The Director

Installation and Configuration

Phinex Informatik AG

Thomas Aeby

Inhalt

Introduction

This document tries to explain step-by-step how you get a Director server (aka. Director management machine) up. It will not explain how you actually install the Director on computers you just want to manage, except for this one mention: just install the „sfidirector"-Package on it and read the Director User manual.

While the Director will work on a number of systems this documentation focusses on Linux systems. Expect the Director to install on other Unix- or Unix-like systems in a similar, but not necessarily identical way.

Prerequisits

In order to successfully install and run the Director server you should first install the following software:

- A Web server. Installation on top of Apache is directly supported, but any other web server that is able to run CGIs will do, too.
- Perl CGI accelerator. Both SpeedyCGI and PPerl are supported. If you wish you can also give mod_perl a chance, but note that you then have to run your web server (or an instance of it) under a special account.
- Perl. The web frontend is written in Perl. In order to ease installation it will not require any special perl libraries except for the CGI libraries which are installed by default on pretty any platform, anyway.
- OpenLDAP. You will want to store management information in some LDAP accessible database. OpenLDAP is an easy to setup one. The Director directly supports SUN's directory server as well (which is more or less identical to the Netscape directory server).
- Name service cache daemon. Due to how the Director binary packages are built you will want to run the nscd service on the Director server machine. Otherwise the Director will not respect your name service configuration (nsswitch.conf) – at least not on Linux systems.

Note: None of this is absolutely necessary. Without Perl or a Web server you will not be able to use the web based frontend, without a CGI accelerator your web frontend will be rather lame, without an LDAP database you can still use the built-in, not very performant file-based database. Nevertheless, the above software set is strongly recommended.

The Director server will use up rather much memory, so you should install it on a machine with plenty of RAM (somewhere around 256MB is ok for a start).

Prerequisite Software installation on Debian

You will want to run

```
% apt-get install apache apache-ssl slapd nscd speedy-cgi-perl
```

Prerequisite Software installation on other Systems

Please consult your software installation tool of choice about the available packages installing the software listed above. The following list of commonly used package names might help:

- Web server: apache, httpd
- CGI Accelerator: SpeedyCGI, speedy-cgi-perl, Pperl
- OpenLDAP: openldap, slapd
- Name service cache daemon: nscd

Installation

Note: You should install the software packages listed in the Prerequisits chapter before you proceed. The available binary packages will usually not force installation via dependencies (since you actually can use the packages without the prerequisits), but will do some effort to configure themselves during installation depending on your environment, e.g. the web frontend will select an available CGI accelerator as the default CGI engine.

Installation from binary packages

Binary packages are currently available for Debian and RPM based systems. Only packages for the x86 architecture are published, though the Director is known to build at least on PowerPC and Sparc architectures as well.

If there are no binary packages for your system then please have a look at the secion Installation from Source.

Installation on Debian system

The most easiest way of getting Director installed on your machin is to first add a line like

```
deb http://software.graeff.com/debian-graeff sarge main
```

to your /etc/apt/sources.list file. Then install the necessary packages via APT:

```
% apt-get update
% apt-get install bigswaf bigclerk sfidirector-server sfidirector-init
```

That's it.

That will work on sarge and sid, we have not tested it on woody. Most probably, the (statically linked) sfidirector packages will work well on woody while the web frontend might fail. Please send us some feedback if you try this out on woody.

Installation on RPM-based systems

If you install the Director on some RPM based systems like SuSE, RedHat or Mandrake, you will have to download the packages

```
bigswaf BigClerk sfidirector sfidirector-server sfidirector-init
```

from the download sites at

```
http://sf.net/project/showfiles.php?group_id=14120&package_id=146922
http://sf.net/project/showfiles.php?group_id=11622
```

Then install them via RPM like

```
% rpm -ivh sfidirector*.rpm BigClerk*.rpm bigswaf*.rpm
```

Installation from Source

If there is no binary package available for your system or you prefer installing software from source you will need to get the source tarballs for

```
bigswaf bigclerk sfidirector
```

from the download sites at

```
http://sf.net/project/showfiles.php?group_id=14120&package_id=146922
http://sf.net/project/showfiles.php?group_id=11622
```

and install them.

Installing the Director

In order to be able to build the Director from source you will need the GCC compiler including Java (GCJ) support. GCC version >= 3.3.2 will do.

Unpack the sfidirector tarball and change to the directory

```
directorNG/director
```

In most cases you can then install the Director via the usual configure/make cycle, e.g.

```
% ./configure --prefix=/usr/local/sfidirector
% make
% make install
```

See

```
% ./configure --help
```

for a short documentation of the available options. Notably,

```
% ./configure --enable-fhs
```

will configure the package for a File-Hierarchy-Standard compliant installation. You can also see how the binary packages configure the package via

```
% grep /configure rpm/sfidirector.spec.in debian/rules
```

The frontend interface – which makes part of the Director – will expect to run under a special user account called

```
frinterface
```

Please create this user (needs a valid shell, but password locked is ok). Also, do a

```
% chown -R frinterface prefix/var
```

or a

```
% chown -R frinterface /var/lib/sfidirector
```

if you chose to install the FHS version.

The make process will install init scripts called

```
sfidirector / frinterface
```

in your system's init directory. Please check if they are matching your policy and enable them for starting both services on boot.

Note: GCC up to and including 4.0.0 contain a bug causing spurious hangs of the director on some systems. If you are hit by this problem please build GCC/GCJ using the configure-time option --disable-hash-synchronization. GCC 4.0.0 was the latest version at the time this remark was written – so it might apply to later versions too.

Installing BigSWAF

BigSWAF will require its own user account it will run under. Usually this is called

```
bigsis
```

You have to create this account manually.

Untar the BigSWAF tarball. Then enter the bigswaf directory and install bigswaf via the usual configure/make cycle:

```
% ./configure --prefix=/usr/local/bigswaf
% make
% make install
```

See

```
% ./configure --help
```

for a list of available options or do a

```
% grep /configure etc/*.spec.in debian/rules
```

in order to see examples. Again, BigSWAF understands the option "--enable-fhs".

Installing BigClerk

BigClerk requires BigSWAF to be installed first. Untar the BigClerk-tarball and install Big-Clerk similar to the instructions for BigSWAF.

Note: BigClerk needs to be installed at the same location as BigSWAF, thus using the same "--prefix" or using "--enable-fhs".

Configuring OpenLDAP and the Director

In this chapter we will set up an initial LDAP database using the OpenLDAP server and make the Director use it.

Configuring OpenLDAP

OpenLDAP is configured via its configuration file called slapd.conf. You will probably find it installed as one of

```
/etc/openldap/slapd.conf
/etc/ldap/slapd.conf
```

In the directory

```
/usr/share/doc/sfidirector-server/examples/openldap
```

(have a look at doc/examples in the source tree if you installed from source) you will find both an example slapd.conf as well as a file

```
initial_db.ldif
```

with initial database entries we need later. The example OpenLDAP-Configuration uses a database suffix of dc=yourcompany,dc=example. This may reflect your DNS domain, the example would match a DNS domain of yourcompany.example. Please change the suffix as you like – and do not forget to change it in all the examples below, too – or just leave it as it is if you do not exactly understand the consequences of this.

Our slapd.conf looks like

```
# This is the main slapd configuration file. See slapd.conf(5) for more
# info on the configuration options.

########################################################################
# Global Directives:

# Schema and objectClass definitions
include         /etc/ldap/schema/core.schema
include         /etc/ldap/schema/cosine.schema
include         /etc/ldap/schema/nis.schema
include         /etc/ldap/schema/misc.schema
include         /etc/ldap/schema/inetorgperson.schema
include         /usr/share/sfidirector/etc/director.schema
include         /usr/share/sfidirector/etc/extraschema/samba.schema

# Schema check allows for forcing entries to
# match schemas for their objectClasses's
schemacheck     on

# Where the pid file is put. The init.d script
# will not stop the server if you change this.
pidfile         /var/run/slapd/slapd.pid

# Read slapd.conf(5) for possible values
loglevel        0

# Where the dynamically loaded modules are stored
modulepath      /usr/lib/ldap
#moduleload     back_bdb
moduleload      back_ldbm
```

```
# we use the LDBM-Backend
backend         ldbm
database        ldbm

# The base of your directory in database #1
suffix          "dc=yourcompany,dc=example"

# Where the database file are physically stored for database #1
directory       "/var/ldap/"

# Save the time that the entry gets modified, for database #1
lastmod         on

# protect passwords
access to attr=userpassword,ntpassword,lmpassword,sambantpassword,\
      sambalmpassword by self write
  by
   dn="uid=root,ou=System  Users,ou=people,dc=yourcompany,dc=example"
   write
  by
   dn="uid=ldaproot,ou=System Users,ou=people,dc=yourcompany,dc=example"
   write
  by * compare

access to dn="dc=yourcompany,dc=example" by self write
  by
    dn="uid=root,ou=System Users,ou=people,dc=yourcompany,dc=example"
    write
  by
   dn="uid=ldaproot,ou=System Users,ou=people,dc=yourcompany,dc=example"
    write
  by * read


index objectClass,uid,gidNumber,sfihostclass,uidNumber eq
index macaddress,apprepository   eq
index   cn,ou,mail,surname,givenname            eq,subinitial
index   appname                                 eq
```

Note: You may need to modify the modulepath, pidfile and include directives in order to reflect your OpenLDAP/Director directories.

Please shut down OpenLDAP (slapd daemon). Then install your slapd.conf file. Before you restart OpenLDAP we will setup some initial database records in the database directory. This is necessary because we told OpenLDAP that the root user (uid=root,ou=Sys...) has got write access to the database. Our database is still empty, so there is no root user there, yet, and therefore no one can write to the database.

Therefore we import the initial database records:

```
% mkdir /var/ldap                      # create the database directory
% slapadd -l initial_db.ldif           # import initial database
% slapindex                            # create database indexes
```

If OpenLDAP is running under a special account (some installations will run them under the account “slapd”), then you have to add a

```
% chown -R slapduser /var/ldap
```

Note: Before importing the initial_db.ldif file you may want to change the database suffix.

Now, you can start the OpenLDAP server. If you have installed the LDAP client command "ldapsearch" together with OpenLDAP (look for a package "ldap-utils" if not), then you can check if everything is ok via issuing

```
% ldapsearch -x -b 'dc=yourcompany,dc=example' objectclass=*
```

You should get some response like

```
# extended LDIF
#
# LDAPv3
# base <dc=yourcompany,dc=example> with scope sub
# filter: objectclass=*
# requesting: ALL
#

# yourcompany.example
dn: dc=yourcompany,dc=example
dc: yourcompany
objectClass: domain
objectClass: top

# People, yourcompany.example
dn: ou=People,dc=yourcompany,dc=example
... ...
...
```

Configuring the Director

Once we have set up the initial LDAP database we can point the Director to the LDAP server. The director's main configuration file is

```
/etc/sfidirector/sfidirector.conf
```

or – if you have not installed the Director in FHS locations –

```
<prefix>/etc/sfidirector.conf
```

For the moment we just make our LDAP server known to the Director. Find the lines

```
Top:                    ldifdir:/var/lib/sfidirector/db
...
Auth:                   ldifdir:/var/lib/sfidirector/db
```

In the configuration file and replace them by

```
Top:                    ldap:localhost:dc=yourcompany,dc=example
Auth:                   ldap:localhost:dc=yourcompany,dc=example
```

The Director will also need to know an account and a password for accessing the database. You can set this by

```
% cd
% mkdir .director
% chmod 700 .director      # this is a secret
% sfidirector storepassword -u root -p root -h localhost -m LDAP
```

The initial passwort of the root account that is set in initial_db.ldif is "root".

If everything went ok, then storepassword should have created a file called .director/cre-

dentials looking like

```
credentials.LDAP_server_localhost.principal: uid=root,ou=System
Users,ou=People,dc=yourcompany,dc=example
credentials.LDAP_server_localhost.password: root
```

You can test if the Director can now talk to the Director server by trying out

```
% sfidirector list -a People
-- uid=root,ou=System Users --

General:
  Base Dir ......... ou=System Users
  First Name ....... System
  Last Name ........ Manager
  User Name ........ System Manager
  Login Name ....... root
```

Loading an initial database

Now that both Director and LDAP are running one might want to populate the database with some example/useful records. In order to do this change to the init directory – if you have installed the sfidirector-init package, then you will find the init directory under

```
/usr/share/sfidirector/init
```

otherwise you will find it in the source tree.

Run

```
% ./init.pl infapp
```

in the init directory and just confirm the suggested settings by typing ENTER. Do not get confused by some strange settings – they will not come into effect if not explicitly requested.

Starting the Director Server

Now that everything is configured the Director server can be started. Since it may have been started automatically during installation it is a good idea to explicitly shut it down before proceeding:

```
% /etc/init.d/sfidirector stop
Stopping SFI Director Admin Server ... STOPPED
% /etc/init.d/frinterface stop
Stopping SFI Director Frontend Interface ... OK
```

Finally, the Director will be started using

```
% /etc/init.d/sfidirector start
Starting SFI Director Admin Server
% /etc/init.d/frinterface start
Starting SFI Director Frontend Interface ... OK
```

Setting up the Web frontend BigClerk

The only task left now is to get the Web frontend running.

Configuring the Web server

BigSWAF will install a file called httpd-bigswaf.conf in its configuration directory

```
/etc/bigsister/httpd-bigswaf.conf
```

if you installed one of the binary packages or installed using FHS standards. Otherwise search for the file under the installation prefix. Either add this file to your Apache configuration or include it from there, e.g. adding a line like

```
Include /etc/bigsister/httpd-bigswaf.conf
```

to your Apache configuration.

The binary packages will look out for an Apache conf.d directory and copy the configuration file therein if it exists. If you installed a binary package, you may already have the necessary configuration be done, then. Please check back.

In any case you will have to (re-)start your web server after this.

Now, you can point your Web browser to

```
http://<yourserver>/bigswaf/BigClerk/
```

and you should get a login prompt. Try

```
Login: root
Password: root
```

Note: You may find that the look of the just-installed BigClerk does not match the screenshots on the Director web page http://sfidirector.sf.net/. This is due to the fact that the "skin" (theme/look) used there cannot be distributed under the GPL. Probably it will be released for free (as in beer, not as in speech) soon under a proprietary licence. Feel free to to provide some nice looking freely distributable icons if you have some.

Enabling CGI accelerators

Provided you installed your preferred CGI accelerator (SpeedyCGI or PPerl) before installing BigSWAF, then you probably do not need to care.

In all other cases you might want to have a look at the CGI directory, which is

```
/usr/share/bigsister/cgi
```

if you installed binary packages or using FHS. Otherwise look for the directory under the installation prefix. In this directory you will find a number of files like

```
bigswaf.cgi
bigswaf.speedy
bigswaf.ppl
bigswaf.mpl
```

and a symbolic link called

```
bigswaf.preferred
```

pointing to one of the above files. Please select the CGI method you prefer and replace the bigswaf.preferred link by a link to your preferred CGI, e.g.:

```
% rm bigswaf.preferred
% ln -s bigswaf.speedy bigswaf.preferred
```

Note: The .mpl variation – using mod_perl – is not covered by this document.

Where to go from here

Your Director server should now be up and running. In order to get familiar with the Director you might want to read additional documentation available from

http://sfidirector.sf.net/