

# Download and Use the Live DVD

 The Live DVD is now obsolete. Please, use the [Using the docker-compose based Moonshot demo](#) instead.

Live DVD creates a Debian instance for testing purposes. On this page you will find instructions on how to get and use the Moonshot Live DVD, which lets you see and test all of the components of Moonshot.

## Contents

- [1. Get the Live DVD](#)
- [2. Boot the Live DVD or Install a new system from the Live DVD](#)
- [3. Upgrade to the latest version of the software](#)
- [4. Configure FreeRADIUS and permissions](#)
  - [4.1. Permissions](#)
  - [4.2. RadSec and FreeRADIUS](#)
- [5. Test Moonshot locally on the VM](#)
  - [5.1. Testing with gss-client and gss-server](#)
  - [5.2. Testing SSH](#)
- [6. Connect your new Service to the Moonshot infrastructure](#)

## LiveDVD version Information

These instructions relate to the 2015.07.22 release of the Live DVD. The previous edition is [2014.10.08](#).

## Notes on this LiveDVD release

This version uses FreeRADIUS 3.0.7 with trust router dynamic realms enabled.

## Known Issues with this LiveDVD release

None.

## 1. Get the Live DVD

You can download an ISO of the Live DVD here - <https://psec.s3.amazonaws.com/moonshot-images/2015.07.22.iso>.

 The sources for the Live DVD are also available in the form of two tarballs: [source.debian.tar](#) and [source.debian-live.tar](#). This is provided for GPL compliance only - if you actually want access to the source, see our [Source Access](#) information for details on how to access our GIT repositories.

## User accounts on this Live DVD release

The Debian Live DVD defaults of user 'user' with password 'live' and no root password apply when running this DVD with the transient 'live' option.

## 2. Boot the Live DVD or Install a new system from the Live DVD

At this point, you can simply boot a machine (physical or virtual) using the ISO and follow the remaining instructions on this page. However, if you do this, any changes you make to the system will be lost upon restarting the system.

For a slightly less transient Live DVD experience, it is suggested that you instead create a new system (physical or virtual) using the Live DVD. To do so, do the following:

1. Either burn the Live DVD to a blank DVD and start your machine with the DVD inserted, or on your virtualisation platform of choice, create a new VM, using the Moonshot ISO you've downloaded as its DVD.
2. Turn on the machine/virtual machine.
3. On the Grub menu that appears, choose the "installgui" option.
4. Run through the Installer.

- a. Choose the correct locale, etc.
  - b. Choose a hostname for your VM, the domain, and set the root password.
  - c. Set up a user account for yourself (note that a "moonshot" account will be set up anyway later - this one is just one for you)
  - d. Set up partitioning, network mirror etc. until installation is complete.
  - e. Hit Finish, eject the virtual CD, boot up your new system.
5. You now have a Moonshot environment containing the client software, an IdP, an RP, and a Moonshot-enabled SSH server.

### 3. Upgrade to the latest version of the software

The software may have been updated since the Live DVD was packaged, so you should run an update to ensure you have the latest updates.

1. Open a terminal window (Applications > Accessories > Terminal).
2. Become the root user (type `su -`). Enter your root password when prompted (if booting directly from the Live DVD, this will be "Moonshot", otherwise, it will be whatever you set it as).
3. Update the packing system by entering the following command:

```
$ apt-get update
```

4. Upgrade by entering the following command:

```
$ apt-get upgrade
```

5. If you have installed a system from the Live DVD, reboot your VM.
6. Congratulations, you are now up to date!

### 4. Configure FreeRADIUS and permissions

The Live DVD is *almost* ready to use out of the box - two manual steps are required.

#### 4.1. Permissions

We need to place the FreeRADIUS user and the Trust Router users into each other's groups to allow them to read each other's shared files.

```
$ adduser freerad trustrouter  
$ adduser trustrouter freerad
```

#### 4.2. RadSec and FreeRADIUS

This step creates a set of certificates for RadSec, that enables the various components on the system to communicate securely, and corrects some permissions in the FreeRADIUS configuration.

1. Open a terminal window (Applications > Accessories > Terminal).
2. Become the root user (type `su -`). Enter your root password when prompted (if booting directly from the Live DVD, this will be "Moonshot", otherwise, it will be whatever you set it as).
3. Change directory into `/etc/freeradius/certs`
4. Clear out any old certificates in the directory:

```
$ make destroycerts
```

5. Run the bootstrap script to generate some new certificates

```
$ ./bootstrap
```

6. Remove the password from the certificate you just generated so that gss-server can start without you having to enter it (when openssl prompts you for a password, it is "whatever").

```
$ openssl rsa -in client.key -out client.noenc && mv client.noenc client.key
```

7. Restart FreeRADIUS.

```
$ /etc/init.d/freeradius restart
```

## 5. Test Moonshot locally on the VM

### 5.1. Testing with gss-client and gss-server

gss-server is a sample server that works in conjunction with gss-client and allows the basic testing of Moonshot functionality and to experience the Identity Selector; these utilities ship with the Live DVD. To use these to test whether Moonshot is working on your VM, do the following:

1. Open a terminal window and run the GSS server using a GSS acceptor name of "gss@[whatever your fully qualified hostname is]". When this starts, the details of any incoming Moonshot connections will be echoed into the terminal.

```
$ gss-server -verbose gss@[hostname of your server]
```



For example, if your hostname was "moonshotdemo.dev.ja.net", you would run `gss-server -verbose gss@moonshotdemo.dev.ja.net`

2. Open a second terminal window run the GSS client instructing it to connect to the GSS server that you just started, using the hostname and GSS acceptor name that you used in the previous step, sending it the message "Hello World".

```
$ gss-client -spnego [hostname of your server] gss@[hostname of your server] "Hello World"
```



For example, if your hostname was "moonshotdemo.dev.ja.net", you would run `gss-client -spnego moonshotdemo.dev.ja.net gss@moonshotdemo.dev.ja.net "Hello World"`

3. At this point, gss-client needs to know which identity you wish to use with the service that you are trying to connect to. To find this out, it invokes the [Moonshot Identity Selector](#).
4. The Moonshot Identity Selector will now pop up, asking which Identity you wish to use with this service. At this point, you have none configured so you will need to add a new one. The Live DVD ships with a fully functioning Identity Provider with one identity for you to use for local testing. To add this identity, click on "File" and "Add ID Card". Enter the following information:
  - Display name: My Test Identity
  - Issuer: example.com
  - Username: steve
  - Password: testing
5. Check "remember password" and then click "Add ID Card". Then click on the ID card you just created and hit "Send".



At this point, if you took a while to add the identity, it's very likely that nothing will happen. This is simply because the transaction between GSS client and GSS server times out very quickly. If this happens, simply run the gss-client command from above as before to restart the attempt.

6. Lots of stuff should now happen in both windows. If it has all worked correctly, in the window you ran the server on, you should see lots of text, ending with something like the following:

```

Attribute local-login-user Authenticated Complete

moonshot

6d6f6f6e73686f74

UID: 501
Accepted connection: "steve"
Sending init_sec_context token (size=150)...continue needed...

context flag: GSS_C_MUTUAL_FLAG
context flag: GSS_C_REPLAY_FLAG
context flag: GSS_C_SEQUENCE_FLAG
context flag: GSS_C_CONF_FLAG
context flag: GSS_C_INTEG_FLAG
"steve" to "host/$hostname", lifetime -1, flags 13e, locally initiated, open
Name type of source name is { 1 2 840 113554 1 2 1 1 }.
Mechanism { 1 3 6 1 5 5 2 } supports 4 names
0: { 1 2 840 113554 1 2 1 1 }
1: { 1 2 840 113554 1 2 1 2 }
2: { 1 2 840 113554 1 2 1 3 }
3: { 1 2 840 113554 1 2 1 4 }
Received message: "Hello World!"
Signature verified.
NOOP token

```

If you do not see this, check the [Testing and Troubleshooting](#) section of this wiki.

## 5.2. Testing SSH

Having tested Moonshot with gss-server and gss-client, you can now test Moonshot with a real application. In this case, SSH.

1. On the Live DVD, the identity "steve" that we just configured into the Identity Selector has an attribute associated with their account that SSH will use to map the login to a local account called "moonshot". So we need to create that local user account.
  - a. Open a terminal window (Applications > Accessories > Terminal).
  - b. Become the root user (type `su -`). Enter your root password when prompted (if booting directly from the Live DVD, this will be "Moonshot", otherwise, it will be whatever you set it as).
  - c. Add the "moonshot" user:

```
$ useradd -m moonshot
```

- d. Exit the root login (type `exit`).
2. The hostname that you use in the `ssh` command must match the output given running `hostname -f` on the server. This is a security feature whereby GSS checks the name the service thinks it has as compared to what you're trying to connect to.
  - a. Open a terminal window (Applications > Accessories > Terminal).
  - b. Become the root user (type `sudo -`). Enter your root password when prompted (if booting directly from the Live DVD, this will be "Moonshot", otherwise, it will be whatever you set it as).
  - c. Run `hostname` and then `hostname -f`.

```
$ hostname
samplehost
$ hostname -f
samplehost.localdomain
```

- d. If the hostnames do not match, set the hostname manually.

```
$ hostname samplehost.localdomain
```

- e. Exit the root login (type `exit`).
3. The SSH server requires some additional information to be passed back by the RADIUS server. This information is a SAML assertion that tells it what the local username is. This is not in the Live DVD by default.
  - a. Open a terminal window (Applications > Accessories > Terminal).
  - b. Become the root user (type `sudo -`). Enter your root password when prompted (if booting directly from the Live DVD, this will be "Moonshot", otherwise, it will be whatever you set it as).
  - c. Follow the instructions to [issue SAML Assertions hard-coded in the RADIUS Server](#). The '[your realm here]' entry in those instructions for your Live DVD is 'example.com'.

d. Restart FreeRADIUS.

```
$ /etc/init.d/freeradius restart
```

e. Exit the root login (type exit).

4. Now try connecting to the local SSH server. The SSH client on the Live DVD is already configured to use Moonshot as its preferred mechanism.

```
$ ssh moonshot@[hostname of your server]
```



For example, if your hostname was "moonshotdemo.dev.ja.net", you would run `ssh moonshot@moonshotdemo.dev.ja.net`

5. The Identity Selector will pop up. This time it will already contain the test account that we configured for the gss-server test - `steve@example.com`. We need to tell it to use this existing identity with the new service it has encountered - simply choose that identity and click "send".
6. If you see several lines of debugging, followed by "\$" where you can type commands... then you're in! You've just authenticated using Moonshot. Verify by typing `whoami` and hitting return, which should return the username you've authenticated as - `moonshot!`

## 6. Connect your new Service to the Moonshot infrastructure

At this point, you've verified that you have a working Moonshot-enabled SSH server. However, you haven't connected it to anything other than itself.

If you wish to take the test one step further and allow external users, with real federated accounts, to authenticate to your service, then you need to connect it to your local RP Proxy. Your organisation might already have one of these in the form of a Moonshot RP Proxy, an eduroam RADIUS server, or for a variety of other reasons. If so, follow these instructions:

1. Speak to your RADIUS server administrator. They will need to configure a new client for you in their RADIUS server configuration; they will give you the IP address of the RADIUS server and a shared secret to use.
2. On your system, open the file `/etc/radsec.conf` for editing:
  - a. Change the hostname parameter from the IP address of 127.0.0.1 to the IP address of your real RADIUS server
  - b. Change the secret to the real shared secret given to you by your RADIUS server admin.
  - c. Save the file.
3. Open the identity selector manually (located in Applications > System Tools > Administration). Choose the `steve@local` identity you created and remove it (hit delete).
4. Test as before - first with `gss-server/gss-client`, then with SSH. When the identity selector pops up, add a new identity. This type, the username, realm, and associated password will be the same as you use for your eduroam account (e.g. `joebloggs@camford.ac.uk / password`). Hit send.
5. If it worked - congratulations, you've just used Moonshot authentication using your real federated credentials! If it didn't... Support is available through Janet and through the [moonshot-community@jiscmail.ac.uk](mailto:moonshot-community@jiscmail.ac.uk) mailing list.