Installation Manual for Sendinel

Sending SMS and Phone Calls to Patients
http://www.sendinel.org

Potsdam, May 2010

[This page intentionally left blank]

Contents

1 What is Sendinel for?

Sendinel aims at improving the communication between clinics and patients. It allows clinics to reach the patients on their cell phones. For example the clinic staff can inform patients if their lab results or medicine have arrived. This quick information transfer avoids useless trips to the clinic for the patients and improves the treatment. The clinic staff can also regularly send messages to a group of people.For example, they can inform all subscribers to "Information about the Clinic" if a specialist is coming the next day.

2 Setting up Sendinel

2.1 Requirements

First, please make sure that the basic system requirements are met:

- a running Linux installation - for the automated installer Ubuntu or Debian is required. In this linux distribution at least Asterisk 1.6 and Python 2.5 have to be supported
- If the asterisk telephony server is not present yet, at least 150MB of free disk space are needed and 50MB have to be downloaded
- a working network connection
- an internet connection if the packages are to be installed from a web resource (at least 60MB have to be downloaded)
- a free USB port to connect the 3G Stick
- a Huawei E169 (also known as Vodafone K3520) - other devices will not work!

2.2 Preparation

Please make sure that you have inserted a working SIM-Card *without PIN protection* into the 3G stick. Connect the stick to the system.

3 The Automatic Installation Script

To use the automatic installation script your system has to run either Debian or Ubuntu. A internet connection is required - as a lot of data has to be downloaded a broadband connection is recommended. On Debian at least 50 MB have to be downloaded.

3.1 Getting the installation package

Either go to http://github.com/Sendinel/Sendinel/ and choose *Download Source* to get the Sendinel archive or use the direct link *http://github.com/Sendinel/Sendinel/tarball/master*.

To download the file you may use your preferred browser, like Firefox. You may also use the command line tool wget.

To do that, you first have to open a *terminal*, sometimes also called *command line*. On Debian it can be found in the *upper menu bar* → *Applications* → *Accessories* → *Terminal*

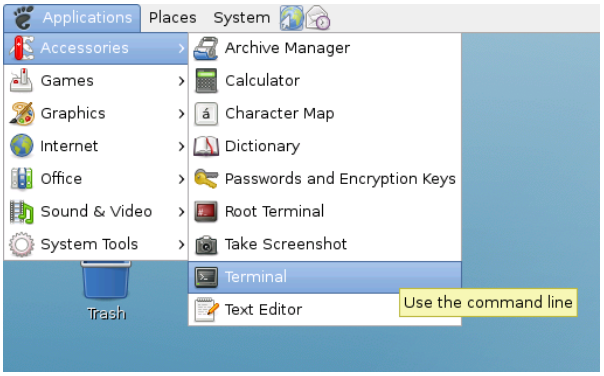

Figure 1: opening a terminal on Debian

After the terminal has been opened, you may create a new folder with *mkdir [Foldername]* or you may just use your home directory.

To download the Sendinel core files with wget, enter

```
wget http://github.com/Sendinel/Sendinel/tarball/master
```

and press the *Return* key on your keyboard.

The download progress will be displayed, as seen in this screenshot.

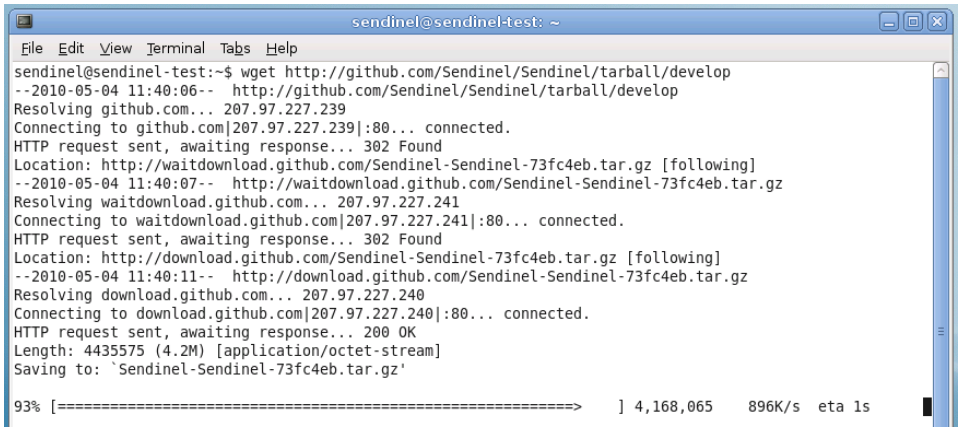

Figure 2: downloading Sendinel with wget

After the file has been downloaded, it has to be extracted. This is to be done with the command

```
tar -xzf [Name of the downloaded file]
```

You may also just enter *tar -xzf* and the first letters of the filename (S in this case - please pay attention to Uppercase/Lowercase) and press the *TAB*-key to let the terminal enter the right name. Afterwards use *cd* to change to the directory where you have just extracted the files. Also here you may use the auto completion feature of the terminal.

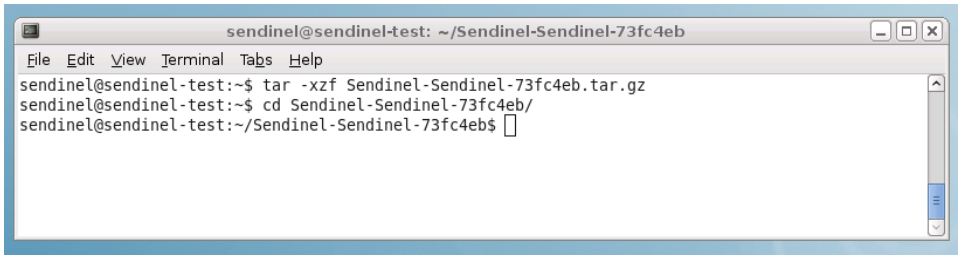

Figure 3: extracting the archive and changing the directory

If your are using debian, use *su* to become the superuser (root), which is equivalent to the *Administrator* on Windows.

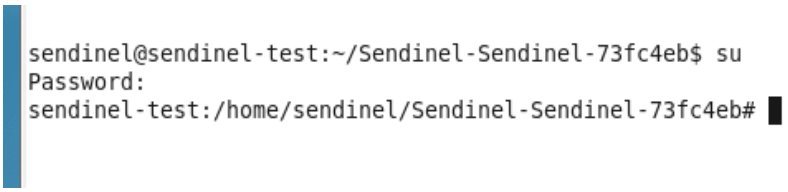

Figure 4: extracting the archive and changing the directory

3.2 Installing Sendinel

To start the installation enter

```
sh install.sh
```

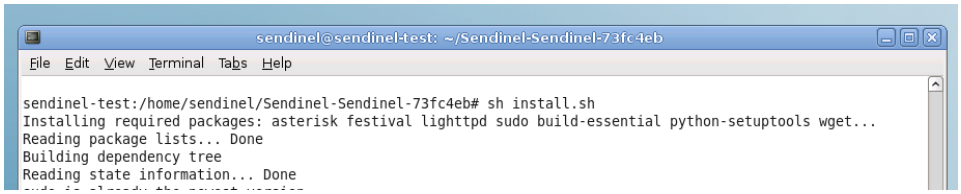

Figure 5: starting the Sendinel installer

After some seconds (this may also take a little bit longer), the system displays how much has to be downloaded and asks for confirmation. If it is okay, simply press the *Return* key and start the download.

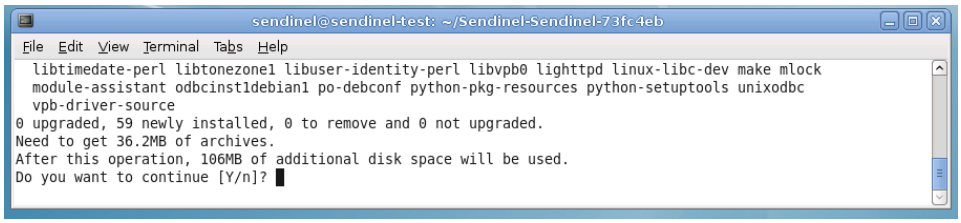

Figure 6: Download Confirmation

During the setup you will be asked for your country code. If you don't know it, it can be found on Wikipedia: http://en.wikipedia.org/wiki/List_of_country_calling_codes . Do not enter the + but only the digits. Later, you will be asked if you want to create an *administrative user*. It is recommended to do so. Enter name, password and e-mail address.

This process may take some time and will displays if errors occur.

4 (manual) off-line installation

If you are either not able to install from online package sources or you want to control the installation by yourself (maybe because you are using another Linux distribution), you should follow this guide

4.1 installing the needed packages

Please make sure the following packages are installed on your system; on Ubuntu/Debian you may use *apt-get install* to install them

- asterisk
- festival
- lighttpd
- sudo
- build-essential
- python-setuptools
- wget

The following python packages also have to be installed. For that you may use *easy_install*

- python-daemon lockfile
- python-daemon lockfile

4.2 Setting up festival text-to-speech synthesis

To get a working festival installation, installing the package should be enough. If you like to change the default voice, please refer to the festival documentation.

4.3 Configuring the Asterisk Telephony Server

4.3.1 the datacard channel

Download the latest revision of chan datacard from github:

`http://github.com/thomasklingbeil/chan\_datacard`

You can either use git or download a tarball of the source code from

```
http://github.com/thomasklingbeil/chan\_datacard/tarball/master
```

If you have downloaded the tarball you've got to untar it, using

```
tar -xvzf [name of the file]
```

Change the directory to the folder where you extracted the files and run the following commands make make install

If you want to configure the datacard by yourself, you can use the datacard.conf from the repository as a template. Else, you can also use the version from the Sendinel repository. In each case the file has to be copied to */etc/asterisk/datacard.conf*

4.3.2 Asterisk configuration files

If you have a plain Asterisk installation it is recommended to use the *extensions.conf* file from the Sendinel repository. It includes all necessary settings for the Asterisk server to run Sendinel.

Manual configuration: If authentication is to be used, an extension in the default call context has to be created which runs a special AGI script for each incoming call; This script created an entry into the Sendinel call log. If such an entry is already existing, extend it to run the AGI script.

```
[default]
...
exten => s,1,AGI(call_log.agi)
exten => s,n,Wait(20)
...
```

For sending outbound short messages, add the following extension into the default context:

```
exten => 2000,1,Answer()
exten => 2000,n,DatacardSendSMS(datacard0,${SmsNumber},${Text})
exten => 2000,n,Hangup()
```

To be able to send automatic outbound calls, the context *"outbound call"* has to be created. Here, you may also adapt the waiting times between the messages.

```
[outbound-call]
exten => s,1,Wait(2)
exten => s,n,Playback(${Salutation})
exten => s,n,Wait(1)
exten => s,n,Playback(${PassedInfo})
exten => s,n,Wait(1)
exten => s,n,Playback(${PassedInfo})
exten => s,n,Hangup()
```

To ensure the system does not hang when there is an incoming SMS on the data card, add the following context. Incoming short messages will be saved to */var/log/asterisk/sms.txt*. You may also change this path.

```
[datacard-incoming]
exten => sms,1,Verbose(Incoming SMS from ${SMSSRC} ${SMSTXT})
exten => sms,n,System(echo '${STRFTIME(${EPOCH},,%Y-%m-%d %H:%M:%S)} -
```

```
        ${CHANNEL} - ${SMSSRC}: ${SMSTXT}' >> /var/log/asterisk/sms.txt)

exten => sms,n,Hangup()

exten => cusd,1,Verbose(Incoming CUSD: ${CUSDTXT})
exten => cusd,n,System(echo '${STRFTIME(${EPOCH},,%Y-%m-%d %H:%M:%S)} -
        ${CHANNEL}: ${CUSDTXT}' >> /var/log/asterisk/cusd.txt)

exten => cusd,n,Hangup()
```

4.3.3 Other

Please ensure that the user the system is running as, is able to write to the asterisk spool directory (by default *ASTERISK_SPOOL_DIR* is configured as /var/spool/asterisk/outgoing). This can either be achieved by grating *any* user the right of writen to that directory (chmod 666) or by adding the user running Sendinel to the group *asterisk*

4.4 Installing Sendinel

Download the source code package from the project page. Extract it and move/copy the contained folder Sendinel to */opt/*, so that the file *manage.py* should reside in */opt/Sendinel/manage.py*

Change the directory to */opt/Sendinel* and run

```
python manage.py syncdb
```

You will be asked if you want to create an *administrative user*. It is recommended to do so. Enter name, password and e-mail address. After this step you may load some basic data into the system by running

```
python manage.py loaddata backend
```

4.5 The lighttpd server

An example configuration for the lighttpd server can be found in the configuration folder. You can either copy those files to the corresponding folder (*/etc/lighttpd/* or adapt your existing configuration.

4.6 Init-Scripts

copy the init scripts *sendinel* and *sendinel-scheduler* to */etc/init.d/* and make them executable by running

```
chmod +x [file name]
```

Please adapt the scripts to the used path
run

```
update-rc.d sendinel defaults
```

and

```
update-rc.d sendinel-scheduler defaults
```

5 Possible settings for Sendinel

To configure the system, create the file *local_settings.py* in the sendinel directory.

5.1 Country Settings

COUNTRY_CODE_PHONE Prefix of the country the system is going to be used in

START_MOBILE_PHONE Beginning of mobile phone numbers for validation purposes.

5.2 Asterisk Settings

ASTERISK_USER and ASTERISK_GROUP Group and user asterisk is running as. In most cases both is *asterisk*

ASTERISK_SPOOL_DIR The directory asterisk uses for scheduling calls. default: /var/spool/asterisk/outgoing/

ASTERISK_DATACARD select whether to use an external datacard for conducting the calls and the short messages. If False is selected, another SIP account has to be configured in the asterisk server manually. Sending short messages is only possible with a connected data card.
possible values: True/False (pay attention to the Upcase first letter) default: True

ASTERISK_EXTENSION asterisk extension to be used for outbound calls
default: *s*

ASTERISK_SIP_ACCOUNT SIP account to be used for asterisk for datacard use, this should be set to *ext-sip-account*, corresponding to the setting in *datacard.conf*

5.3 Text-to-Speech settings

FESTIVAL_CACHE Directory used for caching the speech sound files For use on a plug computer with Debian this should be set to */lib/init/rw*
default: *tmp*

5.4 Sendinel Settings

AUTH Enable/Disable authentication for entered phone numbers. If used, also *AUTH NUMBER* has to be set.
possible values: True, False
default: False

AUTH_NUMBER This setting is only required if authentication is enabled. This entry has to be set to the number of the used SIM-Card/SIP-Account. It is used for display purposes only.

AUTHENTICATION_CALL_TIMEOUT This setting is only required if authentication is enabled. Time after which the authentication request gets invalid.
default: timedelta(minutes = 3)

CALL_SALUTATION This text is played before the information when a user is called.

SMS_SALUTATION This text is put in front of each short message sent through the system.