Communication protocols and services

This chapter describes various protocols and services that may be enabled on Modberry.

SSH Connection

SSH service is started up on boot and already preinstalled and configured. You may access your device through ssh client such as PuTTY on Windows, or ssh command on Linux and OS X.

Configuration

Two files are responsible for *ssh* configuration on the device. These files are */etc/ssh/ssh_config* and */etc/ssh/sshd_config.*

Be wary when you change these files. Wrong modifications may make you lose ability to connect to device over ssh! Remember, you can only change them as root. If you make any changes, remember to restart the service with *service ssh restart* command.

FTP/SFTP

This chapter is about FTP and SFTP protocols on Modberry device.

FTP

By default, Modberry does not have FTP service installed. It is up to user what software uses to setup the FTP service on the device. This manual covers using *vsftpd* as a FTP service provider.

Installation

To install vsftpd on Modberry simple type in this command over ssh:

```
sudo apt-get install vsftpd
```

Configuration

vsftpd needs configuration. To open the configuration file type this command over ssh:

```
sudo nano /etc/vsftpd.conf
```

We need to disable anonymous login. Look for this line:

```
anonymous_enable=YES
```

Then prepend it with "#", effectively turning this line to be a comment and being ignored. You may

also change “YES” to “NO” - effect will be the same
The line should look like this now:

```
#anonymous_enable=YES
```

OR

```
anonymous_enable=NO
```

Let's enable local user to login on FTP. Look for the line:

```
#local_enable=YES
```

Uncomment it (remove the ”#” before the rest). It will allow you to use ftp as the local user.

If you want to write files to FTP, you need to uncomment this line as well:

```
#write_enable=YES
```

Close nano and save changes(ctrl+x then Y, then ENTER).

Then, restart the vsftpd daemon with this command:

```
sudo service vsftpd restart
```

You should be done now and be able to connect with your Modberry over FTP with your favourite FTP client, using credentials that you use to login to device.

Status

You can check service's status using this command:

```
sudo service vsftpd status
```

SFTP

SFTP does not require additional installations as it uses ssh protocol, that is already configured and running on the device. Neither does it need to be configured - all you need is login and password for ssh (read/write permissions are based on that user permissions).

SFTP uses the same port as ssh (by default port 22, but it may be changed by user).

Connection

To connect over SFTP, open your favourite FTP client and change protocol to SFTP then enter your credentials that you use to login into the device.

On Linux there is also a small command-line sftp client called *sftp*.

To open sftp connection using this tool, simply type in this command:

```
sftp username@addressip
```

This command is analogical to ssh command. Enter password and you're in.
To download file use this command:

```
sftp> get FILENAME
```

To send file use this command:

```
sftp> put FULLFILEPATH
```

You can also use *ls* command to list files, *cd* to change directories and *rm* to remove remote files. For more information invoke the *help* command inside of sftp:

```
sftp> help
```

Email

You may configure your device to send emails. You will need *sendmail* application for this.

Installation

To install *sendemail* application, type this command:

```
sudo apt-get install sendemail
```

Here are the options:

```
sendemail-1.56 by Brandon Zehm <caspian@dotconf.net>

Synopsis:  sendemail -f ADDRESS [options]

  Required:
    -f ADDRESS                  from (sender) email address
    * At least one recipient required via -t, -cc, or -bcc
    * Message body required via -m, STDIN, or -o message-file=FILE

  Common:
    -t ADDRESS [ADDR ...]       to email address(es)
    -u SUBJECT                  message subject
    -m MESSAGE                  message body
    -s SERVER[:PORT]            smtp mail relay, default is localhost:25

  Optional:
    -a   FILE [FILE ...]        file attachment(s)
    -cc  ADDRESS [ADDR ...]     cc  email address(es)
    -bcc ADDRESS [ADDR ...]     bcc email address(es)
    -xu  USERNAME               username for SMTP authentication
    -xp  PASSWORD               password for SMTP authentication

  Paranormal:
    -b BINDADDR[:PORT]          local host bind address
    -l LOGFILE                  log to the specified file
    -v                          verbosity, use multiple times for greater effect
    -q                          be quiet (i.e. no STDOUT output)
    -o NAME=VALUE               advanced options, for details try: --help misc
        -o message-content-type=<auto|text|html>
        -o message-file=FILE          -o message-format=raw
        -o message-header=HEADER      -o message-charset=CHARSET
        -o reply-to=ADDRESS           -o timeout=SECONDS
        -o username=USERNAME          -o password=PASSWORD
        -o tls=<auto|yes|no>          -o fqdn=FQDN


  Help:
    --help                      the helpful overview you're reading now
    --help addressing           explain addressing and related options
    --help message              explain message body input and related options
    --help networking           explain -s, -b, etc
    --help output               explain logging and other output options
    --help misc                 explain -o options, TLS, SMTP auth, and more
```

To get this information on this device type:

```
sendemail --help
```

Sending emails

You will need:

- SMTP server address with port (typically 587)
- account name
- account password
- at least one recipient

The simplest mail can be sent with such command:

```
root@modberry:~# sendemail -f SENDERADDRESS -t RECIPIENTADDRESS -u SUBJECT
-m "MESSAGE CONTENT" -s SMTPSERVERADDRESS:PORT -xu USERNAME -xp PASSWORD
```

Of course substitute the CAPS with your informations. You should receive notification, if everything went as expected:

```
Jan 02 13:51:39 modberry sendemail[10668]: Email was sent successfully!
```

You can attach files with *-a* option and instead of *-m*, you can use *-o FILENAME* to use contents of the file specified as a message content.

Apache2 (web server)

This subchapter will tell you how to setup and configure Apache 2 server on Modberry.

Installation

To install apache, type this command:

```
sudo apt-get install apache2 -y
```

The website files goes into /var/www

If everything went properly you can check it by entering Modberry address into browser bar.

Configuration

The full configuration of Apache2 is beyond the scope of this document. Here's link to official documentation: http://httpd.apache.org/docs/2.2/configuring.html

The files are located in /etc/apache2 directory. You can check the default at /etc/apache2/sites-enabled to get a hang of it.

Status

Apache service status can be checked with this command:

```
service apache2 status
```

Plugin: PHP

To allow your Apache server to process PHP files, you'll need to install PHP5 and the PHP5 module for Apache. Type the following command to install these:

```
sudo apt-get install php5 libapache2-mod-php5 -y
```

Now move the index.html file to index.php:

```
sudo mv index.html index.php
```

and edit the file:

```
sudo nano index.php
```

to put some PHP content in it:

```
<?php echo "hello world";
```

Now save and refresh your browser. You should see “hello world”. This is not dynamic but still served by PHP. Try something dynamic:

```
<?php echo date('Y-m-d H:i:s');
```

or show your PHP info:

```
<?php phpinfo();
```

VNC (Virtual Network Computing)

Sometimes it is not convenient to work directly on the Modberry. Maybe you would like to work on it from another computer by remote control.

VNC is a graphical desktop sharing system that allows you to remotely control the desktop interface of one computer from another. It transmits the keyboard and mouse events from the controller, and receives updates to the screen over the network from the remote host.

You will see the desktop of the Modberry inside a window on your computer. You'll be able to control it as though you were working on the Modberry itself.

Installation

On your Modberry (using a monitor or via SSH), install the TightVNC package:

```
sudo apt-get install tightvncserver
```

Next, run TightVNC Server which will prompt you to enter a password and an optional view-only password: tightvncserver Start a VNC server from the terminal. This example starts a session on VNC display zero (:0) with full HD resolution:

```
 vncserver :0 -geometry 1920x1080 -depth 24
```

Now, on your computer, install and run the VNC client:

On a Linux machine install the package xtightvncviewer: sudo apt-get install xtightvncviewer

Otherwise, TightVNC is downloadable from http://tightvnc.com

Automation and run at boot

You can create a simple file with the command to run the VNC server on the Modberry, to save having to remember it:

Create a file containing the following shell script:

```
#!/bin/sh
vncserver :0 -geometry 1920x1080 -depth 24 -dpi 96
```

Save this as vnc.sh (for example)

Make the file executable:

```
chmod +x vnc.sh
```

Then you can run it at any time with:

```
./vnc.sh
```

To run at boot:

Log into a terminal on the Modberry as root:

```
 sudo su
```

Navigate to the directory /etc/init.d/:

```
 cd /etc/init.d/
```

Create a new file here containing the following script:

```
### BEGIN INIT INFO
# Provides: vncboot
# Required-Start: $remote_fs $syslog
# Required-Stop: $remote_fs $syslog
# Default-Start: 2 3 4 5
# Default-Stop: 0 1 6
# Short-Description: Start VNC Server at boot time
# Description: Start VNC Server at boot time.
### END INIT INFO

#! /bin/sh
# /etc/init.d/vncboot

USER=root
```

```
HOME=/root

export USER HOME

case "$1" in
 start)
  echo "Starting VNC Server"
  #Insert your favoured settings for a VNC session
  /usr/bin/vncserver :0 -geometry 1280x800 -depth 16 -pixelformat rgb565
  ;;

 stop)
  echo "Stopping VNC Server"
  /usr/bin/vncserver -kill :0
  ;;

 *)
  echo "Usage: /etc/init.d/vncboot {start|stop}"
  exit 1
  ;;
esac

exit 0
```

Save this file as vncboot (for example)

Make this file executable:

```
chmod 755 vncboot
```

Enable dependency-based boot sequencing:

```
update-rc.d /etc/init.d/vncboot defaults
```

If enabling dependency-based boot sequencing was successful, you will see this: update-rc.d: using dependency based boot sequencing But if you see this:

```
update-rc.d: error: unable to read /etc/init.d//etc/init.d/vncboot
```

then try the following command:

```
update-rc.d vncboot defaults
```

Reboot your Modberry and you should find a VNC server already started. You'll now use a VNC client program on your PC/laptop to connect to the VNC server and take control of it.

rsync

You can use the tool rsync to synchronise folders between computers. You might want to transfer some files from your desktop computer or laptop to your Modberry, and for them to be kept up to date, or you might want the pictures taken by your Modberry transferred to your computer automatically.

Using rsync over SSH allows you to transfer files to your computer automatically.

Here is an example of setting up the sync of a folder of pictures on your Modberry to your computer:

On your computer, create a folder called screenshots:

```
mkdir screenshots
```

Look up the Modberry's IP address by logging in to it and running hostname -I. In this example the Modberry is creating a timelapse by capturing a screenshot every minute, and saving the picture with a timestamp in the local folder sreenshots in its eMMC.

Now run the following command (substituting your own Modberry's IP address):

```
rsync -avz -e ssh modberryuser@192.168.1.10:screenshots/ screenshots/
```

This will copy all files from the Modberry's camera folder to your computer's new camera folder.

In order to keep the folders in sync, run this command in cron.

IPTABLES

IPTABLES is a tool managing IP packet filters in the Linux system. This program can be used as a frame filter or a so-called system state firewall, controlling incoming and outgoing connections to the computer network or work station.

The Netfilter project website containing documentation and examples of using the iptables pack is located under the address:

http://www.netfilter.org/documentation/index.html#documentation-howto

Defining filters

Initiating filters regarding network traffic (independently of the protocol) is made possible by the command:

```
# iptables [-t tables] [-P] [INPUT, OUTPUT, FORWARD, PREROUTING, OUTPUT,
POSTROUTING][ACCEPT, DROP]
```

Main attributes:

- INPUT – describes operations for incoming packets,
- OUTPUT – outgoing packets,
- FORWARD – for packets travelling between interfaces.
- PREROUTING – for incoming packets before routing
- POSTROUTING – for outgoing packets after routing.
- ACCEPT (accepts packet),
- DROP (deletes)
- REJECT (rejects packet and notifies sender).

Adding a filter for a specific protocol is made possible by the command:

```
# iptables [-t table] [-AI] [INPUT, OUTPUT, FORWARD] [-io interface]
[-p tcp, udp, icmp,all] [-s IP/network] [--sport ports] [-d IP/network]
[--dport ports] –j [ACCEPT. DROP]
```

```
Attributes:
• A adds a new rule at the end of the current chain
• I adds more than one rule to a given chain
• i input interface name
• o output interface name
• p protocol to be filtered in a given chain
• s source address
--sport source port number
• d  target address
--dport target port
• j defines actions for packets matching the filter i.e. ACCEPT, DROP, or
LOG.
```

Example of usage:

- Accept all packets from the internal interface lo

```
# iptables –A INPUT –i lo –j ACCEPT
```

- Accept TCP packets sent from address 192.168.0.1

```
# iptables –A INPUT –i eth0 –p tcp –s 192.168.0.1 –j ACCEPT
```

- Accept TCP packets of sent class C network addresses 192.168.1.0/24

```
# iptables –A INPUT –i eth0 –p tcp –s 192.168.1.0/24 –j ACCEPT
```

- Reject TCP packets sent from address 192.168.1.25

```
# iptables –A INPUT –i eth0 –p tcp –s 192.168.1.25 –j DROP
```

- Reject TCP packets addressed to port 21

```
# iptables –A INPUT –i eth0 –p tcp --dport 21 –j DROP
```

- Accept TCP packets sent from address 192.168.0.20 to port 130,...,138,139

```
# iptables –A INPUT –i eth0 –p tcp –s 192.168.0.24 --dport 130:139 –j
ACCEPT
```

- Reject all packets sent from a device with MAC address 11:22:33:44:55:66

```
# iptables –A INPUT –i eth0 –p all –m mac -–mac-source 11:22:33:44:55:66 –j
DROP
```

- Remove a specific existing filter in iptables

```
# iptables –D INPUT –i eth0 –p tcp --dport 21 –j DROP
```

Added rules must not exclude each other. If they do, before setting rules again, it is recommended to clear iptables using the following commands:

iptables -F remove rules from all chains

iptables -X remove selected chain

iptables -t nat -F remove rules from all nat table chains

iptables -t nat -X remove selected chain from nat table

When using many rules, remember the order in which they are to be added:

• starting with the most detailed (filters for individual protocols) to the most general (filters for entire chains).