HOWTO setup a home-server

From Gentoo Linux Wiki

Jump to: navigation, search

This article is part of the **HOWTO** series.

Installation • Kernel & Hardware • Networks • Portage • Software • System • X Server • Gaming • Non-x86 • Emulators • Misc

Contents

[hide]

[edit] Introduction

This HOWTO intends to describe how to configure a clean *Gentoo* box as a home gateway & firewall. While a few additional features are described, this is not intended to be an in-depth explanation of internet security.

[edit] The Gentoo Install

Firstly, follow the Gentoo Handbook installation instructions.

[edit] USE FLAGS

Disable X, as it shouldn't be needed on a firewall. It is a security hazard.

File: /etc/make.conf

```
  USE="-kde -gnome -X -gtk -qt3 -qt4"
```

File: /usr/src/linux/.config

```
# config from a gentoo-sources : 2.6.16-r7
# Loadable module support
#
CONFIG_MODULES=y
# CONFIG_MODULE_UNLOAD is not set
CONFIG_OBSOLETE_MODPARM=y
# CONFIG_MODVERSIONS is not set
# CONFIG_MODULE_SRCVERSION_ALL is not set
CONFIG_KMOD=y

#
# Networking
#
CONFIG_NET=y

#
# Networking options
#
# CONFIG_NETDEBUG is not set
CONFIG_PACKET=y
CONFIG_PACKET_MMAP=y
CONFIG_UNIX=y
# CONFIG_NET_KEY is not set
CONFIG_INET=y
CONFIG_IP_MULTICAST=y
CONFIG_IP_ADVANCED_ROUTER=y
CONFIG_ASK_IP_FIB_HASH=y
# CONFIG_IP_FIB_TRIE is not set
CONFIG_IP_FIB_HASH=y
# CONFIG_IP_MULTIPLE_TABLES is not set
# CONFIG_IP_ROUTE_MULTIPATH is not set
# CONFIG_IP_ROUTE_VERBOSE is not set
# CONFIG_IP_PNP is not set
# CONFIG_NET_IPIP is not set
# CONFIG_NET_IPGRE is not set
# CONFIG_IP_MROUTE is not set
# CONFIG_ARPD is not set
# CONFIG_SYN_COOKIES is not set
# CONFIG_INET_AH is not set
# CONFIG_INET_ESP is not set
# CONFIG_INET_IPCOMP is not set
# CONFIG_INET_TUNNEL is not set
CONFIG_INET_DIAG=y
CONFIG_INET_TCP_DIAG=y
# CONFIG_TCP_CONG_ADVANCED is not set
CONFIG_TCP_CONG_BIC=y

#
# IP: Virtual Server Configuration
#
# CONFIG_IP_VS is not set
# CONFIG_IPV6 is not set
CONFIG_NETFILTER=y
# CONFIG_NETFILTER_DEBUG is not set

#
# Core Netfilter Configuration
#
# CONFIG_NETFILTER_NETLINK is not set
CONFIG_NETFILTER_XTABLES=m
CONFIG_NETFILTER_XT_TARGET_CLASSIFY=m
# CONFIG_NETFILTER_XT_TARGET_CONNMARK is not set
CONFIG_NETFILTER_XT_TARGET_MARK=m
CONFIG_NETFILTER_XT_TARGET_NFQUEUE=m
# CONFIG_NETFILTER_XT_TARGET_NOTRACK is not set
CONFIG_NETFILTER_XT_MATCH_COMMENT=m
# CONFIG_NETFILTER_XT_MATCH_CONNBYTES is not set
# CONFIG_NETFILTER_XT_MATCH_CONNMARK is not set
CONFIG_NETFILTER_XT_MATCH_CONNTRACK=m
CONFIG_NETFILTER_XT_MATCH_DCCP=m
```

```
CONFIG_NETFILTER_XT_MATCH_HELPER=m
CONFIG_NETFILTER_XT_MATCH_LENGTH=m
CONFIG_NETFILTER_XT_MATCH_LIMIT=m
CONFIG_NETFILTER_XT_MATCH_MAC=m
CONFIG_NETFILTER_XT_MATCH_MARK=m
CONFIG_NETFILTER_XT_MATCH_PKTTYPE=m
CONFIG_NETFILTER_XT_MATCH_REALM=m
CONFIG_NETFILTER_XT_MATCH_SCTP=m
CONFIG_NETFILTER_XT_MATCH_STATE=m
CONFIG_NETFILTER_XT_MATCH_STRING=m
CONFIG_NETFILTER_XT_MATCH_TCPMSS=m

#
# IP: Netfilter Configuration
#
CONFIG_IP_NF_CONNTRACK=m
CONFIG_IP_NF_CT_ACCT=y
CONFIG_IP_NF_CONNTRACK_MARK=y
CONFIG_IP_NF_CONNTRACK_EVENTS=y
# CONFIG_IP_NF_CT_PROTO_SCTP is not set
CONFIG_IP_NF_FTP=m
CONFIG_IP_NF_IRC=m
# CONFIG_IP_NF_NETBIOS_NS is not set
# CONFIG_IP_NF_TFTP is not set
# CONFIG_IP_NF_AMANDA is not set
# CONFIG_IP_NF_PPTP is not set
# CONFIG_IP_NF_QUEUE is not set
CONFIG_IP_NF_IPTABLES=m
CONFIG_IP_NF_MATCH_IPRANGE=m
CONFIG_IP_NF_MATCH_MULTIPORT=m
CONFIG_IP_NF_MATCH_TOS=m
CONFIG_IP_NF_MATCH_RECENT=m
CONFIG_IP_NF_MATCH_ECN=m
CONFIG_IP_NF_MATCH_DSCP=m
CONFIG_IP_NF_MATCH_AH_ESP=m
CONFIG_IP_NF_MATCH_TTL=m
CONFIG_IP_NF_MATCH_OWNER=m
CONFIG_IP_NF_MATCH_ADDRTYPE=m
CONFIG_IP_NF_MATCH_HASHLIMIT=m
CONFIG_IP_NF_FILTER=m
CONFIG_IP_NF_TARGET_REJECT=m
CONFIG_IP_NF_TARGET_LOG=m
CONFIG_IP_NF_TARGET_ULOG=m
CONFIG_IP_NF_TARGET_TCPMSS=m
CONFIG_IP_NF_NAT=m
CONFIG_IP_NF_NAT_NEEDED=y
CONFIG_IP_NF_TARGET_MASQUERADE=m
CONFIG_IP_NF_TARGET_REDIRECT=m
CONFIG_IP_NF_TARGET_NETMAP=m
CONFIG_IP_NF_TARGET_SAME=m
CONFIG_IP_NF_NAT_SNMP_BASIC=m
CONFIG_IP_NF_NAT_IRC=m
CONFIG_IP_NF_NAT_FTP=m
CONFIG_IP_NF_MANGLE=m
CONFIG_IP_NF_TARGET_TOS=m
CONFIG_IP_NF_TARGET_ECN=m
CONFIG_IP_NF_TARGET_DSCP=m
CONFIG_IP_NF_TARGET_TTL=m
CONFIG_IP_NF_TARGET_CLUSTERIP=m
CONFIG_IP_NF_RAW=m
# CONFIG_IP_NF_ARPTABLES is not set

#
# DCCP Configuration (EXPERIMENTAL)
#
# CONFIG_IP_DCCP is not set

#
# SCTP Configuration (EXPERIMENTAL)
#
# CONFIG_IP_SCTP is not set

#
# TIPC Configuration (EXPERIMENTAL)
#
# CONFIG_TIPC is not set
# CONFIG_ATM is not set
# CONFIG_BRIDGE is not set
# CONFIG_VLAN_8021Q is not set
# CONFIG_DECNET is not set
# CONFIG_LLC2 is not set
# CONFIG_IPX is not set
# CONFIG_ATALK is not set
# CONFIG_X25 is not set
# CONFIG_LAPB is not set
# CONFIG_NET_DIVERT is not set
# CONFIG_ECONET is not set
# CONFIG_WAN_ROUTER is not set

#
# QoS and/or fair queueing
#
# CONFIG_NET_SCHED is not set
CONFIG_NET_CLS_ROUTE=y

#
# Network testing
#
# CONFIG_NET_PKTGEN is not set
# CONFIG_HAMRADIO is not set
# CONFIG_IRDA is not set
# CONFIG_BT is not set
# CONFIG_IEEE80211 is not set
```

Please use this only as a guideline. It fits my configuration perfectly; but if you know what you're doing, remove or add features as you desire.

For more information, read TIP Kernel Configuration.

[edit] Network

Another requirement is that your gentoo server has proper access to both your local intranet and the Internet. As this is basically covered within the Gentoo Handbook, I will not write too much about it.

[edit] Network Scenario

The scenario which is to be configured in this HOWTO looks like this:

```
 ___________       _____________       ______________       __________
|           |     |             |     |              |     |          |
| internet  |=====| cable modem |===(R) gentoo box  (G)===| intranet |
|___________|     |_____________|     |______________|     |__________|

Interface R (or Red) : eth0, 123.1.1.1.
Interface G (or Green): eth1, 192.168.0.1.
```

The server is getting its IP (123.1.1.1) on the external network interface eth0 via DHCP from the cable modem. The other interface, eth1, has a static ip (192.168.0.1) and is linked with your intranet. To achieve this, you have to edit /etc/conf.d/net with your favourite editor and change it to this:

File: /etc/conf.d/net

```
config_eth0="dhcp"
config_eth1="192.168.0.1 netmask 255.255.255.0"
```

Please comment any other lines (by puting a # on every line). Then you have to add the network devices to startup:

```
cd /etc/init.d
ln -sf net.lo net.eth0
ln -sf net.lo net.eth1
rc-update add net.eth0 default
rc-update add net.eth1 default
# start interfaces
/etc/init.d/net.eth0 start
/etc/init.d/net.eth1 start
```

If everything worked out you now should see something like this when calling *ifconfig*:

```
nexus linux # ifconfig
 eth0      Link encap:Ethernet  HWaddr 00:C5:2C:D5:F5:48
           inet addr:123.1.1.1  Bcast:123.1.1.255  Mask:255.255.255.0
           UP BROADCAST NOTRAILERS RUNNING MULTICAST  MTU:1500  Metric:1
           RX packets:2628042 errors:0 dropped:0 overruns:0 frame:0
           TX packets:1014124 errors:0 dropped:0 overruns:0 carrier:0
           collisions:0 txqueuelen:100
           RX bytes:510846263 (487.1 Mb)  TX bytes:510083552 (486.4 Mb)
           Interrupt:11 Base address:0x6c00

 eth1      Link encap:Ethernet  HWaddr 00:F0:7C:D3:B7:98
           inet addr:192.168.0.1  Bcast:192.168.0.255  Mask:255.255.255.0
           UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
           RX packets:853111 errors:0 dropped:0 overruns:0 frame:0
           TX packets:748590 errors:0 dropped:0 overruns:0 carrier:0
           collisions:0 txqueuelen:100
           RX bytes:500603484 (477.4 Mb)  TX bytes:231023059 (220.3 Mb)
           Interrupt:10 Base address:0x8800

 lo        Link encap:Local Loopback
           inet addr:127.0.0.1  Mask:255.0.0.0
           UP LOOPBACK RUNNING  MTU:16436  Metric:1
           RX packets:549 errors:0 dropped:0 overruns:0 frame:0
           TX packets:549 errors:0 dropped:0 overruns:0 carrier:0
           collisions:0 txqueuelen:0
           RX bytes:43074 (42.0 Kb)  TX bytes:43074 (42.0 Kb)
```

Congratulations! Now, as we are sure, the basic network is functioning, let's get down to business... First of all:

[edit] Firewalling

Ok, this is now very important... if your server is up 24/7 as intended, there will be lots of intrusion attempts. I think in times of W32.blaster-like worms and trojans, I really don't need to explain why a firewall is useful. Here only a basic firewall will be setup, please feel encouraged to modify the iptable-rules as desired for best fitting your needs. Masquerading simply means allowing computers of your intranet to connect to others on the Internet (more or less directly). This is not needed for just browsing the web, as we previously configured squid to do so, but for many applications (like ftp clients, instant messenger, email clients or filesharing tools) a direct connection is needed. Just to refresh your memory, this is the network scenario we talk about:

```
 __________       ______________     _______________     ___________
|          |     |             |    |             |    |          |
| internet |=====| cable modem |===(R) gentoo box (G)===| intranet |
|___________|     |_____________|    |______________|    |__________|

Interface R (or Red) : eth0, 123.1.1.1.
Interface G (or Green): eth1, 192.168.0.1.
```

At the moment, everyone within your intranet (192.168.0.*) can easily browse the Internet. But the web is more than just www. So, if, let's say, 192.168.0.15 (I'll call him foo) wants to establish a ftp connection to ftp.cdrom.com, this won't work because

- foo doesn't know how (no default gateway set yet)
- the server won't let him anyway (no routing enabled yet)
- foo can't access a dns server (you still need dnsmasq)

[edit] What is IP Masquerading?

IP Masquerading is a form of network address translation that many routers already support. The idea behind this implementation is that people running Linux can install the IP masquerading features being developed for Linux and get the features of the high priced routers and NAT boxes without paying the high prices.

IP masquerading lets you use a single Internet-connected computer running Linux with a real IP address as a gateway for non-connected machines with "fake" IP addresses. The Linux box with a real address handles mapping packets from your intranet out to the Internet, and when responses come back, it maps them back to your intranet.

Add the following rules (this is just masquerading): (part of this I took from http://tldp.org/HOWTO/Masquerading-Simple-HOWTO/ thanks very much to John Tapsell, Thomas Spellman and Matthias Grimm for letting me 8))

```
# first line will flush your rules
 iptables -F; iptables -t nat -F; iptables -t mangle -F
 iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
 echo 1 > /proc/sys/net/ipv4/conf/all/forwarding
 iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
 iptables -A INPUT -m state --state NEW -i ! eth0 -j ACCEPT
 iptables -A INPUT -p icmp -j ACCEPT
 iptables -P INPUT DROP
```

[edit] Note for people using pppoe with masquerade

For users connected via pppoe, the MTU value for the ppp0 interface might be set to less than 1500. This might result in problems sharing your network with the LAN computers.

This only applies if you want to share your internet connection directly with your LAN without having to set up a proxy (a proxy can still be set up with this configuration for other reasons).

The symptoms are:

- you can access the internet normally from the server/router
- you can access only certain websites from the LAN (the ones you can't access still seems to be connected but the page doesn't load. The browser says "Waiting for <website>..." and hags there.)
- you can connect only on certain IMs and sometimes partially (e.g. with Yahoo! Messenger, you are connected but your contact list doesn't show any contacts.)

This is due to the fact that certain ISPs or server block the ICMP Fragmentation Needed packets (see "man iptables", section TCPMSS, for more information).

The solution is to use the TCPMSS target rule from iptables that will control the maximum size for those packets.

To do so, add one of the following commands to the list of you iptables rules:

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

or:

```
# iptables -A FORWARD -o ppp0 -p tcp --tcp-flags SYN,RST SYN -j TCPMSS --clamp-mss-to-pmtu
```

[edit] Running Your Firewall/Masquerade Rules "The Gentoo Way"

As of iptables-1.2.9 (the one I have used here), the iptables ebuild installs some nice initscripts for starting the firewall during the boot sequence, in the same way as other daemons like dhcpd, ftpd and apache.

The configuration file for this initscript is `/etc/conf.d/iptables`. Here is how the config should look.

File: /etc/conf.d/iptables

```
# /etc/conf.d/iptables

# Location in which iptables initscript will save set rules on
# service shutdown
IPTABLES_SAVE="/var/lib/iptables/rules-save"

# Options to pass to iptables-save and iptables-restore
SAVE_RESTORE_OPTIONS="-c"

# Save state on stopping iptables
SAVE_ON_STOP="yes"
```

The IPTABLES_SAVE variable is where the daemon expects to find the firewall rules to activate when it starts up.

File: /etc/sysctl.conf

```
# Disables packet forwarding
#net.ipv4.ip_forward = 0
```

You should uncomment the net.ipv4.ip_forward = 0 line and change the value to 1. This will perform the echo 1 > /proc/sys/net/ipv4/ip_forward for you.

To save your configuration:

```
# /etc/init.d/iptables save
```

Then start it up:

```
# /etc/init.d/iptables start
```

Use this command to add iptables to your boot sequence:

```
# rc-update add iptables default
```

Now everything should be working and setup to automatically start up on boot too.

Note: If you install/use another firewalling system (e.g., if you choose a firewalling option in pppoe-setup), it will most likely conflict with the expected iptables initscript save/restore behavior.

You can find more documentation at the following places : HOWTO Iptables for newbies

[edit] Recommended Packages

If you run your gentoo box without screen and keyboard attached I strongly recommend to install OpenSSH, that way you can access your machine while it's running in your closet.

[edit] OpenSSH

See Index:OpenSSH Note: the default SSH config is OK to start with. Start and auto load it at boot with :

1. /etc/init.d/sshd start
2. rc-update add sshd default

[edit] DHCP

Now comes something as comfortable as having an own domain, dhcp. You may ask, what's it good for if I have a dhcp-server running for telling my 2-3 computers their IP addresses? Well, if you sometimes change your infrastructure, i.e. a few friends come around with their notebooks or of course case-modded big machines to play some relaxing 1st-person-shoot-em-ups for hours, it's much more comfy to have their IP addresses, network ranges, gateways and dns-resolver autoassigned. And well, configuration is really quite easy!

You have to choose between ISC DHCPd and dnsmasq. The dnsmasq package is an alternative to using the more common dhcpd daemon. It does everything dhcpd can, with the added coolness factor of being a local DNS server! Now you can avoid that premature aging and hair loss that comes with BIND if you would like a small DNS server for your LAN.

[edit] ISC DHCP

```
# emerge net-misc/dhcp
```

These are the only settings needed in /etc/dhcp/dhcpd.conf (copy sample before) :

File: /etc/dhcp/dhcpd.conf

```
default-lease-time 3600;
max-lease-time 7200;
authorative;
log-facility local7;
ddns-update-style ad-hoc;

subnet 192.168.0.0 netmask 255.255.255.0 {
        range 192.168.0.100 192.168.0.250;

        option subnet-mask 255.255.255.0;
        option netbios-name-servers 192.168.0.1;
        option broadcast-address 192.168.0.255;
        option routers 192.168.0.1;
        option domain-name-servers 192.168.0.1;

        host VeroMars
        {
                hardware ethernet 00:50:8D:6C:AA:BB;
                fixed-address 192.168.0.50;
        }
}
```

The section beginning with host VeroMars just makes sure that the machine with the mac-address 00:50:8D:6C:AA:BB always gets the same IP address. Be sure that this specially assigned ip is not in the range of the dynamically assigned IP addresses (else, I have been told, hell breaks loose!).

Before we start the dhcp daemon, we need to change the interface on which dhcp will listen for dhcp requests. Open /etc/conf.d/dhcp and change IFACE="eth0" to IFACE="eth1". Save and quit your favourite editor.

Okay, now we just have to start the dhcp daemon, and then change all your client PC's settings to retrieve IP addresses and nameservers via dhcp (so that the dhcp-server makes some sense).

```
# rc-update add dhcpd default
# /etc/init.d/dhcpd start
```

[edit] DNSMASQ

One small caveat here - the machine that you install dnsmasq on should have a fully functional /etc/resolv.conf file containing valid "upstream" DNS servers.

Note: You may have to add 127.0.0.1 to the top of the nameserver list in /etc/resolv.conf on the server.

To get installed dnsmasq, simply emerge the package:

```
emerge dnsmasq
```

dnsmasq has a wealth of options, but we'll just concentrate on duplicating the functionality of dhcpd and setting up a small DNS server. Open the /etc/dnsmasq.conf file in your chosen text editor, and look for this section:

File: /etc/dnsmasq.conf

```
# Uncomment this to enable the integrated DHCP server, you need
# to supply the range of addresses available for lease and optionally
# a lease time. If you have more than one network, you will need to
# repeat this for each network on which you want to supply DHCP
# service.
dhcp-range=192.168.0.50,192.168.0.150
```

Note that I have uncommented the line beginning dhcp-range=... - this is the line that does the dynamic IP magic and is much simpler that dhcpd! Just change the range of IP addresses to suit your purposes.

We also want to be able to serve up specific IP addresses to certain machines, depending on their MAC addresses. Scroll down to find this block:

```
# Always allocate the host with ethernet address 11:22:33:44:55:66
# The IP address 192.168.0.60
dhcp-host=11:22:33:44:55:66,192.168.0.60
```

Once again, I've uncommented the line beginning dhcp-host=... You can repeat this line as many times as you want, with the appropiate MAC/IP address combination.

You may want to set the interface that dnsmasq will listen on, especially if you have direct access to the WAN subnet. Here I set it to only serve clients on the subnet connected to eth1.

```
# If you want dnsmasq to listen for requests only on specified interfaces
# (and the loopback) give the name of the interface (eg eth0) here.
# Repeat the line for more than one interface.
interface=eth1
```

[edit] Optional packages

You don't need these packages to reproduce a simple router. But they add features or enhance the security on your network and/or the router. Remember that every **extra** service creates potential leaks. Only typical router services are described indepth. Others are mentioned (like Apache2) and linked to the correct wiki page where you can continue.

[edit] Squid

Squid is an advanced proxy that enables you to restrict, filter and even cache web pages.

Squid is easily configured, don't be scared by the 110kb squid.conf-sample, most of it are very easily understood comments. Here is what I had to change from the squid.conf.sample (my whole config is here, try a *grep -v ^# /etc/squid/squid.conf* | *grep -v ^$* to only see your changes from default values):

```
...
http_port 192.168.0.1:3128
icp_port 0
cache_mem 20 MB

cache_dir ufs /usr/tmp/squid 256 16 256
# change this path to somewhere you have enough diskspace

acl all src 0.0.0.0/0.0.0.0
acl manager proto cache_object
acl allowed_hosts src 192.168.0.0/255.255.255.0

#http_access allow purge localhost
#http_access deny purge

#http_access deny !Safe_ports
#http_access deny CONNECT !SSL_ports

acl our_networks src 192.168.0.0/24
http_access allow our_networks
```

```
icp_access allow allowed_hosts
icp_access deny all

miss_access allow allowed_hosts
miss_access deny all
```

To use squid (prior to version 2.6) as a transparent proxy add also the following:

```
httpd_accel_host virtual
httpd_accel_port 80
httpd_accel_with_proxy on
httpd_accel_uses_host_header on
```

For version 2.6 add:

```
http_port 3128 transparent
```

With this squid.conf, your proxy can only be accessed from your intranet with a storage of 256 MB for its temporary files.

Then we have to let Squid create its directories, add it to startup and run it.

```
# squid -z
# rc-update add squid default
# /etc/init.d/squid start
```

Finally you need to make your clients use the proxy, which means setting their browsers for the proxy. Or if you want transparent proxying, you just need to redirect the traffic to your proxy.

If your server is your LAN network router/firewall, you just need to add this rule to iptables (assuming eth0 is on the LAN side):

```
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 80 -j REDIRECT --to-port 3128
```

Tip: If you want squid to shutdown faster, see the shutdown_lifetime option (default is 30 second, which is good for general use, but in specific cases it may be too long)

[edit] Apache 2

See Index:Apache2

[edit] ProFTPd

See HOWTO ProFTPD

[edit] Configuring dynamic dns

As not everyone is gifted with an excellent memory for numbers, and as it doesn't make any sense to remember dynamic IP addresses or tell every friend of yours, and first of all since it's easy to configure and free, go to www.dyndns.org and register a dynamic domain name.

You can update your dynamic domain using ddclient

Note: If you are using an ADSL modem with pppd, you should move /etc/ddclient/sample-etc_ppp_ip-up.local to /etc/ppp/ip-up.local. This way, pppd will launch ddclient (with the options given in /etc/ddclient/ddclient.conf) at each (re)connection.

And if you use nsswitch.ldap like me be sure to have this line

```
protocols: files ldap
```

[edit] DNS

Since it's easier to remember names than numbers alot of people setup a internal DNS server. That way they can create aliases for their intranet.

A Note on the local Top-Level Domain (TLD) and ZeroConf (Apple Rendezvouz/Bonjour): In principle, you can use any local TLD you like, as long as it is different from the TLDs already used on the internet. (A TLD is the last or right-most part of a domain. Examples of TLDs used on the internet are *.de*, *.com*, *.net*, *.jp* or *.uk*). The configuration exaple below uses *.lan*, which is perfectly fine, just like *.yourname* or anything else not already assigned. However, if you happen to have one or more Apple Macintosh computers or other devices that incorporate the Zeroconf-Standard [1] (Also known as Rendezvous or Bonjour to Macintosh users [2]) connected to your local network, you should strongly consider using *.local* as your local TLD, since Zeroconf uses *.local* as default local TLD for all devices registered via Zeroconf. If you do this you can add Non-Zeroconf Devices via your DNS-Server (BIND does that for me) and have all your computers and devices in one TLD. Otherwise you will have two TLDs floating around in your local lan, which is not a very good idea and might require you to remember which of your two TLDs you need to use to adress what device from what device.

[edit] ISC BIND

First off, BIND is a hightech DNS server. It's not easy to configure and probably create a few headaches. Start by emerging bind.

```
# emerge bind bind-tools
```

Bind-tools is a package which allows you to use dig,nslookup and host. Let's edit the main configuration file of bind found at /etc/bind/named.conf.

File: /etc/bind/named.conf

```
options {
        directory "/var/bind";

        listen-on-v6 { none; };
        listen-on {
                127.0.0.1;
                192.168.0.1;
        };

        allow-query {
                127.0.0.1;
                192.168.0.0/16;
        };

        forwarders {
                195.130.130.132;
                195.130.131.9;
        };

        pid-file "/var/run/named/named.pid";
};

zone "." IN {
        type hint;
        file "named.ca";
};

zone "localhost" IN {
        type master;
        file "pri/localhost.zone";
        allow-update { none; };
        notify no;
};

zone "127.in-addr.arpa" IN {
        type master;
        file "pri/127.zone";
        allow-update { none; };
        notify no;
};

zone "mars.lan" IN {
        type master;
        file "pri/mars.lan.zone";
        allow-update { none; };
        notify no;
};

zone "0.168.192.in-addr.arpa" IN {
        type master;
        file "pri/0.168.192.zone";
        allow-update { none; };
        notify no;
};
```

This configuration file needs a little explanation. listen-on and allow-query define on what ip-address bind is listening and from what ip-address' it accepts DNS queries. forwarders defines to where DNS requests are sent when BIND hasn't an answer. Change them to your ISP's DNS servers. The ones in the example are from Telenet (Belgium). The zone statement 'mars.lan' defines a zone 'mars.lan' . The configuration file for that zone is found in : /var/bind/pri/mars.lan.zone. The reverse DNS zone file is called 0.168.192.zone. It's placed in /var/bind/pri also.

File: /var/bind/pri/mars.lan.zone

```
$TTL 86400
@               IN SOA  boxname.mars.lan        dnsadmin@mars.lan (
                2005101003      ;serial
                10800           ;refresh
                7200            ;retry
                36000000        ;expire
                86400)          ;default minimum ttl
                IN NS   boxname.mars.lan.

smoothwall      IN A            192.168.0.10
anlaug          IN A            192.168.0.11
printserver     IN A            192.168.0.20
3com4300        IN A            192.168.0.30
h4x0r           IN A            192.168.0.50

areabitchslap   IN A            192.168.1.2

ntp             IN CNAME        anlaug
proxy           IN CNAME        anlaug
sw              IN CNAME        smoothwall
it              IN CNAME        anlaug
ps              IN CNAME        printserver
3com            IN CNAME        3com4300
switch          IN CNAME        3com4300

abs             IN CNAME        areabitchslap
www             IN CNAME        areabitchslap
```

A bit of explanation: this config file says that smoothwall.mars.lan is equal to 192.168.0.10. But sw.mars.lan is an alias to the same IP-address.

File: /var/bind/pri/0.168.192.zone

```
$ORIGIN 0.168.192.in-addr.arpa.
$TTL 86400
@               IN SOA  boxname.mars.lan        dnsadmin@mars.lan (
                2005101003      ;serial
                10800           ;refresh
                7200            ;retry
                36000000        ;expire
                86400)          ;default minimum ttl
                IN NS   boxname.mars.lan.

10              IN PTR  smoothwall.mars.lan.
11              IN PTR  anlaug.mars.lan.
30              IN PTR  3com4300.mars.lan.
20              IN PTR  printserver.mars.lan.
50              IN PTR  h4x0r.mars.lan.
```

Adjust the files to suit your needs and save them. Now we're going to add BIND to the default system start and launch it for the first time.

```
# rc-update add named default
# /etc/init.d/named start
```

[edit] DNSMASQ

[edit] Static /etc/hosts File

If the machine running dnsmasq has a nice, full /etc/hosts file, other machines on your LAN will be able to look up each other without maintaining their own hosts file - a nice centralization.

Here's my own hosts file on my dnsmasq box:

```
127.0.0.1       localhost
192.168.0.1     tux.homenetwork tux
192.168.0.2     idontcare.homenetwork idontcare
192.168.0.3     goblin.homenetwork goblin
192.168.0.100   gateway.homenetwork gateway
192.168.0.10    ultra.homenetwork ultra
192.168.0.11    sparc20.homenetwork sparc20
```

Now I can be logged into tux, and ping ultra and dnsmasq will dig out the right IP address!

[edit] Dynamic via the -l Option

Since you have dnsmasq working as a DHCP server, why not use the hostnames your clients pass as part of their DHCP requests?

1. Alter /etc/dnsmasq.conf to ensure that we know where the leases file will be

File: /etc/dnsmasq.conf

```
# The DHCP server needs somewhere on disk to keep its lease database.
# This defaults to a sane location, but if you want to change it, use
# the line below.
dhcp-leasefile=/var/lib/misc/dnsmasq.leases
```

2. Alter /etc/conf.d/dnsmasq

File: /etc/conf.d/dnsmasq

```
DNSMASQ_OPTS="-l /var/lib/misc/dnsmasq.leases"
```

The '-l' option tells dnsmasq to look to the leases file for hostnames. So now you don't have to worry about maintaining a static hosts file.

[edit] Starting dnsmasq

Finally, you can start dnsmasq and add it to your boot sequence with:

```
# /etc/init.d/dnsmasq start
# rc-update add dnsmasq default
```

[edit] Core Software

Depending on how you install Gentoo you may have some outdated packages after the install. Before you proceed, you should do **emerge --sync** and **emerge -NuavD world** to make sure your system is up to date.

[edit] Software Manifest

- rcs - for config file version control
- snort - packet sniffer, logger, lightweight IDS
- AIDE - Advanced Intrusion Detection Environment
- rrdtool - used by ntop to store graphs
- ntop- to get statistics on your traffic.
- iptables - the heart of the matter.
- logsentry - keep tap on logs.
- ntp - get the correct time.
- openssh
- sSMTP - an extremely simple MTA
- Squid - caching web proxy

[edit] Install

So we don't re-emerge software and to make sure everything is updated:

emerge -uav rcs snort aide rrdtool ntop iptables logsentry ntp openssh squid.

[edit] sSMTP

sSSMTP must be compiled with the **mailwrapper** useflag for some things to work. To do that type in **echo "mail-mta/ssmtp mailwrapper" >> /etc/portage/package.use** and then emerge it.

1. **cd /etc/ssmtp**
2. **mkdir RCS**
3. **ci -l ssmtp.conf**
4. **$EDITOR ssmtp.conf**
5. update /etc/ssmtp/revaliases
6. update /etc/passwd
7. Send a test mail: **echo "Is this working?" | mail -s "Test" myaunt@spamcity.world**

change roots name from 'root' to whatever, This will then go into the from field And look better in your e-mails.

File: /etc/passwd

```
From:
root:x:0:0:root:/root:/bin/bash
To:
root:x:0:0:Benjamin Sisko:/root:/bin/bash
```

To configure sSMTP you can use the ssmtp-config command. Run it and answer the questions posed.

or update; 'mailhub' and 'hostname', 'rewriteDomain' in

File: /etc/ssmtp/ssmtp.conf

```
#
# /etc/ssmtp.conf -- a config file for sSMTP sendmail.
#
# The person who gets all mail for userids < 10
root=postmaster
# The place where the mail goes. The actual machine name is required
# no MX records are consulted. Commonly mailhosts are named mail.domain.com
# The example will fit if you are in domain.com and you mailhub is so named.
mailhub=mail.bar.baz
# Where will the mail seem to come from?
rewriteDomain=bar.baz
# The full hostname
hostname=foo.bar.baz
# Set this to never rewrite the "From:" line (unless not given) and to
# use that address in the "from line" of the envelope.
#FromLineOverride=YES
```

If 'rewriteDomain=bar.baz' is uncommented, ssmtp always rewrites the 'From' envelope and the 'From:' line so that the domain name is set to bar.baz.

File: /etc/ssmtp/revaliases

```
# sSMTP aliases
#
# Format: local_account:outgoing_address:mailhub
#
# Example: root:your_login@your.domain:mailhub.your.domain:[port]
# where [port] is an optional port number that defaults to 25.
root:bens@bar.baz:mail.bar.baz
```

[edit] Discussion

If the option 'FromLineOverride' is set to YES, ssmtp uses the same 'From' in the mail envelope as you have written into the 'From:' line of your mail.

In case you send mail through another server than the one providing your e-mail address, it is not likely that your mails come through unless the envelope from presents you as a valid user at the server. That is, you need the From: and From_ lines to differ. (For example, if the cygwin user cygwinuser wishes to send a mail with the From: line reading a.user@some.domain through the mail server mail.bar.baz, he probably needs the envelope from to be user@bar.baz.) You can do this by editing the revaliases file.

[edit] Related links

- HOWTO Gmail and sSMTP

[edit] Logsentry

1. **cd /etc/cron.hourly**
2. **mkdir RCS**
3. **ci -l logsentry.cron**
   - Give an explanation: logsentry hourly run, cron job.
4. **$EDITOR logsentry.cron**
   - remove the '#' from the line

Please note that this is for vixie-cron.

[edit] AIDE

1. **cd /etc/aide/**
2. **mkdir RCS**
3. **ci -l aide.conf**

4. **`$EDITOR aide.conf`**
5. **`/usr/bin/aide -i`**
   - This takes about 10 minutes.
6. **`mv aide.db.new aide.db`**
7. **`/usr/bin/aide -C`**
8. **`cd /etc/cron.daily`**
9. **`mkdir RCS`**
10. **`$EDITOR aide.cron`**

File: /etc/aide/aide.conf

```
#
# AIDE 0.10
#
# Base configuration taken from the Gentoo security handbook.

# $Id: aide.conf,v 1.3 2005/09/02 04:40:17 root Exp root $

verbose=20
#p:     permissions
#i:     inode
#n:     number of links
#u:     user
#g:     group
#s:     size
#b:     block count
#m:     mtime
#a:     atime
#c:     ctime
#S:     check for growing size
#md5:   md5 checksum
#sha1:  sha1 checksum
#rmd160:     rmd160 checksum
#tiger:     tiger checksum
#R:     p+i+n+u+g+s+m+c+md5
#L:     p+i+n+u+g
#E:     Empty group
#>:     Growing logfile p+u+g+i+n+S
#The following are available if you have mhash support enabled.
#haval:         haval checksum
#gost:          gost checksum
#crc32:         crc32 checksum

# define the Top directory.
@@ifndef TOPDIR
@@define TOPDIR /
@@endif

# define where aide specific stuff is storred.
@@ifndef AIDEDIR
@@define AIDEDIR /etc/aide
@@endif

# Not used here.
@@ifhost smbserv
@@define smbactive
@@endif

# The location of the database to be read.
database=file:@@{AIDEDIR}/aide.db

# The location of the database to be written.
database_out=file:@@{AIDEDIR}/aide.db.new

# Don't know what the verbosity level means.
verbose=20

# Where to send the output.
report_url=stdout


# warn about dead symlinks.
warn_dead_symlinks=true

# Rule definition
All=R+a+sha1+rmd160
Norm=s+n+b+md5+sha1+rmd160
```

```
# Do include everything.

@@{TOPDIR} Norm
# Dont barf about the new db file. Perhaps this should be removed
#  once the system is stable ?
!@@{TOPDIR}etc/aide/*.new
# directories not to include.
!@@{TOPDIR}dev
!@@{TOPDIR}proc
!@@{TOPDIR}root
!@@{TOPDIR}tmp
!@@{TOPDIR}var/log
!@@{TOPDIR}var/run
# This one might be interesting once we have a stable box.
!@@{TOPDIR}usr/portage
# the rrd db is continously being updated.
!@@{TOPDIR}var/lib/ntop/rrd
!@@{TOPDIR}var/lib/ntop/addressQueue.db
!@@{TOPDIR}var/lib/ntop/dnsCache.db

# I'm not sure if this is a good idea but I get lot of errors in that dir.
!@@{TOPDIR}sys


# NTP writes to this.
!@@{TOPDIR}etc/adjtime



# I dont know what this does, since we are starting at / this should be included.
=@@{TOPDIR}home Norm
```

File: /etc/cron.daily/aide.cron

```
#!/bin/sh
#
# $Id:$
#
# script concept from /etc/logcheck/logcheck.sh


SYSADMIN=odo@spamcity.world
MAIL=mail
# Shouldn't need to touch these...
HOSTNAME=`hostname`
DATE=`date +%m/%d/%y:%H.%M`

# Set the flag variables
FOUND=0
ATTACK=0



szOutPut=`/usr/bin/aide --update`

# here must be some grep stuff to identify the state of security.
echo "$szOutPut" | $MAIL -s "$HOSTNAME $DATE file system check" $SYSADMIN


# remove the new df if it is identical except for the date generated.
```

[edit] Discussion

If your aide.conf is syntactically incorrect aide will segfault. So use RCS to keep taps on the changes.

[edit] NTOP

1. **cd /etc/conf.d**
2. **mkdir RCS**
3. **ci -l -m "Configuration file for ntop." ntop**
4. **$EDITOR /etc/conf.d/ntop**
   - add: NTOP_OPTS="--http-server 3000 --https-server 0 --interface eth0,eth1"
5. **/usr/bin/ntop --http-server 3000 --https-server 0 --interface eth0,eth1**
   - both NICs have to be configured for ntop to run.

6. Enter password: Please enter the password for the admin user:
7. **/etc/init.d/ntop start**
8. **rc-update add ntop default**

[edit] NTP

See HOWTO NTP

[edit] Snort

- http://www.gentoo.org/doc/en/security/security-handbook.xml?part=1&chap=13
- http://www.snort.org/docs/

[edit] Extras for LAN with Many Gentoo Boxes

[edit] Local RSYNC Mirror

If you have a few gentoo boxes in your LAN, you can be a good netizen and set up a **local RSYNC mirror**. This means that only one box needs to go out to one of the main RSYNC mirrors, and the rest can use this local mirror. The syncing for the internal clients will then happen at LAN speeds!

Check out : HOWTO: Local RSYNC Mirror.

[edit] Share /usr/portage with NFS

When security is not a big issue in your LAN, you should share your /usr/portage directory via NFS. That way you only need to emerge --sync on 1 box and share the portage tree with all your machines. Distfiles are also only downloaded once (since /usr/portage/distfiles is shared, too).

See HOWTO: Using a shared portage via NFS

[edit] DistCC compile farm

With many computers on the LAN doing nothing most of the time, why not set up a distcc compile farm and set portage to use all the computers in its compiles?

See The official gentoo distcc guide

[edit] Extras for LAN with Apple Macintosh Computers

[edit] NetATalk

(This Tutorial deals with the Netatalk 2.0.3-r2 ebuild)

NetATalk provides the Apple File Protocol (AFP) which is used natively by Mac OS and Mac OS X Systems, and avoids much of the hassle and trouble SMB/Samba on Mac OS brings. By Installing NetATalk you'll be able to tightly integrate your Gentoo Server into Mac OS (X) as Network Attached Storage (Fileserver) and you'll be able to tinker with (configuration) files on your Server from your Mac OS X Desktop.

- **emerge**:
  - **emerge -v netatalk**

- **Add AFP-Volumes to */etc/netatalk/AppleVolumes.default*** :
  - **cd /etc/netatalk**
  - **vi AppleVolumes.default**

Add a line for every volume you want to show up via AFP at the end of AppleVolumes.default. Every line consists of:

```
<local path> <volume name> allow:<user1>[,<user2>,...]
```

A typical line would be:

```
/home/foo/ Foo allow:foo
```

This would map the directory */home/foo* on your server to the AFP-Volume *Foo* and allow the user *foo* to connect to it. You can add more users, separated by commas after allow.

Note: The user names after *allow* are user names on your Mac OS systems, not on your gentoo server

So after mapping the home directories of users foo, bar and john to similarily named volumes only accessible by their owner and the dircetory */home/share* to the volume *Shared*

Space acessible by all three, AppleVolumes.default would look like this (The added lines are at the end):

File: /etc/netatalk/AppleVolumes.default

```
# This file looks empty when viewed with "vi".  In fact, there is one
# '~', so users with no AppleVolumes file in their home directory get
# their home directory by default.
#
# volume format:
# :DEFAULT: [all of the default options except volume name]
# path [name] [casefold:x] [options:z,l,j] \
#   [allow:a,@b,c,d] [deny:a,@b,c,d] [dbpath:path] [password:p] \
#   [rwlist:a,@b,c,d] [rolist:a,@b,c,d] [limitsize:value in bytes]\
#   [preexec:cmd] [root_preexec:cmd] [postexec:cmd]  [root_postexec:cmd]
#
#
# name:      volume name. it can't include the ':' character and is limited
#            to 27 characters in length.
#
# variable substitutions:
# you can use variables for both <path> and <name> now. here are the
# rules:
#     1) if you specify an unknown variable, it will not get converted.
#     2) if you specify a known variable, but that variable doesn't have
#        a value, it will get ignored.
#
# the variables:
# $b   -> basename of path
# $c   -> client's ip or appletalk address
# $d   -> volume pathname on server
# $f   -> full name (whatever's in the gecos field)
# $g   -> group
# $h   -> hostname
# $i   -> client ip without tcp port or appletalk network
# $s   -> server name (can be the hostname)
# $u   -> username (if guest, it's whatever user guest is running as)
# $v   -> volume name (either ADEID_NAME or basename of path)
# $z   -> zone (may not exist)
# $$   -> $
#
# casefold options [syntax: casefold:option]:
# tolower    -> lowercases names in both directions
# toupper    -> uppercases names in both directions
# xlatelower -> client sees lowercase, server sees uppercase
# xlateupper -> client sees uppercase, server sees lowercase
#
# allow/deny/rwlist/rolist format [syntax: allow:user1,@group]:
# user1,@group,user2  -> allows/denies access from listed users/groups
#                        rwlist/rolist control whether or not the
#                        volume is ro for those users.
# preexec             -> command to be run when the volume is mounted,
#                        ignore for user defined volumes
# root_preexec        -> command to be run as root when the volume is mounted,
#                        ignore for user defined volumes
# postexec             -> command to be run when the volume is closed,
#                        ignore for user defined volumes
# root_postexec        -> command to be run as root when the volume is closed,
#                        ignore for user defined volumes
#
# codepage options [syntax: options:charsetname]
# volcharset           -> specifies the charset to be used as the volume codepage
#                         e.g. "UTF8", "UTF8-MAC", "ISO-8859-15"
# maccharset           -> specifies the charset to be used as the mac client codepage
#                         e.g. "MAC_ROMAN", "MAC_CYRILLIC"
#
# miscellaneous options [syntax: options:option1,option2]:
# prodos              -> make compatible with appleII clients.
# crlf                -> enable crlf translation for TEXT files.
# noadouble           -> don't create .AppleDouble unless a resource
#                        fork needs to be created.
# ro                  -> mount the volume as read-only.
# mswindows           -> enforce filename restrictions imposed by MS
#                        Windows. this will also invoke a default
#                        codepage (iso8859-1) if one isn't already
#                        specified.
# nohex               -> don't do :hex translations for anything
#                        except dot files. specify usedots as well if
#                        you want that turned off. note: this option
#                        makes the / character illegal.
# usedots            -> don't do :hex translation for dot files. note: when
#                       this option gets set, certain file names
#                       become illegal. these are .Parent and
#                       anything that starts with .Apple. also, dot
#                       files created on the unix side are marked
#                       invisible.
# limitsize          -> limit disk size reporting to 2GB. this is
#                       here for older macintoshes using newer
#                       appleshare clients. yucko.
# nofileid           -> don't advertise createfileid, resolveid, deleteid
#                       calls
# root_preexec_close -> a non-zero return code from root_preexec close the
#                       volume being mounted.
# preexec_close      -> a non-zero return code from preexec close the
#                       volume being mounted.
# nostat             -> don't stat volume path when enumerating volumes list
# upriv              -> use unix privilege.
#
#
# dbpath:path        -> store the database stuff in the following path.
# password:password  -> set a volume password (8 characters max)
# cnidscheme:scheme  -> set the cnid scheme for the volume, default is [cdb]
#                       available schemes: [cdb dbd last]
#
# The "~" below indicates that Home directories are visible by default.
# If you do not wish to have people accessing their Home directories,
# please put a pound sign in front of the tilde or delete it.
~
/home/foo Foo allow:foo
/home/bar Bar allow:bar
/home/john John allow:john
/home/shared "Shared Space" allow:foo,bar,john
```

[edit] Zeroconf (Bonjour/Rendezvous) Server

[edit] Avahi

1. USE="-gtk -qt3 -qt4" emerge --pretend avahi
   - You should only see a few packages.
   - The -gtk/-qt3/4 is to avoid getting the whole X deal onto the server
   - Remove the '--pretend' when you are ready.
   - To avoid having avahi rebuilt later with gtk/qt3/4 USE flag, do "echo net-dns/avahi -gtk -qt3 -qt4 >> etc/portage/package.use"

[edit] Related Pages

- http proxy
- DNS server
- NAT
- HOWTO Setup Samba

[edit] External References

- Gentoo Security Handbook
- IP Masquerading Howto
- The official gentoo distcc guide

Retrieved from "http://gentoo-wiki.com/HOWTO_setup_a_home-server"

Category: Network

Browse categories > Browse categories > Applications > Network

Views

- Article
- Discussion and Bugs
- Edit This Page
- History

Personal tools

- Log in / create account

Navigation

- Main Page
- Wiki News
- Recent changes
- Random page
- Help

Search

Go Search

Indexes

- HOWTO Index
- Tips & Tricks
- MAN Pages
- Hardware
- Security
- FAQ
- Screenshots
- Misc
- Other Indexes

- Wiki Pages
- Wanted Articles
- Categories

Toolbox

- What links here
- Related changes
- Upload file
- Special pages
- Printable version
- Permanent link

Powered By MediaWiki

- This page was last modified 15:26, 15 January 2008.
- This page has been accessed 196,351 times.

- Privacy policy
- About Gentoo Linux Wiki
- Disclaimers