LESSLINUX Handbook

Mattias Schlenker
August-Bebel-Str. 74

04275 Leipzig
GERMANY

ms@mattiasschlenker.de

Contents

Abstract

LessLinux is a free Linux system designed to be light and easily modifiable. It is based on Linux from Scratch and was started by Mattias Schlenker in 2009. Since then it has been used as a base for dozens of security and rescue systems published by computer magazines all over the world. It's simple architecture makes it easy to build LessLinux based systems for use as thinclient, software deployment or the demonstration of software. This book covers the possibilities of LessLinux and tells you how small changes can make LessLinux the lever you need to move your world.

Chapter 1

About LessLinux and this handbook

Chapter 2

LessLinux for users

Chapter 3

LessLinux for admins

This chapter covers using LessLinux for typical administrative tasks in heterogenous networks. Topics will be remote accessing a macine running LessLinux, using LessLinux as a simple yet flexible thinclient and booting LessLinux via the network. It will further cover modifying lesslinux with payloads: An easy way to add scripts (like for backing up and restoring), thus using LessLinux as a platform for your own netbootable toolbox - without having to touch LessLinux containers.

3.1 Remote access

Since one of the main target groups for LessLinux are network admins, easy remote access to running LessLinux instances was one of the goals during development. Currently access via SSH and VNC is supported, both together result in encrypted VNC. Reverse VNC allows for easy remote support on machines that are behind NAT routers and firewalls. RDP access is work in progress.

3.1.1 SSH

LessLinux is usually shipped with the OpenSSH secure shell server. However this is disabled in most builds by adding it to the `skipservices=|servic1|service2|` boot command line. To start the OpenSSH server, remove `ssh` from `skipservices`.

Warning:

Placing the hash for the root password in the boot command line, transferring it over a hostile network or netbooting with SSH machine keys (or root's private SSH keys) make eavesdropping easy. Use those options just in networks that are considered safe. Also consider locking down the local consoles when LessLinux is netbooted in order to do adminstrative tasks via SSH to prevent local users from manipulating the machine.

Login with password

To login with password use the possibility to add files to the initramfs by adding `/etc/lesslinux/root.hash` that contains the MD5 hash of the root password. You can generate this hash with the command

```
openssh passwd -1
```

You might also specify the base64 encoded MD5 hash of the root password via boot command line. Since the = character which is used for indicating padding in base64 is not valid, you will have to pad the hash with one or more spaces (usually one) until the base64 string does not end on =. Those spaces are removed after decoding:

```
hash=` openssl passwd -1 `
echo "$hash " | base64
```

The result must look like:

```
JDEkTktlL3NSRHIkd1owLzJ5UjJDTVdlMnZPY0NaSGRuMSAK
```

The resulting parameter for the boot command line:

```
roothash=JDEkTktlL3NSRHIkd1owLzJ5UjJDTVdlMnZPY0NaSGRuMSAK
```

Public key login

Use the possibility to add files to the initramfs by preparing a CPIO archive that contains all the host keys placed in `/etc/openssh` as well as the needed `/root/.ssh/authorized_keys`. Make sure permissions fit and add this CPIO archive as the last file to the bootloader or concatenate it to the initramfs if you use a bootloader that allows just one initramfs. If permissions are correct and the OpenSSH daemon is started you can connect with public key login afterwards.

3.1.2 VNC

Virtual Network Computing or *VNC* is a simple, easy to implement protocol used to gain access to remote computers. It is the basis for Apple's remote administration tool. Clients are available for nearly every operating system as well as for smartphones and as Java applet. In LessLinux VNC can be used to mirror a local X server - which can be either a graphical console or an invisible virtual frame buffer.

Warning:

The VNC protocol does not use encryption! This means all keyboard input and all screen output can easily be eavesdropped. Combine VNC with an SSH tunnel if you need some minimum security.

VNC for incoming connections

The easiest way to access a running LessLinux system is mirroring the local Xserver. Specify

```
x11vnc=|remote|
```

to enable VNC access without password. You might specify a clear text password as second argument:

```
x11vnc=|remote|password|
```

VNC uses port 5900 as default. The current IP address is usually shown on the first console (`Ctrl+Alt+F1`).

Reverse VNC

If a host that booted LessLinux is running behind a firewall or NAT router, incoming connections might not be possible. With reverse VNC the LessLinux host tries to establish an outgoing connection to the IP address or hostname specified as target (on port 5500):

```
x11vnc=|reverse|target|
```

On a Linux target machine you usually run

```
vncviewer -listen
```

or use the graphical application Remmina to work with reverse VNC connections.

Headless access

Specify `x11vnc=...` as mentioned above to enable VNC access. To run headless access instead of the local X server you have to disable it and replace it by Xvfb - the virtual frame buffer X server, which is done with the additional parameter

```
xvfb=1280x800x32
```

The `1280x800x32` tells Xvfb to start with an 1280x800 resolution and 32 bit color depth. All values that make sense (they should resemble real world resolutions) are allowed.

Tunnel VNC over SSH

The security implecations of VNC are described earlier in this chapter. If you want to use VNC in hostile environments, enable SSH access. Then bind the listening port for VNC to localhost only by specifying:

```
x11vnc=|local|
```

On the machine that is used to access the LessLinux with running VNC you then have to run SSH with port forwarding, on unix like operating systems:

```
ssh -L 5900:localhost:5900 lesslinux-host
```

On the machine connecting via SSH, VNC's port 5900 is now available at `localhost`, point your VNC viewer to `localhost` or `127.0.0.1`. Depending on the viewer it may be necessary to specify the priority of encodings (if you are using `vncviewer` from TightVNC, see the man page regarding `-encodings`.

3.1.3 RDP

Remote access via Microsofts Remote Desktop Protocol based on `xrdp` is currently worked on and will be available in a few weeks. The main goal of implementing accessibility via RDP is to enable admins in windows focused networks to access machines running LessLinux without having to install additional tools.

3.2 Netbooting LessLinux

Netbooting LessLinux is a convenient way of providing a rescue system to medium scale networks or converting existing machines to thin clients. To maximize the possiblities you might want to combine netboot with remote access or LessLinux as thin client.

This chapter will not describe how to completely setup a netboot environment. Please view the respective documentation of your preferred DHCP server and TFTP daemon to see how to get kernel and initramfs delivered to your PXE capable clients. For the system's boot command line boot LessLinux from CD and then run

```
cat /proc/cmdline
```

to retrieve a command line that can be used as a basis for the the following examples. Make sure that `earlynet` is not included in the list of services to skip, since this brings up network interfaces before the system image is found. To speed up boot you might add `dhcpcd` and `wicd` to the list of services to skip.

3.2.1 CIFS or NFS boot

With the command line

```
nfs=12.34.56.78:/path/to/share
```

respectively

```
cifs=//12.34.56.78/share
```

the specified share will be mounted during boot. The share will then be recursively searched for an ISO image that matches the LessLinux build running. This ISO will then be loopback mounted. Please note that when using `cifs` the share must be mountable as User `guest` with an empty password.

Depending on speed and reliability of your network and RAM of your clients you might want to use `toram=1` to force loading the system completely to memory upon boot. This might cost about one minute during startup, but will accelerate the start of programs considerably in congested networks.

This boot method is defined in `/etc/rc.d/0107-nfssys.sh`.

3.2.2 HTTP, FTP or TFTP boot

Use the command line parameter

```
wgetiso=http://12.34.56.78/path/to/image.iso
```

to use BusyBox' `wget` to download LessLinux' ISO image. The complete ISO image will be saved in memory, so this option requires sufficient RAM. Using `tftp` as alternative protocol allows netbooting without having to setup another daemon on your server. However `tftp` is far less reliable, so this should be just used as an option if no other protocol is available.

This boot method is defined in `/etc/rc.d/0108-wgetsys.sh`.

Warning:

Always make sure the parameter `toram=0` prevents that LessLinux will be copied to RAM again. Activated `toram` would double the amount of RAM used for the system!

3.3 LessLinux as thinclient

Since autumn 2012 LessLinux includes programs and scripts to run as a thinclient with minimal overhead. This targets primarily environments where LessLinux is booted via the network and converted to a thinclient OS with just a few boot parameters. For now the functionality to comfortably access RDP logins is implemented, audio redirection works good if optional parameters are specified (chooser only). Access to printers, local USB drives and other local USB devices is work in progress.

3.3.1 Booting to Remmina

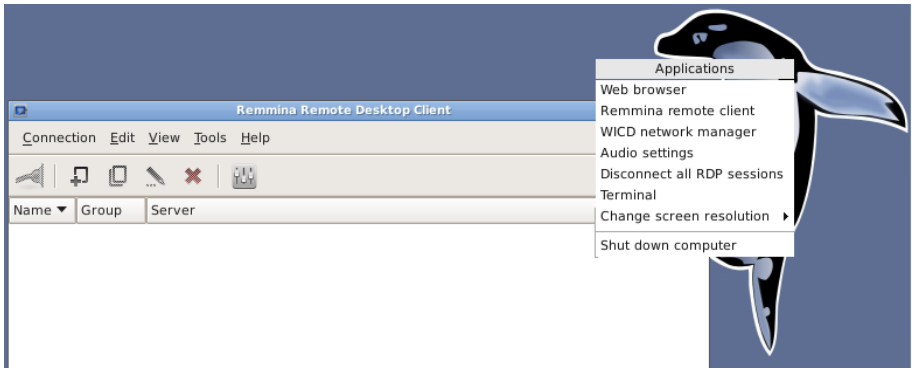

Figure 3.1: Just specifying `xinitrc=/etc/lesslinux/xinitrc_remote` will open Remmina on an otherwise empty desktop. Right click on the desktop background to get a menu with some useful programs (also available when booting to chooser or RDP login mask).

If you just pass the parameters

```
xinitrc=/etc/lesslinux/xinitrc_remote
```

you will end up in an empty desktop with a Remmina window open. Remmina allows for access to RDP, VNC, reverse VNC and SSH servers. Since all settings are lost upon reboot, this option is primarily interestinng for admins who want to be able to remotely login to servers from any machine in the network that boots via PXE.

With the `xinitrc_remote` the default window manager is OpenBox. Right click on the desktop to get a simple menu from where you can start Firefox, a terminal window or the mixer (for remote audio).

3.3.2 Booting to an RDP login mask

The easiest way is booting to a simple login mask: To enable thinclient mode specify

```
xinitrc=/etc/lesslinux/xinitrc_remote
```

and additionally to enter the RDP login mask

```
rdesktop=|12.34.56.78|username|domain|2|
```

Those parameters are IP address of the RDP server, user name and domain name (optional). The user name might be encoded as in HTTP URIs: `Kemal%20Atat%C3%BCrk` is unescaped to `Kemal Atatürk` - however due to character set limitiations you should try to avoid umlauts

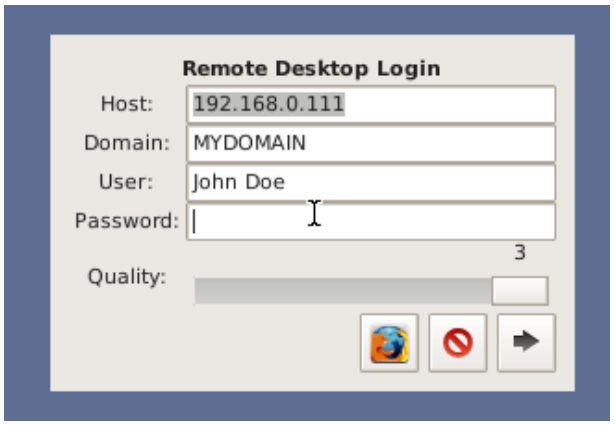

Figure 3.2: The RDP login mask is configured via boot command line and does not need a separate configuration file.

in user names. The optional integer specifies the quality of the connection: 0 is low quality, but fast, 3 is highest quality, but slower. Use 0 for very slow connections (modem, 3G), 1 in broadband connections (or 3.5G, 4G), 2 (default) in typically congested ethernets and 3 for very fast ethernets with low congestion. Those numerical values map to the following parameters upon execution of `xfreerdp`

- **0** `-x m -a 16` - 16 bit color depth, modem experience
- **1** `-x b -a 16` - 16 bit color depth, broadband experience
- **2** `-x l -a 32` - 32 bit color depth, LAN experience
- **3** `-x 180 -a 32` - 32 bit color depth, LAN experience plus font smoothing and Desktop Composition

RDP Sessions start in full screen. You might toggle full screen with `Ctrl+Alt+Enter`, e.g. to access the local OpenBox menu or start a local Firefox instance.

3.3.3 Booting to a chooser

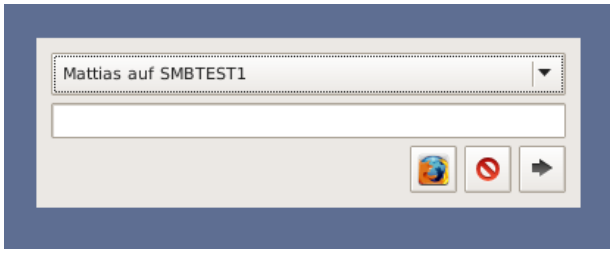

Figure 3.3: The chooser is configured by an XML file that is loaded from the boot medium or via the network.

More comfort and more flexibility is provided by the optional chooser. To use it, you have to prepare an XML file with one block per entry. This XML file has to be accessible via FTP, HTTP or TFTP. Currently only RDP connections are available via chooser, VNC, Nomachine NX, XDMCP and Citrix ICA will follow eventually. Add the parameters

```
xinitrc=/etc/lesslinux/xinitrc_remote
```

and

```
chooser=tftp://server/path/chooser.xml
```

to boot into chooser mode. If you are booting from a USB drive, you can put the `chooser.xml` on the boot partition (usually the second partition on the thumb drive) and specify

```
chooser=file:///lesslinux/boot/chooser.xml
```

Alternatively when remastering a CD with `chooser.xml` in the root directory of the ISO 9660 file system, specify

```
chooser=file:///lesslinux/cdrom/chooser.xml
```

In both cases make sure to specify `toram=0` to prevent LessLinux from unmounting `/lesslinux/boot` or `/lesslinux/cdrom` before the chooser.xml there can be used.

A sample XML file `chooser.xml` could look like:

```
<chooser>
  <connect nicename="John Doe" host="192.168.0.23"
           proto="rdp" user="john" domain="MYDOMAIN">
    <default inet="192.168.0.17" />
    <default ether="00:24:8c:6b:5d:e4" />
    <param>-x l -a 32</param>
  </connect>
  <connect nicename="Jane Doe" host="192.168.0.23"
           proto="rdp" user="jane" domain="MYDOMAIN">
    <default inet="192.168.0.18" />
    <param>-x 180 -a 32</param>
    <param>--plugin rdpsnd --data tsmf:audio:alsa:plughw:0,0 --</param>
  </connect>
  <connect nicename="Alice X" host="192.168.0.24"
           proto="rdp" user="alice" domain="OTHERDOMAIN">
    <default inet="192.168.0.19" />
    <param>-x l -a 32</param>
  </connect>
  <connect nicename="Admininstrator on Server 1" host="192.168.0.23"
           proto="rdp" user="Administrator" />
  <connect nicename="Admininstrator on Server 2"
           host="192.168.0.24" proto="rdp" user="Administrator" />
</chooser>
```

The first to entries specifiy connections to the RDP server `192.168.0.23`, with the domain `MYDOMAIN`. The parameters specify LAN experience for John Doe and remote audio as well as font smoothing and compositing for Jane Doe. The `default` tag specifies on which client machines an entry is selected as default. On the client machines with local MAC adress `00:24:8c:6b:5d:e4` or local IP `192.168.0.17` John Doe will be shown as default, on `192.168.0.18` Jane Doe will be shown as default. Feel free to play with parameters for `xfreerdp` to get the best performance according to your network's latency, bandwidth and congestion.

3.3.4 Using XDMCP

The X Display Manager Control Protocol that is used in several unix only environments is work in progress and not yet reflected in any cheatcodes. It will become available soon with the sole limitation that the keymap of the initially started X server will be US english. Depending on your Display Manager some special character in passwords won't be at the used keys. As soon as you are logged in most desktop environments take care for setting the user's keyboard layout.

3.3.5 Local printers

Figure 3.4: Local USB printers can easily be shared: This screenshot shows adding USB printer `/dev/usb/lp0` on LessLinux host 10.76.23.231 to a CUPS server that centralizes spooling.

Minimal support for local printers is now provided with BusyBox' LPD. This is considered experimental, it might work or not, and parameters will change in the future without further notice. Specify

```
printers=|lp0|lp1|
```

to enable printing to the USB printers available at `/dev/usb/lp0` and `/dev/usb/lp1`. This will start a line printer daemon with queues having the same names as the printers `lp0` and `lp1`. Since no filtering is done, the clients printing have to send raw output in a format the printer understands.

Chapter 4

LessLinux for builders and contributors

The build system used by LessLinux is closely based on Linux from scratch. Building is done in three stages. The first stage is used to cross compile a toolchain that resides under `/tools`. In the second stage we use this toolchain to build all LessLinux packages in a chroot environment. Stage three is the assembly of the final boot image by taking the required files from the second stage. Although the fiirst stage involves cross compiling, the rest of the build process has to be done on the target architecture – which currently only is i686. This means you should be able to build on any recent 32 Bit linux. Our preferred build environment are the „FULL" ISO images of LessLinux Search and Rescue that are self containing, you should also get good results using the latest Ubuntu LTS version. The next sections in this chapter tell you how to build using the LessLinux „FULL" live ISO.

The examples in this chapter use the „FULL" ISO image with it's defaults. We assume that you can leave the defaults, although some might seem odd: The build partition is mounted at `/mnt/archiv` (german diction without trailing ‚e') and the unprivileged user for stage01 is called „mattias". You can adjust these values in the settings contained in the `config` directory, but we do not recommend so for the first steps.

4.1 Preparation

4.1.1 Prepare a drive

We strongly recommend building LessLinux on a SSD with at least 40 Gigabyte you can dedicate to LessLinux - random access to small files is the most costly operation when building LessLinux. However, for the fist steps a notebook and a 32 Gigabyte USB thumb drive might do – albeit much slower. Building on a virtual machine also works very good as long as you try to avoid fragmented hard disk images on rotational drives.

Prepare two partitions for the LessLinux build. They do not necessarily have to reside on the same hard disk, nor is a special partition scheme necessary - GPT and MBR will both work. Recommended size for the swap partition is 4 Gigabyte, the build partition itself should be between 20 and 60 Gigabytes. Larger sizes do not make any sense. Format them using a label that will be used to identify them when mounting:

```
mkswap -L LessLinux_swap /dev/sdb1
mkfs.btrfs -L LessLinux_build /dev/sdb2
```

After preparing the disk(s), either reboot LessLinux with the „FULL" ISO or run the following command:

```
/usr/share/lesslinux/auxiliary-scripts/prepare-lesslinux-build.sh
```

This command will be run everytime upon boot in the „FULL" ISO, so with the right labels you do not have to care about mounting anymore.

4.1.2 Create some directories

Some directories are not automatically created during the build process. This is intentional since you might want to share the sources directory between several machines building LessLinux. Create them manually as soon as `/mnt/archiv/LessLinux` is available:

```
mkdir -p /mnt/archiv/LessLinux/src
mkdir -p /mnt/archiv/LessLinux/llbuild
mkdir -p /mnt/archiv/LessLinux/llbuilder
```

4.1.3 Download the „sources"

The „sources" are really just build definitions - shell script fragments embedded in XML files - and download locations. Currently those download locations are also backed up at `http://distfiles.lesslinux.org/`. This location is provided for build convenience and satisfaction of the GPL/LGPL only. If you distribute modified builds of LessLinux you have to provide your own sets of build definitions and sources (either downloads or physically), so do not count on me.

Build definitions are currently just available as tarballs that unpack in `/mnt/archiv/LessLinux/llbuilder`, subversion access might or might not follow. Just take the build definitions that seem to fit best to your planned LessLinux derivative. If unsure, take the latest source used to build LessLinux Search and Rescue:

```
wget -O /tmp/llsrc.tar.xz \
http://download.lesslinux.org/src/\
lesslinux-search-and-rescue-uluru-YYYYMMDD-HHMMSS-buildscripts.tar.xz
```

Unpack the build scripts in the newly created directory llbuilder:

```
cd /mnt/archiv/LessLinux/llbuilder
tar xvJf /tmp/llsrc.tar.xz
```

4.2 Build the first stage

Index