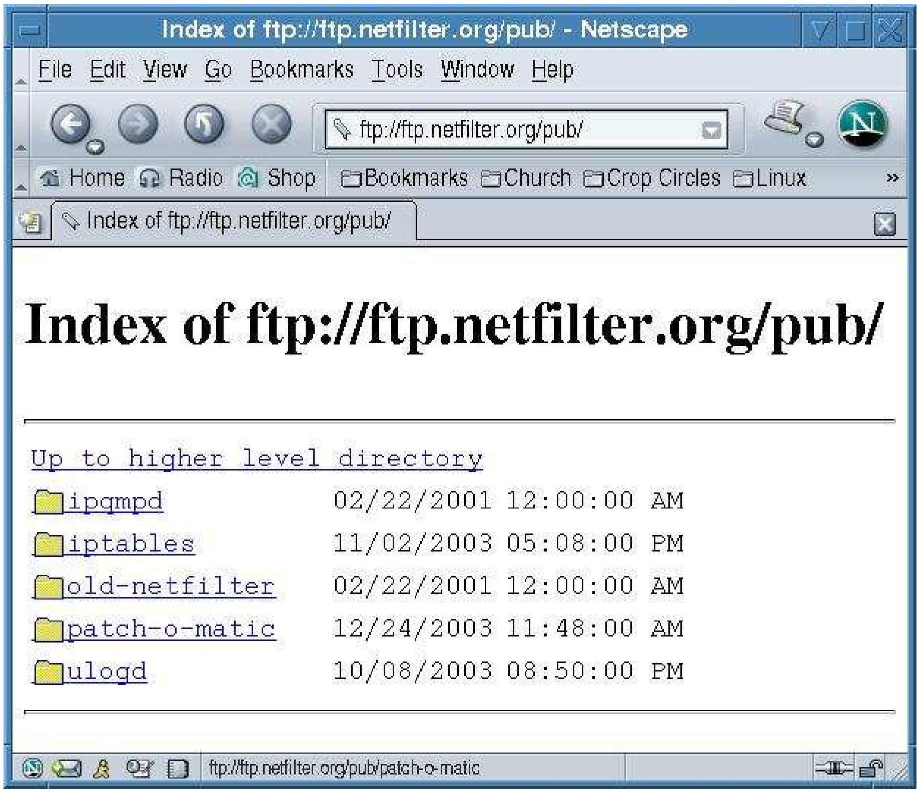
2 Patching Your Kernel Source Code

There are many options and pending patches that have not made it into the kernel tree. Some of these are included in the 2.6.x Linux kernel tree, while others are currently being tested or are alpha patches. Some of the patches are not compatible with others.

2.1 Using Patch-O-Matic

Patch-O-Matic is a separate package in the netfilter suite. You need to grab the latest version from netfilter's ftp site ftp://ftp.netfilter.org/pub/patch-o-matic.

[1] Niels Ferguson and Bruce Schneier, Practical Cryptography (Indianapolis: Wiley Publishing, Inc., 2003), p. 9.


Index of ftp://ftp.netfilter.org/pub/ - Netscape
File Edit View Go Bookmarks Tools Window Help
ftp://ftp.netfilter.org/pub/
Home Radio Shop Bookmarks Church Crop Circles Linux
Index of ftp://ftp.netfilter.org/pub/
ftp://ftp.netfilter.org/pub/patch-o-matic

Index of ftp://ftp.netfilter.org/pub/

```
Up to higher level directory
ipqmpd          02/22/2001 12:00:00 AM
iptables        11/02/2003 05:08:00 PM
old-netfilter   02/22/2001 12:00:00 AM
patch-o-matic   12/24/2003 11:48:00 AM
ulogd           10/08/2003 08:50:00 PM
```

After unpacking the tarball read the README file and block the line:

```
KERNEL_DIR=<<where-you-built-your-kernel>>  ./runme pending
```

Paste it into an xterm editing the place where your kernel source code is located as indicated.

```
KERNEL_DIR=/usr/local/src/linux/v2.4/linux-2.4.23  ./runme pending
```

Before you run this script you should have a backup of your kernel source or the original tarball. This script changes the kernel source code. As you run the script you will notice that there are numerous patches already installed. As of December 31, 2003, patch-o-matic does not work with the 2.6.0 kernel. Heed the warning about not applying all of the patches. After running pending, you can run base and extra portions of patch-o-matic.

!

The problem with the 2.6.x kernel is that the subdirectory config files have changed from Config and Config.help to Kconfig which produces the following error.

```
Do you want to apply this patch [N/y/t/f/a/r/b/w/q/?] y
Testing patch submitted/01_2.4.19.patch...
Warning - no help text file could be found in either
/usr/local/src/linux/v2.6/linux-2.6.0/net/ipv4/netfilter/Config.help
or /usr/local/src/linux/v2.6/linux-2.6.0/Documentation/Configure.help
Failed to patch copy of /usr/local/src/linux/v2.6/linux-2.6.0
```

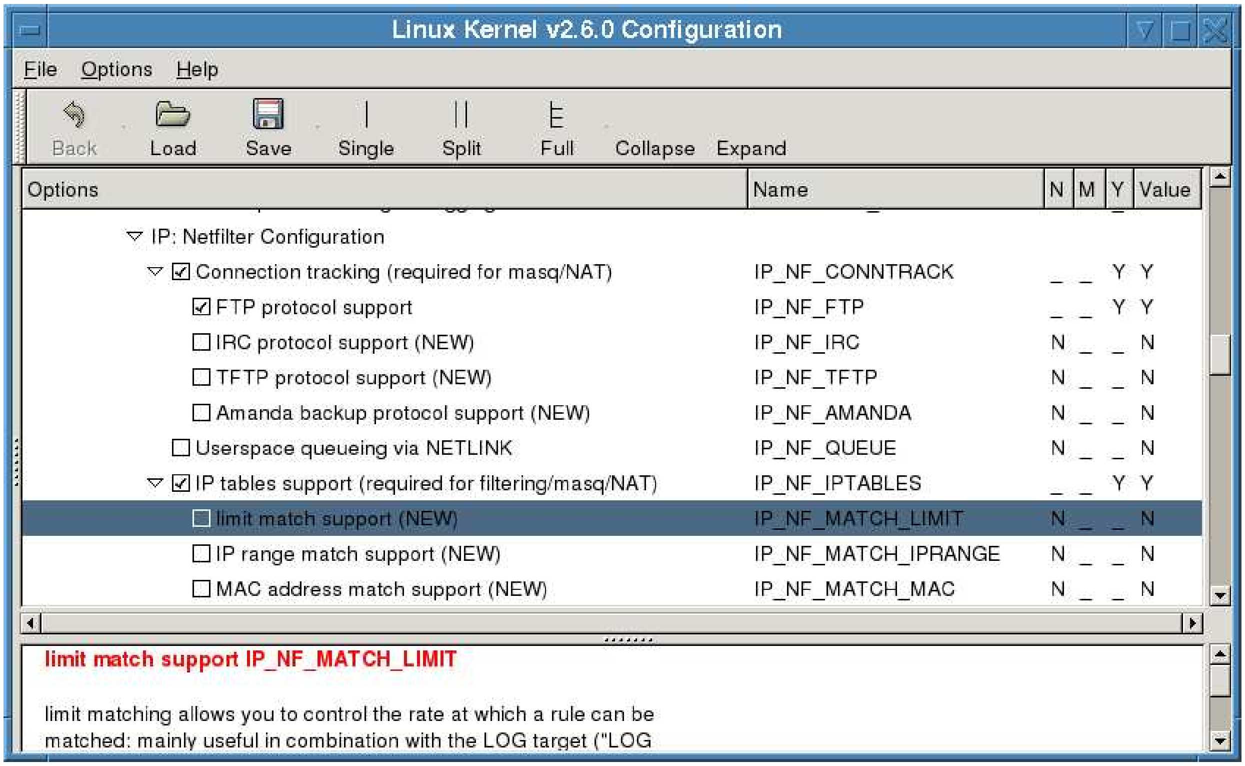
```
TEST FAILED: patch NOT applied.

[Press enter to continue]
```

There is a patch-o-matic converter at http://www.stearns.org/pom26convert/ along with others that are in testing. Hopefully, one of these will be on the netfilter site in the near future.

3 Compiling the Kernel

Since kernel 2.6.0 has been released I will talk about using it. The old xconfig using tcl/tk is no more. Now if you want to use xconfig you need the QT development stuff installed. On the other hand there is a gconfig that uses the gtk development which on a Debian system is easier for me to figure out.

The netfilter configuration using gconfig looks like:



Options	Name	N	M	Y	Value
▽ IP: Netfilter Configuration					
▽ ☑ Connection tracking (required for masq/NAT)	IP_NF_CONNTRACK	_	_	Y	Y
☑ FTP protocol support	IP_NF_FTP	_	_	Y	Y
□ IRC protocol support (NEW)	IP_NF_IRC	N	_	_	N
□ TFTP protocol support (NEW)	IP_NF_TFTP	N	_	_	N
□ Amanda backup protocol support (NEW)	IP_NF_AMANDA	N	_	_	N
□ Userspace queueing via NETLINK	IP_NF_QUEUE	N	_	_	N
▽ ☑ IP tables support (required for filtering/masq/NAT)	IP_NF_IPTABLES	_	_	Y	Y
□ limit match support (NEW)	IP_NF_MATCH_LIMIT	N	_	_	N
□ IP range match support (NEW)	IP_NF_MATCH_IPRANGE	N	_	_	N
□ MAC address match support (NEW)	IP_NF_MATCH_MAC	N	_	_	N

limit match support IP_NF_MATCH_LIMIT

limit matching allows you to control the rate at which a rule can be matched: mainly useful in combination with the LOG target ("LOG

One of the nice things is that the help is always displayed at the bottom of the window instead of having to click on the help button. The key strokes of using 'y', 'm', or 'n' are the same as in the past. Of course you still use menuconfig for a text based configuration.

With all of the patches that you have added to the kernel along with what is already part of the kernel source code there are a lot of options that you can put into your firewall kernel. Another item to consider for firewalls is not using modules. Though it has not happened yet, there is a possibility that someone could write a module that would infect a running kernel. With module support turned off I feel that the potential of this happening is eliminated.

If you are using one of the pre made iptables scripts, run the script by hand to make sure that all options that the script requires are compiled in your kernel. If you get any error messages from running the script then you need to remake the kernel. I have never used one of these packages, but it is possible that the standard kernel does not have all of the netfilter patches that the iptables script requires.

Personally, I will stick with the 2.4 kernel for a firewall until 2.6 becomes rock solid. Presently, I have one machine that I am testing with the 2.6 kernel and there seems to be some problems. Already 2.6.0 has been patched with some serious security fixes. If you want to use the cryptography features in the kernel, I would suggest using the 2.6 kernel.

4 Iptables

Here we come to the interesting portion of the creation of a firewall. We have patched and compiled a custom kernel with all of the drivers included for our specific hardware along with all of the netfilter items that we think we will need included in the kernel.

A iptables firewall script is a shell script that is run as a system startup script usually located in /etc/init.d/. Depending on the distribution that you are using there will be a symbolic link in /etc/rc2.d/ or whatever the default runtime is specified in /etc/inittab.

Since we are creating a shell script it is wise to create several variables at the beginning of the script like:

- Real_IP_Address
- Internet_eth
- DMZ_eth
- LAN_eth
- LAN_network
- DMZ_network
- Config_file

You can use these variables to make your code portable for numerous machines making your life easier.

```
RealIP="66.92.109.218"
Internet="eth0"
DMZ="eth1"
LAN="eth2"
conf="/usr/local/etc/firewall.conf"
```

As you can see you have the option of either setting up and parsing a config file or by having the variables directly in the script. You as the author of this script should decide on what method you want to use.

There are some security concerns that you might want to consider in creating your script. For instance, do you want the Ethernet card that is connected to the Internet to be up? Couldn't this lead to an attack against the system prior to the completion of the shell script? Or as part of the shell script we drop all packets from the Internet opening up the ports one at a time as the shell script runs. The choice is yours, but I would not accept any packets from the Internet until I am ready. One of the problems with either dropping all packets or running ifdown on the various Ethernet cards would be with remote administration. Let's say if you are about 100 miles away from the box that you are running your script on and there is a bug in the script which causes the system to hang or not bring up the Internet connection you are out of luck. So please use a little caution and thought on the creation of your script.

!

Another concern is that you should severely limit who and where can log in to the firewall via ssh. Also, I like all packets from the Internet to be either forwarded via NAT or dropped. Nothing from the Internet should have direct access to the firewall. This includes yourself. I realize that this makes it a little harder for you to remote administer the box in question, but if you are selling a secure firewall to a client you want it to be as secure as possible. I port forward all ssh connections from the Internet to a box on the DMZ network. The more restrictive that you can make your system the better it will be for your client.

Since last year the netfilter code has grown, so there are more options and targets that you can use. One of the newer targets is TARPIT where the packets gets stuck in a TARPIT instead of just dropping the packet. There are more matches now along with various states for a more robust stateful firewall.

4.1 Default Tables

There are three default tables with associated chains.

- Filter
  - INPUT
  - OUTPUT
  - FORWARD
- Nat
  - PREROUTING
  - OUTPUT
  - POSTROUTING
- Mangle
  - PREROUTING
  - OUTPUT

4.2 Iptables Targets

The following definitions are from the kernel source help. I just included the targets and matches that are part of the standard 2.6.0 kernel and have not included any of the additional patches from netfilter. To use any of these they must be compiled into the kernel. Also, the version of the iptables executable needs to have the support built in. I have found that the Debian version of iptables has all of the support for whatever you compile into the kernel.

- ACCEPT
  Accepts the packet.

- DROP
  Drops the packet into a black hole. This is one of the most used targets.

- QUEUE
  Pass the packet to userspace.

- LOG
  This option adds a 'LOG' target, which allows you to create rules in any iptables table which records the packet header to the syslog.

- ULOG
  This option adds a 'ULOG' target, which allows you to create rules in any iptables table. The packet is passed to a userspace logging daemon using netlink multicast sockets; unlike the LOG target which can only be viewed through syslog.

  The apropriate userspace logging daemon (ulogd) may be obtained from
  http://www.gnumonks.org/projects/ulogd

- REJECT
  The REJECT target allows a filtering rule to specify that an ICMP error should be issued in response to an incoming packet, rather than silently being dropped.

- MASQUERADE
  Masquerading is a special case of NAT: all outgoing connections are changed to seem to come from a particular interface's address, and if the interface goes down, those connections are lost. This is only useful for dialup accounts with dynamic IP address (ie. your IP address will be different on next dialup).

- SNAT
  Source NAT which changes the source IP address to a real IP address. This is useful for private networks access the Internet.

  ```
  iptables -t nat -A POSTROUTING -o eth0 -p tcp \
   -s 192.168.1.0/24  --dport 22   -j SNAT --to $RealIP
  ```

- DNAT
  Destination NAT which changes the destination address of a packet. Suppose your Apache web server is behind a firewall on a private network. You only have one real IP address and all of your other boxes are on a private network.

  ```
  iptables -t nat -A PREROUTING  -i eth0 -p tcp --dport 80 \
   -j DNAT --to 192.168.10.1
  ```

- REDIRECT
  REDIRECT is a special case of NAT: all incoming connections are mapped onto the incoming interface's address, causing the packets to come to the local machine instead of passing through. This is useful for transparent proxies.

- NETMAP
  NETMAP is an implementation of static 1:1 NAT mapping of network addresses. It maps the network address part, while keeping the host address part intact. It is similar to Fast NAT, except that Netfilter's connection tracking doesn't work well with Fast NAT.

- SAME
  This option adds a 'SAME' target, which works like the standard SNAT target, but attempts to give clients the same IP for all connections.

- TOS
  This option adds a 'TOS' target, which allows you to create rules in the 'mangle' table which alter the Type Of Service field of an IP packet prior to routing.

- ECN
  This option adds a 'ECN' target, which can be used in the iptables mangle table.

  You can use this target to remove the ECN bits from the IPv4 header of an IP packet. This is particularly useful, if you need to work around existing ECN blackholes on the internet, but don't want to disable ECN support in general.

- DSCP
  This option adds a 'DSCP' match, which allows you to match against the IPv4 header DSCP field (DSCP codepoint).

  The DSCP codepoint can have any value between 0x0 and 0x4f.

- MARK
  This option adds a 'MARK' target, which allows you to create rules in the 'mangle' table which alter the netfilter mark (nfmark) field associated with the packet prior to routing. This can change the routing method (see 'Use netfilter MARK value as routing key') and can also be used by other subsystems to change their behavior.

- CLASSIFY
  This option adds a 'CLASSIFY' target, which enables the user to set the priority of a packet. Some qdiscs can use this value for classification, among these are:

  atm, cbq, dsmark, pfifo_fast, htb, prio

- TCPMSS
  This option adds a 'TCPMSS' target, which allows you to alter the MSS value of TCP SYN packets, to control the maximum size for that connection (usually limiting it to your outgoing interface's MTU minus 40).

  This is used to overcome criminally braindead ISPs or servers which block ICMP Fragmentation Needed packets. The symptoms of this problem are that everything works fine from your Linux firewall/router, but machines behind it can never exchange large packets: 1) Web browsers connect, then hang with no data received. 2) Small mail works fine, but large emails hang. 3) ssh works fine, but scp hangs after initial handshaking.

  Workaround: activate this option and add a rule to your firewall configuration like:

  ```
          iptables -A FORWARD -p tcp --tcp-flags SYN,RST SYN  \
          -j TCPMSS --clamp-mss-to-pmtu
  ```

4.3 Iptables Matches

The following comments are from the kernel compile help.

- limit
  limit matching allows you to control the rate at which a rule can be matched: mainly useful in combination with the LOG target ("LOG target support", below) and to avoid some Denial of Service attacks.

- IP Range
  This option makes possible to match IP addresses against IP address ranges.

- MAC Address
  MAC matching allows you to match packets based on the source Ethernet address of the packet.

- Packite Type
  Packet type matching allows you to match a packet by its "class", eg. BROADCAST, MULTICAST, ...

  Typical usage:

  ```
          iptables -A INPUT -m pkttype --pkt-type broadcast -j LOG
  ```

- Netfilter MARK
  Netfilter mark matching allows you to match packets based on the 'nfmark' value in the packet. This can be set by the MARK target (see below).

- Multiple Port
  Multiport matching allows you to match TCP or UDP packets based on a series of source or destination ports: normally a rule can only match a single range of ports.

- TOS
  TOS matching allows you to match packets based on the Type Of Service fields of the IP packet.

- Recent
  This match is used for creating one or many lists of recently used addresses and then matching against that/those list(s).

  Short options are available by using 'iptables -m recent -h'
  Official Website: http://snowman.net/projects/ipt_recent/

- Ecn
  This option adds a 'ECN' match, which allows you to match against the IPv4 and TCP header ECN fields.

- DSCP
  This option adds a 'DSCP' match, which allows you to match against the IPv4 header DSCP field (DSCP codepoint).

  The DSCP codepoint can have any value between 0x0 and 0x4f.

- AH/ESP
  These two match extensions ('ah' and 'esp') allow you to match a range of SPIs inside AH or ESP headers of IPSec packets.

- Length
  This option allows you to match the length of a packet against a specific value or range of values.

- TTL
  This adds CONFIG_IP_NF_MATCH_TTL option, which enabled the user to match packets by their TTL value.

- Tcpmss
  This option adds a 'tcpmss' match, which allows you to examine the MSS value of TCP SYN packets, which control the maximum packet size for that connection.

- Helper
  Helper matching allows you to match packets in dynamic connections tracked by a conntrack-helper, ie. ip_conntrack_ftp

- Connection State
  Connection state matching allows you to match packets based on their relationship to a tracked connection (ie. previous packets). This is a powerful tool for packet classification.

- Connection Tracking
  This is a general conntrack match module, a superset of the state match.

  It allows matching on additional conntrack information, which is useful in complex configurations, such as NAT gateways with multiple internet links or tunnels.

- Owner
  Packet owner matching allows you to match locally-generated packets based on who created them: the user, group, process or session.

4.4 Iptables Options

A look at the man page for Iptables lists a huge number of options. When you start to write your script it is a good idea to keep the man page open on a separate xterm. I also keep an xterm open with /etc/services. There are too many options to discuss them all here.

4.4.1 General Options

The only options that I use normally outside of a script is:

- −L, −−list

  List all rules in the selected chain. If no chain is selected, all chains are listed. It is legal to specify the -Z (zero) option as well, in which case the chain(s) will be atomically listed and zeroed. The exact output is affected by the other arguments given.

- −t, −−table

  This option specifies the packet matching table which the command should operate on. If the kernel is configured with automatic module loading, an attempt will be made to load the appropriate module for that table if it is not already there.

- $-v$, −−verbose

  Verbose output. This option makes the list command show the interface address, the rule options (if any), and the TOS masks. The packet and byte counters are also listed, with the suffix 'K', 'M' or 'G' for 1000, 1,000,000 and 1,000,000,000 multipliers respectively (but see the -x flag to change this). For appending, insertion, deletion and replacement, this causes detailed information on the rule or rules to be printed.

- −n, −−numeric

  Numeric output. IP addresses and port numbers will be printed in numeric format. By default, the program will try to display them as host names, network names, or services (whenever applicable).

- −−line-numbers

  When listing rules, add line numbers to the beginning of each rule, corresponding to that rule's position in the chain.

- $-x$, −−exact

  Expand numbers. Display the exact value of the packet and byte counters, instead of only the rounded number in K's (multiples of 1000) M's (multiples of 1000K) or G's (multiples of 1000M). This option is only relevant for the -L command.

- −−modprobe=<command>

  When adding or inserting rules into a chain, use command to load any necessary modules (targets, match extensions, etc).

4.4.2 Commands

- −A, −−append

  Append one or more rules to the end of the selected chain. When the source and/or destination names resolve to more than one address, a rule will be added for each possible address combination.

- −D, −−delete

  Delete one or more rules from the selected chain. There are two versions of this command: the rule can be specified as a number in the chain (starting at 1 for the first rule) or a rule to match.

- −R, −−replace

  Replace a rule in the selected chain. If the source and/or destination names resolve to multiple addresses, the command will fail. Rules are numbered starting at 1.

- −I, −−insert

  Insert one or more rules in the selected chain as the given rule number. So, if the rule number is 1, the rule or rules are inserted at the head of the chain. This is also the default if no rule number is specified.

- −L, −−list

  List all rules in the selected chain. If no chain is selected, all chains are listed. It is legal to specify the −Z (zero) option as well, in which case the chain(s) will be atomically listed and zeroed. The exact output is affected by the other arguments given.

- −F, −−flush

  Flush the selected chain. This is equivalent to deleting all the rules one by one.

- −Z, −−zero

  Zero the packet and byte counters in all chains. It is legal to specify the −L, −−list (list) option as well, to see the counters immediately before they are cleared. (See above.)

- −N, −−new-chain

  Create a new user-defined chain by the given name. There must be no target of that name already.

- −X, −−delete-chain

  Delete the specified user-defined chain. There must be no references to the chain. If there are, you must delete or replace the referring rules before the chain can be deleted. If no argument is given, it will attempt to delete every non-builtin chain in the table.

- −P, −−policy

  Set the policy for the chain to the given target. See the section TARGETS for the legal targets. Only non-user-defined chains can have policies, and neither built-in nor user-defined chains can be policy targets.

- −E, −−rename-chain

  Rename the user specified chain to the user supplied name. This is cosmetic, and has no effect on the structure of the table.

- −h −−Help

  Give a (currently very brief) description of the command syntax.

4.4.3 Parameters

The following parameters make up a rule specification (as used in the add, delete, insert, replace and append commands).

- −p, −−protocol [!] protocol

  The protocol of the rule or of the packet to check. The specified protocol can be one of tcp, udp, icmp, or all, or it can be a numeric value, representing one of these protocols or a different one. A protocol name from /etc/protocols is also allowed. A ”!” argument before the protocol inverts the test. The number zero is equivalent to all. Protocol all will match with all protocols and is taken as default when this option is omitted.

- −s, −−source [!] address[/mask]

  Source specification. Address can be either a hostname, a network name, or a plain IP address. The mask can be either a network mask or a plain number, specifying the number of 1’s at the left side of the network mask. Thus, a mask of 24 is equivalent to 255.255.255.0. A ”!” argument before the address specification inverts the sense of the address. The flag −−src is a convenient alias for this option.

- −d, −−destination [!] address[/mask]

  Destination specification. See the description of the −s (source) flag for a detailed description of the syntax. The flag −−dst is an alias for this option.

- −j, −−jump target

  This specifies the target of the rule; i.e., what to do if the packet matches it. The target can be a user-defined chain (other than the one this rule is in), one of the special builtin targets which decide the fate of the packet immediately, or an extension (see EXTENSIONS below). If this option is omitted in a rule, then matching the rule will have no effect on the packet's fate, but the counters on the rule will be incremented.

- −i, −−in-interface [!] [name]

  Optional name of an interface via which a packet is received (for packets entering the INPUT, FORWARD and PREROUTING chains). When the ”!” argument is used before the interface name, the sense is inverted. If the interface name ends in a ”+”, then any interface which begins with this name will match. If this option is omitted, the string ”+” is assumed, which will match with any interface name.

- −o, −−out-interface [!] [name]

  Optional name of an interface via which a packet is going to be sent (for packets entering the FORWARD, OUTPUT and POSTROUTING chains). When the ”!” argument is used before the interface name, the sense is inverted. If the interface name ends in a ”+”, then any interface which begins with this name will match. If this option is omitted, the string ”+” is assumed, which will match with any interface name.

- [!] −f, −−fragment

  This means that the rule only refers to second and further fragments of fragmented packets. Since there is no way to tell the source or destination ports of such a packet (or ICMP type), such a packet will not match any rules which specify them. When the ”!” argument precedes the ”−f” flag, the rule will only match head fragments, or unfragmented packets.

- −c, −−set-counters PKTS BYTES

  This enables the administrater to initialize the packet and byte counters of a rule (during INSERT, APPEND, REPLACE operations)

5 Packet Flow

We have already seen the beginnings of our script where we turn off IP forwdarding, and initialize our chains. One thing to remember is to turn on IP forwarding at the end of your script.

The thing to remember is the sequence of events on how iptables works. http://iptables-tutorial.frozentux.net/iptables-tutorial.html has an excellent discussion on how a packet traverses iptables. In short, here is a diagram of what happens when a packet enters the firewall.

Mangle Table

PREROUTING chain
This is the first table that a packet encounters. If there are any packet mangeling rules, this is where the packet gets changed in the prerouting chain

Nat Table

PREROUTING Chain
Let's say that your web server is on a private network behind the firewall. All port 80 packets should have their destination changed to the private IP address of the web server.
This is accomplished with using DNAT Target. Other items I have found useful is that you can use the DROP Target here to drop some packets early in the game. Like if you are port forwarding email to your email server, you can drop some email packets in your firewall script, thus making your firewall script a windows virus scanner and blocker.

Filter Table

INPUT FORWARD OUTPUT Chains
This is the default table where most of the filtering takes place. This is where you normally ACCEPT, REJECT, or DROP packets. If a packet has been prerouted to another box, it never gets to the Filter table.

DO NOT use the input chain for any packets that have been forwarded via the Nat table! Any packets that are forwarded are only handled by the forward chain of the Filter table. Thus any ports that are forwarded to another box, it is best to test those packets prior to changing the destination.

Mangle Table

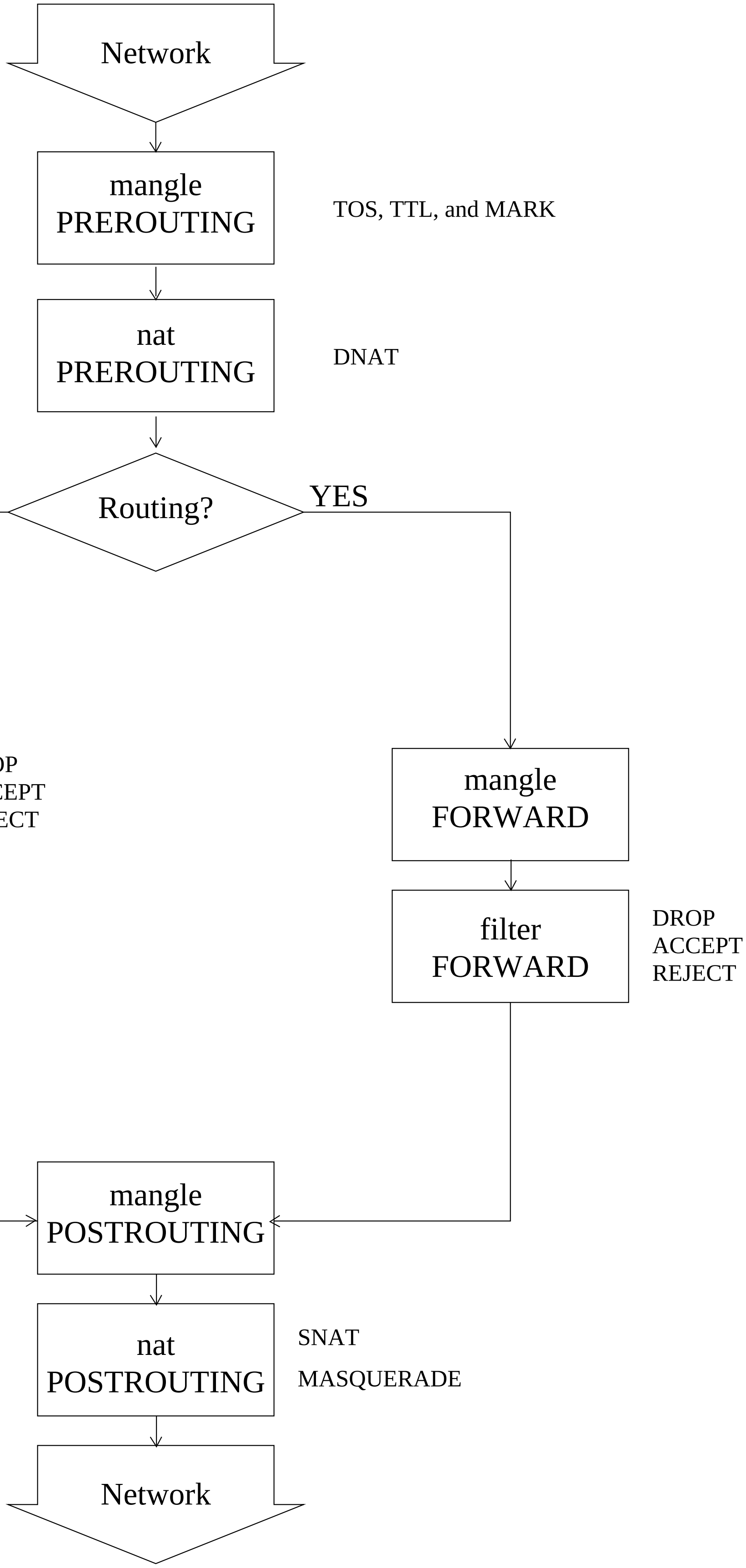
POSTROUTING Chain

Nat Table

POSTROUTING Chain
This is where the targets SNAT and MASAQUERADE are accomplished. If we have a private network, this is where the private IP Address gets changed to the real IP Address.

5.1 Packet Flowchart

The following flowchart will help you understanding the sequence of events. You can use this flowchart when you begin to write your own firewall script.


Network
mangle
PREROUTING
TOS, TTL, and MARK
nat
PREROUTING
DNAT
NO
Routing?
YES
mangle
INPUT
filter
INPUT
DROP
ACCEPT
REJECT
mangle
OUTPUT
nat
OUTPUT
filter
OUTPUT
mangle
FORWARD
filter
FORWARD
DROP
ACCEPT
REJECT
mangle
POSTROUTING
nat
POSTROUTING
SNAT
MASQUERADE
Network

!

As you can see from the above flowchart if a packet is to be routed to another box, using the filter input chain to drop a packet will not work. There are several approaches that you can use to make sure all packets are checked. The first option would be to create a new chain that has both input and forward point to it. An alternative method is to filter packets prior to the routing decision in the nat prerouting chain. According to the various iptables documentation this is not the best method. I like creating a chain called block and making both input and forward jump to it.

```
iptables -A INPUT -j block
iptables -A FORWARD -j block
```

This causes both the INPUT and the FORWARD chains of the filter table use the same rules. This method came from the early documentation by Rusty. Also you do not have to be concerned if a packet is being forwarded or not when you create your rules.

6 Writing a Firewall Script

Now that we know what the targets, matches, states, and how a packet traverses iptables, we are ready to write a script to protect our network. This script will not only protect us from the outside, but will also protect us from unauthorized use of the Internet. Thus we can set up and determine who has access to the Internet and who does not.

```
#!/bin/bash

# Turn off IP Forwarding
echo 0 > /proc/sys/net/ipv4/ip_forward

# Set up some variables
MailIP="192.168.10.1"
WebIP="192.168.10.2"
DMZ_IP="192.168.10.0"
LAN_IP="192.168.1.0"
RealIP=""
Internet="eth0"
DMZ="eth1"
LAN="eth2"


             ###########################################
             #                                         #
             #                  NAT                    #
             #                                         #
             ###########################################


iptables -t nat -F PREROUTING
iptables -t nat -F POSTROUTING



# Set up the ip forwarding for the local network to get to the outside:
# Use /etc/services for the destination port
iptables -t nat -A POSTROUTING -o $Internet -p tcp -s $LAN_IP/16 --dport 20  \
        -j SNAT --to $RealIP
iptables -t nat -A POSTROUTING -o $Internet -p tcp -s $LAN_IP/16 --dport 21  \
        -j SNAT --to $RealIP
iptables -t nat -A POSTROUTING -o $Internet -p tcp -s $LAN_IP/24 --dport 22  \
        -j SNAT --to $RealIP

# Block these nasties from being forwarded
# Firewall acting as a virus scanner.  Notice that these are being dropped
# on all ports, states, addresses, and interfaces.
# Also you should notice that our log file will tell us that we are are being
# attacked.
# For this to work you need the string match included in your kernel.
iptables -t nat -A PREROUTING -p tcp --dport http -m state                   \
        --state NEW,ESTABLISHED,RELATED -m string --string "/default.ida"    \
        -m limit --limit 1/hour -j LOG --log-prefix "CodeRed virus "
iptables -t nat -A PREROUTING -p tcp --dport http -m state                   \
        --state NEW,ESTABLISHED,RELATED -m string --string "/default.ida"    \
        -j DROP

# Block Port Scans

iptables -t nat -A PREROUTING            -m psd -m limit --limit 1/hour      \
        -j LOG --log-prefix "Port Scan "
iptables -t nat -A PREROUTING -s ! 192.168.1.1 -i ! $LAN  -m psd -j DROP

# Forward These

iptables -t nat -A PREROUTING  -i $Internet -p tcp --dport 80 -m limit       \
        --limit 1/hour  -j LOG --log-level info                              \
        --log-prefix "Forward WWW Request "
iptables -t nat -A PREROUTING  -i $Internet -p tcp --dport 80  -j DNAT       \
        --to $WebIP


# Turn on IP Forwarding
echo 1 > /proc/sys/net/ipv4/ip_forward
```

These few lines of code are just examples for you to get started. You should DROP everything that is not specifically allowed. If you only have one real IP address, the only thing that you might allow access to your firewall is DNS if you are running named. Everything else gets forwarded or dropped.

The fewer things that you allow the safer you will be.

When writing your firewall include a log entry for every rule with a good −−log−prefix statement. This will not only help in the debugging process, but will also show you what your firewall is doing. This is very vital for accepted packets. You might find that you are allowing packets through that you do not want. Also keep tabs on your /var/log/messages file.

7 GNU Free Documentation License

Version 1.2, November 2002

PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document “free” in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of “copyleft”, which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

7.1 APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The “Title Page” means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, “Title Page” means the text near the most prominent appearance of the work’s title, preceding the beginning of the body of the text.

A section “Entitled XYZ” means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as “Acknowledgements”, “Dedications”, “Endorsements”, or “History”.) To “Preserve the Title” of such a section when you modify the Document means that it remains a section “Entitled XYZ” according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

7.2 VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 7.3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

7.3 COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document’s license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

7.4 MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 7.2 and 7.3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

1. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

2. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

3. State on the Title page the name of the publisher of the Modified Version, as the publisher.

4. Preserve all the copyright notices of the Document.

5. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

6. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

7. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

8. Include an unaltered copy of this License.

9. Preserve the section Entitled “History”, Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled “History” in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

10. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the “History” section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

11. For any section Entitled “Acknowledgements” or “Dedications”, Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

12. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

13. Delete any section Entitled “Endorsements”. Such a section may not be included in the Modified Version.

14. Do not retitle any existing section to be Entitled “Endorsements” or to conflict in title with any Invariant Section.

15. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version’s license notice. These titles must be distinct from any other section titles.

You may add a section Entitled “Endorsements”, provided it contains nothing but endorsements of your Modified Version by various parties–for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

7.5 COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 7.4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled “History” in the various original documents, forming one section Entitled “History”; likewise combine any sections Entitled “Acknowledgements”, and any sections Entitled “Dedications”. You must delete all sections Entitled “Endorsements”.

7.6 COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7.7 AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an “aggregate” if the copyright resulting from the compilation is not used to limit the legal rights of the compilation’s users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 7.3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document’s Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

7.8 TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 7.4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled “Acknowledgements”, “Dedications”, or “History”, the requirement (section 7.4) to Preserve its Title (section 7.1) will typically require changing the actual title.

7.9 TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

7.10 FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License “or any later version” applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

> Copyright © YEAR YOUR NAME. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with...Texts." line with this:

with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.