How To

Gentoo chroot on Medion P89626

(v0.5)

(ATTENTION: THIS IS STILL WORK IN PROGRESS!)

Version History

v0.5: Some clean-up and more Q&A
Command for device node copy was missing
Disclaimer

Disclaimer

By implementing the steps in this HowTo you are using your NAS in a way that might actually damage it in some way. This might result in anything, from simply a higher power consumption of the device, might lead to loss of data, or – in the worst case – a burnt down house or the universe being ruled over by a cute, but giant killer kitten, devouring our very souls!

YOU WILL NOT HOLD ME RESPONSIBLE TO ANY DAMAGE CAUSED TO THE DEVICE, YOU, YOUR DATA, YOUR POSSESIONS OR ANY PERSONS IN YOUR HOUSHOLD!

Think before typing! If anything doesn't make sense to you, don't do it! Use your common sense! This HowTo is aimed at an experienced Linux audience; if you don't know what a device node is or how a pipe works and what the frak a bash is, stop now!

Motivation

As always, there is no more space left on the disc. So we went to your local Aldi discounter and tried to get that neat little NAS system they advertised for. But, alas, they were gone in a matter of minutes – it was a good deal, and apparently others thought so, too.

Then, the next week we bought some food at the same shop only to find they had put up said NAS in the display case in the store. They must have taken one of them from the original shipment but didn't put it on display until later, when the rush was gone. So we bought that one and considered ourselves lucky! ;-)

Until now we had an old Via C3 PC as our “Server”, that replaced an even older AMD K6 system, but it never really worked and it is getting older and older as I write this, awaiting fatal failure. So, why not host all the services we need on the NAS? It's always on, it's considerably faster, it's perfect!

What we need is basically this: DNS, DHCP, SSH, and maybe a web server. The first two are marked as must-have, the third would be nice, and the last one, well, if everything else is working and the system is not overwhelmed by the other stuff I might give it a try.[1]

So, how do I get these?

It didn't take me long to realize that for the love of Gods I wouldn't be able to find binary ARM packages of these programmes that would work on the preinstalled system without any modification. As long as I have a warranty on the box I won't fiddle with the firmware, so that is out, too.

I've been using Gentoo as my primary desktop and server distribution for some time now and I

1 For now, I have not tried to install apache. It might be a bit too much strain on the 128M of RAM. lighthttpd seems to be the better choice.

believe to have some knowledge about the internals of that system. And there is an ARM Stage3 package waiting for me on their mirrors.

So, I will use that!

Here Is The How To

First of all, log on to the NAS by opening its web interface. Freshly from the factory the *admin* password is *1234*. Now set your admin password to something you remember and that is a little more secure! Set up the system so that it can connect to the internet. See if there is a new firmware. Do all the things you ever wanted with you new NAS until you get bored! ;-)

Now go on and create a share called *gentoo*. Then go to *http://x.x.x.x/r32694,/adv,/cgi-bin/remote_help-cgi?type=backdoor*[2] (where the x.x.x.x stands for the IP of you NAS). From now on until the next reboot you will be able to telnet to your NAS.

Use telnet to get to your NAS's shell and log in as *root* with the password set above. Change directory:

```
cd /i-data/6764ac2f/gentoo
```

Get the Gentoo Stage 3 ARM tar.bz file from one of the mirrors[3] like this one: http://de-mirror.org/gentoo/releases/arm/autobuilds/current-stage3-armv5tel/. While you are there, get the portage tree as well: http://de-mirror.org/gentoo/snapshots/. (You can use wget on the NAS, but you'll have to use a browser on your desktop to find out what the files are actually called since they include a time stamp.)

Issue the following commands in your telnet shell:

```
tar -xjvf stage3*
tar -xjvf port*
mv portage usr/
cp /etc/resolv.conf etc/
cp -a /dev/* dev/
mkdir dev/pts
echo "#!/bin/sh" > setupchroot.sh
echo "mount -t proc none /i-data/6764ac2f/gentoo/proc" >>
    setupchroot.sh
echo "mount -t devpts devpts /i-data/6764ac2f/gentoo/dev/pts"
    >> setupchroot.sh
chmod ugo+x setupchroot.sh
./setupchroot.sh
chroot gentoo /bin/bash
TERM=linux
export TERM
env-update
mv /etc/fstab /etc/fstab.orig
touch /etc/fstab[4]
```

2 Since Firmware Version 1.01(UZD.0)_20120203 this has changed to: *http://x.x.x.x/**r34814**,/adv,/cgi-bin/remote_help-cgi?type=backdoor* This might change in the future with new firmware releases! It might be that the number after the r is the software revision?

3 http://www.gentoo.org/main/en/mirrors2.xml

4 Deleting the original fstab, because as soon as we start services, init will try to mount file systems. But they are already mounted, we are in a chroot. But without a fstab file someone might get jumpy, so I touch a new one.

```
touch /lib/rc/init.d/softlevel[5]
passwd
```

Enter a password for the root user of your chroot. I suggest you use the same as for the admin user of the NAS.

Before you try to emerge something you might want to edit the /etc/make.conf file to include MAKEOPTS="-j2" to make use of the two processor cores of your NAS[6]. However, I chose to not do so, because I wanted the system to stay responsive to normal operations during compile time. And I wasn't in a hurry. ;-) Also, compiling two or more threads at a time puts a heavy load on the RAM, too, of which our NAS only has 128M, already maxed-out by all the NAS services running! (There is a 512M swap somewhere on the disk, so you won't run out of memory, but it might get slow.) So, in effect, turning on -j2 might actually lead to a longer compile time.

This is what I'm setting in my /etc/make.conf:

```
CFLAGS="-O2 -pipe -march=armv5te"
CXXFLAGS="${CFLAGS}"
CHOST="armv5tel-softfloat-linux-gnueabi"
MAKEOPTS="-j1"
FEATURES="parallel-fetch"
```

You might want to set some use-flags. I didn't. Now *joe /etc/bin/bash/profile*. (As long as we have no joe you can use nano.) Add to the end of the file:

```
TERM=linux
export TERM
```

Now edit */root/.bashrc*:

```
PS1="\[\033[01;30m\]\d \t \[\033[01;36m\]\# \[\033[01;31m\]\h\
[\033[00m\]:\n\[\033[01;34m\]\w \[\033[01;35m\]#\[\033[00m\] "
alias l="ls -la --color"
alias emerge="nice emerge"
alias emsync="emerge --sync"
alias emworld="emerge -avu --keep-going world"
```

All these things a not really necessary, but I like some colour in my life, and it's always a good idea to run emerge with lower priority so it won't block other processes.

At this point we are (almost) ready to go. Enter on the command line:

```
source /etc/bin/bash/profile
source /root/.bashrc
emerge --sync
emerge -av screen joe ssh bind dhcp
```

So far, so good. Compiling will take a while, so go and get yourself some coffee or something.

---

5 This one is needed to be able to start services on a system not using openrc. The manual says this might be risky. I'm willing to take that risk. Yes, I'm a risk taker. What's the worst that could happen? It won't work? Fine by me! ;-)

6 About whether or not this system actually supports two CPUs as advertised, have a look in the Q&A section below.

After everything has finished successfully, you can start the first service. I started with ssh because it was a lower priority and if that didn't work I wouldn't be that disappointed or frustrated.

```
/etc/init.d/sshd start
```

Test it by ssh-ing from a different machine to your NAS. Use *root* as user and the password you set above.

If everything is working, you can begin to add entries to */etc/bind/...* and */etc/dhcp/...* to configure these services, too. Since I have working configurations from my old server I won't go into details about that. I just changed every occurrence of the old server's IP to the new NAS's IP and it worked, no problem. If you have trouble, there are enough HowTos out there, a lot of man pages, and the example configuration files are a good place to start.

So, after configuring, just...

```
/etc/init.d/named start
/etc/init.d/dhcpd start
```

...and see what happens. Don't forget to change the */etc/resolv.conf* to use 127.0.0.1 as a name server from now on.

And now, the conclusion

So far we have done nothing to start these services automatically on boot. Well, that's not entirely true, some might have wondered why I echoed some mount commands to a sh-file first right at the start. That's because we now will build us some scripts to start the chroot and the services we need. In the / directory of your chroot, do a *joe startservices.sh*.

```
#!/bin/bash

services=( udev sshd named dhcpd )

for service in ${services[@]}
do
  echo "Restarting" $service
 /etc/init.d/$service stop
 /etc/init.d/$service zap
 /etc/init.d/$service start
done
```

From now on, whenever you want to add a service to the chroot, do not forget to add them to this file.

Now *chmod u+x startservices.sh*, then *joe setupchroot.sh* and add:

```
chroot . /startservices.sh
```

Now all we need is a mechanism to actually start *setupchroot.sh* (which in turn starts *setupservices.sh*) automatically at boot up of the NAS. Sadly, as far as I can tell from */etc/init.d/rcS*, this option is only available for USB flash drives connected to the NAS. However, it looks like this method has been removed in recent firmware updates.

A different way would be to add a script in */usr/local/zy-pkgs/etc/init.d*. Lets call it *startchroot.sh*:

```
#!/bin/sh

# Open Telnet Backdoor:
/usr/local/btn/open_back_door.sh
# Copy Profile
cp /i-data/6764ac2f/gentoo/root/.profile /root/.profile [7]
# Copy Joe
cp /i-data/6764ac2f/gentoo/usr/bin/joe /bin [8]
# Start Services
/bin/sh /i-data/6764ac2f/gentoo/setupchroot.sh >
    /root/start.log &
# Make sure original Apache is started
/etc/init.d/httpd.sh start

exit 0
```

This seems to work. I still don't know why I have to manually start the build-in Apache now. As soon as I figure it out I'll tell you.

From now on the chroot and all its configured daemons will start up at boot time.

lighttpd & PHP

It took me two days of compiling, but it works. And it's rather fast, I am a bit astonished. Here's what I did:

I opened */etc/make.conf* and added the following use-flages:

```
USE="php cgi"
```

Then I emerged lighttp, which now automatically pulls in PHP, among some other stuff.

```
emerge -av www-servers/lighttpd
```

Then I waited for over one day. Building PHP is not the easiest thing to do, or so it would seem. There was a lot of swapping going on and the NAS was a bit unresponsive at times, so be patient! Best to start this merge, when you know you won't be needing the NAS. (That's the reason I installed *screen* in the first place, so I can detach from the shell and still keep it running in the background.)

After emerge was finished, I started editing */etc/lighttp/lighttpd.conf*. I uncommented the following lines in the modules-section: *mod_access, mod_userdir, mod_accesslog*.[9] I changed the *server port* variable to 81 (because on 80 there already is the NAS's apache running, serving you the configurations interface; `server.port = 81`) In the cgi includes section I uncommented include "*mod_fastcgi.conf*".

Finally I started the service:

```
/etc/init.d/lighttpd
```

7 Make sure you have a .profile file in your root's home of your chroot. Else leave this line out. I put it in for some basic aliases like *l=”ls -la --color”* that are gone after every reboot. And I rebooted a lot lately!

8 As I might have said before, I don't like vi. No guarantee joe – or any other programme for that matter – will work outside of the chroot, though. Static linking might help.

9 Actually I can't remember which of these were activated beforehand, so I just list all that are uncommented. ;-)

...and tested it with a browser on my desktop. Keep in mind that the server answers on port 81, not 80 as usual. As you can change the default setting of the NAS to serve the configuration interface on a different port than 80 you might be able to change that, but I'm lazy and as it runs perfectly this way and our router can forward external accesses to it's port 80 to the NAS's port 81 I see no problem.

Remember to add the new service to */startservices.sh*! Just put it with all the others in that array at the beginning.

Questions and Answers

/dev Related Questions

Q: You copy /dev/*? Why don't you mount -o bind?
A: It didn't work. I don't know why. Maybe it's because there is no udev running on the NAS, or it's a quirk of busybox's mount. In the end, it works with copied device nodes. They don't change, anyway, and Gentoo starts it's own udevd as soon as you start any service, so if they would change (if you plug in something to the USB ports) udevd hopefully takes care of it.

Q: I can't start my services. Instead I get messages like “/dev/zero does not exist” (or urandom) or “PRNG is not seeded”. Help?
A: Take a look at your chroot's /dev directory. Did you copy the device nodes from the NAS's /dev? If not, do so. (Don't forget the -a switch, else you end up with a very large regular file called urandom and that's not what you want!) (It seems I forgot to mention this in an earlier version of this HowTo.)

Questions Related to Services

Q: In startservices.sh you keep zapping services after you already stopped them?
A: When the NAS crashes or is rebooted, there will be pid files left over from these daemons and they won't restart until they are removed. By first trying to stop them and then zapping them, we can be sure they start.

Q: I can't start any services since they all depend on net. What can I do?
A: I recently ran into this problem after updating the *sysvinit* package. It seems there is some more checking going on now. As a quick and dirty fix you can edit the init-scripts in */etc/init.d/* to not depend on net being started – just comment out all the lines *need net* in the *depend()* function. Until now I found no other, more elegent way to fix this. The network device does not need to be started by out chroot anyway, since it is already running, otherwise you wouldn't be able to telnet into the NAS.

General Questions

Q: Why not cross compile? Wouldn't that be faster?
A: Well, I thought about it, and it actually might be. But I'd have to install all the cross compiling tools and in the end it would take me more time to figure out the cross compiler than it took me to set up a chroot and compile inside of that. If you like you can certainly try your luck with cross compiling, but as far as I've heard, it's a bitch!

Q: Why joe?
A: It uses the same keyboard short cuts I've been accustomed to since my time in school, back in the '90s. We still used some ancient version of WordStar, if anyone can remember that, on some senile 8086s someone must have forgotten still existed. (The lucky (?) ones in our class had to use some even older Apple IIe.) So, you can use whatever *you* like. The Stage3 gets shipped with *nano*, that's a fine editor and you can use that out of the box. You might even want to use *vi* or *emacs*, but you'd have to emerge them yourself.

Q: My NAS forgot it's IP address after reboot or disconnection from the network. It seems like it defaulted to 0.0.0.0 when there is no DHCP server running. How do I get back my access?

A: Only way I found was to boot Windows, install the software, wait until the NAS gets found, and re-enter the correct IP in the system settings. It might work if you set up temporary dhcpd somewhere in the net (like on a router or something) and reboot the NAS, but I didn't test that.

Q: Why use *armv5tel-softfloat-linux-gnueabi*? Isn't this an ARMv6?

A: Yes, it is. I tested the ARMv6 Stage 3 package, but some programmes produced “Segfault” and “Illegal Instruction” errors, so I guess it is compiled for some processor that actually understands some more instructions. With fine tuning, patience, and recompilation you might get it to work, but I am lazy (didn't I already mention that?) and used the ARMv5 package that worked out of the box. Anyways, I never really understood the naming convention of ARM processors, so I just use what is working! ;-)

Q: All those additional work seems to get my NAS to swap a lot, the disk won't spin down any more. Is there something I can do to prevent this?

A: Swapping during compilation can't be helped. There are only 128M of physical memory build in and compiling things may take a lot more. As for normal operations, with the four services described in this document (ssh, bind, dhcpd, and lighttpd) it takes 40M of swap space, while the NAS itself tells me there are about 45M of free RAM available. (Take a look at *cat /proc/meminfo* where you can see about 15M of free RAM and 30M of cache.) So – theoretically – it would be possible to deactivate swapping altogether. BUT: There WILL be side effects! Sooner or later something will request more memory than available and then things will start to frak up! So, don't! If you do, don't blame me!
The thing that takes by far the most RAM is lighttpd in combination with php. So, you might want to consider not installing those two.
On the other hand you might want to kill the telnetd after you are done and have a working sshd. It's not that large, though, so I keep it running in case sshd unexpectedly dies.

Q: The packaging says “Dual Core”, but *cat /proc/cpu* shows only one CPU! What the...?

A: So it would seem. Since FW Version 101UZD0D0_20120203 however, there seems to be some progress: There are two CPU metres now in the administration interface and a *dmesg | grep CPU* confirms that two CPUs have been activated by the kernel. There is some discussion about that issue on the net.

Q: I have a Philips TV, and since the last firmware update it won't play any AVI files from my NAS. I know, this has nothing to do with this HowTo, but...

A: They changed from Twonky 5 to 6 and there is a known bug. I guess we'll see another firmware release as soon as Twonky is fixed[10]. Whenever that'll be, it seems this bug is well aged. In the meantime, open your browser, input http://x.x.x.x:9001/configpage/config-content.html, scroll down to where the media playback devices are listed, find your “Philips TV Renderer” and change it to “Philips Streamium AV”. Then scroll back up an hit the “save changes” button. There might be side effects but your AVIs will hopefully play again.

Q: Does this HowTo work with the newer Medion NAS that only has a 1TB hard disk?

A: I haven't got the foggiest! I think it is the same hardware and I think it runs the same software. But I do not know for sure! If you try it and it works you are welcome to tell me so I can update this HowTo.

Q: Haven't you missed something?

A: I probably have. If you think it is important, please comment on my blog so I can fix it!

10 I've been waiting for over a year now, so don't hold your breath...

Acknowledgements

You can find many useful informations about this NAS on the German web page http://www.mikrocontroller.net/articles/P89626. It has been a good source for me and may be for others.

25 Oct 2013, 17:49:48 CET