Linux at the SOSHGIC

Documentation of Implementation

Joachim Breitner

5th December 2006

Abstract

During my semester at the SOS Hermann Gmeiner International College[1], my goal was to deploy Linux and Free Software as an alternative to Windows and proprietary software, for both staff and students. Although the project was a failure in the end, the technical work by itself was successful and should be documented here.

The targeted audience is on the one hand anyone who would want to continue my work at SOSHGIC and, on the other hand, anyone who plans to create a similar setup and can make use of the guidance and tips herein.

This document is no introduction to Linux, nor a real tutorial. It assumes proper knowledge of the Linux system and the used tools and only describes what goes beyond that.

Please excuse the linguistically horrible mixing of first and third person, singular and plural, imperative and indicative mood, passive and active, past and present tenses.

Joachim Breitner
mail@joachim-breitner.de

[1] `http://www.soshgic.edu.gh/`

Contents

1. Notation

Because a lot of files described here are saved in the rather long path `/usr/local/share/fai` on the FAI installation server, and since this path is mounted in `/fai` on the installation client, whereas the environment variable `$FAI` points to that directory, in the following text, `$FAI` refers to `/usr/local/share/fai`.

Lines in listings that would be too long for the page are wrapped. This indicated by the sign ↲.

2. The existing infrastructure

At the beginning of the 2006/07 school year, the school had a purely Windows based setup. In the server room is active and relevant for us:

- One domain controller, Windows 2003 based, also acting as DNS and DHCP server. (`main-server`, 10.0.0.1), and a secondary domain controller on 10.0.0.3.
- One mail server, Exchange based. (`mail-server`, 10.0.0.5)
- One file server, serving per-user-directories (the "h-drive") and others. User profiles are not stored in the network. (`nas-server`, 10.0.0.9)
- Two routers, one for staff and one for student computers, which also acts as a web filter (10.0.0.120, 10.0.0.204)

Besides this, there are two computer labs in the college, a few computers in the staff room as well as in several offices, all connected by wire. There is also a brand new lab on the hostel premises, connected internally and to the college by WLAN. Computers are all part of the domain, installation and maintenance is done manually.

3. Goals for the Linux setup

My technical goals for the Linux deployment were as follows:

- Automatic installation of the Linux system, even on machines with Windows already installed.
- User authentication via the Active Directory
- Access to the network services (shared drives, printers)
- Shared user home directories between Linux setups

4. Linux network outline

I was provided with a spare workstation, on which I installed Debian sarge, serving as a “general server” for now. On this machine, named `occ`, the following services are installed.

- Web server, to
  - be a local Debian mirror, for fast installations.
  - provide a blog and a wiki to the computer club I'm running.
  - run the Nagios network monitoring software.
- FAI installation server
- NFS server for the user's home directories
- SSH box to let users use terminal-based Linux software from Windows machines
- Authentication against the Active Directory using winbind
- Mail delivery using `mail-server` as the smarthost.

Later, an even older workstation served as a gateway to provide these services:

- Turning internet access on and off per classroom.
- Caching web proxy to save bandwidth and speed up browsing.

I also worked on the workstations, also equipped with Debian sarge. Installation works automatically by booting from a diskette/CD, and then installing from `occ`. These provide:

- GNOME based graphical user interface, with various applications. Most notably
  - OpenOffice
  - Firefox
  - GIMP
- Automatically mounted “h-drive”
- Home directories on the server
- Dual boot using Grub, with Windows as the default.
- Authentication against the Active Directory using winbind

5. Joining the Windows Domain

5.1. Custom Samba package

This section is obsolete, as we are using a backport from testing, as described in section 7.3.

Because the version of samba included in Debian Sarge does not support the `idmap_rid` method of mapping Windows user ids to Unix user ids, I had to recompile it. After getting the source (`apt-get source samba`) and installing the build dependencies (`apt-get build-dep samba`), I applied the changes (two lines added and a changelog entry) as described in listing 1. Building the package works by running `dpkg-buildpackage -rfakeroot`. The deb files are then put into our local repository (see section 7.2).

5.2. Configuration and joining

For the Linux machine to join the Windows domain, and to activate authentication against it, the following steps were taken.

1. Installation of the packages: `samba` `winbind` `ntp-simple`

2. Configuration of the file `/etc/samba/smb.conf`:

   - Add `realm = SOSHGIC.EDU.GH`, set `workgroup = SOSHGIC`
   - Enable WINS resolution with `wins server = 10.0.0.1`
   - Set `security = ads`
   - Append these to the `[global]` section to configure winbind: (Not all might be required)

     ```
     idmap backend = rid:SOSHGIC=2000-100000000
     idmap uid = 2000-100000000
     idmap gid = 2000-100000000
     template shell = /bin/bash
     allow trusted domains = No
     winbind use default domain = Yes
     password server= *
     client schannel = no
     winbind nested groups = Yes
     ```

     Note that we are using the `imap_rid` option. This plug-in ensures that the same user gets the same Unix user id on every machine, which is a prerequisite for the sharing of the home directory via NFS (see section 8).

   - Uncomment everything related to printing, sharing, the `passwd backend` line and otherwise uninteresting stuff.

   This configuration file can be used by all workstation and `occ` (until there will be special services there).

3. Configuration of the file `/etc/krb5.conf`

   - Set `default_realm = SOSHGIC.EDU.GH` in the section `[libdefaults]`.

Listing 1: Samba patch

```
diff -u samba-3.0.14a/debian/changelog samba-3.0.14a/debian/changelog
--- samba-3.0.14a/debian/changelog
+++ samba-3.0.14a/debian/changelog
@@ -1,3 +1,9 @@
+samba (3.0.14a-3sarge2.soshgic.1) stable-security; urgency=low
+
+  * enbled idmap_rid
+
+ -- Joachim Breitner <nomeata@debian.org>  Tue, 29 Aug 2006 08:22:59 +0000
+
 samba (3.0.14a-3sarge2) stable-security; urgency=high
 
   * Non-maintainer upload by the Security Team:
diff -u samba-3.0.14a/debian/rules samba-3.0.14a/debian/rules
--- samba-3.0.14a/debian/rules
+++ samba-3.0.14a/debian/rules
@@ -63,6 +63,7 @@
                --with-pam_smbpass \
                --with-libsmbclient \
                --with-winbind \
+              --with-shared-modules=idmap_rid \
                --with-msdfs \
                --with-automount \
                --with-tdbsam \
diff -u samba-3.0.14a/debian/winbind.files samba-3.0.14a/debian/winbind.files
--- samba-3.0.14a/debian/winbind.files
+++ samba-3.0.14a/debian/winbind.files
@@ -1,6 +1,7 @@
 usr/sbin/winbindd
 usr/bin/wbinfo
 usr/bin/ntlm_auth
+usr/lib/samba/idmap/idmap_rid.so
 usr/share/man/man1/wbinfo.1
 usr/share/man/man1/ntlm_auth.1
 usr/share/man/man8/winbindd.8
```

- In section [realms], add:

  ```
  SOSHGIC.EDU.GH = {
          kdc = 10.0.0.1
          admin_server = 10.0.0.1
  }
  ```

4. Start winbind and samba

5. Join the domain, using `net join -U some-admin`, whereas some-admin is an account on the domain with sufficient rights. Enter the password when asked.

6. Verify that winbind is working: `wbinfo -u` should list all domain users, `wbinfo -g` all groups.

7. Set up authentication:

   Listing 2: **File:** /etc/pam.d/common-auth

   ```
   auth    sufficient      pam_winbind.so
   auth    required        pam_unix.so nullok_secure use_first_pass
   ```

   As the user might log in for the first time and probably does not have a home directory, we use libpam-mkhomedir to create that. See also section 10.7.

   Listing 3: **File:** /etc/pam.d/common-session

   ```
   session required        pam_mkhomedir.so skel=/home/skel/ umask=0022 ↵
      silent
   session sufficient      pam_winbind.so
   session required        pam_unix.so
   ```

8. To make the system generally aware of the users, we need to configure the Name Service Switch. In /etc/nsswitch.conf, add `winbind` after `compat` to the `passwd` and `group` settings.

Now users with an account on the Active Directory should be able to login and find a home directory. Note that these are created in /home/SOSHGIC/$USERNAME, as that is what winbind sets the home directory to.

As none of these files contain the hostname, they can be identical on every system, which eases maintenance.

6. Web server on occ

The web server is installed using the packages `apache`, for the wiki `python-moinmoin` and for the blog `php4 libapache2-mod-php4 php4-mysql imagemagick`. For the blog we also need `mysql-server`.

Since both the blog and the wiki were hardly used, I will not describe the configuration here.

6.1. General changes to the web server configuration

In the file /etc/apache2/apache.conf, the ServerAdmin directive should be set to someone's e-mail address. Since we have no proper start page, I redirect everyone that tries to access / to /wiki/ using:

```
        RewriteEngine on
        RewriteRule ^/$ /wiki/ [R]
```

6.2. Configuration of Nagios

Nagios[2] is a sophisticated network monitoring software. It provides a choice of very useful overviews of the network (see figure 1)

Installation is easy, as Debian provides the packages nagios-text and nagios-plugins. The configuration is rather extensive if you have a lot of hosts, so I will only list examples here. The main configuration file /etc/nagios/nagios.cfg can be left as it is.

6.2.1. Hosts

Every host that should be monitored (that includes access points, routes, managed switches) needs an entry in /etc/nagios/hosts.cfg. Virtual entries for servers, printers and access points were created to store common settings, and the real entries can inherit them. The entry for the occ server would be:

```
define host{
        use                     generic-server
        host_name               occ
        alias                   occ Linux Server
        address                 10.0.0.19
}
```

Furthermore, every host is a member of one host group, defined in hostgroups.cfg. Here we can define what contact groups are resonsible for this set of hosts.

```
define hostgroup {
        hostgroup_name  linux-server
        contact_groups  admins
        alias           Linux servers
        members         occ, netswitch
}
```

For nice icons in the web front end, we can add "secondary" information to /etc/nagios/hostextinfo.cfg. The images used are stored in /usr/share/nagios/htdocs/images/logos/ and partly downloaded from the internet.

```
define hostextinfo{
      host_name       occ, netswitch
      icon_image      base/linux40.png
      icon_image_alt  Linux
```

[2] http://www.nagios.org/

```
      vrml_image      base/linux40.png
      statusmap_image base/linux40.gd2
}
```

6.2.2. Contacts

Nagios allows a quite detailed setup with regard to who gets notified how if some thing is happening at some time. Every administrator has an entry in /etc/nagios/contacts.cfg that specifies when this contact should be notified, what kind of events should be relayed and how the contact should be made. In my case, during class hours, by e-mail, on any kind of event.

```
define contact{
        contact_name                    joachim
        alias                           Joachim Breitner
        service_notification_period     classhours
        host_notification_period        classhours
        service_notification_options    w,u,c,r
        host_notification_options       d,u,r
        service_notification_commands   notify-by-email
        host_notification_commands      host-notify-by-email
        email                           joachim@soshgic.edu.gh
}
```

Contacts are put together to contact groups in /etc/nagios/contactgroups.cfg:

```
define contactgroup{
        contactgroup_name       admins
        alias                   General Admins
        members                 joachim,maslino
}
```

Notification times are specified in /etc/nagios/timeperiods.cfg

```
define timeperiod{
        timeperiod_name classhours
        alias           Class Hours
        monday          07:30-17:00
        tuesday         07:30-17:00
        wednesday       07:30-17:00
        thursday        07:30-17:00
        friday          07:30-17:00
}
```

6.2.3. Services

Every service in the network (e.g. a web server, availability to pings) is put into /etc/nagios/services.cfg. The entry for http servers is:

```
define service{
        use                             generic-service
        host_name                       www-server,mail-server,netswitch,occ↵
           ,cache-raq
        service_description             HTTP
        is_volatile                     0
        check_period                    24x7
        max_check_attempts              3
        normal_check_interval           5
        retry_check_interval            1
        contact_groups                  admins
        notification_interval           120
        notification_period             24x7
        notification_options            w,u,c,r
        check_command                   check_http
}
```

Besides HTTP, we are monitoring the services ping, http cache, internet connectivity, DHCP, DNS (internal & external), POP3, IMAP, NTP, SMTP (spam filter & exchange), free space on `nas-server` and `occ`, the mail queue size on `occ`, SSH and Telnet.

`/usr/lib/nagios/plugins/check_disk_smb` was modified to take a `-A` parameter to specify a samba credentials file, instead of entering the password directly. The file is saved as `/etc/nagios/smbclient.auth` with permission `600` and owned by the user `nagios`.

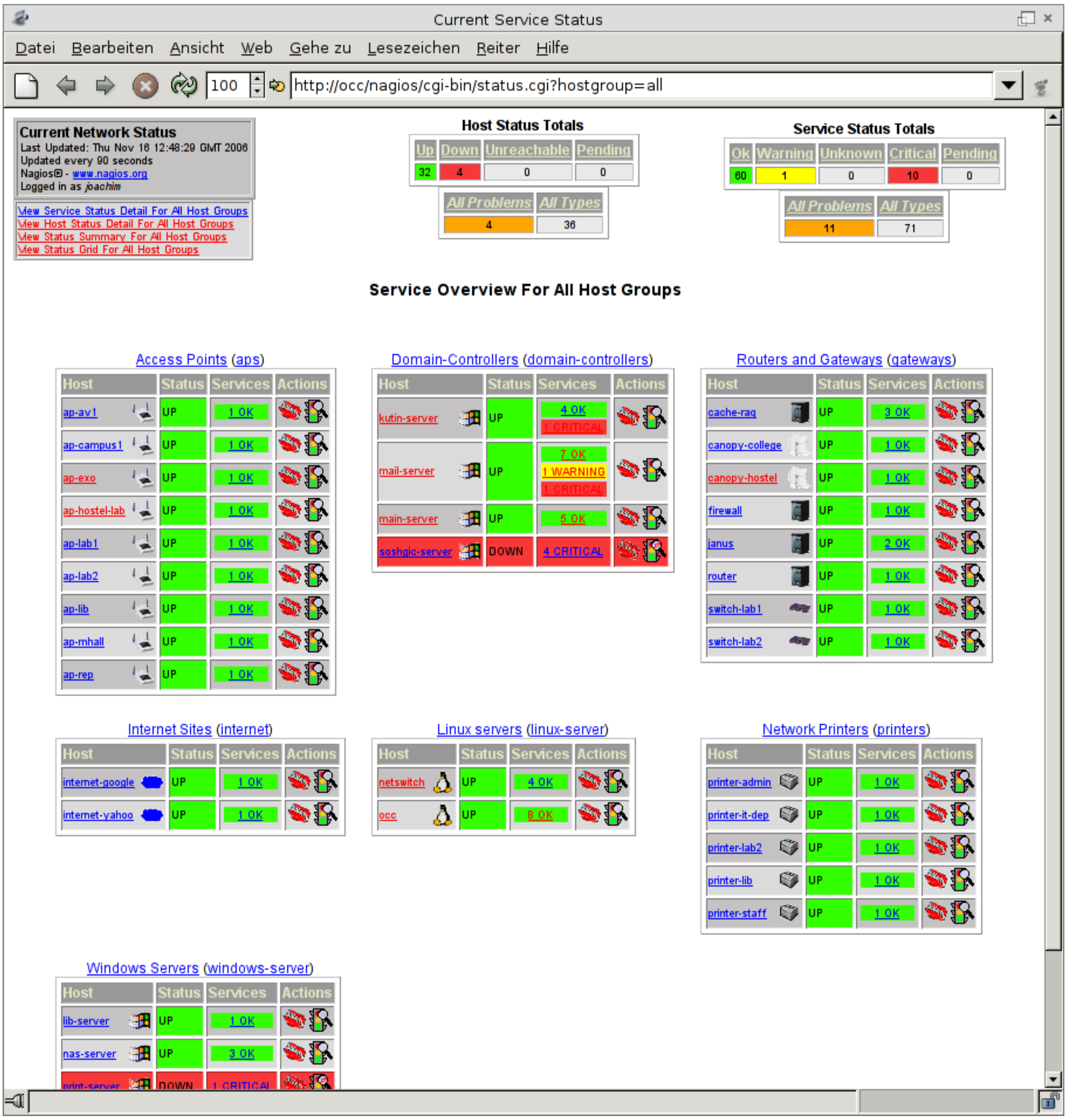
Notification by windows popup message is made possible through this entry to `/etc/nagios/misccommands.cfg`. Note that the destination host is hard coded into this file:

```
define command{
        command_name    notify-smb-msg
        command_line    /usr/bin/printf "%b" "***** Nagios  *****\n\↵
           nNotification Type: $NOTIFICATIONTYPE$\n\nService: $SERVICEDESC$\↵
           nHost: $HOSTALIAS$\nAddress: $HOSTADDRESS$\nState: $SERVICESTATE$↵
           \n\nDate/Time: $DATETIME$\n\nAdditional Info:\n\n$OUTPUT$\n(End ↵
           of Message)\n\n" | smbclient -U nagios@occ -M IDCARD
}
```

6.3. http-Authentication against AD using ldap

Some parts of the web server, for example the monitoring software, should only be accessible for IT staff. This can easily be achieved by connecting Apache's authentication mechanisms to the ldap server which is provided by Microsoft's Active Directory. For this and other uses of the AD from a Linux service, a user "linux-system" with full name "Linux System Account" was created in the AD with sufficient rights.

The general ldap configuration for apache was put in `/etc/apache2/ldap.conf` (listing 4 and activated by putting `Include /etc/apache2/ldap.conf` into `/etc/apache2/apache2.conf`. It can read the data from any of the tree domain controllers. Because the file contains a password, it's access mode was set to `600`.



Current Service Status

Datei Bearbeiten Ansicht Web Gehe zu Lesezeichen Reiter Hilfe
100

http://occ/nagios/cgi-bin/status.cgi?hostgroup=all

Current Network Status
Last Updated: Thu Nov 16 12:48:29 GMT 2006
Updated every 90 seconds
Nagios® - www.nagios.org
Logged in as *joachim*

View Service Status Detail For All Host Groups
View Host Status Detail For All Host Groups
View Status Summary For All Host Groups
View Status Grid For All Host Groups

Host Status Totals

Up	Down	Unreachable	Pending
32	4	0	0

All Problems	*All Types*
4	36

Service Status Totals

Ok	Warning	Unknown	Critical	Pending
60	1	0	10	0

All Problems	*All Types*
11	71

Service Overview For All Host Groups

Access Points (aps)

Host	Status	Services	Actions
ap-av1	UP	1 OK	
ap-campus1	UP	1 OK	
ap-exo	UP	1 OK	
ap-hostel-lab	UP	1 OK	
ap-lab1	UP	1 OK	
ap-lab2	UP	1 OK	
ap-lib	UP	1 OK	
ap-mhall	UP	1 OK	
ap-rep	UP	1 OK	

Domain-Controllers (domain-controllers)

Host	Status	Services	Actions
kutin-server	UP	4 OK 1 CRITICAL	
mail-server	UP	7 OK 1 WARNING 1 CRITICAL	
main-server	UP	5 OK	
soshgic-server	DOWN	4 CRITICAL	

Routers and Gateways (gateways)

Host	Status	Services	Actions
cache-raq	UP	3 OK	
canopy-college	UP	1 OK	
canopy-hostel	UP	1 OK	
firewall	UP	1 OK	
janus	UP	2 OK	
router	UP	1 OK	
switch-lab1	UP	1 OK	
switch-lab2	UP	1 OK	

Internet Sites (internet)

Host	Status	Services	Actions
internet-google	UP	1 OK	
internet-yahoo	UP	1 OK	

Linux servers (linux-server)

Host	Status	Services	Actions
netswitch	UP	4 OK	
occ	UP	8 OK	

Network Printers (printers)

Host	Status	Services	Actions
printer-admin	UP	1 OK	
printer-it-dep	UP	1 OK	
printer-lab2	UP	1 OK	
printer-lib	UP	1 OK	
printer-staff	UP	1 OK	

Windows Servers (windows-server)

Host	Status	Services	Actions
lib-server	UP	1 OK	
nas-server	UP	3 OK	
print-server	DOWN	1 CRITICAL	

Figure 1: The service overview in nagios

Listing 4: **File:** (/etc/apache2/ldap.conf

```
<Directory />
  AuthLDAPAuthoritative on
  AuthType Basic
  AuthName "SOSHGIC Webserver"
  AuthLDAPBindDN "cn=Linux System Account,cn=Users,dc=soshgic,dc=edu,dc=gh"
  AuthLDAPBindPassword "%PASSWORD%"
  AuthLDAPURL "ldap://10.0.0.1:3268 10.0.0.3:3268 10.0.0.5:3268/dc=soshgic,↲
     dc=edu,dc=gh?sAMAccountName?sub?(objectClass=*)"
</Directory>
```

7. Debian mirror

7.1. Mirroring the archive

For further installations of Debian I copied the whole archive of Debian software, stable version, to the host, using the `debmirror` program and this command:

```
debmirror -a i386 -s main -s contrib -s non-free \
          -h ftp.no.debian.org -d sarge          \
          --nosource --progress                  \
          /var/www/debian/
```

I added a symbolic link from `sarge` to `stable` in `/var/www/debian/dists/`.

A similar command mirrored the debian security updates:

```
debmirror -d stable/updates -s main -s contrib -s non-free -a i386  \
          -h security.debian.org -r debian-security                 \
          --nosource --progress                                     \
          /var/www/debian-sec/
```

And a not-so-different command to mirror the FAI packages (which are just 5 MB):

```
debmirror -d sarge -s koeln -a i386                              \
          -e http -h www.informatik.uni-koeln.de -r fai/download \
           --nosource --progress  -v --ignore-release-gpg        \
          /var/www/debian-fai/
```

These commands are run every night via cron, from the `/etc/cron.daily/debmirror`.

7.2. Repository for local packages

Sometimes, one wants to put locally modified, upgraded or created packages on the Debian machines. For that, we use a simple local repository.

Any Debian packages that should be distributed are put in `/var/www/debian-soshgic/`, and then the following commands has to be run in that directory after any changes.

```
dpkg-scanpackages . /dev/null |gzip > Packages.gz
dpkg-scansources  .           |gzip > Sources.gz
```

7.3. Backports

Of some programs we want a version newer than in the current Debian stable distribution. Often, such packages can be found on `http://www.backports.org`. In `/var/www/debian-backports`, the script `update.sh` will copy interesting packages to that directory and recreate the `Packages.gz` file. This script is also run by `/etc/cron.daily/debmirror`. Currently, backports of the following packages are used:

1. `openoffice-org` and related packages.
2. Sun Java JRE and JDK, for OpenOffice and programming
3. The `samba` suite, because of the Kerberos support in `pam_winbind`, `smbspool`, as well as the `idmap_rid` plugin. This depends on a newer version of `lsb-base`, which is also retrieved from `http://www.backports.org`.

7.4. The apt sources

Now we can use the apt-sources file as in listing 5 in all of SOSHGIC:

Listing 5: **File:** `/etc/apt/sources.list`

```
# General debian sarge packages, mirrored
deb http://occ.soshgic.edu.gh/debian/ stable main contrib non-free
deb-src http://ftp.no.debian.org/debian/ stable main contrib non-free

# Security updates, mirrored
#deb http://security.debian.org/ stable/updates main contrib non-free
deb http://occ.soshgic.edu.gh/debian-sec stable/updates main contrib non-↵
    free
deb-src http://security.debian.org/ stable/updates main contrib non-free

# Local packages
deb http://occ.soshgic.edu.gh/debian-soshgic/ ./
deb-src http://occ.soshgic.edu.gh/debian-soshgic/ ./

# Backports (copied by update.sh)
deb http://occ.soshgic.edu.gh/debian-backports/ ./

# FAI packages
deb http://occ.soshgic.edu.gh/debian-fai/ sarge koeln
```

8. NFS-Server on occ

We want the home directories of the user shared across all installations, and for that we rely on the usual Unix network file system NFS. It is also required for FAI (see section 9).

The necessary packages on the server for NFS are `nfs-kernel-server` and it's dependencies. To export ("share") a file system, an entry has to be added to `/etc/exports` (see listing 6).

Listing 6: **File:** `/etc/exports`

```
# /etc/exports: the access control list for filesystems which may be
#               exported NFS clients.  See exports(5).

# Switching to NFSv4 or kerberized NFSv3 is very desireable but
# hard to do with Active Directory as the Kerberos server
# no_root_squash because of mkhomedir :-(
/home           10.0.0.0/24(rw,no_root_squash,async,no_subtree_check,mp)
/var/games      10.0.0.0/24(rw,root_squash,async)

# FAI stuff
/usr/local/share/fai 10.0.0.0/24(async,ro)
/usr/lib/fai/nfsroot 10.0.0.0/24(async,ro,no_root_squash)
```

We also want the GNOME desktop to be notified about file changes. The daemon for that, famd of the package `fam` ("File Alteration Monitor") is network capable after you set `local_only = false` in `/etc/fam.conf`.

9. Fully Automatic Installer

To make the installation of Linux workstation no hassle, I decided to deploy FAI, the Fully Automatic Installer[3].

9.1. Quick overview of FAI

Usually with FAI, the workstations that are to be installed boot over the network, get a bit of configuration from the DHCP server (network configuration, host name, from what server to install). They then start an installation system (which is Debian, too) over NFS on the server. A script takes over, automatically partitioning and formatting the hard drive, installing and setting up all software. After a reboot, the system is ready to use.

9.2. Problems in SOSHGIC

Due to the current setup of windows at the SOSHGIC, we have a few issues to attend to when deploying FAI:

1. The DHCP server is running on windows. It would probably be possible to actually add the FAI configuration there, but probably not worth the hassle. At least unless there is

[3] `http://www.informatik.uni-koeln.de/fai/`

a scripting possibility. Also, the machines don't boot over the network. And if they did, we do not have a proper DHCP server, so it won't help.

2. The host name can't be the one that is in the DNS, as the machines will have to join the Active Directory Domain (see section 5.2), and they have to use a different host name for that than the already running windows machine.

3. There is already windows installed on the machines, and this installation must not be destroyed.

4. Some of the pre-installed Windowses and/or BIOSes care about the non existing and obsolete disk geometry, while Linux does not care anymore. In some cases, the installation will re-write the partition table assuming 16 heads geometry, instead of 255, which can cause either the BIOS or Windows to freeze at start.

To solve these problems, I decided to do the following:

1. can be avoided by using boot floppies with the Linux kernel to boot. The kernel itself then gets its network configuration via DHCP, the NFS server is hard coded on the floppy. The floppy is actually burned on CD, as some machines don't have a floppy drive (including the laptop I'm working on).

2. I chose to append "l" to the host name for the Linux installations. The host name is set manually right after it starts booting from the floppy, at the `grub` menu.

3. I wrote an FAI hook that, before FAI partitions the disk, resizes Windows' NTFS partition to free at least 10GB. That partition is marked as `preserve` in FAI's configuration, so FAI won't touch it.

4. After the installation, using a fai script (listing 10), the partition table is unconditionally re-written with 255 heads.

9.3. Setting up FAI on the server

FAI is sufficiently well documented, so I will only describe the changes to the configuration that I did.

In the file `/etc/fai/make-fai-nfsroot.conf`, I added `ntfsprogs` to the list of packages (`NFSROOT_PACKAGES`). In `/etc/fai/fai.conf`, I set both `installserver` and `mirrorhost` to `occ.soshgic.edu.gh`. Other changes were not required.

The `/etc/fai/sources.list` can be the same as `/etc/apt/sources.list`. If the Internet connection is very slow, one might remove the `deb-src` lines, then everything is taken from the local network.

Now, the fai nfsroot and default configuration can be created with `fai-setup`. This will also append two lines to `/etc/exports`, so `nfs-kernel-server` has to be installed and running on `occ`.

The fai boot floppy and boot CD images are created by

```
make-fai-bootfloppy -f fai-img.floppy -i fai-img.iso -v    \
                    nfsroot=10.0.0.19:/usr/lib/fai/nfsroot \
                    FAI_FLAGS=createvt FAI_ACTION=install  \
                    HOST=setmeonboot
```

the file `fai-img.iso` can then be burned to a CD and used for the installation. The parameters will be passed to the kernel, the `HOST` parameter needs to be changed after `grub` comes up, using “e”, “e” and after the change “b”.

For the wireless PCs, this does not work. Instead, I created a boot CD custom made for the SOSHGIC, see chapter 11 about that. This CD can then also be used to replace the CD described here, and offers a menu to set the host name.

9.4. FAI scripts

The installation on the client itself is controlled by scripts in the directory `/usr/local/share/fai`, here referred to as `$FAI`. In listing 7 you see what classes are defined for what host names.

Listing 7: **File:** `$FAI/class/50-host-classes`

```
#! /bin/sh

# assign classes hosts

# SOSHGIC Defaults
echo "FAIBASE DHCPC WINBIND XFREE DESKTOP EDU PRINTING"

# Room classes
case $HOSTNAME in
    sw*)
        echo "SW" ;;
    lib*)
        echo "LIB" ;;
    clh*)
        echo "LAB1" ;;
    clk*)
        echo "LAB2" ;;
    hl*)
        echo "GAMES HL" ;;
esac

(ifclass I386 || ifclass AMD64) && echo GRUB
exit 0
```

In `$FAI/class/FAIBASE.var`, I set the `rootpw` to a value created by `mkpasswd -H md5` and the `time_zone` to `Africa/Accra`

The file `$FAI/disk_config/FAIBASE`, see listing 8, is important, as it defines the partition layout. For the workstation, the trivial layout of using one partition (and one for swap) was chosen.

The directory `$FAI/gconf` with the file `soshgic-panel.entries` was created by me, see the section 10.8 about preconfiguring the desktop.

Listing 8: **File:** `$FAI/disk_config/FAIBASE`

```
# SOSHGIC disk configuration
# will leave the NTFS partition alone and use up to 11GB of space
# <type> <mountpoint> <size in mb> [mount options]     [;extra options]

disk_config disk1
primary  -             preserve1
primary  /boot         100-500          rw,errors=remount-ro ; boot -j ext3
logical  /             3000-10000       rw,errors=remount-ro ; -j ext3
logical  swap          512-1024    rw
logical  -             512-1024
# logical swap preserve5 rw
# logical /    preserve6 rw,errors-ro ; -j ext3
```

The NTFS resizing happens in `$FAI/hooks/partition.DEFAULT`, in full length printed here in listing 33.

Some machines are connected by wireless. Because of that, the script in listing 9 copies the module saved in `$FAI/modules/rt61.ko` as well as a workaround for a problem with the driver.

Listing 9: **File:** `$FAI/scripts/FAIBASE/60-rt61`

```
#!/bin/bash

cp $FAI/modules/rt61.ko ${target}/lib/modules/*/
$ROOTCMD depmod -a

cat > $target/etc/network/if-pre-up.d/ra-fix <<’EOF’
#!/bin/sh
if [ "$IFACE" == "ra0" ]
then
  ifconfig "$IFACE" up
else
  exit 0
fi
EOF

chmod +x $target/etc/network/if-pre-up.d/ra-fix
```

As already explained, we have to fix the partition table after the installation. This happens in the script in listing 10.

To enable passwordless logins for the administrator, we copy a public ssh key to root's account, in the script listed in 11. The corresponding private key is in `occ`'s root's home directory.

The selection of the software happens in the files in `$FAI/package_config`. The standard

Listing 10: **File:** `$FAI/scripts/FAIBASE/70-fixdisk`

```
#!/bin/bash

[ "$FAI_ACTION" == "softupdate" ] && exit 0

echo "Fixing disk geometry"

if [ -e /sys/block/sda ]
then
        echo "SCSI (SATA?) device found, aborting"
        exit 0
else
        echo "SCSI (SATA?) device not found, sticking to /dev/hda"
        disk=/dev/hda
fi

echo "Forcing table to 255 Heads"
sfdisk -d $disk | sfdisk -f -H255 $disk
```

Listing 11: **File:** `$FAI/scripts/XFREE/80-sshkey`

```
#!/bin/bash
echo Installing SSH-Key
fcopy -v root/.ssh/authorized_keys
```

$FAI/package_config/FAIBASE file was modified: jove, nscd and rsh-client are not installed, vim was added. These are the other package lists:

- Listing 12: Winbind related package
- Listing 13: XFREE related packages
- Listing 14: Desktop related packages.
- Listing 15: a selection of educational packages.
- Listing 16: printing related packages

Listing 12: **File:** $FAI/package_config/WINBIND

```
PACKAGES install
samba
winbind
ntp-simple
libpam-mount
smbfs
smbclient
krb5-user
```

After the installation, FAI runs the scripts in the directory $FAI/scripts. These and other files in $FAI are explained in their appropriate section of "Workstation Configuration".

10. Workstation configuration

This section describes how the workstations are set up. As all the configuration happens automatically during the installation, the files referred here are actually on the FAI installation server, but run on the workstations during installation.

10.1. GRUB configuration

To the standard file $FAI/scripts/GRUB/10-setup I added line fcopy -v boot/grub/splash .xpm.gz to install a nice background for the boot manager, which I have saved in $FAI/files /boot/grub/splash.xpm.gz/GRUB. The GRUB menu file in listing 34 loads the boot splash file, sets the password for grub's advanced features and has a static entry that loads Windows, which is default, meaning that it will be loaded after timeout seconds, if the user does not do anything.

As GRUB has code installed in the master boot record, this has to be recovered when an Installation of Windows happens after the Linux system is installed. The easiest way is by booting from a live CD (e.g. Knoppix[4]), running mount /dev/hda5 /mnt and chroot /mnt grub-install /dev/hda. Alternatively, the custom boot CD in section 11 has a comfortable menu entry for that.

[4] http://www.knoppix.org

Listing 13: **File:** `$FAI/package_config/XFREE`

```
PACKAGES install
# No x-window-system, as we don't want xprint
x-window-system-core
xterm
xfs
twm
xnest

xlockmore-gl
ttf-freefont
type1inst
xscreensaver-nognome
xscreensaver-gl
gdm

# packages for auto-configuration
mdetect
hwinfo
xresprobe
hwinfo
read-edid
xdebconfigurator
```

Listing 14: **File:** $FAI/package_config/DESKTOP

```
PACKAGES install

mozilla-browser-
gnome-desktop-environment
mozilla-firefox-gnome-support
mozilla-firefox
evolution
evolution-exchange
menu
gnumeric
gimp
eject
gthumb
gaim
totem
flashplugin-nonfree

openoffice.org
openoffice.org-l10n-en-us
openoffice.org-l10n-en-gb
openoffice.org-help-en-us
myspell-en-gb
myspell-en-us

rdesktop

libc6-dev
gcc
g++
make
```

Listing 15: **File:** $FAI/package_config/EDU

```
PACKAGES install

gperiodic
openuniverse
geg
drgeo
xabacus
chemtool
xmaxima
```

Listing 16: **File:** `$FAI/package_config/PRINTING`

```
PACKAGES install

cupsys
cupsys-client
cupsys-bsd
foomatic-filters
```

10.2. GDM configuration

To enable the user to shut down the computer without superuser rights, the script in listing 17 is run, modifying the GDM settings. This also enables the graphical installer using our custom theme and disables the configuration of GDM and starting the XDMCP chooser from the login screen.

A custom gdm theme for SOSHGIC has been created, sporting a Ghana-coloured tux and a help text for the user. The theme is saved in `$FAI/gdm-themes` and copied by the script in 18.

10.3. X configuration

Configuration of the X Windowing system can be tricky, especially automatic ones. We rely on the `xdebconfigurator` from the debian-edu project, and we run this in a fai script (see listing 19).

10.4. Winbind configuration

Winbind is also configured, as described in section 5.2. Three scripts are at work: The one in listing 22 will make sure the home directories are mounted from the server (see section 8). Listing 23 copies the various configuration files to the client. `/etc/pam.d/common-auth`, listed in 20, and `/etc/pam.d/common-session`, listed in 21, are different: The `pam_winbind` stores the user's kerberos ticket and `pam_mount` automatically mounts the h-drive. The last script, listing 24 finally joins the domain, if necessary.

10.5. DHCP configuration

To make the Linux computer tell the windows DHCP server to update the windows DNS, a few extra entries in the file DHCP client config file `/etc/dhcp3/dhclient.conf` (listing 25) were required. A FAI script (listing 26) copies that configuration file and sets the hostname in it.

10.6. Printing configuration

Printing is managed by CUPS[5]. One challenge is to make sure that although there is a system wide spooler on the workstation all print requests are correctly authenticated to the print server, for accounting purposes. Luckily, the `smbspool` program, which is CUPS' back end to

[5] `http://www.cups.org/`

Listing 17: **File:** `$FAI/scripts/XFREE/10-gdm-config`

```
#! /usr/bin/cfengine

control:
   OutputPrefix = ("cfengine")
   actionsequence = ( files directories editfiles )
   EditFileSize = ( 30000 )

files:
   any::

directories:
   any::

editfiles:
   any::
        { ${target}/etc/gdm/gdm.conf
          ReplaceAll "^#?Greeter=.*"
                With "Greeter=/usr/bin/gdmgreeter"

          ReplaceAll "^#?SecureSystemMenu=.*"
                With "SecureSystemMenu=false"

          ReplaceAll "^#?ChooserButton=.*"
                With "ChooserButton=false"

          ReplaceAll "^#?ConfigAvailable=.*"
                With "ConfigAvailable=false"

          ReplaceAll "^#?Welcome=.*"
                With "Welcome=Welcome to Linux"

          ReplaceAll "^#?GraphicalTheme=.*"
                With "GraphicalTheme=soshgic"

          ReplaceAll "^#?UseCirclesInEntry=.*"
                With "UseCirclesInEntry=true"

        }
```

Listing 18: **File:** $FAI/scripts/XFREE/10-gdm-themes

```
#!/bin/bash

for dir in $FAI/gdm-themes/*
do
  echo "Copying gdm theme $dir"
  cp -rva $dir $target/usr/share/gdm/themes/
done
```

Listing 19: **File:** $FAI/scripts/XFREE/30-xconfig

```
#!/bin/bash

# Recreate X config database
$ROOTCMD xdebconfigurator
$ROOTCMD debconf-set-selections <<__END__
xserver-xfree86 xserver-xfree86/config/display/modes    multiselect     1280↲
   x1024, 1024x768, 800x600, 640x480
xserver-xfree86 xserver-xfree86/config/inputdevice/mouse/port   select  /dev↲
   /input/mice
__END__
# We don't have the nvidia driver and use nv instead (xdebconfigurator bug)
if $ROOTCMD debconf-get-selections | grep xserver-xfree86/config/device/↲
   driver | grep -q  nvidia
then
        $ROOTCMD debconf-set-selections <<__END__
        xserver-xfree86 xserver-xfree86/config/device/driver  select nv
__END__
fi
$ROOTCMD dexconf
```

Listing 20: **File:** $FAI/files/etc/pam.d/common-auth/WINBIND

```
# /etc/pam.d/common-auth - authentication settings common to all services

auth    required        pam_mount.so
auth    sufficient      pam_winbind.so use_first_pass krb5_auth ↲
   krb5_ccache_type=FILE
auth    required        pam_unix.so nullok_secure use_first_pass
```

Listing 21: **File:** `$FAI/files/etc/pam.d/common-session/WINBIND`

```
#
# /etc/pam.d/common-session - session-related modules common to all services
#
# This file is included from other service-specific PAM config files,
# and should contain a list of modules that define tasks to be performed
# at the start and end of sessions of *any* kind (both interactive and
# non-interactive).  The default is pam_unix.
#
session required        pam_mkhomedir.so silent skel=/home/skel/ umask=0022
session required        pam_mount.so
session sufficient      pam_winbind.so
session required        pam_unix.so
```

Listing 22: **File:** `$FAI/scripts/WINBIND/10-home`

```
#! /usr/bin/cfengine

control:
   OutputPrefix = ("cfengine")
   actionsequence = ( files directories editfiles )
   EditFileSize = ( 30000 )

files:
   any::

directories:
   any::

editfiles:
   any::
        { ${target}/etc/fstab
          AppendIfNoSuchLine "10.0.0.19:/home  /home  nfs  defaults,bg,tcp 0↵
               0"
          AppendIfNoSuchLine "10.0.0.19:/var/games  /var/games  nfs  ↵
              defaults,bg 0 0"

        }
```

Listing 23: **File:** `$FAI/scripts/WINBIND/20-auth`

```
#!/bin/bash

fcopy -v /etc/security/pam_mount.conf
fcopy -v /etc/samba/smb.conf
fcopy -v /etc/krb5.conf
fcopy -v /etc/nsswitch.conf
fcopy -v /etc/pam.d/common-auth
fcopy -v /etc/pam.d/common-session
fcopy -v /etc/pam.d/common-account
fcopy -v /etc/pam.d/ssh
```

Listing 24: **File:** `$FAI/scripts/WINBIND/30-join`

```
#!/bin/bash

if ! $ROOTCMD wbinfo -t
then
        echo "Joining SOSHGIC Domain"
        $ROOTCMD net join -U linux-system%PASSWORD%
else
        echo "Already in the Domain"
fi
```

Listing 25: **File:** `$FAI/files/etc/dhcp3/dhclient.conf/FAIBASE`

```
# Make DHCP server update the Hostname
send fqdn.server-update on;
send fqdn.no-client-update on;
send fqdn.fqdn "HOSTNAME.soshgic.edu.gh";
send fqdn.encoded off;

# dhclient.conf defaults
request subnet-mask, broadcast-address, time-offset, routers,
        domain-name, domain-name-servers, host-name,
        netbios-name-servers, netbios-scope, interface-mtu;
```

Listing 26: **File:** `$FAI/scripts/FAIBASE/50-dhclient`

```
#!/bin/bash

# Install dhclient.conf file and set hostname therein
fcopy -Bv /etc/dhcp3/dhclient.conf
sed -i "s/HOSTNAME/${HOSTNAME}/" ${target}/etc/dhcp3/dhclient.conf
```

access printers shared by Windows Servers, looks for a valid kerberos ticket of the users that originally sent the print job and tries to authenticate with that, which is exactly what we need. This was the main reason to use a backport of Samba instead of the version in Debian sarge, as that did not yet contain this functionality. Note that CUPS 1.2 will have problems with this setup, see Debian bug #372270.

The default settings of CUPS have not been modified much: The `Port` directive in `/etc/cups/cupsd.conf` has been replaced with `Listen 127.0.0.1:631` for increased security and `Browsing` has been switched off. Printers are created as usual. PPD-files for all printers are put in `$FAI/ppd`, and the script in listing 27 copies the relevant ones to `/etc/cups/ppd`. CUPS saves it's printer settings (e.g. default tray option) in these files, so to have such a setting on all computers, it can be created by any CUPS installation first.

This script also sets the default printer to whatever is specified in the variable `$DEFAULT_PRINTER` , which in turn is set per class in "var"-files in `$FAI/classes/`.

Listing 27: **File:** `$FAI/files/etc/cups/printers.conf/postinst`

```
#! /bin/bash

error=0 ; trap "error=$((error|1))" ERR

set -a

for printer in $(perl -n -e 'print "$1\n" if /Printer (.+)>/' ${target}/etc/↵
    cups/printers.conf)
do
  cp -v ${FAI}/ppd/$printer.ppd  ${target}/etc/cups/ppd
done

if [ -n "$DEFAULT_PRINTER" ]
then
  echo "Setting default printer to $DEFAULT_PRINTER"
  sed -i "s/<DefaultPrinter/<Printer/;
          s/<Printer $DEFAULT_PRINTER/<DefaultPrinter $DEFAULT_PRINTER/" \
          ${target}/etc/cups/printers.conf
else
  echo "Warning: no default printer set!"
fi

exit $error
```

10.7. Home directory creation

It is not advisable to pre-create home directories for all existing users, mostly because this would have to be done when new users are added, and might be forgotten. Instead, when a user logs in the very first time, a home directory should be created. There is a pam module for that, `pam-mkhomdir`, which is configured in `/etc/pam.d/common-session` (listing 21). It will copy the content of /home/skel to the user's directory. So far, that directory only contains the desktop director with a symbolic link to the h_drive directory, where the users's "H-Drive" will be mounted.

Because the workstation will have to create the user's home directory on the server, which only the super user can do, the export in listing 6 has the no_root_squash option set.

When the user logs in, his "H-Drive" will be mounted automatically, using his just typed-in credentials. The relevant line in /etc/security/pam_mount.conf is volume * smb 10.0.0.9 &$ ~/h_drive uid=&,gid=&,dmask=0750,workgroup=SOSHGIC - -

10.8. GNOME pre-configuration

The GNOME desktop stores the users' configuration in XML files handled by gconf. It allows setting default and mandatory settings system wide, so the FAI script in listing 28 will load any gconf setting files in $FAI/gconf as default settings. The file $FAI/gconf/soshgic-panel.entries contains small modifications to the icons that the panel presents.

Listing 28: **File:** $FAI/scripts/DESKTOP/10-gconf

```
#!/bin/bash

for file in $FAI/gconf/*
do
  echo "Applying GConf file $file"
  cp $file $target/tmp/$(basename $file)
  $ROOTCMD gconftool-2 --direct --config-source xml:readwrite:/etc/gconf/↵
      gconf.xml.defaults --load /tmp/$(basename $file)
done
```

Two more FAI scripts, listed in 29 and 30, fix a few file dependencies. Debian usually allows access to USB devices, CD-ROMs and audio based upon group membership. This does not fit to our setup, so we remove these requirements.

Listing 29: **File:** $FAI/scripts/DESKTOP/20-pmount

```
#!/bin/bash
# Make pmount usable by everyone:
$ROOTCMD dpkg-statoverride --remove /usr/bin/pmount
$ROOTCMD dpkg-statoverride --update --add root plugdev 4755 /usr/bin/pmount
$ROOTCMD dpkg-statoverride --remove /usr/bin/pumount
$ROOTCMD dpkg-statoverride --update --add root plugdev 4755 /usr/bin/pumount
```

Listing 30: **File:** `$FAI/scripts/DESKTOP/30-perms`

```
#!/bin/bash
# Make cdrom ejectable/writeable by all
sed -i 's/GROUP="cdrom"/MODE="0644"/' ${target}/etc/udev/permissions.rules
# Audio for everyone
sed -i 's/^SUBSYSTEM="sound".*/SUBSYSTEM="sound",    MODE="0666"/' ${target↲
   }/etc/udev/permissions.rules
```

11. Boot CD

The special treatment needed by those workstations who are connected by wireless network made a custom boot CD with the following tasks:

- Boot and present a menu to the user.
- Set up networking, either over wireless or regular network.
- Load FAI's nfsroot and start the installation.
- Fix the partition table or the grub installation, if needed.
- Run a shell as a simple rescue system.

See the list of files on the ramdisk on the CD (listing 31): The hand-picked selection of binaries in `/bin` contains, among some network tools, `busybox`, which can serve as a multitude of binaries, thus saving space. The symbolic links to that binary are skipped in the listing, as are the devices files in `/dev`. `/sbin` is a symbolic link to `/bin`. `/var` to `/tmp` and `/etc/resolv.conf` to `/tmp/resolv.conf`. `rt61.ko` is the kernel module for the wireless cards in use.

The intelligence of the boot disk lies in `/linuxrc` (listing 38). This shell script presents the menu, detects the presence of a wireless card, knows how to mount the nfsroot and start the installation, after having asked the user for the hostname of the machine to install.

The initrd is created using `/usr/sbin/mkcramfs ramdisk/ cdimage/initrd`, assuming that the ramdisk files are in the directory `ramdisk/`. In `cdimages/` are these additional files:

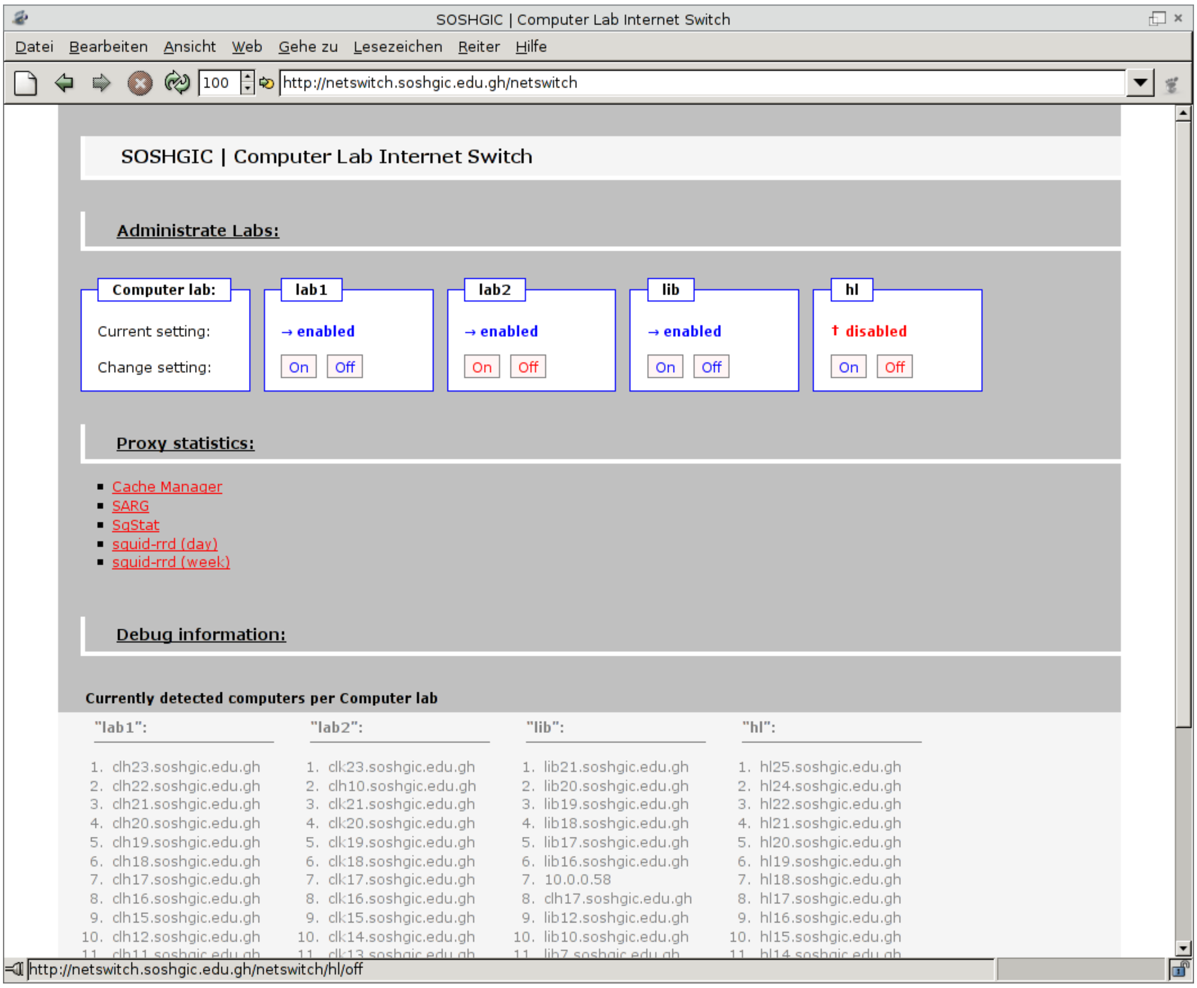
- `/vmlinuz`, which is the compiled kernel for the boot disk. It is configured to not require any external modules.
- `/cdimage/isolinux.bin` and `/cdimage/isolinux.cfg` are part of the isolinux boot loader[6] which enable to start the linux kernel when the computer boots from this CD.

To finally put everything together, use `mkisofs -o cd-image.iso -b isolinux/isolinux.bin -no-emul-boot -boot-load-size 4 -boot-info-table cdimage/` and burn the resulting file `cd-image.iso` to a CD-R.

[6] `http://syslinux.zytor.com/iso.php`

Listing 31: Bootdisk files

```
/rt61.ko
/dev
/dev/input
/dev/rd
/dev/ida
/dev/ataraid
/dev/cciss
/dev/i2o
/linuxrc
/bin
/bin/busybox
/bin/reboot
/bin/chvt
/bin/insmod
/bin/lsmod
/bin/udhcpc
/bin/iwconfig
/bin/bash
/bin/openvt
/bin/iwlist
/bin/lspci
/bin/portmap
/bin/pivot_root
/bin/rpc.lockd
/bin/rpc.statd
/bin/whiptail
/bin/grub
/bin/sfdisk
/lib
/lib/libc.so.6
/lib/ld-linux.so.2
/lib/libiw.so.27
/lib/libncurses.so.5
/lib/libdl.so.2
/lib/libm-2.3.2.so
/lib/libm.so.6
/lib/libconsole.so.0
/lib/libctutils.so.0
/lib/libpci.so.2
/lib/libwrap.so.0
/lib/libnsl.so.1
/lib/libnewt.so.0.51
/lib/libslang.so.1-UTF8
/lib/libpopt.so.0
/proc
/etc
/etc/udhcpc
/etc/udhcpc/default.deconfig
/etc/udhcpc/default.renew
/etc/udhcpc/default.script
/etc/udhcpc/default.nak
/etc/udhcpc/default.bound
/etc/hosts
/etc/nsswitch.conf
/sys
/tmp
/mnt
```


SOSHGIC | Computer Lab Internet Switch
Datei Bearbeiten Ansicht Web Gehe zu Lesezeichen Reiter Hilfe
100
http://netswitch.soshgic.edu.gh/netswitch
http://netswitch.soshgic.edu.gh/netswitch/hl/off

SOSHGIC | Computer Lab Internet Switch

Administrate Labs:

Computer lab:	lab1	lab2	lib	hl
Current setting:	→ enabled	→ enabled	→ enabled	↑ disabled
Change setting:	On Off	On Off	On Off	On Off

Proxy statistics:

- Cache Manager
- SARG
- SqStat
- squid-rrd (day)
- squid-rrd (week)

Debug information:

Currently detected computers per Computer lab

"lab1":

1. clh23.soshgic.edu.gh
2. clh22.soshgic.edu.gh
3. clh21.soshgic.edu.gh
4. clh20.soshgic.edu.gh
5. clh19.soshgic.edu.gh
6. clh18.soshgic.edu.gh
7. clh17.soshgic.edu.gh
8. clh16.soshgic.edu.gh
9. clh15.soshgic.edu.gh
10. clh12.soshgic.edu.gh
11. clh11.soshgic.edu.gh

"lab2":

1. clk23.soshgic.edu.gh
2. clh10.soshgic.edu.gh
3. clk21.soshgic.edu.gh
4. clk20.soshgic.edu.gh
5. clk19.soshgic.edu.gh
6. clk18.soshgic.edu.gh
7. clk17.soshgic.edu.gh
8. clk16.soshgic.edu.gh
9. clk15.soshgic.edu.gh
10. clk14.soshgic.edu.gh
11. clk13.soshgic.edu.gh

"lib":

1. lib21.soshgic.edu.gh
2. lib20.soshgic.edu.gh
3. lib19.soshgic.edu.gh
4. lib18.soshgic.edu.gh
5. lib17.soshgic.edu.gh
6. lib16.soshgic.edu.gh
7. 10.0.0.58
8. clh17.soshgic.edu.gh
9. lib12.soshgic.edu.gh
10. lib10.soshgic.edu.gh
11. lib7.soshgic.edu.gh

"hl":

1. hl25.soshgic.edu.gh
2. hl24.soshgic.edu.gh
3. hl22.soshgic.edu.gh
4. hl21.soshgic.edu.gh
5. hl20.soshgic.edu.gh
6. hl19.soshgic.edu.gh
7. hl18.soshgic.edu.gh
8. hl17.soshgic.edu.gh
9. hl16.soshgic.edu.gh
10. hl15.soshgic.edu.gh
11. hl14.soshgic.edu.gh

Figure 2: Self-developed internet connection control

12. Netswitch

Independently of the Linux client setup, I created a “internet switch” for the computer labs, dubbed “netswitch”. The aim is to allow the teacher to turn of access to the internet during classes, if it is bad for the student’s attention.

12.1. Network integration

This is implemented as another gateway, running on Debian, which is then made the default gateway using DHCP. Normally, it will simply forward any packages to the gateway at 10.0.0.204. If the teacher turns off the internet using a simple web front end, though, packages from that lab will be rejected. As the next hop is on the same IP subnet, the traffic coming from the internet will go directly to the client, and local traffic is not affected either.

12.2. Implementation

The software consists of two parts. The first part, /opt/netswitch/resolve.pl is a Perl script that will look up all lab computer host names in DNS, and add iptables rules to put them in the queue of their lab. This is needed as IP addresses are given out dynamically, so there is no fixed IP-to-lab mapping. This script is run every five minutes from cron.

The second part, /opt/netswitch/cgi.pl, provides the web front end, displaying the current settings of the package filter tables, and allowing the teacher to add or remove a "drop all" rule to any of the hostel lab queues. A screenshot is printed in figure 2.

The web front end is password protected, using the same ldap settings as occ, see section 6.3. The relevant lines to /etc/apache2/sites-available/default are, added to the VirtualHost stanza:

```
ScriptAlias /netswitch  /opt/netswitch/cgi.pl
RewriteEngine on
RewriteRule ^/$ /netswitch [R]
<Directory /opt/netswitch/>
require group CN=Department - IT,OU=Soshgic Groups,DC=soshgic,DC=edu,DC=gh
</Directory>
```

It was noticed that students would skip lunch to surf the internet. To stop them I have set up a cron job that will prevent it, by turning internet off on weekday from 13:15 to 13:50. It uses a small script in /opt/netswitch/all.pl.

The entries to cron /etc/crontab are:

```
*/5 *  * * *           root /opt/netswitch/resolve.pl 2>/dev/null
15  13 * * 1,2,3,4,5   root /opt/netswitch/all.pl off >/dev/null
50  13 * * 1,2,3,4,5   root /opt/netswitch/all.pl on >/dev/null
```

The code of these files are in listings 35, 36 and 37.

13. Caching proxy

13.1. Setup

Also on the netswitch computer, but otherwise dependent, runs a caching proxy based on squid. The setup is mostly the default Debian setup, with these changes in /etc/squid/squid.conf:

1. Some settings were tweaked to suit the hardware. The proxy can use 6 gigabytes of hard disk space (more would be desirable)

   ```
   maximum_object_size 32 MB
   cache_dir ufs /var/spool/squid 8000 16 256
   ```

2. A simple filtering system is set up. One list of domain names is always blocked, the other one only during daytime (7:30 to 17:30 on weekdays):

   ```
   acl blocked dstdomain "/opt/blocklists/blocked.list"
   deny_info SOSHGIC_INAPPROPRIATE blocked
   acl blocked_day dstdomain "/opt/blocklists/blocked_day.list"
   ```

```
deny_info SOSHGIC_DAYTIME blocked_day
acl daytime time MTWHF 7:30-17:30
http_access deny blocked
http_access deny daytime blocked_day
```

3. Access is granted to the local network:

```
acl our_networks src 10.0.0.0/24
http_access allow our_networks
http_access allow localhost
```

4. To allow squid to act as a transparent proxy, meaning that the clients do not have to select it as the proxy, these settings are required:

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

5. A custom error messages directory is selected:

```
error_directory /opt/blocklists/errors/
```

6. Certain sites are not blocked, but to make sure that they do not use all the bandwidth, they are limited to 10kbytes/s, with an initial burst of 30kbytes/s:

```
acl slowsites dstdomain "/opt/blocklists/slowdown.list"
delay_pools 1
delay_class 1 1
delay_access 1 allow slowsites
delay_access 1 deny all
delay_parameters 1 10000/30000
```

The three blocking lists /opt/blocklists/blocked.list, /opt/blocklists/blocked_day.list and /opt/blocklists/slowdown.list are domain match lists. To block a domain like example.com, including all subdomains, put .example.com on a line of it's own in that file. The leading dot is important.

To actually intercept the HTTP connections, and to do so after a restart, the init script in listing 32 is used.

13.2. Monitoring

A caching proxy collects a lot of information on the users, and to easily read this data, various programs for logging are deployed.

13.2.1. sqstat

To view the currently open connection on the proxy, sqstat[7] is installed in /var/www/sqstat, and IP resolving is activated in /var/www/sqstat/config.inc.php.

[7] http://samm.kiev.ua/sqstat/

Listing 32: **File:** /etc/init.d/intercept

```
#! /bin/sh

set -e

PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

case "$1" in
  start)
        echo -n "Transparently intercepting HTTP traffic"
        iptables -t nat -A PREROUTING -i eth0 -p tcp -d ! 10.0.0.0/24 \
                                --dport 80 -j REDIRECT --to-port 3128
        echo "."
        ;;
  stop)
        echo -n "Letting HTTP traffic pass"
        iptables -t nat -F PREROUTING
        echo "."
        ;;
  restart|force-reload)
        echo -n "Re-Setting transparent interception of HTTP traffic"
        iptables -t nat -F PREROUTING
        iptables -t nat -A PREROUTING -i eth0 -p tcp -d ! 10.0.0.0/24 \
                                --dport 80 -j REDIRECT --to-port 3128
        echo "."
        ;;
  *)
        echo "Usage: $SCRIPTNAME {start|stop|restart|force-reload}" >&2
        exit 1
        ;;
esac

exit 0
```

13.2.2. sarg

The reporting tool `sarg` comes with Debian. In `/etc/squid/sarg.conf`, both `resolve-ip` and `user-ip` were set to `yes`. The generation of the daily report happens at night, so `/etc/squid/sarg-maint.sh` is symlinked to `/etc/cron.daily`.

13.2.3. squid-rrd

For nice graphs of various diagnosis values, squid-rrd[8] is used. Some minor modifications to display the current rate of transfer were made, as well as a second view with weekly data.

13.2.4. Security

Only IT staff should be able to see these reports, to the following configuration was added to `/etc/apache2/sites-available/default`:

```
<Location /cgi-bin/cachemgr.cgi>
  require group CN=Department - IT,OU=Soshgic Groups,DC=soshgic,DC=edu,DC=gh
</Location>

<Location /sqstat>
  require group CN=Department - IT,OU=Soshgic Groups,DC=soshgic,DC=edu,DC=gh
</Location>

<Location /squid-reports>
  require group CN=Department - IT,OU=Soshgic Groups,DC=soshgic,DC=edu,DC=gh
</Location>

<Directory /var/www/squid-rrd>
  AllowOverride Indexes
  Options ExecCGI
  AddHandler cgi-script .cgi
  require group CN=Department - IT,OU=Soshgic Groups,DC=soshgic,DC=edu,DC=gh
</Directory>
```

[8] `http://www.squid-cache.org/~wessels/squid-rrd/`

A. Long Listings

Listing 33: **File:** $FAI/hooks/partition.DEFAULT

```
#!/bin/bash

fail () {
    sleep 1000000000
    faireboot
}

echo "NTFS resizer for SOSHGIC"
echo "Loosely based on http://svn.debian.org/wsvn/d-i/trunk/packages/partman↵
   /partman-partitioning/resize.sh"

echo
echo "Trying to find out the state of affairs..."

if [ -e /sys/block/sda ]
then
        echo "SCSI (SATA?) device found, using that"
        disk=/dev/sda
else
        echo "SCSI (SATA?) device not found, sticking to /dev/hda"
        disk=/dev/hda
fi

device=${disk}1

# Blocksize 1024. Is that always the case?
cursize=$(sfdisk -s $device)
cursize=$((cursize * 1024))
disksize=$(sfdisk -s $disk)
disksize=$((disksize * 1024))

echo "Detected that the NTFS partition uses $cursize of the $disksize disk"

if [ -z "$cursize" -o -z "$disksize" ]
then
echo "Could not detect some of the information"
fail
fi

#if [ $((cursize)) -gt $((28*1000*1000*1000)) ]
#then
#       echo "Windows partition takes more than 28GB, might cause boot ↵
   problems, resizing"
```

```
#elif [ $((disksize - cursize)) -gt $((8*1000*1000*1000)) ]
if [ $((disksize - cursize)) -gt $((8*1000*1000*1000)) ]
then
        echo "We have more than 8GB free, nothing to resize"
        exit 0
fi

echo "We need resizing..."

if file -s $device | grep -q NTFS
then

   minsize=$(ntfsresize -f -i $device \
        | grep ’^You might resize at’ \
        | sed ’s/^You might resize at \([0-9]*\) bytes.*/\1/’ \
        | grep ’^[0-9]*$’)

   if [ $((disksize - minsize)) -lt $((15*1000*1000*1000)) ]
   then
       echo "We don’t have 15GB free on the disk, sorry"
       fail
   fi

#   if [ $((minsize)) -gt $((23*1000*1000*1000)) ]
#   then
#       echo "Partition larger than 23GB, sorry"
#       fail
#   fi

   target=$((disksize - 10*1000*1000*1000))
#   if [ $((target)) -gt $((28*1000*1000*1000)) ]
#   then
#       target=$((27*1000*1000*1000))
#   fi

   echo "The current file system uses $minsize bytes, which leaves us with $↵
       ((disksize-minsize)) bytes, of which we take $((target)) bytes."

   echo "Figuring out disk geometry"
   heads=$(sfdisk -g $disk | awk ’{print $4}’)
   secperhead=$(sfdisk -g $disk | awk ’{print $6}’)
   cylsize=$((heads * secperhead)) # in sectors
   #heads=$(fdisk -l $disk | grep heads | awk ’{print $1}’ )
   #secperhead=$(fdisk -l $disk | grep heads | awk ’{print $3}’ )
   #cylsize=$((heads * secperhead)) # in sectors
   echo "Heads: $heads Sec/Head: $secperhead Cylsize: $cylsize"
   echo "Waiting for a while, abort with Ctrl-C"
   sleep 10 || fail


   echo "Resizing NTFS partition to $target bytes"
   echo y | ntfsresize --force --size "$target" "$device"
   echo "Waiting for a while, abort with Ctrl-C"
   sleep 10 || fail

   # Sectors are 512 bytes
   size=$((target/512 + 200)) # 200MB cyl saftey padding
   start=$(sfdisk -l -uS|fgrep $device|awk '{print $3}')
   end=$(( (start + size)/cylsize * cylsize + cylsize)) # align
   size=$((end-start))
   echo "Repartitioning using this table:"
   echo ", $size," | sfdisk $disk -uS -N1 -f -H$heads
   echo ", $size, | sfdisk $disk -uS -N1 -f -H$heads"
   echo "Waiting for a while"
   sleep 10 || fail
   echo "Re-Resizing partition"
   echo y | ntfsresize --force "$device"
   echo "Waiting for a while"
   sleep 10 || fail

else
    echo "No NTFS on $device, aborting"
    fail
fi
```

Listing 34: **File:** $FAI/files/boot/grub/menu.lst/GRUB

```
timeout 10
default 0
color blue/cyan black/white

splashimage=#GROOT#/boot/grub/splash.xpm.gz

password --md5 $1$iIVMc1$aDf5aj7YWb3vb2py0C4.Q.

# Load windows on the first partition
title           Microsoft Windows XP Professional
root            (hd0,0)
savedefault
makeactive
chainloader     +1

### BEGIN AUTOMAGIC KERNELS LIST
## lines between the AUTOMAGIC KERNELS LIST markers will be modified
## by the debian update-grub script except for the default options below
## DO NOT UNCOMMENT THEM, Just edit them to your needs

## ## Start Default Options ##
## default kernel options
## default kernel options for automagic boot options
## If you want special options for specifiv kernels use kopt_x_y_z
## where x.y.z is kernel version. Minor versions can be omitted.
## e.g. kopt=root=/dev/hda1 ro
# kopt=root=#ROOT_PARTITION# ro

## default grub root device
## e.g. groot=(hd0,0)
# groot=#GROOT#

## should update-grub create alternative automagic boot options
## e.g. alternative=true
##      alternative=false
# alternative=false

## should update-grub lock alternative automagic boot options
## e.g. lockalternative=true
##      lockalternative=false
# lockalternative=false

## altoption boot targets option
## multiple altoptions lines are allowed
## e.g. altoptions=(extra menu suffix) extra boot options
##      altoptions=(recovery mode) single
# altoptions=(recovery mode) single

## controls how many kernels should be put into the menu.lst
## only counts the first occurence of a kernel, not the
## alternative kernel options
## e.g. howmany=all
##      howmany=7
# howmany=all

## should update-grub create memtest86 boot option
## e.g. memtest86=true
##      memtest86=false
# memtest86=true

## ## End Default Options ##

### END DEBIAN AUTOMAGIC KERNELS LIST
```

Listing 35: **File:** /opt/netswitch/resolve.pl

```
#!/usr/bin/perl

#    2006 Joachim Breitner
# Written for the SOS Hermann Gmeiner International College, Tema, Ghana

use warnings;
use strict;

use Socket;

my $domain = "soshgic.edu.gh";

my %labs = (
        lab1 => [
                (map "clh${_}", 1..23),
                (map "clh${_}l", 1..23),
        ],
        lab2 => [
                (map "clk${_}", 1..23),
                (map "clk${_}l", 1..23),
        ],
        lib => [
                (map "lib${_}", 7..21),
                (map "lib${_}l", 7..21),
        ],
        hl => [
                (map "hl${_}", 1..25),
                (map "hl${_}l", 1..25),
        ],
);

sub run ($) {
        my $cmd = shift;
        print STDERR "Running $cmd\n";
        system($cmd);
}

run "iptables -F FORWARD";
run "iptables -F INPUT";

for my $lab (keys %labs) {
        run "iptables -N $lab";
        for my $host (@{$labs{$lab}}) {
                my $hostname =  $host.".".$domain;
                print STDERR "$hostname: ";
                my $networkip = inet_aton($hostname);
                my $ip= inet_ntoa($networkip) if defined $networkip;
                if ($ip) {
                        my $dnsname = gethostbyaddr(inet_aton($ip), AF_INET);
                        if ($dnsname && $dnsname ne $hostname) {
                                print STDERR "resolves to $ip, held by $dnsname\n";
                        } else {
                                run "iptables -A FORWARD -s $ip -j $lab";
                                run "iptables -A INPUT -s $ip -p tcp --dport 3128 -j $lab";
                        }
                } else {
                        print STDERR "not found\n"
                }
        }

}
```

Listing 36: **File:** /opt/netswitch/cgi.pl

```
#!/usr/bin/perl

#    2006 Joachim Breitner
# Written for the SOS Hermann Gmeiner International College, Tema, Ghana

use warnings;
use strict;

use CGI;
use CGI::Carp qw(fatalsToBrowser);


my $cgi = new CGI;

my $dir = "/opt/netswitch/";
my @labs = qw/lab1 lab2 lib hl/;

if ($ENV{PATH_INFO}) {
        unless ($ENV{PATH_INFO} =~ m!/(.*)/(on|off)!) {
                die "Not understandable command $ENV{PATH_INFO}"
        }
        my $lab = $1;
        my $cmd = $2;

        unless (grep /^$lab$/,@labs) {
                die "Lab $lab unknown\n";
        }

        system("sudo /sbin/iptables -F $lab");
        if ($cmd eq 'off') {
                system("sudo /sbin/iptables -A $lab -j REJECT");
        }

        print $cgi->redirect("/netswitch");


        exit;
}

print $cgi->header(-charset=>'UTF-8');

print <<'END';
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
    <head>
        <title>SOSHGIC | Computer Lab Internet Switch</title>
        <meta http-equiv="Content-Type" content="text/html; charset=utf-8" ↵
            />
<style type="text/css">
/*<![CDATA[*/

*       {
        margin: 0;
        padding: 0;
        font-family: Verdana, sans-serif;
        font-size: 12px;
        line-height: 140%;
        }

h1, h2, h3      { font-weight: bold; margin-top: 2em; padding: 0.4em 2em;}
h3                      { font-size: 12px; }
h2                      { font-size: 14px;  text-decoration: underline; }
h1                      { font-size: 16px; margin-top: 1em; background-color↵
   : #f5f5f5;}


html, body {
        height: 101%;
        }

#body {
        height: 101%;
        margin: 0 4em;
        padding: 1em 0 1em 0;
        background-color: silver;
        }

h1, h2, h3, #fsclab, table, #debug h4, #debug ol {
        margin-left: 20px;
        }

h1, h2 {
        clear: both;
        border-width: 0 0 4px 4px;
        border-style: solid;
        border-color: white;
        }

.hrwrp {
        height: 1px;
        clear: both;
        }
.hrwrp hr {
        display: none;
        }

form p {
        line-height: 240%;
        }

fieldset,
legend  {
        background-color: white;
}
fieldset {
        float: left;
        position: relative;
        padding: 1em 1.2em 0.4em 1.2em;
        width: 10em;
        border: 1px solid blue;
        margin: 2em 1em 0 0;
        }
legend {
        position: absolute;
        top: -.8em;
        left: .5em;
        padding: 0.1em 1em;
        font-weight: bold;
        border: 1px solid blue;
        }
fieldset p {
        display: block;
        height: 2.6em;
        }
fieldset a {
        text-align: center;
        border: 1px solid gray;
        padding: 0.1em 0.5em;
        background-color: #fff5f5;
        margin-right: 0.4em;
        text-decoration: none;
        }
fieldset a:hover {
        background-color: blue;
        color: white;
        border-color: white;
        font-weight: bold;
        }
.enabled,
.disabled {
        font-weight: bold;
        }
.enabled {
        color: blue;
        }
.disabled {
        color: red;
        }

ul {
        list-style-type: square;
        margin: 1em 0 1em 4em ;
        }



#debug {
        float: left;
        width: 100%;
        }
#debug h3 {
        margin-left: 0;
        }
#debug div {
        background-color: #f5f5f5;
        width: 100%;
        float: left;
        }
#debug div div {
        float: left;
        width: 15em;
        padding: 0 0 0 1em;
        }
#debug h4 {
        padding: 0.4em 0;
        border-bottom: 1px solid gray;
        }
#debug ol, #debug h4 {
        color: gray;
        }
#debug ol {
        margin:1em 1em 1em 3em;
        }

#imprint {
        background-color: silver;
        clear: both;
        padding: 1em;
        }
#imprint p {
        text-align: right;
        }

/*]]>*/
</style>
</head>
<body>
<!-- body /-->
<div id="body">
<h1>SOSHGIC | Computer Lab Internet Switch</h1>

<h2>Administrate Labs:</h2>

<form action="#">
        <fieldset id="fsclab">
        <legend>Computer lab:</legend>
        <p>Current setting:</p>
        <p>Change setting:</p>
        </fieldset>
END



my %current;
```

```
$current{$_} = [] for @labs;
my @forward = ‘sudo /sbin/iptables -L FORWARD‘;
for (@forward) {
        my @words = split;
        next if $words[0] eq "Chain";
        next if $words[0] eq "target";
        unshift @{$current{$words[0]}},$words[3];
}



for my $lab (@labs) {
        my $setting = ‘sudo /sbin/iptables -L $lab‘;
        print <<"";
        <fieldset>
        <legend>$lab</legend>

        if ($setting =˜ /REJECT/) {
                print '<p><span class="disabled">&dagger; disabled</span></p↵
                    >';
        } else {
                print '<p><span class="enabled">&rarr; enabled</span></p>';
        }
        print <<"";
        <p>
        <a href="http://netswitch.soshgic.edu.gh/netswitch/$lab/on">On</a>
        <a href="http://netswitch.soshgic.edu.gh/netswitch/$lab/off">Off</a
        </p>
        </fieldset>

}
print <<'END';
</form>

<div class="hrwrp"><hr /></div>

<h2>Proxy statistics:</h2>

<ul>
        <li><a href="http://netswitch.soshgic.edu.gh/cgi-bin/cachemgr.cgi">↵
           Cache Manager</a></li>
        <li><a href="http://netswitch.soshgic.edu.gh/squid-reports/">SARG</a↵
           ></li>
        <li><a href="http://netswitch.soshgic.edu.gh/sqstat/">SqStat</a></li↵
           >
        <li><a href="http://netswitch.soshgic.edu.gh/squid-rrd/1day.cgi">↵
           squid-rrd (day)</a></li>
        <li><a href="http://netswitch.soshgic.edu.gh/squid-rrd/1week.cgi">↵
           squid-rrd (week)</a></li>
</ul>



<div id="debug">

<h2>Debug information:</h2>

<h3>Currently detected computers per Computer lab</h3>
<div id="debugclabs">
END

for my $lab (@labs) {
        print "<div><h4>   $lab   :</h4>\n";

        print "<ol>\n";
        print "<li>$_</li>\n" for @{$current{$lab}};
        print "</ol>\n";

        print '</div>';
}

print <<'END';
</div>
<div class="hrwrp"><hr /></div>
</div>

<div id="imprint">
        <p>Programmed by <a href="mailto:mail@joachim-breitner.de">Joachim ↵
           Breitner</a> | Design by <a href="mailto:tombreit@gmail.com">↵
           Thomas Breitner</a></p>
</div>

</div>
</body>
</html
END
```

Listing 37: **File:** /opt/netswitch/all.pl

```
#!/usr/bin/perl

#    2006 Joachim Breitner
# Written for the SOS Hermann Gmeiner International College, Tema, Ghana

use warnings;
use strict;
use Socket;

my $cmd = $ARGV[0];

unless ($cmd eq "on" or $cmd eq "off") {
        print "Usage: $0 on|off\n";
        exit 1;
}

my @labs = qw/lab1 lab2 lib hl/;

for my $lab (@labs) {
        print "Turning $lab $cmd\n";
        system("/sbin/iptables -F $lab");
        system("/sbin/iptables -A $lab -j REJECT") if ($cmd eq 'off');
}
```

Listing 38: **File:** `/linuxrc`

```
#!/bin/sh


function start_shell {
  echo "You got a shell.."
  openvt -c 1 -f -s -w /bin/bash
}

function  WHIPTAIL {
   whiptail --backtitle "SOSHGIC Installation CD" "$@"
}

function setup_network {
  echo "Starting lo..."
  ifconfig lo up 127.0.0.1

  if lspci -n | grep -q 1814:
  then

    echo "Selecting wireless"
    export WIRELESS=yes

    echo "Loading kernel module..."
    insmod rt61.ko

    #echo "Ok, we should have a ra0 device... see for yourself"
    #ifconfig -a

    ifconfig ra0 up
    APS=$(iwlist ra0 scan | grep ESSID | sed 's/^.*ESSID:"\(.*\)".*$/\1/')

    #WHIPTAIL --nocancel --menu "Please select the WLAN." 20 50 10 \
    #  $APS \
    #  2>/tmp/answer
    #export ESSID=$(</tmp/answer)

    for choice in "lab-2" LIB Hostel_Lab
    do
      if echo $APS | grep -q $choice
      then
        export ESSID=$choice
        break
      fi
    done

    if [ -z "$ESSID" ]
    then
      WHIPTAIL --msgbox "No WLAN detected!" 20 50
      return 1
    fi

    WEP_APS=LIB
    if echo $WEP_APS | grep -q "$ESSID"
    then
      echo "Network is WEP encrypted"
      export WEP_KEY=be5018fc8d
    fi

    echo "Configuring wireless..."
    iwconfig ra0 essid $ESSID
    [ -n "$WEP_KEY" ] && iwconfig ra0 key $WIRELESS_KEY
    udhcpc -n -i ra0

  else
    echo "Selecting wired"
    udhcpc -n -i eth0
  fi

  echo "Testing network"
  ping -c 3 10.0.0.19
}

function setup_chroot () {
    portmap
    mount -o ro,tcp 10.0.0.19:/usr/lib/fai/nfsroot /mnt
    mount -o bind /dev /mnt/dev
    mount -o bind /sys /mnt/sys
    mount -o bind /proc /mnt/proc
    killall portmap
    echo "nfs-chroot auf /mnt eingerichtet"
}


echo
echo
echo "SOSHGIC Custom Boot disk"
echo

echo "Nice, I have booted!"

echo "Mounting /proc..."
mount proc -t proc  /proc
mount none -t sysfs /sys

echo "Making tmpfs..."
mount none -t tmpfs -o size=16M /tmp


# SOSHGIC-Menu
backtitle="SOSHGI Installation CD"
WHIPTAIL --nocancel --menu "What do you want to do?" 20 50 10 \
  install "Install Linux on this machine." \
  shell  "Go to the command line." \
  netshell  "Go to the command line. (w/ network)" \
  chroot  "Go to the command line. (w/ chroot)" \
  fixgrub "Fix GRUB in the MBR." \
  fixtbl "Fix partition table (255 head problem)." \
  2>/tmp/answer

case "$(< /tmp/answer)" in
fixtbl)
  sfdisk -d /dev/hda | sfdisk -H255 -f /dev/hda
  start_shell
;;
fixgrub)
  grub --batch <<-END
        root (hd0,1)
        setup (hd0)
END
  start_shell
;;
install)
```

```
  # Asking input
  WHIPTAIL --nocancel --inputbox  "What is the hostname?" 20 50 2> /tmp/↵
      answer
  export HOST=$(< /tmp/answer)
  export FAI_ACTION=install
  export FAI_FLAGS=createvt

  if setup_network
  then
    echo "Mounting nfs-root"
    portmap
    mount -o ro,tcp 10.0.0.19:/usr/lib/fai/nfsroot /mnt
    killall portmap


    # Starting FAI
    cd /mnt
    #pivot_root . oldroot
    exec chroot . /etc/init.d/rcS
  else
    whiptail --backtitle "$backtitle" --msgbox "Network failed, rebooting" ↵
        20 50
  fi
;;
shell)
  start_shell
;;
netshell)
  setup_network
  start_shell
;;
chroot)
  setup_network
  setup_chroot
  start_shell
;;
*)
  echo Bad Answer $(</tmp/answer)
  echo "Rebooting..."
esac

echo "Rebooting...."
reboot  -f -i -d
```

B. Index of all Listings

C. List of Figures

D. General Index