

FreeBSD Administration - Support #535

Install Prosody XMPP Server on FreeBSD

01/18/2015 04:27 PM - Daniel Curtis

Status:	Closed	Start date:	08/21/2013
Priority:	Normal	Due date:	
Assignee:	Daniel Curtis	% Done:	100%
Category:	XMPP Server	Estimated time:	1.00 hour
Target version:	FreeBSD 9	Spent time:	3.50 hours

Description

Install Prosody

- Install the Prosody server. Use the following command:

```
pkg install prosody openssl
```

- Enable prosody to start at boot:

```
echo 'prosody_enable="YES"' >> /etc/rc.conf
```

When pkg finishes, the Prosody server will have been successfully installed, and will be ready for configuration. Prosody provides an init script that allows you to reload the configuration file, start, stop, or restart the XMPP server. Issue one of the following commands as appropriate:

```
service prosody reload
service prosody start
service prosody stop
service prosody restart
```

Configure Prosody Server

The configuration file for Prosody is located in `/usr/local/etc/prosody/prosody.cfg.lua`, and is written in Lua syntax.

Note that in the Lua programming language, comments (lines that are ignored by the interpreter) are preceded by two hyphen characters (e.g. `--`). The default config has some basic instructions in Lua syntax, which can be helpful if you're unfamiliar with the language.

- To allow Prosody to provide XMPP/jabber services for more than one domain, insert a line in the following form into the configuration file. This example defines one virtual host.

```
VirtualHost "example.com"
```

- Following a Host line there are generally a series of host-specific configuration options. If you want to set options for all hosts, add them below the "Host "" entry in your config file. For instance, to ensure that Prosody behaves like a proper Linux server daemon make sure that the "posix;" option is included in the "modules_enabled = { }" table.

```
modules_enabled = {
  -- [...]
  "posix";
  -- [...]
}
```

NOTE: that there should be a number of global modules included in this table to provide basic functionality.

- To disable a host without removing it from your configuration file, add the following line to its section of the file:

```
enabled = false
```

- To specify administrators for your server, add a line in the following format to your `prosody.cfg.lua` file.

```
admins = { "admin1@example.com", "admin2@example.com" }
```

To add server-wide administrators, add the `admins` line to the "Hosts "" section. To grant specific users more granular control to administer particular hosts, you can add an `admins` line, or more properly tables in Lua, to specific hosts.

- Specify a PID file:

```
pidfile = "prosody.pid"
```

Do not forget to reload the configuration for the Prosody server after making any changes to your `/usr/local/etc/prosody/prosody.cfg.lua` file, by issuing the following command:

```
service prosody reload
```

XMPP Federation and DNS

To ensure that your Prosody instance will federate properly with the rest of the XMPP network, particularly with Google's "GTalk" service (i.e. the "@gmail.com" chat tool,) we must set the SRV records for the domain to point to the server where the Prosody instance is running. We need three records, which can be created in the DNS Management tool of your choice:

- Service: `_xmpp-server._tcp` and Port: **5269**
- Service: `_xmpp-client._tcp` and Port: **5222**
- Service: `_jabber._tcp` and Port: **5269**

The "target" of the SRV record should point to the publicly routable hostname for that machine (e.g. `"*squire.example.net"`). The priority and weight should both be set to 0.

Enabling Components

- In the XMPP world, many services are provided in components, which allows for greater ease of customization within a basic framework. A common example of this is the MUC or multi-user chat functionality. To enable MUC services in Prosody you need to add a line like the following to your `/usr/local/etc/prosody/prosody.cfg.lua` file.

```
Component "prosody.example.com" "muc"
```

In this example, **conference.example.net** is the domain where the MUC rooms are located, and will require an "DNS A record," that points to the IP Address where the Prosody instance is running. MUCs will be identified as JIDs (Jabber IDs) at this hostname, so for instance the "rabbits" MUC hosted by this server would be located at rabbits@example.com.

- MUC, in contrast to many other common components in the XMPP world, is provided internally by Prosody. Other components, like transports to other services, run on an external interface. Each external component has its own host name, and provides a secret key which allows the central server to authenticate to it. See the following **aim.bucknell.net** component as an example.

```
Component "aim.example.com"  
component_secret = "mysecretcomponentpassword"
```

NOTE: That external components will need to be installed and configured independently of Prosody.

Typically, Prosody listens for connections from components on the localhost interface (i.e. on the 127.0.0.1 interface;). If you're connected to external resources that are running on an alternate interface, specify the following variables as appropriate in the "Host "" section of the file config file.

```
Host ""
```

```
component_interface = "192.168.0.10"  
component_ports = { 8888, 8887 }
```

Administration with prosodyctl

The XMPP protocol supports "in-band" registration, where users can register for accounts with your server via the XMPP interface. However, this is often an undesirable function as it doesn't permit the server administrator the ability to moderate the creation of new accounts and can lead to spam-related problems. As a result, Prosody has this functionality disabled by default. While you can enable in-band registration, we recommend using the prosodyctl interface at the terminal prompt.

If you're familiar with the ejabberdctl interface from ejabberd, prosodyctl mimics its counterpart as much as possible.

- To use prosodyctl to register a user, in this case *lollipop@*ducklington.org**, issue the following command:

```
prosodyctl adduser lollipop@example.com
```

- To set the password for this account, issue the following command and enter the password as requested:

```
prosodyctl passwd lollipop@example.com
```

- To remove this user, issue the following command:

```
prosodyctl deluser lollipop@example.com
```

- Additionally, prosodyctl can provide a report on the status of the server in response to the following command:

```
prosodyctl status
```

NOTE: That all of the prosodyctl commands require root privileges, unless you've logged in as the same user that Prosody runs under (not recommended).

Securing Prosody

- Generate a strong SSL private key and CSR:

```
openssl req -sha512 -out prosody.example.com.csr -new -newkey rsa:4096 -nodes -keyout prosody.  
example.com.key
```

- Once the SSL certificate has been created, edit `/usr/local/etc/prosody/prosody.cfg.lua` and change the ssl parameters:

```
ssl = {  
    key = "/usr/local/etc/prosody/certs/prosody.example.com.key";  
    certificate = "/usr/local/etc/prosody/certs/prosody.example.com.crt";  
}  
  
https_ssl = {  
    certificate = "/usr/local/etc/prosody/certs/prosody.example.com.crt";  
    key = "/usr/local/etc/prosody/certs/prosody.example.com.key";  
}
```

```
}  
  
c2s_require_encryption = true  
  
s2s_secure_auth = true
```

- Restart prosody

```
service prosody restart
```

Enabling BOSH

- Edit `/usr/local/etc/prosody/prosody.cfg.lua`, and add the following to the `modules_enable` section:

```
modules_enabled = {  
    -- HTTP modules  
    "bosh"; -- Enable BOSH clients, aka "Jabber over HTTP"  
};
```

- **NOTE:** Make sure the ports **5280** and **5281** are open.
- Now use <http://example.com:5280/http-bind> as the BOSH URL in the messaging client, in my case Pidgin.

Nginx Proxy

- Setup the web server that serves the app script also act as a proxy to the real BOSH server at some URL:

```
vi /usr/local/etc/nginx/conf.d/example.com.conf
```

- And add the following, adjusting the parameters as necessary:

```
server {  
    listen 80;  
    server_name prosody.example.com;  
    access_log /var/log/nginx/prosody.example.com-access.log;  
    error_log /var/log/nginx/prosody.example.com-error.log;  
    root /usr/local/www/sites/prosody.example.com;  
  
    location /http-bind {  
        proxy_pass http://prosody.example.com:5280/http-bind;  
        proxy_set_header Host $host;  
        proxy_buffering off;  
        tcp_nodelay on;  
    }  
}  
  
server {  
    listen 443 ssl;  
    server_name prosody.example.com;  
    access_log /var/log/nginx/prosody.example.com-access.log;  
    error_log /var/log/nginx/prosody.example.com-error.log;  
  
    ssl_certificate /usr/local/etc/nginx/ssl/prosody.example.com.crt;  
    ssl_certificate_key /usr/local/etc/nginx/ssl/prosody.example.com.key;  
  
    # Configure Strong SSL  
    ssl_ciphers 'AES128+EECDH:AES128+EDH:!aNULL';  
    ssl_protocols TLSv1 TLSv1.1 TLSv1.2;  
    ssl_session_cache builtin:1000 shared:SSL:10m;  
    ssl_stapling on;
```

```

ssl_stapling_verify on;
ssl_prefer_server_ciphers on;
ssl_dhparam /usr/local/etc/nginx/dhparam.pem;
add_header Strict-Transport-Security max-age=63072000;
add_header X-Frame-Options SAMEORIGIN;
add_header X-Content-Type-Options nosniff;

root /usr/local/www/sites/prosody.example.com;

location /http-bind {
    proxy_pass https://prosody.example.com:5281/http-bind;
    proxy_set_header Host $host;
    proxy_buffering off;
    tcp_nodelay on;
}
}

```

This will allow prosody BOSH server to transfer files using <http://prosody.example.com/http-bind> and <https://prosody.example.com/http-bind>

Enable File Transfer Proxy

- Edit `/usr/local/etc/prosody/prosody.cfg.lua`, and add the following to the **modules_enable** section:

```

modules_enabled = {
    "proxy65"
};

proxy65_ports = { 5000 }

```

Adding 3rd Party Modules

- Install mercurial and rsync:

```
pkg install mercurial rsync
```

- Clone the latest 3rd party into the home folder:

```

cd
hg clone https://code.google.com/p/prosody-modules/

```

- Copy the 3rd party modules to the prosody module directory:

```
rsync -a prosody-modules/ /usr/local/lib/prosody/modules/
```

- Edit `/usr/local/etc/prosody/prosody.cfg.lua`, and add the following to the **modules_enable** section:

```

modules_enabled = {
    "log_auth"; Logs failed authentication attempts IP (for fail2ban)
};

```

NOTE: Documentation for each module can be found [here](#)

Resources

You may wish to consult the following resources for additional information on this topic. While these are provided in the hope that they will be useful, please note that we cannot vouch for the accuracy or timeliness of externally hosted materials.

- [The official Prosody server website](#)
- [XMPP Standards Foundation](#)
- [XMPP Client Software](#)
- <http://wiki.prosody-modules.googlecode.com/hg/>
- http://prosody.im/doc/setting_up_bosh

Related issues:

Copied from GNU/Linux Administration - Support #183: Installing Prosody XMPP ...

Closed

08/21/2013

History

#1 - 01/18/2015 04:27 PM - Daniel Curtis

- Copied from Support #183: Installing Prosody XMPP Server on Debian 7 added

#2 - 01/18/2015 04:45 PM - Daniel Curtis

- Description updated

- Status changed from New to In Progress

- % Done changed from 0 to 50

#3 - 01/18/2015 05:26 PM - Daniel Curtis

- Description updated

- % Done changed from 50 to 100

#4 - 01/18/2015 05:29 PM - Daniel Curtis

- Description updated

#5 - 01/19/2015 12:13 PM - Daniel Curtis

- Description updated

#6 - 01/19/2015 02:27 PM - Daniel Curtis

- Description updated

#7 - 01/20/2015 04:32 PM - Daniel Curtis

- Status changed from In Progress to Resolved

#8 - 01/22/2015 07:52 AM - Daniel Curtis

- Description updated

#9 - 01/22/2015 10:39 AM - Daniel Curtis

- Description updated

#10 - 01/22/2015 10:52 AM - Daniel Curtis

- Description updated

#11 - 01/22/2015 10:53 AM - Daniel Curtis

- Description updated

#12 - 01/22/2015 10:55 AM - Daniel Curtis

- Description updated

#13 - 01/22/2015 01:25 PM - Daniel Curtis

- Description updated

#14 - 01/23/2015 09:42 AM - Daniel Curtis

- Description updated

#15 - 01/27/2015 11:18 AM - Daniel Curtis

- *Status changed from Resolved to Closed*

#16 - 02/14/2015 10:47 AM - Daniel Curtis

- *Target version set to FreeBSD 9*

#17 - 02/14/2015 12:03 PM - Daniel Curtis

- *Category set to XMPP Server*

#18 - 03/17/2015 01:56 PM - Daniel Curtis

- *Subject changed from Installing Prosody XMPP Server on FreeBSD to Install Prosody XMPP Server on FreeBSD*

- *Description updated*

#19 - 03/17/2015 02:01 PM - Daniel Curtis

- *Description updated*

#20 - 03/02/2016 05:01 PM - Daniel Curtis

- *Description updated*

#21 - 08/07/2016 08:51 PM - Daniel Curtis

- *Description updated*

#22 - 08/07/2016 09:15 PM - Daniel Curtis

- *Description updated*