
Understanding and Using SuSE Firewall2

Togan Muftuoglu

Copyright © 2002 Togan Muftuoglu

Copyright (c) 2002 Togan Muftuoglu. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.1 or any later version published by the Free Software Foundation; with no Invariant Sections, with no Front-Cover Texts, and with no Back-Cover Texts. A copy of the license is included in the Appendix.

Revision History

Revision 0.8

16/10/2002

Draft for review

Table of Contents

Preface	1
Introduction to SuSEfirewall2 package	3
Technical Background of SuSEFirewall2	3
Variables used in SuSEfirewall2	6
SuSEfirewall2 Command parameters	15
Basic Configuration of SuSEFirewall2	15
Configuration of SuSEFirewall2 for Proxy masquerading	17
Configuration of SuSEfirewall2 for Firewall Masquearading	19
Configuring the SuSEfirewall2 for using DMZ	20
Expert Level Configuration of SuSEfirewall2	25
Configuring SuSEfirewall2 for VPN	32
Using custom rules with SuSEfirewall2	33
Understanding SuSEfirewall2 log messages	35
Cookbook Recipies	38
SuSEfirewall2 Frequently asked questions	40
Resources on the Web	41
Colophon	42
A. GNU Free Documentation License	43
GNU Free Documentation License	43
How to use this License for your documents	47

Preface

This document tries to explain how to configure the excellent SuSEfirewall2 packet filtering script in a laymans level. Most of the information you will find on these pages are basicly collected by reading the documentation, the SuSEfirewall2 script itself and **searching the fine web** STFW for SuSEfirewall2 related questions.

Although naming the program as a firewall is theoreticly wrong as it is a highly configurable and effective packet filter

says Marc Heuse yet I will refer it as a *firewall* just to make things easier for readabilty.

Why read this article

In the hope of making SuSEfirewall2 usage easier and to avoid fighting simple issues encountered by SuSE Linux users this article tries to cover the following

- Understanding how SuSEfirewall2 works

- Understanding the parameters of SuSEfirewall2 and assigning the needed variables.
- How to do troubleshooting

Acknowledgements

- Actually the whole credit should go to Marc Heuse since I just reformatted what was in the script and the configuration file
- SuSE-Security mailinglist has valuable members. Many of the cookbook recipies came from that mailinglist

Conventions used in this article

In the hope of making ease of readability the following are used:

Choices

In order to clarify the choices available for the selected variable these are blue colored if there is a default setting this is highlighted with bold black color.

Choice "yes" or "no" defaults to "no"

Requires

Sometimes in order for the parameter to function another variable needs to be configured also. This is identified as follows.

REQUIRES:FW_ROUTE

with the required variable linked

Note

In order to identify **Notes** the following notation is used.



This is informative note

Tip

In order to identify **Tip** the following notation is used.



This is a helpful hint

Warning

In order to point out a **Warning**, the following notation is used.

Warning

This is a warning message. Be sure to read it carefully.

Caution

In order to identify a **Caution** information, the following notation is used.

Caution

Be carefull this is a **Caution** area

Introduction to SuSEfirewall2 package

Where to get from

The most current version of the SuSE Firewall2 can be found at <http://www.suse.de/~marc/suse.html>. The version that comes with SuSE 7.3 CD's is old and has some bugs which were fixed later and hence the reason to get the latest from Marc's page.

If you are using SuSE 8.0 due to the fact of `/etc/sysconfig` directory structure, the correct version is the one that comes with the *SuSE 8.0 CD/DVD*

This article will cover the version compatible with SuSE 8.0

How to install SuSEfirewall2

Depending on where you get the SuSEFirewall2 the installation method could be via YaST or via `rpm -Uhv packagename.rpm` or via `tar zxvf package.tar.gz` and changing in to the directory and issuing the `./INSTALL` command

Regardless of the method of installation you should read the documentation that comes with the firewall package which will walk you through how it should be configured for your specific case.

Here is the list of **documentation files** located at `/usr/share/doc/packages/SuSEfirewall2`. You may want read these before configuring the firewall

- SuSEfirewall2.conf.EXAMPLE
- FAQ
- PROBLEMS

The configuration file is placed under the `/etc/sysconfig/SuSEfirewall2` and it does include explanations of the various parameters.

Technical Background of SuSEFirewall2

How does SuSEfirewall2 starts at boot time

This section will try to explain what happens when you have configured `/etc/sysconfig/SuSEfirewall2` and the program SuSEfirewall2 starts.

1. Important is the initialization of the firewall during the boot up. First `/etc/init.d/SuSEfirewall2_init` is called, making sure no incoming network traffic allowed except `dhcpd +ping` (used for boot security).

```
case "$1" in
  start)
    echo -n "Starting Firewall Initialization: "
    echo -n '(phase 1 of 3) '
    ( $SUSEFWALL close ) > /dev/null 2>&1 || return=$rc_failed
    echo -e "$return" ;;
```



`/etc/init.d/SuSEfirewall2_init start` calls `/sbin/SuSEfirewall2` with the `close` parameter.

```
test "$1" = close && {
  ( rmmod ipfwadm; rmmod ipchains; modprobe ip_tables; ) > /dev/null 2>&1
  ( echo 0 > /proc/sys/net/ipv4/ip_forward > /dev/null 2>&1 ) > /dev/null
  2>&1
  $IPTABLES -F INPUT
```

```

$IPTABLES -F OUTPUT
$IPTABLES -F FORWARD
$IPTABLES -P INPUT DROP
$IPTABLES -P FORWARD DROP
$IPTABLES -P OUTPUT ACCEPT
$IPTABLES -F
$IPTABLES -X
$IPTABLES -t nat -F
$IPTABLES -t nat -X
$IPTABLES -t mangle -F
$IPTABLES -t mangle -X
$IPTABLES -A INPUT -j ACCEPT -p udp --sport 67 -d 255.255.255.255/32 --
dport 68
$IPTABLES -A INPUT -j ACCEPT -p icmp --icmp-type echo-request
$IPTABLES -A INPUT -j ACCEPT -i lo
$IPTABLES -A INPUT -j REJECT
test -x "$LOGGER" && \
    $LOGGER -p kern.info -t SuSEfirewall2 "Firewall rules set to CLOSE all
network traffic."
exit 0
}

```

2. Next when all other related init scripts have started (SuSEfirewall2_init network route dhclient) **/etc/init.d/SuSEfirewall2_setup** is called which will read and parse the `/etc/sysconfig/SuSEfirewall2` configuration file generating firewall rules for all interfaces available at this time
3. However, because more dynamic connections and services might be run later (ie. rpc, named, sshd, inetd, dhcp, nsd, nessusd, wpmd, squid, ipsec, after all these have started then **/etc/init.d/SuSEfirewall2_final** is called. All possible firewall rules are generated and also if there are error messages are given back.

It is important that from dynamic dialup scripts, the firewall should always be called from `/etc/ppp/ip-up` so that firewall rules are reloaded. This is already done by SuSE so no worries here.

Just by configuring these settings and using the SuSEfirewall2 you are not secure per se! There is not such a thing you install and hence you are saved from all (security) hazards.

Enhanced Security

The below list is placed at the very beginning of `/etc/sysconfig/SuSEfirewall2` and it says ensure your security, you need also:

- Secure all services you are offering to untrusted networks (internet) You can do this by using software which has been designed with security in mind (like postfix, apop3d, ssh), setting these up without misconfiguration and praying, that they have got really no holes. SuSEcompartment can help in most circumstances to reduce the risk.
- Do not run untrusted software. (philosophical question, can you trust SuSE or any other software distributor?)
- Harden your server(s) with the `harden_suse` package/script
- Recompile your kernel with the openwall-linux kernel patch <http://www.openwall.com> (former secure-linux patch, from Solar Designer)
- Check the security of your server(s) regularly
- If you are using this server as a firewall/bastion host to the internet for an internal network, try to run proxy services for everything and disable routing on this machine.
- If you run DNS on the firewall: disable untrusted zone transfers and either don't allow access to it from the internet or run it split-brained.

How SuSEfirewall2 works

This section will try to explain what happens when you have configured `/etc/sysconfig/SuSEfirewall2` and the program SuSEfirewall2 starts.

In this section is often talked about filter rules on internal interfaces and networks. These will only be set up if you defined those networks and interfaces and they are up and running when you are starting the firewall. (If not, no traffic on/to these is allowed)

This is how the SuSEfirewall2 script, which resides in `/sbin`, works:

- a. First it reads the configuration file `/etc/sysconfig/SuSEfirewall2`. You must first configure this configuration file before anything can happen.
- b. All tools like `sed`, `awk`, `grep`, `ifconfig`, `netstat` and of course `iptables` are searched for and if not available it terminates with an error message. If the kernel version can not be determined, it prints a warning; if it is not a 2.4.x kernel it terminates with an error message.
- c. Now all the configuration options are processed, some logical settings done and some data is read from the system, e.g. IP addresses and netmasks of interfaces. If an interface is not up, there will only some small protection rules generated. After any new connection with a dynamic IP address, this script should be run again.



If you are connected via `ppp` or `ipp` the script `/etc/ip-up` takes care of this restart automatically.

- d. The script starts! All former rules are flushed and defaults for incoming packets and those which has to be routed are set to `DROP`. This means that if there's no rule found for a special packet, it will be thrown away. All packets which leave the firewall are *allowed*.
- e. If the firewall is configured to route, the routing is activated. (You need a kernel configured for this, however, the default kernel from SuSE is.)
- f. The `/proc` system allows easy online kernel configuration. The script uses this possibility to add some security settings. However, to set most of these, you have to set the option `FW_KERNEL_SECURITY` in the configuration file to `yes`, because the results can be very complex ...
- g. Very important: all traffic is allowed via the `localhost` interface.
- h. Now the IP spoofing and circumvention rules are set, which prevents attacks in which an intruder tries to disguise his data as coming from the internal network, and direct access to the internal network.
- i. The redirection rules, which come next, can be used to direct access to the internal network or local ports to special port defined on the firewall.
- j. Is an internal (trusted?) network connected to the firewall? If this is the case, and an internal interface is defined, now the configuration option `FW_PROTECT_FROM_INTERNAL` is processed, which allows all traffic from the internal network to the firewall, if set to `yes`. Otherwise, all other filter rules notes below are also done against the internal network, unless noted otherwise.
- k. Now all ICMP, TCP and UDP rules are generated.

The ICMP rules are generated in two waves, first special ICMP packets are allowed or denied, depending on the configuration, e.g. ping to the firewall, sourcequench messages from the connected routers and replies to traceroute like programs (TTL exceeded). Next the general config which denied dangerous ICMP packets and allows important ones are set. The internal network is always allowed to ping the firewall.

- l. For the TCP rules, first all configured services which are allowed to the outside, to trusted networks and the internal one are allowed. Then port 113 is set to send `REJECT` with TCP connection resets to all incoming connections (this is important to prevent long mail sending delays).

Then a fine automation process is done. If the option `FW_AUTOPROTECT_GLOBAL_SERVICES` was set to `yes` in the configuration file, all services which are listening on all interfaces (e.g. public services, using `0.0.0.0` or `INADDR_ANY`) will be protected. This is useful if you have to open your high ports to incoming connection, but you want to ensure that e.g. your database running on port 4545 is protected.

Finally, it is decided from the configuration if/how the high/unprivileged ports (between 1024 and 65535) may be accessed: not at all, only DNS packets from name servers which are defined in `/etc/resolv.conf`, only packets which come from a special source port (not recommended, this protection is easy to subvert), or everyone. Internal networks are always allowed to access unprivileged ports (except to those protected by the AUTOPROTECT routine).

- m. The same is done for UDP, but the connection reset for port 113 is not needed here.
- n. Then come the routing rules, those, which defined to which internal systems external machines are allowed to have access to. This should of course not be used!!
- o. Now are the masquerading rules set.
- p. Very important for the configuration are additional logging mechanisms for special packets. E.g. logging all incoming TCP packets which want to initiate a connection. If a `LOG_*_ALL` option is set, all packets are logged! This is for debugging filter problems.

Last, some optimization rules are installed, to make the transfer with ssh, ftp, www, syslog and snmp faster or more reliable.

This is the end of the script. If an error occurred, it returns 1, if everything went fine, it returns 0.

Variables used in SuSEfirewall2

1. Should the Firewall be started?

This setting is done via the links in the `/etc/init.d/rc?.d` runlevel directories, which can be tweaked with a runlevel editor (or manually)

Interfaces

2. **FW_DEV_EXT**

Which is the interface that points to the internet/untrusted networks?

Enter all the network devices here which are untrusted.

Choice: any number of devices, seperated by a space e.g. "eth0", "ipp0 ipp1 eth0:1"

3. **FW_DEV_INT**

Which is the interface that points to the internal network?

Enter all the network devices here which are trusted. If you are not connected to a trusted network (e.g. you have just a dialup) leave this empty.

Choice: leave empty or any number of devices, seperated by a space e.g. "tr0", "eth0 eth1 eth1:1" or ""

4. **FW_DEV_DMZ**

Which is the interface that points to the dmz or dialup network?

Enter all the network devices here which point to the dmz/dialups. A "dmz" is a special, seperated network, which is only connected to the firewall, and should be reachable from the internet to provide services, e.g. WWW, Mail, etc. and hence are at risk from attacks. See `/usr/share/doc/packages/SuSEfirewall2/EXAMPLES` for an example.



You have to configure `FW_FORWARD` to define the services which should be available to the internet and set `FW_ROUTE` to "yes".

Choice: leave empty or any number of devices, seperated by a space e.g. "tr0", "eth0 eth1 eth1:1" or ""

5. FW_ROUTE

Should routing between the internet, dmz and internal network be activated?

REQUIRES: FW_DEV_INT or FW_DEV_DMZ

You need only set this to "yes", if you either want to masquerade internal machines or allow access to the dmz (or internal machines, but this is not a good idea). This option supersedes IP_FORWARD from /etc/sysconfig/network/options

Setting this option one alone doesn't do anything. Either activate massquerading with FW_MASQUERADE below if you want to masquerade your internal network to the internet, or configure FW_FORWARD to define what is allowed to be forwarded!

Choice: "yes" or "no", defaults to "no"

Masquearding

6. FW_MASQUERADE

Do you want to masquerade internal networks to the outside?

REQUIRES: FW_DEV_INT orFW_DEV_DMZ, FW_ROUTE

"Masquerading" means that all your internal machines which use services on the internet seem to come from your firewall.



Please note that it is more secure to communicate via proxies to the internet than masquerading. This option is required for FW_MASQ_NETS and FW_FORWARD_MASQ.

Choice: "yes" or "no", defaults to "no"

FW_MASQ_DEV

You must also define on which interface(s) to masquerade on. This is normally your external device(s) to the internet.

Most users can leave the default below.

e.g. "ipp0" or \$FW_DEV_EXT

FW_MASQ_NETS

Which internal computers/networks are allowed to access the internet directly (not via proxys on the firewall)?

Only these networks will be allowed access and will be masqueraded!

Choice: leave empty or any number of hosts/networks seperated by a space.

Every host/network may get a list of allowed services, otherwise everything is allowed. A target network, protocol and service is appended by a comma to the host/network. e.g. 10.0.0.0/8 allows the whole 10.0.0.0 network with unrestricted access. 10.0.1.0/24,0/0,tcp,80 10.0.1.0/24,0/0tcp,21 allows the 10.0.1.0 network to use www/ftp to the internet. 10.0.1.0/24,tcp,1024:65535 10.0.2.0/24 is OK too.

Set this variable to "0/0" to allow unrestricted access to the internet.

General protection

7. FW_PROTECT_FROM_INTERNAL

Do you want to protect the firewall from the internal network?

REQUIRES: FW_DEV_INT

If you set this to **"yes"**, internal machines may only access services on the machine you explicitly allow. They will be also affected from the FW_AUTOPROTECT_SERVICES option.

If you set this to **"no"**, any user can connect (and attack) any service on the firewall.

Choice: "yes" or "no", defaults to "yes"

8. FW_AUTOPROTECT_SERVICES

Do you want to autoprotect all running network services on the firewall?

If set to **"yes"**, all network access to services TCP and UDP on this machine will be prevented (except to those which you explicitly allow, see below: FW_SERVICES_{EXT,DMZ,INT}_{TCP,UDP})

Choice: "yes" or "no", defaults to "yes"

9. FW_SERVICES_**

Which services **ON THE FIREWALL** should be accessible from either the internet (or other untrusted networks), the dmz or internal (trusted networks)?



(see FW_FORWARD & FW_FORWARD_MASQ if you want to route traffic through the firewall)

Enter all ports or known portnames below, separated by a space. TCP services (e.g. SMTP, WWW) must be set in FW_SERVICES_*_TCP, and UDP services (e.g. syslog) must be set in FW_SERVICES_*_UDP. e.g. if a webserver on the firewall should be accessible from the internet:

```
FW_SERVICES_EXT_TCP="www"
```

e.g. if the firewall should receive syslog messages from the dmz:

```
FW_SERVICES_DMZ_UDP="syslog"
```

For IP protocols (like GRE for PPTP, or OSPF for routing) you need to set FW_SERVICES_*_IP with the protocol name or number (see /etc/protocols)

Choice: leave empty or any number of ports, known portnames (from /etc/services) and port ranges separated by a space. Port ranges are written like this: allow port 1 to 10 -> "1:10" e.g. " ", "smtp", "123 514", "3200:3299", "ftp 22 telnet 512:514" For FW_SERVICES_*_IP enter the protocol name (like "igmp") or number ("2")

FW_SERVICES_EXT_TCP

Common: smtp domain

```
FW_SERVICES_EXT_TCP=" "
```

FW_SERVICES_EXT_UDP

Common: domain

```
FW_SERVICES_EXT_UDP
```

FW_SERVICES_EXT_IP

For VPN/Routing which END at the firewall!!

```
FW_SERVICES_EXT_IP= " "
```

FW_SERVICES_DMZ_TCP

Common: smtp domain

```
FW_SERVICES_DMZ_TCP= " "
```

FW_SERVICES_DMZ_UDP=""

Common: domain

```
FW_SERVICES_DMZ_UDP= " "
```

FW_SERVICES_DMZ_IP

For VPN/Routing which END at the firewall!!

```
FW_SERVICES_DMZ_IP= " "
```

FW_SERVICES_INT_TCP

Common: ssh smtp domain

```
FW_SERVICES_INT_TCP= " "
```

FW_SERVICES_INT_UDP

Common: domain syslog

```
FW_SERVICES_INT_UDP= " "
```

FW_SERVICES_INT_IP

For VPN/Routing which END at the firewall!!

```
FW_SERVICES_INT_IP= " "
```

10. FW_TRUSTED_NETS

Which services should be accessible from trusted hosts/nets?

Define trusted hosts/networks (doesn't matter if they are internal or external) and the TCP and/or UDP services they are allowed to use. Please note that a trusted host/net is **not** allowed to ping the firewall until you set it to allow also icmp!

Choice: leave `FW_TRUSTED_NETS` empty or any number of computers and/or networks, separated by a space. e.g. `"172.20.1.1 172.20.0.0/16"`

Optional, enter a protocol after a comma, e.g. `"1.1.1.1,icmp"`

Optional, enter a port after a protocol, e.g. "2.2.2.2,tcp,22"

11. **FW_ALLOW_INCOMING_HIGHPORTS_***

How is access allowed to high (unprivileged [above 1023]) ports?

FW_ALLOW_INCOMING_HIGHPORTS_TCP
FW_ALLOW_INCOMING_HIGHPORTS_UDP

You may either allow everyone from anyport access to your highports ("yes"), disallow anyone ("no"), anyone who comes from a defined port (portnumber or known portname) [note that this is easy to circumvent!], or just your defined nameservers ("DNS").



Note that you can't use rpc requests (e.g. rpcinfo, showmount) as root from a firewall using this script (well, you can if you include range 600:1023 in FW_SERVICES_EXT_UDP ...).

Please note that with v2.1 "yes" is **not mandatory** for active FTP from the firewall anymore.

Choice: "yes", "no", "DNS", portnumber or known portname, defaults to "no" if not set

FW_ALLOW_INCOMING_HIGHPORTS_TCP

Common: "ftp-data", better is "yes" to be sure that everything else works :-)

FW_ALLOW_INCOMING_HIGHPORTS_UDP

Common: "DNS" or "domain" "ntp", better is "yes" to be sure ...

12. **FW_SERVICE_AUTODETECT**

Are you running some of the services below?

They need special attention - otherwise they won't work!

Set services you are running to "yes", all others to "no", defaults to "no" if not set.

FW_SERVICE_DNS

If you are running bind/named set to yes. Remember that you have to open port 53 (or "domain") as udp/tcp to allow incoming queries.

Also FW_ALLOW_INCOMING_HIGHPORTS_UDP needs to be "yes"

FW_SERVICE_DHCLIENT

if you use dhclient to get an ip address you have to set this to "yes" !

FW_SERVICE_DHCPD

set to "yes" if this server is a DHCP server

FW_SERVICE_SQUID

set to "yes" if this server is running squid. You still have to open the tcp port 3128 to allow remote access to the squid proxy service.

FW_SERVICE_SAMBA

set to "yes" if this server is running a samba server. You still have to open the tcp port 139 to allow remote access to SAMBA.

13. **FW_FORWARD**

Which services accessed from the internet should be allowed to the dmz (or internal network - if it is not masqueraded)?

REQUIRES: FW_ROUTE

With this option you may allow access to e.g. your mailserver. The machines must have valid, non-private, IP addresses which were assigned to you by your ISP. This opens a direct link to your network, so only use this option for access to your dmz!!!!

Choice: leave empty (good choice!) or use the following explained syntax of forwarding rules, seperated each by a space.

A forwarding rule consists of:

- source IP/net
- destination IP seperated by a comma. e.g. 1.1.1.1,2.2.2.2 3.3.3.3/16,4.4.4.4/24

Optional is a protocol, seperated by a comma, e.g. 5.5.5.5,6.6.6.6,igmp Optional is a port after the protocol with a comma, e.g. 0/0,0/0,udp,514

14. FW_FORWARD_MASQ

Which services accessed from the internet should be allowed to masqueraded servers (on the internal network or dmz)?

REQUIRES: FW_ROUTE

With this option you may allow access to e.g. your mailserver. The machines must be in a masqueraded segment and may not have public IP addresses!



If FW_DEV_MASQ is set to the external interface you have to set FW_FORWARD from internal to DMZ for the service as well to allow access from internal!

Please note that this should **not** be used for security reasons! You are opening a hole to your precious internal network. If e.g. the webserver there is compromised - your full internal network is compromised!!

Choice: leave empty (good choice!) or use the following explained syntax of forward masquerade rules, seperated each by a space.

A forward masquerade rule consists of:

- source IP/net
- destination IP (dmz/intern)
- a protocol (tcp/udp only!)
- destination port, seperated by a comma (" , ")

, e.g. 4.0.0.0/8,1.1.1.1,tcp,80

Optional is a port after the destination port, to redirect the request to a different destination port on the destination IP, e.g. 4.0.0.0/8,1.1.1.1,tcp,80,81

15. FW_REDIRECT

Which accesses to services should be redirected to a localport on the firewall machine?

This can be used to force all internal users to surf via your squid proxy, or transparently redirect incoming webtraffic to a secure webserver.

Choice: leave empty or use the following explained syntax of redirecting rules, seperated by a space.

A redirecting rule consists of:

- source IP/net
- destination IP/net,
- protocol (tcp or udp)
- original destination port
- local port to redirect the traffic to, seperated by a colon.

e.g.: 10.0.0.0/8,0/0,tcp,80,3128 0/0,172.20.1.1,tcp,80,8080

16. FW_LOG_*

Which logging level should be enforced?

You can define to log packets which were accepted or denied. You can also the set log level, the critical stuff or everything.



Note that logging *_ALL is only for debugging purpose ...

Choice: "yes" or "no", FW_LOG_*_CRIT defaults to "yes", FW_LOG_*_ALL defaults to ""no""

- FW_LOG_DROP_CRIT="yes"
- FW_LOG_DROP_ALL="no"
- FW_LOG_ACCEPT_CRIT="yes"
- FW_LOG_ACCEPT_ALL="no"

FW_LOG=

```
FW_LOG="--log-level warning --log-tcp-options --log-ip-option --log-prefix
SuSE-FW"
```

Only change/activate this if you know what you are doing!. To have a better undertanding of SuSEfirewall2 log process see the section called "Understanding SuSEfirewall2 log messages"

17. FW_KERNEL_SECURITY

Do you want to enable additional kernel TCP/IP security features? If set to yes, some obscure kernel options are set.

- icmp_ignore_bogus_error_responses
- icmp_echoreply_rate
- icmp_destunreach_rate
- icmp_paramprob_rate
- icmp_timeexeed_rate
- ip_local_port_range
- log_martians
- mc_forwarding
- mc_forwarding
- rp_filter
- routing flush



This is actually what is happening when you set this option and it could be very problematic in some cases. For example if you have this option turned on at first, but later you change your mind and turn it

off since changes have been implemented in the `/proc` area the results you will be getting from SuSE-firewall2 may not be what you are after

```
test "$FW_KERNEL_SECURITY" = no || {
  echo 1 > /proc/sys/net/ipv4/icmp_ignore_bogus_error_responses 2> /dev/null
  echo 5 > /proc/sys/net/ipv4/icmp_echo_reply_rate 2> /dev/null
  echo 5 > /proc/sys/net/ipv4/icmp_dest_unreach_rate 2> /dev/null
  echo 5 > /proc/sys/net/ipv4/icmp_paramprob_rate 2> /dev/null
  echo 6 > /proc/sys/net/ipv4/icmp_time_exceed_rate 2> /dev/null
  echo 20 > /proc/sys/net/ipv4/ipfrag_time 2> /dev/null
  echo 1 > /proc/sys/net/ipv4/igmp_max_memberships 2> /dev/null
  echo "1024 29999" > /proc/sys/net/ipv4/ip_local_port_range 2> /dev/null
  for i in /proc/sys/net/ipv4/conf/*; do
    echo 1 > $i/log_martians 2> /dev/null
    echo 0 > $i/bootp_relay 2> /dev/null
    test "$FW_ROUTE" = yes || ( echo 0 > $i/forwarding ) > /dev/null 2>>&1
    echo 0 > $i/proxy_arp 2> /dev/null
    echo 1 > $i/secure_redirects 2> /dev/null
  done
  echo 1 > /proc/sys/net/ipv4/route/flush
}
```



Set this to "no" until you have verified that you have got a configuration which works for you. Then set this to "yes" and keep it if everything still works. (It should!) ;-)

Choice: "yes" or "no", defaults to "yes"

18. FW_STOP_KEEP_ROUTING_STATE

Keep the routing set on, if the firewall rules are unloaded?

REQUIRES: FW_ROUTE

If you are using diald, or automatic dialing via ISDN, if packets need to be sent to the internet, you need to turn this on. The script will then not turn off routing and masquerading when stopped.

You *might* also need this if you have got a DMZ.



Please note that this is **insecure** ! If you unload the rules, but are still connected, you might your internal network open to attacks!

The better solution is to remove `"/sbin/SuSEfirewall2 stop"` or `/sbin/SuSEfirewall2 stop` from the ip-down script!

Choices: "yes" or "no", defaults to "no"

19. FW_ALLOW_PING_*

Allow (or don't) ICMP echo pings on either the firewall or the dmz from the internet? The internet option is for allowing the DMZ and the internal network to ping the internet.

REQUIRES: FW_ROUTE for FW_ALLOW_PING_DMZ and FW_ALLOW_PING_EXT

Choice: "yes" or "no", defaults to "no" if not set

```
FW_ALLOW_PING_FW="yes"
```

```
FW_ALLOW_PING_DMZ="no"
```

```
FW_ALLOW_PING_EXT="no"
```

20. FW_ALLOW_FW_TRACEROUTE

Allow (or don't) ICMP `time-to-live-exceeded` to be send from your firewall. This is used for traceroutes to your firewall (or traceroute like tools).

Please note that the unix traceroute only works if you say **"yes"** to `FW_ALLOW_INCOMING_HIGH-PORTS_UDP`, and Windows™ traceroutes only if you say *additionally* **"yes"** to `FW_ALLOW_PING_FW`

Choice: "yes" or "no", defaults to "no" if not set.

21. FW_ALLOW_FW_SOURCEQUENCH

Allow ICMP sourcequench from your ISP?

If set to yes, the firewall will notice when connection is choking, however this opens yourself to a denial of service attack. Choose your poison.

Choice: "yes" or "no", defaults to "yes"

22. FW_*_FW_BROADCAST

Allow/Ignore IP Broadcasts?

If set to yes, the firewall will not filter broadcasts by default. This is needed e.g. for Netbios/Samba, RIP, OSPF where the broadcast option is used.

If you do not want to allow them however ignore the annoying log entries, set `FW_IGNORE_FW_BROADCAST` to **"yes"**.

Choice: "yes" or "no", defaults to "no" if not set.

```
FW_ALLOW_FW_BROADCAST="no"
```

```
FW_IGNORE_FW_BROADCAST="yes"
```

23. FW_ALLOW_CLASS_ROUTING

Allow same class routing per default?

REQUIRES: `FW_ROUTE`

Do you want to allow routing between interfaces of the same class (e.g. between all internet interfaces, or all internal network interfaces) be default (so without the need setting up `FW_FORWARD` definitions)?

Choice: "yes" or "no", defaults to "no"

24. FW_CUSTOMRULES

Do you want to load customary rules from a file?

Warning

This is really an expert option. **NO HELP WILL BE GIVEN FOR THIS! READ THE EXAMPLE CUSTOMARY FILE AT** `/etc/sysconfig/scripts/SuSEfirewall2-custom`

FW_CUSTOMRULES="/etc/sysconfig/scripts/SuSEfirewall2-custom"

You will find more information on this topic in the section called "Using custom rules with SuSEfirewall2".

SuSEfirewall2 Command parameters

Although if you have configured SuSEfirewall2 properly, it will start during the system initialization time and you would not need the possible command parameters. In this section you will find the various options `/sbin/SuSEfirewall2` can take and how they can help you.

Table 1. SuSEfirewall2 command options

Option	Explanation
start	generate and load the firewall filter rules from <code>/etc/sysconfig/SuSEfirewall2</code>
stop	unload all filter rules
close	no incoming network traffic except bootp+ping (used for boot security)
file <i>FILENAME</i>	same as <i>start</i> but load alternate config file <i>FILENAME</i>
test	generate and load the filter rules but do not drop any packet but log to syslog anything which would be denied
status	print the output of <code>iptables -L -nv</code> , <code>iptables -nat -L -nv</code> and <code>iptables -t mangle -L -nv</code>
debug	print the iptables command to stdout instead of executing them
help	the output of above options



Calling `/sbin/SuSEfirewall2` without any option is the same as the *start* option. The *file FILENAME* option may be used with the *start*, *test* and *debug* options.

Configuring SuSEfirewall2 for different rules via cron

One possibility is to make use of cron to load a different configuration file.

For example during night time you may disable access to your proxy so your employees can not use your highbandwidth to download software or MP3's

```
# run at 8 pm to disable proxy and ftp access
0 20 * * * * /sbin/SuSEfirewall2 start file /home/admin/conf/firewall-nightrules
# normal firewall rules apply
30 7 * * * * /sbin/SuSEfirewall2 start
```

Basic Configuration of SuSEFirewall2

This section will try to explain the basics of configuring `/etc/sysconfig/SuSEfirewall2`. Please consult the file as this page covers very basic details.

Single user

If you are an end-user who is NOT connected to two networks (read: you have got a single user system and are using a dialup or cable modem to the internet and you are not offering any services to the world) you just have to configure the following (all other settings are OK): `FW_DEV_EXT`.

FW_DEV_EXT=""

Which is the interface that points to the internet/untrusted networks?

If you have a dialup connection or ADSL with PPPOE then your internet connection is Point to Point Protocol and you have `ppp+` device. For ISDN users the device is `ipp`

```
FW_DEV_EXT="ppp+"
```

If you have cable modem or direct Ethernet connection then you need to have your device as `ethX` where X is the ethernet number

```
FW_DEV_EXT="ethX"
```



When you have PPPOE although you are connecting the ethernet device to the ADSL modem you are actually connected to the Internet via virtual device `ppp` so ADSL users who have PPPOE connections must use `ppp+` as their connecting device

FW_STOP_KEEP_ROUTING_STATE="no"

If you are using `diald`, or automatic dialing via ISDN, if packets need to be sent to the internet, you need to turn this on. The script will then not turn off routing and masquerading when stopped.

Choices "yes" or "no", defaults to "no"

```
FW_STOP_KEEP_ROUTING_STATE="yes"
```

Examples for basic configuration

Example 1. Single User ADSL

A User with his nice SuSE Linux PC wants to be protected when connected to the internet via the ADSL dialup of his ISP. He wants to offer NO services to the internet. He is NOT connected to any other network, nor are any other network cards active.

```
FW_DEV_EXT="ppp0" # this is the adsl interface, which is using pppoe
```

Example 2. Single User ISDN

A User with his nice SuSE Linux PC wants to be protected when connected to the internet via the ISDN dialup of his ISP. He has *dial on demand* configuration He wants to offer NO services to the internet. He is NOT connected to any other network, nor are any other network cards active.

```
FW_DEV_EXT="ipp0" # this is the isdn interface
```

```
FW_STOP_KEEP_ROUTING_STATE="yes"
```

Configuration of SuSEFirewall2 for Proxy masquerading

If this server is a firewall, which should act like a proxy (no direct routing between both networks), or you are an end-user connected to the internet and to an internal network, you have to setup your proxys and reconfigure (all other settings are OK): FW_DEV_EXT, FW_DEV_INT, FW_EXT_SERVICES_*_* and maybe FW_PROTECT_INTERNAL, FW_ALLOW_INCOMING_HIGH, FW_FORWARD_MASQ

FW_DEV_INT=""

Which is the interface that points to the internal network?

Enter all the network devices here which are trusted. If you are not connected to a trusted network (e.g. you have just a dialup) leave this empty.

```
FW_DEV_INT= " "
```

If you have a LAN and other PC's are connected to the *firewall PC* then enter the device for the LAN connection. Again you can enter multiple variables seperated with space if that is your case.

```
FW_DEV_INT= "eth1 "
```



You can also enter virtual interfaces ie. eth0:1

FW_PROTECT_FROM_INTERNAL=""

Do you want to protect the firewall from the internal network?

Warning

REQUIRES: FW_DEV_INT

If you set this to *yes*, internal machines may only access services on the machine you explicitly allow. They will be also affected from the FW_AUTOPROTECT_SERVICES option.



The way you set FW_AUTOPROTECT_SERVICES may require additional configurations.

For instance if you have chosed "**yes**" for both of the parameters, the you have to state the services that are running on the firewall which local users may access. For that you will need to enter the necessary ports to FW_SERVICES_INT_TCP and FW_SERVICES_INT_UDP and maybe FW_SERVICES_INT_IP

One thing to make clear is the services mentioned here are like SSH server or SMTP server running on the *firewall machine*

If you set this to *no*, any user can connect (and attack) any service on the firewall.

Choice: "yes" or "no", defaults to "yes"

```
FW_PROTECT_FROM_INTERNAL="yes" # "yes" is a good choice
```

FW_ALLOW_INCOMING_HIGHPORTS

How is access allowed to high (unprivileged [above 1023]) ports?

You may either allow everyone from anyport access to your highports ("yes"), disallow anyone ("no"), anyone who comes from a defined port (portnumber or known portname) [note that this is easy to circumvent!], or just your defined nameservers ("DNS").



Note that if you want to use normal (active) ftp, you have to set the TCP option to **ftp-data**. If you use passive ftp, you don't need that. Note that you can't use rpc requests (e.g. rpcinfo, showmount) as root from a firewall using this script (well, you can if you include range 600:1023 in FW_SERVICES_EXT_UDP ...).

Choice: "yes", "no", "DNS", portnumber or known portname, defaults to "no" if not set

```
FW_ALLOW_INCOMING_HIGHPORTS_TCP="no"      # Common: "ftp-data"
FW_ALLOW_INCOMING_HIGHPORTS_UDP="DNS"     # Common: "DNS" or "domain ntp"
```

FW_FORWARD_MASQ=

Which services accessed from the internet should be allowed to masqueraded servers (on the internal network or dmz)?

Warning

REQUIRES: FW_ROUTE

With this option you may allow access to e.g. your mailserver. The machines must be in a masqueraded segment and may not have public IP addresses!



If dev_masq is set to the external interface you have to set FW_FORWARD from internal to DMZ for the service as well to allow access from internal!

Caution

Please note that this should **not** be used for security reasons! You are opening a hole to your precious internal network. If e.g. the webserver there is compromised - your full internal network is compromised!!

Choice: leave empty (good choice!) or use the following explained syntax of forward masquerade rules, seperated each by a space. A forward masquerade rule consists of

1. source IP/net
2. destination IP(dmz/intern)
3. a protocol (tcp/udp only!)
4. destination port, seperated by a comma (","), e.g. **4.0.0/8,1.1.1.1,tcp,80** Optional is a port after the destination port, to redirect the request to a different destination port on the destination IP, e.g. **"4.0.0/8,1.1.1.1,tcp,80,81"**

```
FW_FORWARD_MASQ="0/0,10.0.1.2,tcp,80" # Beware to use this!
```

Examples for Proxy Configuration

Example 3. Proxy Masquearding with ISDN

A company uses it's SuSE Linux PC to access the internet via an ISDN dialup of it's ISP.

It has got a web server running on the PC plus it's the mail-/pop3-server for the company. Squid is running to cache www traffic. No internal PC should have direct access to the internet.

The network address of the internal LAN is 192.168.1.0 netmask 255.255.255.0

TODO: users have to configure their mail server, pop3server and DNS to the IP of the firewall, and their web client software to use the firewall on port 3128.

TODO: The services mail, squid and pop3 have to be set up (securely).

```
FW_DEV_EXT="ipp0"  
FW_DEV_INT="eth0"  
FW_ROUTE="yes"  
FW_SERVICES_EXT_TCP="25 80"  
FW_SERVICES_INT_TCP="25 53 80 110 3128"  
FW_SERVICES_INT_UDP="53"  
FW_SERVICE_DNS="yes"  
FW_STOP_KEEP_ROUTING_STATE="yes"
```

Configuration of SuSEfirewall2 for Firewall Masquearding

If this server is a firewall, and should do routing/masquearding between the untrusted and the trusted network, you have to reconfigure (all other settings are OK): FW_DEV_EXT, FW_DEV_INT, FW_ROUTE, FW_MASQUERADE, FW_MASQ_NETS, FW_SERVICES

If you have applications running on your system that are providing services to the Internet (for example webserver) then you have to configure also the following

FW_SERVICES_EXT_TCP=""

Enter all ports or known portnames, seperated by a space. Please note that if you use service names, that they exist in `/etc/services`. There is no service dns, it's called domain; email is called smtp etc. TCP services (e.g. SMTP, WWW) must be set in FW_SERVICES_*_TCP e.g. if a webserver on the firewall should be accessible from the internet:

```
FW_SERVICES_EXT_TCP="www"
```

or if a webserver on the firewall should be reachable along with SSH you have to edit like

```
FW_SERVICES_EXT_TCP="www ssh"
```



This means the services you will be listing here are the actual services running on the firewall. If you want to use services that are available on other sites ie ftp.suse.com **do not** enter 21 here thinking you are providing access permission to your local lan to access the ftp servers which are located on the internet.

Any service type and port should be actually running on the firewall machine

FW_SERVICES_EXT_UDP

Enter all ports or known portnames, seperated by a space. UDP services (e.g. domain) must be set in FW_SERVICES_EXT_UDP

```
FW_SERVICES_EXT_UDP="53 123 " # Common: domain
```

Examples for Masquerading firewall configuration

Example 4. Masquearding Only

A small university unit wants to use masquerading to access the internet directly but wants the client PCs not directly reachable from the outside (and hence provide limited protection through this). The Firewall provides no services whatsoever.

- external fw interface=eth1
- internal fw interface=eth0
- internal LAN: 192.168.10.0/24

```
FW_DEV_EXT="eth1 "  
FW_DEV_INT="eth0 "  
FW_ROUTE="yes "  
FW_MASQUERADE="yes "  
FW_MASQ_NETS="192.168.10.0/24"
```

Configuring the SuSEfirewall2 for using DMZ

If you want to run a DMZ in either of the above three standard setups, you just have to configure **additionally**

- FW_DEV_DMZ
- FW_SERVICES_DMZ
- FW_SERVICES_AUTODETECT
- FW_FORWARD
- FW_KERNEL_SECURITY
- FW_ALLOW_PING_*

The very first variable you need to set is FW_DEV_DMZ. Once that is set you need to define the services that DMZ will be offering.

Setting up DMZ Services for TCP, UDP and IP

This is where you define the services you run on the DMZ network. it could be webserver, mailserver or an ftpserver or a name server.

- **FW_SERVICES_DMZ_TCP**
Enter the services you provide on the DMZ server. For example to provide access to your webserver you may enter

```
FW_SERVICES_DMZ_TCP="80 443 "
```



When you are providing the service on the DMZ you do not enter the same service on the FW_SERVICES_EXT_TCP

FW_SERVICES_DMZ_UDP

Enter the services you provide on the DMZ server. For example to provide access to your webserver you may enter

```
FW_SERVICES_DMZ_UDP="domain"
```



When you are providing the service on the DMZ you do not enter the same service on the FW_SERVICES_EXT_UDP

FW_SERVICES_DMZ_IP

For IP protocols (like GRE for PPTP, or OSPF for routing)

Services Autoprotect

Actually Autoprotect services feature is a complicated part of the SuSEfirewall2 even says so the author.

"... these long lines of code try to identify which services were allowed via the config file (including the DNS port for UDP) and which others are open which have to be protected. This could even be more optimised by resolving name->number and just protecting ports > 1023. I know it looks ... weird ... but it works! ;-)"

```
test "$FW_AUTOPROTECT_SERVICES" = no || {
  PROTECT_GLOBAL=`$NETSTAT -an | \
  $GREP -E '^tcp .* 0.0.0.0:[1-9].*LISTEN|^tcp .* :::[1-9].*LISTEN' | \
  $AWK '{print $4}' | $SED 's/.*/`
  for IP in $DEV_EXT; do
    PROTECT_EXT="$PROTECT_EXT ` $NETSTAT -an | \
    $AWK '/^tcp .* "$IP":[1-9].*LISTEN/ {print $4}' | $SED 's/.*/`"
  done
  for IP in $DEV_DMZ; do
    PROTECT_DMZ="$PROTECT_DMZ ` $NETSTAT -an | \
    $AWK '/^tcp .* "$IP":[1-9].*LISTEN/ {print $4}' | $SED 's/.*/`"
  done
  test "$FW_PROTECT_FROM_INTERNAL" = no || {
    for IP in $DEV_INT; do
      PROTECT_INT="$PROTECT_INT ` $NETSTAT -an | \
      $AWK '/^tcp .* "$IP":[1-9].*LISTEN/ {print $4}' | $SED 's/.*/`"
    done
    PROTECT=`for S in $PROTECT_INT $PROTECT_GLOBAL; do echo $$S; done | $SORT
  -n`
    OPENED_INT=`echo "$FW_SERVICES_INT_TCP" | $SED 's/ /|/g'`
    PROTECT_INT=`for S in $PROTECT; do echo $$S; done | grep -Evw "$OPENED_INT"`
    for PORT in $PROTECT_INT; do
      test -z "$LDC" -o -z "$LDA" && $IPTABLES -A input_int -j LOG ${LOG}"-
    DROP " -p tcp --dport $PORT --syn
      $IPTABLES -A input_int -j "$DROP" -p tcp --dport $PORT --syn
    done
  }
  PROTECT=`for S in $PROTECT_DMZ $PROTECT_GLOBAL; do echo $$S; done | $SORT
  -n`
  OPENED_DMZ=`echo "$FW_SERVICES_DMZ_TCP" | $SED 's/ /|/g'`
  PROTECT_DMZ=`for S in $PROTECT; do echo $$S; done | grep -Evw "$OPENED_DMZ`
```

```

    for PORT in $PROTECT_DMZ; do
        test -z "$LDC" -o -z "$LDA" && $IPTABLES -A input_dmz -j LOG ${LOG}"-
DROP " -p tcp --dport $PORT --syn
        $IPTABLES -A input_dmz -j "$DROP" -p tcp --dport $PORT --syn
    done
    PROTECT=`for S in $PROTECT_EXT $PROTECT_GLOBAL; do echo $S; done | $SORT
-n`
    OPENED_EXT=`echo "$FW_SERVICES_EXT_TCP" | $SED 's/ /|/g'`
    PROTECT_EXT=`for S in $PROTECT; do echo $S; done | grep -Ew "$OPENED_EXT"`
    for PORT in $PROTECT_EXT; do
        test -z "$LDC" -o -z "$LDA" && $IPTABLES -A input_ext -j LOG ${LOG}"-
DROP " -p tcp --dport $PORT --syn
        $IPTABLES -A input_ext -j "$DROP" -p tcp --dport $PORT --syn
    done
}

```

This basically searches for the listening ports on the server using netstat and then using grep and awk and sed identifies the ports to protect

```

test "$FW_SERVICE_SQUID" = yes -o "$START_SQUID" = yes && SQUID_PORT=`$LSOF -
i -n -P | \
    $GREP '^squid .* UDP \*:|' $AWK -F: '{print $2}' | \
    $AWK '{print $1}' | $SORT -un`

```

Here we also see the use of lsof in identification of the port used

```

test "$FW_SERVICE_SQUID" = yes && {
    test -z "$SQUID_PORT" && \
        echo 'Warning: FW_SERVICE_SQUID defined, but no Squid server found run-
ning!'
    test -z "$SQUID_PORT" || {
        for PORT in $SQUID_PORT; do
            for CHAIN in input_int input_dmz input_ext; do
                $LAA $IPTABLES -A $CHAIN -j LOG ${LOG}"-ACCEPT " -p udp --dport $PORT
                $IPTABLES -A $CHAIN -j "$ACCEPT" -m state --state NEW,ESTABLISHED,RE-
LATED -p udp --dport $PORT
            done
        done
    }
}

```

FW_FORWARD

Since with this option you may allow access to e.g. your mailserver. The machines must have valid, non-private, IP addresses which were assigned to you by your ISP. This opens a direct link to your network, so only use this option for access to your dmz!!!!

Lets say your web server has got an official IP address of 1.1.1.1 which you received from your ISP. You would just configure FW_FORWARD_TCP like this:

```
FW_FORWARD="0/0,1.1.1.1,tcp,80"
```

In the case you have only one official IP address that's your external Firewall IP address and you have got a web-server with a private ip placed in the dmz, you can use backward masquerading.

For this you need to set FW_ROUTE and FW_MASQUERADE to **"yes"**, and additionally FW_FORWARD_MASQ for the web servers private IP (lets say it is 10.0.0.1):

```
FW_FORWARD_MASQ="0/0,10.0.0.1,tcp,80"
```

Kernel security options

Due to the fact that when you enable this option after you disable it the results may be not exactly what you want. It is a good idea to have this option to be disabled until you are sure that everything works.

```
FW_KERNEL_SECURITY="no"
```



Inorder to refresh what happens when enabled please refer to ???

Routing state

You *might* also need to enable FW_KEEP_ROUTING_STATE got a DMZ.

This is defined in the functions part of the SuSEfirewall2

```
function reset_rules() {
    echo -n "SuSEfirewall2: clearing rules now ..."
    test "$FW_STOP_KEEP_ROUTING_STATE" = "yes" || (
        echo 0 > /proc/sys/net/ipv4/ip_forward
    ) > /dev/null 2>&1
    ....
}
```

```
FW_STOP_KEEP_ROUTING_STATE="yes"
```

Ping

Allow (or don't) ICMP echo pings on the dmz from the internet? In order to have this parameter to work FW_ROUTE should have a value of "yes"

```
# FORWARD ICMP rules
test "$FW_ALLOW_PING_DMZ" = yes -a "$FW_ROUTE" = yes && {
    for DEV in $FW_DEV_DMZ; do
        for CHAIN in forward_ext forward_int; do
            $LAA $IPTABLES -A $CHAIN -j LOG ${LOG}"-ACCEPT-PING " -p icmp --
icmp-type echo-request -o $DEV
            $IPTABLES -A $CHAIN -j "$ACCEPT" -m state --state NEW -p icmp --
icmp-type echo-request -o $DEV
        done
    done
    $LAA $IPTABLES -A forward_dmz -j LOG ${LOG}"-ACCEPT-PING " -p icmp --icmp-
type echo-reply
    $IPTABLES -A forward_dmz -j "$ACCEPT" -m state --state ESTABLISHED -p icmp
--icmp-type echo-reply
}
```

```
FW_ALLOW_PING_DMZ="no"
```



```
FW_REDIRECT="192.168.1.0/24,0/0,udp,53,53" # all DNS is done by the firewall
FW_ALLOW_PING_DMZ="yes"
```



the redirect statements here are gimmicks to show how to use it. in this example they send *any* traffic from the internal network, which go via the firewall and are destined to a target port of 53 (DNS) or 25 (Mail) to the local servers on the firewall.

Expert Level Configuration of SuSEfirewall2

If you know what you are doing, you may also change

- FW_AUTOPROTECT_SERVICES
- FW_ALLOW_HIGHPORTS_*
- FW_REDIRECT
- FW_LOG_*

and the expert options at the far end, but you should **NOT**.

- FW_ALLOW_PING_*
- FW_TRACEROUTE
- FW_ALLOW_FW_SOURCEQUENCH)
- FW_*_FW_BROADCAST)
- FW_ALLOW_CLASS_ROUTING)

Understanding how FW_AUTOPROTECT_SERVICES works

Basically the easiest way for configuring SuSEfirewall2 is to have this option set to "yes". If there are services that the internal lan needs to reach than you have to specify them on FW_SERVICES_*_* parameter

You also need to adjust the services FW_SERVICE_AUTODETECT accordingly. SuSEfirewall2 will check these adjust them in the memory and give you a warning message so you can correct your configuration

```
Warning: detected activated named, enabling FW_SERVICE_DNS!
You still have to allow tcp/udp port 53 on internal, dmz and/or external.
```

```
test "$FW_SERVICE_AUTODETECT" = yes && {
    test "$FW_SERVICE_DNS" = no && check_srv named && {
        echo -e 'Warning: detected activated named, enabling FW_SERVICE_DNS!
You still have to allow tcp/udp port 53 on internal, dmz and/or external.'
        FW_SERVICE_DNS=yes
        FW_ALLOW_INCOMING_HIGHPORTS_UDP=yes
    }
    test "$FW_SERVICE_SQUID" = no && check_srv squid && {
        echo -e 'Warning: detected activated squid, enabling FW_SERVICE_SQUID!
You still have to allow tcp port 3128 on internal, dmz and/or external.'
        FW_SERVICE_SQUID=yes
    }
    test "$FW_SERVICE_DHCPD" = no && check_srv dhcpd && {
        echo 'Warning: detected activated dhcpd, enabling FW_SERVICE_DHCPD!'
```

```

    FW_SERVICE_DHCPD=yes
  }
  test "$FW_SERVICE_SAMBA" = no && check_srv smb && {
    echo -e 'Warning: detected activated samba, enabling FW_SERVICE_SMB!
You still have to allow tcp port 139 on internal, dmz and/or external.'
    FW_SERVICE_SAMBA=yes
  }
  test "$FW_SERVICE_DHCLIENT" = no && {
    test "$IFCONFIG_0" = dhcpclient -o "$IFCONFIG_1" = dhcpclient \
      -o "$IFCONFIG_0" = bootp -o "$IFCONFIG_1" = bootp && {
      echo 'Warning: detected BOOTP/DHCLIENT usage for interfaces in
/etc/sysconfig/network/config, enabling FW_SERVICE_DHCLIENT!'
      FW_SERVICE_DHCLIENT=yes
    }
  }
}

```

Configuring FW_ALLOW_HIGHPORTS_*

By configuring the two variables

FW_ALLOW_INCOMING_HIGHPORTS_TCP

FW_ALLOW_INCOMING_HIGHPORTS_UDP

in this area you decide how access is allowed to high (unprivileged [above 1023]) ports

You may either allow everyone from anyport access to your highports ("yes"), disallow anyone ("no"), anyone who comes from a defined port (portnumber or known portname) [note that this is easy to circumvent!], or just your defined nameservers ("DNS").

```

test -e /etc/resolv.conf && NAMESERVERS=`$AWK '/^nameserver/ {print $2}'
/etc/resolv.conf | \
  $GREP -iv yast 2> /dev/null`

```

```

DONE_ALL=no
test "$FW_ALLOW_INCOMING_HIGHPORTS_TCP" = yes || {
  $LAC $IPTABLES -A $CHAIN -j LOG ${LOG}"-ACCEPT " -p tcp --dport 1024:65535
  --syn
  $LAA $IPTABLES -A $CHAIN -j LOG ${LOG}"-ACCEPT " -p tcp --dport 1024:65535
  $IPTABLES -A $CHAIN -j "$ACCEPT" -m state --state ESTABLISHED,RELATED -p
  tcp --dport 1024:65535
}
for j in $FW_ALLOW_INCOMING_HIGHPORTS_TCP; do
  case "$j" in
    no) ;;
    yes)
      for CHAIN in input_int input_dmz input_ext; do
        $LAC $IPTABLES -A $CHAIN -j LOG ${LOG}"-ACCEPT " -p tcp -
        -dport 1024:65535 --syn
        $LAA $IPTABLES -A $CHAIN -j LOG ${LOG}"-ACCEPT " -p tcp -
        -dport 1024:65535
        $IPTABLES -A $CHAIN -j "$ACCEPT" -m state --state NEW,ES-
        TABLISHED,RELATED -p tcp --dport 1024:65535
      done
      DONE_ALL=yes
      ;;
    [Dd][Nn][Ss])
      test -z "$NAMESERVERS" && \
        echo 'Warning: No nameservers in /etc/resolv.conf!'
      test "$DONE_ALL" = yes || for k in $NAMESERVERS; do
        test "$k" = 127.0.0.1 || for CHAIN in input_int input_dmz
        input_ext; do
          $LAA $IPTABLES -A $CHAIN -j LOG ${LOG}"-ACCEPT " -p

```

```

tcp -s $j --sport 53 --dport 1024:65535
        $IPTABLES -A $CHAIN -j "$ACCEPT" -m state --state ES-
TABLISHED,RELATED -p tcp -s $k --sport 53 --dport 1024:65535
        done
    done
    ;;
*)
    test "$DONE_ALL" = yes || for CHAIN in input_int input_dmz in-
put_ext; do
        $LAC $IPTABLES -A $CHAIN -j LOG ${LOG}"-ACCEPT " -p tcp -
-sport $j --dport 1024:65535 --syn
        $LAA $IPTABLES -A $CHAIN -j LOG ${LOG}"-ACCEPT " -p tcp -
-sport $j --dport 1024:65535
        $IPTABLES -A $CHAIN -j "$ACCEPT" -m state --state NEW,ES-
TABLISHED,RELATED -p tcp --sport $j --dport 1024:65535
        done
    done
    ;;
esac
done

```

So if you enter DNS as the allowed port you are basically allowing communication with your defined nameservers in `/etc/resolv.conf`. If you are only querying other nameservers for name resolution you don't need to have DNS in `FW_ALLOW_HIGHPORTS_TCP` to have DNS.



Note that you can't use rpc requests (e.g. `rpcinfo`, `showmount`) as root from a firewall using this script (well, you can if you include range `600:1023` in `FW_SERVICES_EXT_UDP` ...).



Please note that with v2.1 "yes" is **not mandatory** for active FTP from the firewall anymore. If you want to have active FTP then you should enter "**ftp-data**"

Configuring FW_REDIRECTT

This can be used to force all internal users to surf via your squid proxy, or transparently redirect incoming webtraffic to a secure webserver.

Transparent Proxy Configuration for FTP

Lets' assume the company is using SuSE-FTP proxy application and wants to implement it as a transparent proxy so the IT supervisor does not have configure the employees' ftp clients and force the users to the SuSE-FTP proxy

How does Transparent Proxy works?

Its very simple: the ip filter of you kernel redirects all packages to the ftp port (21) in "*external networks*" to the proxy and the proxy connects the server you wanted to go using informations from the ip package of your request.

Example 6. Transparent Proxy

Here is a example network configuration:

```

internal network - 192.168.1.0/24 (and others)
|
| eth0: 192.168.1.1
Firewall/Gateway + Proxy
| ippp0: 200.200.200.1
|

```

```

      |
I N T E R N E T

```

First, you want to enable the AllowTransProxy flag in the ftp-proxy.conf(5) file and start the proxy.

```

# grep -v ^# /etc/proxy-suite/ftp-proxy.conf | grep -v ^$
[-Global-]
ServerType                standalone
LogDestination            daemon
DestinationTransferMode  passive
PortResetsPasv           yes
Listen                   192.168.1.1
# may be needed in NAT'ed/Masqueraded environments
#TranslatedAddress       200.200.200.1
UseMagicChar              %
AllowMagicUser            yes
AllowTransProxy           yes

```

Second, you want to enable the transparent proxy feature in your kernel and set up the redirect rules



Default SuSE kernel has the transparent proxy feature enabled so kernel reconfiguration is not necessary

```
FW_REDIRECT="192.168.1.0/24,!192.168.1.1,tcp,21,21"
```

Transparent Proxy Configuration for Squid

Example 7. Transparent Squid Proxy

As you have seen in How does Transparent Proxy works? first prepare /etc/squid.conf for transparent proxy support and the as seen in Network drawing configure FW_REDIRECT parameter

```

# this is needed for transparent proxy:
httpd_accel_host virtual
httpd_accel_port 80
httpd_accel_with_proxy on
httpd_accel_uses_host_header on

```

and configure the firewall for using Squid

```

FW_SERVICE_SQUID="yes"

FW_REDIRECT="192.168.1.0/24,0/0,tcp,80,3128"

```

Expert log Format

The configuration of FW_LOG_* options results to the noise of your logging process:

- FW_LOG_DROP_CRIT="yes"
- FW_LOG_ACCEPT_CRIT="yes"

If you turn these to "no" the amount of logging will be less. It would be wiser to only change FW_LOG_DROP_CRIT to "no".

If for some reason you are having problems with the configuration, ie SuSEfirewall2 is preventing an access whereas you would like to permit you may change the following to "yes" for debugging reasons

- FW_LOG_DROP_ALL="no"
- FW_LOG_ACCEPT_ALL="no"



This will log everything so your logs will grow really fast

Another option is turn on kernel logging of matching packets. When this option is set for a rule, the Linux kernel will print some information on all matching packets (like most IP header fields) via the kernel log (where it can be read with dmesg or syslogd).

You can read more about the options in the section called "Understanding Iptables log parameters".

```
FW_LOG="--log-level warning --log-tcp-options --log-ip-option --log-prefix
SuSE-FW"
```

FW_ALLOW_PING_* options

ping uses the ICMP protocol's mandatory ECHO_REQUEST datagram to elicit an ICMP ECHO_RESPONSE from a host or gateway. ECHO_REQUEST datagrams ("pings") have an IP and ICMP header, followed by a struct timeval and then an arbitrary number of "pad" bytes used to fill out the packet. ¹

```
test "$FW_ROUTE" = no -a "$FW_ALLOW_PING_DMZ" = yes -o "FW_ROUTE" = no -a
"$FW_ALLOW_PING_EXT" = yes && \
    echo 'Warning FW_ROUTE needs to be set to yes, so that the FW_ALLOW_PING_EXT
and/or FW_ALLOW_PING_DMZ works!'
```

```
test "$FW_ALLOW_PING_FW" = yes && for CHAIN in input_ext input_dmz input_int;
do
    $LAA $IPTABLES -A $CHAIN -j LOG ${LOG}"-ACCEPT-PING " -p icmp --icmp-type
echo-request
    $IPTABLES -A $CHAIN -j "$ACCEPT" -p icmp --icmp-type echo-request
done
for TYPE in echo-reply destination-unreachable time-exceeded \
parameter-problem timestamp-reply address-mask-reply; do
    for CHAIN in input_ext input_dmz input_int; do
        $LAA $IPTABLES -A $CHAIN -j LOG ${LOG}"-ACCEPT-ICMP " -p icmp --icmp-
type $TYPE
        $IPTABLES -A $CHAIN -j "$ACCEPT" -m state --state ESTABLISHED,RELATED
-p icmp --icmp-type $TYPE
    done
done
# DROP rules for input ICMP are after trusted handling (see below)
....
....

# FORWARD ICMP rules
test "$FW_ALLOW_PING_DMZ" = yes -a "$FW_ROUTE" = yes && {
    for DEV in $FW_DEV_DMZ; do
        for CHAIN in forward_ext forward_int; do
            $LAA $IPTABLES -A $CHAIN -j LOG ${LOG}"-ACCEPT-PING " -p icmp --
icmp-type echo-request -o $DEV
            $IPTABLES -A $CHAIN -j "$ACCEPT" -m state --state NEW -p icmp --
icmp-type echo-request -o $DEV
        done
    done
}
```

¹refer to man ping(8) for more information

```

done
$LAA $IPTABLES -A forward_dmz -j LOG ${LOG}"-ACCEPT-PING " -p icmp --icmp-
type echo-reply
$IPTABLES -A forward_dmz -j "$ACCEPT" -m state --state ESTABLISHED -p icmp
--icmp-type echo-reply
}
test "$FW_ALLOW_PING_EXT" = yes -a "$FW_ROUTE" = yes && {
  for DEV in $FW_DEV_EXT; do
    for CHAIN in forward_int forward_dmz; do
      $LAA $IPTABLES -A $CHAIN -j LOG ${LOG}"-ACCEPT-PING " -p icmp --
icmp-type echo-request -o $DEV
      $IPTABLES -A $CHAIN -j "$ACCEPT" -m state --state NEW -p icmp --
icmp-type echo-request -o $DEV
    done
  done
  $LAA $IPTABLES -A forward_ext -j LOG ${LOG}"-ACCEPT-PING " -p icmp --icmp-
type echo-reply
  $IPTABLES -A forward_ext -j "$ACCEPT" -m state --state ESTABLISHED -p icmp
--icmp-type echo-reply
}
# drop rule for forwarding chains are at the end of the forwarding rules

```

Configuring the SuSEFirewall2 for Traceroute

What is traceroute ? The Internet is a large and complex aggregation of network hardware, connected together by gateways. Tracking the route one's packets follow (or finding the miscreant gateway that's discarding your packets) can be difficult. Traceroute utilizes the IP protocol `time to live` field and attempts to elicit an ICMP `TIME_EXCEEDED` response from each gateway along the path to some host.²

```

# OUTPUT ICMP rules
test -z "$LDC" -o -z "$LDA" -o -z "$LAC" -o -z "$LAA" && $IPTABLES -A OUTPUT
-j LOG ${LOG}"-TRACEROUTE-ATTEMPT " -p icmp --icmp-type time-exceeded
test "$FW_ALLOW_FW_TRACEROUTE" = yes && {
  $IPTABLES -A OUTPUT -j "$ACCEPT" -p icmp --icmp-type time-exceeded
  $IPTABLES -A OUTPUT -j "$ACCEPT" -p icmp --icmp-type port-unreachable
}
test "$FW_ALLOW_FW_TRACEROUTE" = yes || \
  $IPTABLES -A OUTPUT -j "$DROP" -p icmp --icmp-type time-exceeded
for TYPE in fragmentation-needed network-prohibited host-prohibited communica-
tion-prohibited; do
  $IPTABLES -A OUTPUT -j "$ACCEPT" -p icmp --icmp-type $TYPE
done
$IPTABLES -A OUTPUT -j "$DROP" -p icmp --icmp-type destination-unreachable #
we deny all other icmp type 3 codes

```

To get programs like traceroute to your firewall to work is a bit tricky, you have to set the following options in items

- `FW_ALLOW_INCOMING_HIGHPORTS_*`
- `FW_ALLOW_PING_*`
- `FW_ALLOW_FW_TRACEROUTE`

```

FW_ALLOW_INCOMING_HIGHPORTS_UDP=yes
FW_ALLOW_FW_PING=yes          #( this is for Windows PC's)
FW_ALLOW_FW_TRACEROUTE=yes

```

²for more information look at traceroute man or info pages by issuing the `man traceroute` command

Understanding FW_ALLOW_FW_SOURCEQUENCH

ICMP message types are not listed in `/etc/services`. However you may find a helpful file in `/usr/include/netinet/ip_icmp.h` that defines the names of the ICMP message numbers. ICMP, the Internet Control Message Protocol, is not exploited to break in to your site's computer systems. However, it is being used, for numerous denial of service attacks. ICMP was designed as a network health indicator



Time exceeded and port unreachable messages are also a potential result of running the traceroute program from one of your site's host computers. Traceroute sends out packets to probe for the identities of all the routers along a network path and gathers the data it needs from the ICMP messages.

SuSEfirewall2 will let only your isp to send you this when you enable `FW_ALLOW_FW_SOURCEQUENCH`

```
test "$FW_ALLOW_FW_SOURCEQUENCH" = no || for NET in $DEV_EXT_NET; do
    test -z "$LAC" -o -z "$LAA" && $IPTABLES -A input_ext -j LOG ${LOG}"-ACCEPT-
SOURCEQUENCH " -p icmp -s $NET --icmp-type source-quench
    $IPTABLES -A input_ext -j "$ACCEPT" -p icmp -s $NET --icmp-type source-quench
done
```

Understanding FW_*_FW_BROADCAST options

If set to yes, the firewall will not filter broadcasts by default. This is needed e.g. for Netbios/Samba, RIP, OSPF where the broadcast option is used.

```
test "$FW_ALLOW_FW_BROADCAST" = yes && {
    for NET in $DEV_EXT_BCAST; do
        for DEV in $FW_DEV_EXT; do
            $IPTABLES -A INPUT -j input_ext -i $DEV -d $NET
            $IPTABLES -A INPUT -j input_ext -i $DEV -d 255.255.255.255
        done
    done
    for NET in $DEV_DMZ_BCAST; do
        for DEV in $FW_DEV_DMZ; do
            $IPTABLES -A INPUT -j input_dmz -i $DEV -d $NET
            $IPTABLES -A INPUT -j input_dmz -i $DEV -d 255.255.255.255
        done
    done
    for NET in $DEV_INT_BCAST; do
        for DEV in $FW_DEV_INT; do
            $IPTABLES -A INPUT -j input_int -i $DEV -d $NET
            $IPTABLES -A INPUT -j input_int -i $DEV -d 255.255.255.255
        done
    done
done
}
```

If you do not want to allow them however ignore the annoying log entries, set `FW_IGNORE_FW_BROADCAST` to "yes"

```
broadcast stuff
test "$FW_IGNORE_FW_BROADCAST" = yes && {
    for NET in $DEV_EXT_BCAST; do
        for DEV in $FW_DEV_EXT; do
            $IPTABLES -A INPUT -j "$DROP" -i $DEV -d $NET
            $IPTABLES -A INPUT -j "$DROP" -i $DEV -d 255.255.255.255
        done
    done
    for NET in $DEV_DMZ_BCAST; do
        for DEV in $FW_DEV_DMZ; do
```

```

        $IPTABLES -A INPUT -j "$DROP" -i $DEV -d $NET
        $IPTABLES -A INPUT -j "$DROP" -i $DEV -d 255.255.255.255
    done
done
for NET in $FW_DEV_INT_BCAST; do
    for DEV in $FW_DEV_INT; do
        $IPTABLES -A INPUT -j "$DROP" -i $DEV -d $NET
        $IPTABLES -A INPUT -j "$DROP" -i $DEV -d 255.255.255.255
    done
done
done
}

```

Routing interfaces of same class

If you have two internal network cards, communication between the two networks connected to the firewall is not possible. This works as designed. For security reasons, no network may communicate to another until configured otherwise. Even if both are *trusted* internal networks. You can allow full traffic with `FW_ALLOW_CLASS_ROUTING` or specifying all allowed traffic with `FW_FORWARD`

```

test "$FW_ALLOW_CLASS_ROUTING" = yes && {
    for DEV1 in $FW_DEV_INT; do
        for DEV2 in $FW_DEV_INT; do
            test "$DEV1" = "$DEV2" || {
                $LAA $IPTABLES -A forward_int -j LOG ${LOG}"-ACCEPT-CLASS "
            }
        done
    done
    for DEV1 in $FW_DEV_DMZ; do
        for DEV2 in $FW_DEV_DMZ; do
            test "$DEV1" = "$DEV2" || {
                $LAA $IPTABLES -A forward_dmz -j LOG ${LOG}"-ACCEPT-CLASS "
            }
        done
    done
    for DEV1 in $FW_DEV_EXT; do
        for DEV2 in $FW_DEV_EXT; do
            test "$DEV1" = "$DEV2" || {
                $LAA $IPTABLES -A forward_ext -j LOG ${LOG}"-ACCEPT-CLASS "
            }
        done
    done
done
}

```

Configuring SuSEfirewall2 for VPN

Ipssec interface must be on the same `FW_DEV_*` as the real interface accepting the ipsec traffic. in this examples it's eth0, so ipsec must be in `FW_DEV_EXT`.

If you encounter problems you may refer to http://www.freeswan.org/freeswan_trees/freeswan-1.98b/doc/ for Freeswan documentation. Also there is an article by Nadeem Hasan at [Building a VPN with FreeS/WAN, SuSEfirewall2 and SSH Sentinel](#)

Example 8. Sample VPN configuration

A small company wants access to the internet for it's client PCs and additionally IPSEC with another office on another continent.

external fw interface=eth0

internal fw interface=eth1

freese/wan ipsec device=ipsec0

internal LAN: 192.168.0.0/16

remote LAN: 10.0.0.0/16

the incoming freese/wan traffic is accepted on eth0, the external interface

```
FW_DEV_EXT="eth0 ipsec0"
FW_DEV_INT="eth1"
FW_ROUTE="yes"
FW_MASQUERADE="yes"
FW_MASQ_NETS="192.168.0.0/16"
FW_FORWARD="192.168.0.0/16,10.0.0.0/16 10.0.0.0/16,192.168.0.0/16"
```

Using custom rules with SuSEfirewall2

If you are in need of some special rules for your firewall you can have SuSEfirewall2 load it automatically. This section will try explain the `/etc/sysconfig/scripts/SuSEfirewall2-custom`

Warning

This file is for SuSEfirewall2 and is an example for using the hooks which are supplied to load customized iptables rules.

THERE IS NO HELP FOR USING HOOKS EXCEPT THIS FILE ! SO READ CAREFULLY ! IT IS USEFUL TO CROSS-READ /sbin/SuSEfirewall2 TO SEE HOW HOOKS WORK !

Enabling FW_CUSTOM

In order to use custom rules together with SuSEfirewall2 you should first enable the `FW_CUSTOM` by removing the comment character `#`

```
FW_CUSTOMRULES="/etc/sysconfig/scripts/SuSEfirewall2-custom"
```

```
test -z "$FW_CUSTOMRULES" || {
    test -r "$FW_CUSTOMRULES" || {
        echo "Error: Can not read custom rules file: $FW_CUSTOMRULES"
        test -x "$LOGGER" && \
            $LOGGER -p kern.error -t SuSEfirewall2 "Firewall customary rules
file can not be read: $FW_CUSTOMRULES"
        exit -1
    }
    . "$FW_CUSTOMRULES"
    test -x "$LOGGER" && \
        $LOGGER -p kern.info -t SuSEfirewall2 "Firewall customary rules loaded
from $FW_CUSTOMRULES"
}
```

Understanding how FW_CUSTOM works

fw_custom_before_antispoofing()

these rules will be loaded before any anti spoofing rules will be loaded. Effectively the only filter lists already effective are

1. allow any traffic via the loopback interface
2. allow DHCP stuff,
3. allow SAMBA stuff [2 and 3 only if FW_SERVICE_... are set to "yes"]

You can use this hook to prevent logging of uninteresting broadcast packets or to allow certain packet through the anti-spoofing mechanism.

```
fw_custom_before_antispoofing() {
#example: allow incoming multicast packets for any routing protocol
#iptables -A INPUT -j ACCEPT -d 224.0.0.0/24

true
}
```

fw_custom_before_port_handling()

could also be named "after_antispoofing()" these rules will be loaded after the anti-spoofing and icmp handling but before any IP protocol or TCP/UDP port allow/protection rules will be set.

You can use this hook to allow/deny certain IP protocols or TCP/UDP ports before the SuSEfirewall2 generated rules are hit.

```
fw_custom_before_port_handling() { # could also be named "after_antispoofing()"
#example: always filter backorifice/netbus trojan connect requests and log
them.
#for target in LOG DROP; do
#   for chain in input_ext input_dmz input_int forward_int forward_ext for-
ward_dmz; do
#       iptables -A $chain -j $target -p tcp --dport 31337
#       iptables -A $chain -j $target -p udp --dport 31337
#       iptables -A $chain -j $target -p tcp --dport 12345:12346
#       iptables -A $chain -j $target -p udp --dport 12345:12346
#   done
#done

true
}
```

fw_custom_before_masq()

could also be named "after_port_handling()" these rules will be loaded after the IP protocol and TCP/UDP port handling, but before any IP forwarding (routing), masquerading will be done.



NOTE: reverse masquerading is before directly after fw_custom_before_port_handling !!!!

You can use this hook to ... hmmm ... I'm sure you'll find a use for this ..

```
fw_custom_before_masq() { # could also be named "after_port_handling()"
true
}
```

fw_custom_before_denyall()

could also be named "after_forwardmasq()" these are the rules to be loaded after IP forwarding and masquerading but before the logging and deny all section is set by SuSEfirewall2. You can use this hook to prevent the logging of annoying packets.

```
fw_custom_before_denyall() { # could also be named "after_forwardmasq()"
#example: prevent logging of talk requests from anywhere
#for chain in input_ext input_dmz input_int forward_int forward_ext forward_dmz;
do
# iptables -A $chain -j DENY -p udp --dport 517:518
#done
true
}
```

Understanding SuSEfirewall2 log messages

SuSEfirewall2 will create log files which are typically written to /var/log/firewall. The level of logging is based on your choice in FW_LOG_* parameter.

```
# Logging setup
LOG="--log-level warning --log-tcp-options --log-ip-options --log-prefix SuSE-FW"
test -z "$FW_LOG" || LOG="$FW_LOG"
test "$FW_LOG_DROP_CRIT" = no -o "$FW_LOG_DROP_ALL" = yes && LDC=":"
test "$FW_LOG_ACCEPT_CRIT" = no -o "$FW_LOG_ACCEPT_ALL" = yes && LAC=":"
test "$FW_LOG_DROP_ALL" = yes || LDA=":" # it might look weird - a ":"
test "$FW_LOG_ACCEPT_ALL" = yes || LAA=":" # disables logging of this type
```

Understanding Iptables log parameters

Before we can go in analyzing the logs of SuSEfirewall2 a basic understanding of iptables log target options is necessary.

Table 2. LOG target options

Option	--log-level
Example	iptables -A FORWARD -p tcp -j LOG --log-level debug
Explanation	This is the option to tell iptables and syslog which log level to use. For a complete list of log levels read the <code>syslog.conf</code> manual. Normally there are the following log levels, or priorities as they are normally referred to: debug, info, notice, warning, warn, err, error, crit, alert, emerg and panic. The keyword error is the same as err, warn is the same as warning and panic is the same as emerg. Note that all three of these are deprecated, in other words do not use error, warn and panic. The priority defines the severity of the message being logged. All messages are logged through the kernel facility. In other words, setting kern.=info /var/log/iptables in your <code>syslog.conf</code> file and then letting all your LOG messages in iptables use log level info, would make all messages appear in the /var/log/iptables file. Note that there may be other messages here as well from other parts of the kernel that uses the info priority. For more information on logging I recommend you to read the syslog and <code>syslog.conf</code> manpages as well as other HOWTOs etc.
Option	--log-prefix
Example	iptables -A INPUT -p tcp -j LOG --log-prefix "INPUT packets"

Explanation	This option tells iptables to prefix all log messages with a specific prefix, which can be easily be combined with grep or other tools to track specific problems and output from different rules. The prefix may be up to 29 letters long, including whitespaces and other special symbols.
Option	--log-tcp-sequence
Example	iptables -A INPUT -p tcp -j LOG --log-tcp-sequence
Explanation	This option will log the TCP Sequence numbers, together with the log message. The TCP Sequence number are special numbers that identify each packet and where it fits into a TCP sequence, as well as how the stream should be reassembled. Note that this option constitutes a security risk if the logs are readable by unauthorized users, or by the world for that matter. As does any log that contains output from iptables .
Option	--log-tcp-options
Example	iptables -A FORWARD -p tcp -j LOG --log-tcp-options
Explanation	The --log-tcp-options option logs the different options from the TCP packet headers and can be valuable when trying to debug what could go wrong, or what has actually gone wrong. This option does not take any variable fields or anything like that, just as most of the LOG options don't.
Option	--log-ip-options
Example	iptables -A FORWARD -p tcp -j LOG --log-ip-options
Explanation	The --log-ip-options option will log most of the IP packet header options. This works exactly the same as the --log-tcp-options option, but instead works on the IP options. These logging messages may be valuable when trying to debug or track specific culprits, as well as for debugging - in just the same way as the previous option.

Variables used in the SuSEfirewall2 generated logs

SuSEfirewall2 has a complex and helpful logging options. Before we start analyzing the logs it would be better to understand the remarks used in the logs. All logs have the default **SuSE-FW** in addition to the prefixes shown below. You can change the **SuSE-FW** by editing the `--log-prefix` in the `FW_LOG` parameter

Obviously you will not be seeing all those unless you are debugging your configuration

Table 3. SuSEfirewall2 LOG Prefixes

Remark	Explanation
ACCEPT	Packet is accepted
DROP	Packet is dropped
REJECT	Packet is rejected. This is used when there is an <code>ident</code> request
ACCEPT-CLASS	This is used when <code>FW_ALLOW_CLASS_ROUTING</code> is enabled and you have enabled <code>LOG_ACCEPT_ALL</code> for debugging purposes
ACCEPT-ICMP	This is used when <code>FW_ALLOW_PING_FW</code> is enabled and <code>LOG_ACCEPT_ALL</code> is set to <code>yes</code> for debugging purposes
ACCEPT-ALL-INTERNAL	This is used when <code>FW_PROTECT_FROM_INTERNAL</code> is set to <code>"no"</code> and <code>LOG_ACCEPT_ALL</code> is set to <code>yes</code> for debugging purposes
ACCEPT-MASQ	This is used when you have defined <code>FW_MASQ</code> and <code>LOG_ACCEPT_ALL</code> is set to <code>yes</code> for debugging purposes
ACCEPT-REVERSE-MASQ	This is used with <code>FW_FORWARD_MASQ</code> and <code>LOG_ACCEPT_ALL</code> is set to <code>yes</code> for debugging purposes or <code>LOG_ACCEPT_CRITICAL</code> is enabled
ACCEPT-PING	This is used when <code>FW_ALLOW_PING_DMZ</code> and/or <code>FW_ALLOW_PING_EXT</code> is enabled and <code>LOG_ACCEPT_ALL</code> is set to <code>yes</code> for debugging purposes

Remark	Explanation
ACCEPT-SOURCEQUENCH	This is used when you have enabled FW_ALLOW_SOURCEQUENCH and LOG_ACCEPT_ALL is set to yes for debugging purposes or LOG_ACCEPT_CRITICAL is enabled
ACCEPT-REDIRECT	This is used when you have defined FW_REDIRECT and LOG_ACCEPT_ALL is set to yes for debugging purposes or LOG_ACCEPT_CRITICAL is enabled
ACCEPT-TRUST	This is used when you have defined FW_TRUSTED_NETS and LOG_ACCEPT_ALL is set to yes for debugging purposes or LOG_ACCEPT_CRITICAL is enabled
DROP-ANTI-SPOOF	This is used when LOG_DROP_CRITICAL or LOG_DROP_ALL is enabled and if IP spoofing is encountered on the interface
DROP-CIRCUMVENION	This used when FW_ROUTE is enabled and LOG_DROP_CRITICAL or LOG_DROP_ALL is enabled
DROP-ICMP-CRIT	If LOG_DROP_CRITICAL is enabled then critical ICMP requests are logged <pre># ICMP drop rules must be here to allow trusted rules for ICMP for CHAIN in input_ext input_dmz input_int; do \$LDC \$IPTABLES -A \$CHAIN -j LOG \${LOG}"-DROP-ICMP-CRIT " -p icmp --icmp-type redirect \$LDC \$IPTABLES -A \$CHAIN -j LOG \${LOG}"-DROP-ICMP-CRIT " -p icmp --icmp-type source-quench \$LDC \$IPTABLES -A \$CHAIN -j LOG \${LOG}"-DROP-ICMP-CRIT " -p icmp --icmp-type timestamp-request \$LDC \$IPTABLES -A \$CHAIN -j LOG \${LOG}"-DROP-ICMP-CRIT " -p icmp --icmp-type address-mask-request \$LDC \$IPTABLES -A \$CHAIN -j LOG \${LOG}"-DROP-ICMP-CRIT " -p icmp --icmp-type 2</pre>
DROP-DEFAULT	These are the packet dropped by default when LOG_DROP_CRITICAL is enabled
DROP-DEFAULT-INVALID	These are the packet dropped by default if state is <i>INVALID</i> when LOG_DROP_CRITICAL is enabled
FORWARD-RELATED	This is used if you have FW_ROUTE enabled and LOG_ACCEPT_ALL is set to yes for debugging purposes
NO-ACCESS-INT->FWEXT	This is used to identify Anti Spoofing/Circumvention protection - interface dependent
TRACEROUTE-ATTEMPT	This is used with all logging options if you have enabled FW_ALLOW_FW_TRACEROUTE
UNAUTHORIZED-ROUTING	This is used to identify Anti Spoofing/Circumvention protection - interface dependent seen if FW_ROUTE is enabled and either LOG_DROP_CRITICAL or LOG_DROP_ALL is enabled
UNAUTHORIZED-TARGET	This is used to identify Anti Spoofing/Circumvention protection - interface dependent and either LOG_DROP_CRITICAL or LOG_DROP_ALL is enabled

Analyzing SuSEfirewall2 generated logs

Example 9. Drop Default

```
Oct 1 14:21:32 zeus kernel: SuSE-FW-DROP-DEFAULT IN=eth0 OUT=
MAC=00:10:4b:10:69:c1:00:20:6f:13:82:d2:08:00 SRC=111.222.333.444
DST=555.666.777.888 LEN=60 TOS=0x00 PREC=0x00 TTL=51 ID=10094 DF PROTO=TCP
SPT=1332 DPT=443 WINDOW=5840 RES=0x00 SYN URGP=0 OPT
(020405B40402080A03E4463C0000000001030300).
```

Table 4. SuSEfirewall2 log explanations

Option	Explanation
SUSE-FW-DROP-DE-FAULT	Log title produced by the SuSEfirewall2 script describing the action taken
IN	interface the packet came in on
OUT	interface packet went out on. In this case, nada
MAC	Combined mac address of sender and recipient
SRC	Source IP. "this is the ip of the attacker"
DST	Destination IP
LEN, TOS, PREC, TTL, ID	various stuff in the TCP/IP headers
PROTO	protocol of the packet
SPT	Source port "this is the port they're coming from"
DPT	Destination Port "this is the port they're trying to gain access to"
WINDOW, RES	more packet header stuff
SYN	The packet was a SYN packet, i.e. the first packet in a TCP negotiation.



The details of the header fields can be found in the RFC documents on TCP and IP ([rfc793](#), [rfc791](#)).

Cookbook Recipes

Although SuSEfirewall2 is a straightforward application there are times when you may have difficulty in setting up things. In this section you will find quick fixes for your problems.

Configuring SuSEfirewall2 for pcAnywhere

Here is an example 192.168.0.23 is an internal IP of the system behind firewall. this will accept connections from anywhere

```
FW_FORWARD_MASQ="0/0,192.168.0.23,tcp,5631 0/0,192.168.0.23,udp,5632"
```

Another example is to specify the source IP

```
FW_FORWARD_MASQ="12.45.26.3,192.168.0.23,tcp,5631 \
12.45.26.3,192.168.0.23,udp,5632"
```

In the case you want to use pcanywhere in reaching more than one Windows™ PC. Obviously, we can't use ports 5631 and 5632 on the firewall, those are now port-forwarded to the 192.168.0.15 machine. So... We'll pick a different pair (5633, and 5634), and forward them to 5631 and 5632 on 192.168.0.30.

Now our forward statement will look like this:

```
FW_FORWARD_MASQ="1.2.3.0/24,192.168.0.15,tcp,5631 \
1.2.3.0/24,192.168.0.15,udp,5631 \
1.2.3.0/24,192.168.0.15,tcp,5632 \
1.2.3.0/24,192.168.0.15,udp,5632 \
5.6.7.8/32,192.168.0.30,tcp,5633,5631 \
5.6.7.8/32,192.168.0.30,udp,5633,5631 \
```

```
5.6.7.8/32,192.168.0.30,tcp,5634,5632 \  
5.6.7.8/32,192.168.0.30,udp,5634,5632"
```

Configuring SuSEfirewall2 for edonkey

Here is an example for port forwarding edonkey to 192.168.0.2

```
FW_FORWARD_MASQ="0.0.0.0/0,192.168.0.2,tcp,4662 0.0.0.0/0,192.168.0.2,udp,4665"
```

Configuring SuSEfirewall2 for Kazaa

To enable Kazaa clients to share with other internet users; Internal network is 192.168.0.0/24

Use following Rule with SuSEfirewall2 and **No Masquerading**:

```
FW_FORWARD="192.168.0.0/24,0.0.0.0/0,tcp,1024: \  
192.168.0.0/24,0.0.0.0/0,udp,1024: 0.0.0.0/0,192.168.0.0/24,tcp.1024: \  
0.0.0.0/0,192.168.0.0/24,udp,1024: "
```

Use following Rule with SuSEfirewall2 and **Masquerading**:

```
FW_FORWARD_MASQ="192.168.0.0/24,0.0.0.0/0,tcp,1024: \  
192.168.0.0/24,0.0.0.0/0,udp,1024: 0.0.0.0/0,192.168.0.0/24,tcp.1024: \  
0.0.0.0/0,192.168.0.0/24,udp,1024: "
```



Interfaces have to be set correct in FW_DEV_EXT, FW_DEV_INT

Configuring Masquerading for specific port numbers

1. Which options I have to change in /etc/sysconfig/SuSEfirewall2, that a Client behind the wall can access the Ports 2401(tcp/udp) and 21(tcp) on the Internet directly?

try, for your example,

```
FW_MASQ_NETS=" 192.168.0.0/24,0/0,tcp,21 \  
192.168.0.0/24,0/0,tcp,2401 \  
192.169.0.0/24,0/0,udp,2401"
```



(change 192.168.0.0/24 to suite your needs)

Configuring SuSEfirewall2 for use with time servers

To configure Network time server access and response which is using UDP protocol and port number 123 you have couple of options. NTP uses source and destination port 123.

```
FW_SERVICES_EXT_UDP="123"
```

or

```
FW_TRUSTED_NETS="IP_address_of_timeserver,udp,123"
```

SuSEfirewall2 Frequently asked questions

1. Please name me some good books about TCP/IP and Firewalls.

Here are the most important books about the topics:

- Chapman/Zwicky: Building Internet Firewalls 2nd Ed., O'Reilly
- Bellovin/Cheswick: Firewalls and Internet Security, Addison Wesley
- Stevens: TCP/IP Illustrated I, Addison Wesley

2. I want to allow access to my application XYZ on my firewall

These need to be set in `FW_SERVICES_EXT_TCP`

The common problem is about what port the application uses. Let's say you are running an irc daemon and want to allow this service.

Execute `lsuf -i -n -P` and look for irc. You will see a line like this:

```
ircd      1275 irc      5u  IPv4    3097 TCP *:6667 (LISTEN)
```

This 6667 is the number you are looking for. Put this into e.g. `FW_SERVICES_EXT_TCP` and execute SuSEfirewall2 again.

3. I want to allow access to application XYZ on my internal windows machine

For this you have to use `FW_FORWARD_MASQ` and again, you need to find out the port numbers to put in there. Read the documentation of the application or run something like `tcpdump` to find this out.

If you still have got problems - sorry, you are on your own here. Ask your friends. The `EXAMPLES` file shows you the syntax and some uses.

4. I want to do IPSEC - what do I have to configure?
 - a. put the ipsec* devices into the same interface group (e.g. `FW_DEV_EXT`) where the real ipsec traffic ends on.
 - b. you might want to set `FW_FORWARD` rules to allow access
5. How can I reduce the generated rule set as most as possible?
 1. Only put in the network interfaces you really need.
 2. Disable Logging
 3. Set `FW_PROTECT_FROM_INTERNAL` to *no*
 4. Disable the service autoprotecting feature
 5. Set all `FW_ALLOW_*` and `FW_SERVICE_*` to *no*
 6. Do not use routing or masquerading :-)

7. Only enable routing/services you really need and make the statements as general as possible to reduce the number of definitions.

Then you will have got much less rules, but also a lesser security. Better spend 50\$ on an old pentium processor + board and don't use an old 486 as firewall!

6. How can I be sure that after I dial-in to the internet, that the firewall rules are active?

If you have SuSE 7.3+, the `/etc/ppp/ip-up` will take care of this, however the `INSTALL` file of SuSE-firewall2 exchanges the `ip-up` anyway if no SuSEfirewall2 support is detected. so don't care.

7. I'm still not feeling sure if my SuSEfirewall2 setup really protects me!

Run a port scanner against your firewall from the internet.

8. How many interfaces are supported for each region (EXTERNAL/DMZ/INTERNAL)?

Any number you want

9. When I connect to the internet via dial-on-demand (e.g. diald, or ipp0/ISDN) the command which activated the dial-on-demand feature fails (e.g. ping `www.suse.com`) with an error message, however the next try succeeds. Why?

Most time this is because it's the DNS lookup which fails. It fails because **expert talk on** your local resolver opens a UDP port to wait for a UDP answer from DNS servers it asks. When the feature `FW_AUTOPROTECT_SERVICES` is turned on, this will bite you, because after the dial-in is completed, the SuSEfirewall2 will protect that UDP port. hence no DNS answers can be processed, hence the initial command (e.g. ping) fails. **expert talk off**

Solution: Set `FW_AUTOPROTECT_SERVICES` to **"no"** (BAD!!) or better set up a DNS server on the firewall which just acts as a cache or forwarder.

10. I get some service not working. However it works if I disable the firewall.

Run SuSEfirewall2 in test mode: **SuSEfirewall2 test**. Then try to connect to the service in a way which failed before. It will work because SuSEfirewall2 will **not** filter any packets. However, it will log all packets to syslog it normally would have filtered.

So just check out the last lines in `/var/log/firewall` to see which ports you have to open/forward to get the service running with SuSEfirewall2.

Resources on the Web

Netfilter and SuSEfirewall2 related resources

- [Unix SysAdm Resources: Firewalls & Unix Security](#)
- [Guide to IP Layer Network Administration with Linux](#)
- [SuSEFirewall2](#)
- [sfirescan SuSEfirewall log parser](#)
- [FAQ: Firewall Forensics \(What am I seeing?\)](#)
- [Netfilter Extensions HOWTO](#)
- [Linux Networking-concepts HOWTO](#)
- [Linux 2.4 NAT HOWTO](#)
- [Iptables: Connection tracking](#)

- [Netfilter Log Format](#)

Colophon

This document was written entirely in XML, using Docbook 4.1.2 XML DTD. The PDF file was generated using Docbook XSL Stylesheets 1.56.1. Transforming the XML file to Fo was done using Saxon 6.5.2 and the resulting Fo file was converted to PDF by XEP 2.77

Emacs was used as the editing processor along with psgml extensions. SuSE Linux 8.0 was the operating platform for all of the tasks

The Unofficial SuSEFAQ project is aiming to produce documents which are supplementary to those that come with the distribution itself with an overall objective to provide answers to frequently asked questions

You are more than welcomed to contribute to the project. Details may be find in the project website <http://sf.net/projects/susefaq> hosted at Sourceforge

DRAFT

A. GNU Free Documentation License

GNU Free Documentation License

Version 1.1, March 2000

Copyright (C) 2000 Free Software Foundation, Inc. 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

PREAMBLE

The purpose of this License is to make a manual, textbook, or other written 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. Secondly, 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.

APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you".

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. (For example, 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.

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 "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, whose contents can be viewed and edited directly and 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 has been designed to thwart or discourage subsequent modification by readers is not Transparent. 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

designed for human modification. Opaque formats include PostScript, PDF, 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 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.

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 3.

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

COPYING IN QUANTITY

If you publish printed copies 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 publicly-accessible computer-network location containing a complete Transparent copy of the Document, free of added material, which the general network-using public has access to download anonymously at no charge using public-standard network protocols. 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.

MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 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:

- A. 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.

- B. 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 less than five).
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. 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.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.
- I. Preserve the section entitled "History", and 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.
- J. 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.
- K. In any section entitled "Acknowledgements" or "Dedications", preserve the section's title, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. 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.
- M. Delete any section entitled "Endorsements". Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section as "Endorsements" or to conflict in title with any Invariant Section.

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.

COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 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.

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."

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.

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, does not as a whole count as a Modified Version of the Document, provided no compilation copyright is claimed for the compilation. Such a compilation is called an "aggregate", and this License does not apply to the other self-contained works thus compiled with the Document, on account of their being thus compiled, if they are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one quarter of the entire aggregate, the Document's Cover Texts may be placed on covers that surround only the Document within the aggregate. Otherwise they must appear on covers around the whole aggregate.

TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 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 provided that you also include the original English version of this License. In case of a disagreement between the translation and the original English version of this License, the original English version will prevail.

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.

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 [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.

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 (c) YEAR YOUR NAME. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.1 or any later version published by the Free Software Foundation; with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST. A copy of the license is included in the section entitled "GNU Free Documentation License".

If you have no Invariant Sections, write "with no Invariant Sections" instead of saying which ones are invariant. If you have no Front-Cover Texts, write "no Front-Cover Texts" instead of "Front-Cover Texts being LIST"; likewise for Back-Cover Texts.

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.