How to forward X over SSH from Ubuntu machine?

I have a machine running Ubuntu which I SSH to from my Fedora 14 machine. I want to forward X from the Ubuntu machine back so I can run graphical programs remotely. Both machines are on a LAN.

I know that the `-x` option enables X11 forwarding in SSH, but I feel like I am missing some of the steps.

What are the required steps to forward X from a Ubuntu machine to Fedora over SSH?

/ ssh / xorg / xforwarding

edited May 6 '11 at 20:26

Gilles
340k 58 606 1044

4 I know this is rather common, but I am having issues. A definitive answer for this question would be helpful for many. Lots of examples around seem omit important details. – Mr. Shickadance May 6 '11 at 17:41

6 Answers

X11 forwarding needs to be enabled on both the client side and the server side.

On the client side, the `-X` (capital X) option to `ssh` enables X11 forwarding, and you can make this the default (for all connections or for a specific conection) with `ForwardX11 yes` in `~/.ssh/config`.

On the server side, `X11Forwarding yes` must specified in `/etc/ssh/sshd_config`. Note that the default is no forwarding (some distributions turn it on in their default `/etc/ssh/sshd_config`), and that the user cannot override this setting.

The `xauth` program must be installed on the server side. If there are any X11 programs there, it's very likely that `xauth` will be there. In the unlikely case `xauth` was installed in a nonstandard location, it can be called through `~/.ssh/rc` (on the server!).

Note that you do not need to set any environment variables on the server. `DISPLAY` and `XAUTHORITY` will automatically be set to their proper values. If you run ssh and `DISPLAY` is not set, it means ssh is not forwarding the X11 connection.

To confirm that ssh is forwarding X11, check for a line containing `Requesting X11 forwarding` in the `ssh -v -X` output. Note that the server won't reply either way.

answered May 6 '11 at 20:26

Gilles
340k 58 606 1044

10 As others have mentioned, Gilles must be some sort of manifestation of Linux itself. I'd buy you a pint if I could. Much appreciated. – Mr. Shickadance May 6 '11 at 20:38

16 @user: No, you never need `xhost +`. `xhost` is from a gentler era when having a machine connected to the network meant you were trustworthy. `xhost +` means anyone who can spoof your IP can take control of your X server session. `ssh -X` will set up all the required authorizations. If X11 forwarding disabled in the server config, talk to your administrator; if that doesn't work, see Forwarding X11 over SSH if the server configuration doesn't allow it. – Gilles May 6 '11 at 22:52

2 Thanks for mentioning xauth! Lack of that on a barebones server was causing me trouble. – vasi Apr 9 '13 at 6:53

1 @KhurshidAlam It doesn't matter whether the server is also running a GUI environment. Check the permissions on the `.Xauthority` file. If using Red Hat or other system with SELinux, check the SELinux context, see unix.stackexchange.com/questions/36540/… – Gilles Jan 6 '14 at 12:30

1 @KhurshidAlam Your setup sounds fine, you can share the same keypair (it's purely a matter of security vs convenience). The permissions on `.Xauthority` have nothing to do with SSH. There's something unusual in your setup, post a new question and be sure to mention the permissions on `.Xauthority` (output of `ls -l ~/.Xauthority`), your distribution, and anything else that seems relevant. – Gilles Jan 6 '14 at 13:36

To get X11 forwarding working over ssh, you'll need 3 things in place.

1. Your client must be set up to forward X11.
2. Your server must be set up to allow X11 forwarding.
3. Your server must be able to set up X11 authentication.

If you have both #1 and #2 in place but are missing #3, then you'll end up with an empty DISPLAY environment variable.

Soup-to-nuts, here's how to get X11 forwarding working.

1. On your server, make sure /etc/ssh/sshd_config contains:

   ```
   X11Forwarding yes
   X11DisplayOffset 10
   ```

   You may need to SIGHUP sshd so it picks up these changes.

   ```
   cat /var/run/sshd.pid | xargs kill -1
   ```

2. On your server, make sure you have xauth installed.

   ```
   belden@skretting:~$ which xauth
   /usr/bin/xauth
   ```

   If you don't have xauth installed, you'll run into the "empty DISPLAY environment variable" problem.

3. On your client, connect to your server. Be certain to tell ssh to allow X11 forwarding. I prefer

   ```
   belden@skretting:~$ ssh -X blyman@the-server
   ```

but you may like

```
belden@skretting:~$ ssh -o ForwardX11=yes blyman@the-server
```

or you can set this up in your ~/.ssh/config.

I was running into this empty DISPLAY environment variable earlier today when ssh'ing into a new server that I don't administer. Tracking down the missing xauth part was a bit fun. Here's what I did, and what you can do too.

On my local workstation, where I am an administrator, I verified that /etc/ssh/sshd_config was set up to forward X11. When I ssh -X back in to localhost, I do get my DISPLAY set correctly.

Forcing DISPLAY to get unset wasn't too hard. I just needed to watch what sshd and ssh were doing to get it set correctly. Here's the full output of everything I did along the way.

```
blyman@skretting:~$ mkdir ~/dummy-sshd
blyman@skretting:~$ cp -r /etc/ssh/* ~/dummy-sshd/
cp: cannot open `/etc/ssh/ssh_host_dsa_key' for reading: Permission denied
cp: cannot open `/etc/ssh/ssh_host_rsa_key' for reading: Permission denied
```

Instead of using sudo to force copying my ssh_host_{dsa,rsa}_key files into place, I used ssh-keygen to create dummy ones for myself.

```
blyman@skretting:~$ ssh-keygen -t rsa -f ~/dummy-sshd/ssh_host_rsa_key
Generating public/private rsa key pair.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/blyman/dummy-sshd/ssh_host_rsa_key.
Your public key has been saved in /home/blyman/dummy-sshd/ssh_host_rsa_key.pub.
```

Rinse-and-repeate with -t dsa:

```
blyman@skretting:~$ ssh-keygen -t dsa -f ~/dummy-sshd/ssh_host_dsa_key
# I bet you can visually copy-paste the above output down here
```

Edit ~/dummy-sshd/sshd_config to point to the correct new ssh_host key files.

```
# before
blyman@skretting:~$ grep ssh_host /home/blyman/dummy-sshd/sshd_config
HostKey /etc/ssh/ssh_host_rsa_key
HostKey /etc/ssh/ssh_host_dsa_key

# after
blyman@skretting:~$ grep ssh_host /home/blyman/dummy-sshd/sshd_config
HostKey /home/blyman/dummy-sshd/ssh_host_rsa_key
HostKey /home/blyman/dummy-sshd/ssh_host_dsa_key
```

Fire up sshd on a new port in non-detach mode:

```
blyman@skretting:~$ sshd -p 50505 -f ~/dummy-sshd/sshd_config -d
sshd re-exec requires execution with an absolute path
```

Whoops, better correct that path:

```
blyman@skretting:~$ /usr/sbin/sshd -p 50505 -f ~/dummy-sshd/sshd_config -d
debug1: sshd version OpenSSH_5.5p1 Debian-4ubuntu6
debug1: read PEM private key done: type RSA
debug1: Checking blacklist file /usr/share/ssh/blacklist.RSA-2048
debug1: Checking blacklist file /etc/ssh/blacklist.RSA-2048
debug1: private host key: #0 type 1 RSA
debug1: read PEM private key done: type DSA
debug1: Checking blacklist file /usr/share/ssh/blacklist.DSA-1024
debug1: Checking blacklist file /etc/ssh/blacklist.DSA-1024
debug1: private host key: #1 type 2 DSA
debug1: setgroups() failed: Operation not permitted
debug1: rexec_argv[0]='/usr/sbin/sshd'
debug1: rexec_argv[1]='-p'
debug1: rexec_argv[2]='50505'
debug1: rexec_argv[3]='-f'
debug1: rexec_argv[4]='/home/blyman/dummy-sshd/sshd_config'
debug1: rexec_argv[5]='-d'
Set /proc/self/oom_adj from 0 to -17
debug1: Bind to port 50505 on 0.0.0.0.
Server listening on 0.0.0.0 port 50505.
debug1: Bind to port 50505 on ::.
Server listening on :: port 50505.
```

Pop a new terminal and ssh in to localhost on port 50505:

```
blyman@skretting:~$ ssh -p 50505 localhost
The authenticity of host '[localhost]:50505 ([::1]:50505)' can't be established.
RSA key fingerprint is 81:36:a5:ff:a3:5a:45:a6:90:d3:cc:54:6b:52:d0:61.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '[localhost]:50505' (RSA) to the list of known hosts.
Linux skretting 2.6.35-32-generic #67-Ubuntu SMP Mon Mar 5 19:39:49 UTC 2012 x86_64
GNU/Linux
Ubuntu 10.10

Welcome to Ubuntu!
 * Documentation:  https://help.ubuntu.com/

1 package can be updated.
0 updates are security updates.

Last login: Thu Aug 16 15:41:58 2012 from 10.0.65.153
Environment:
  LANG=en_US.UTF-8
```

```
  USER=blyman
  LOGNAME=blyman
  HOME=/home/blyman
  PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games
  MAIL=/var/mail/blyman
  SHELL=/bin/bash
  SSH_CLIENT=::1 43599 50505
  SSH_CONNECTION=::1 43599 ::1 50505
  SSH_TTY=/dev/pts/16
  TERM=xterm
  DISPLAY=localhost:10.0
Running /usr/bin/xauth remove unix:10.0
/usr/bin/xauth add unix:10.0 MIT-MAGIC-COOKIE-1 79aa9275ced418dd445d9798b115d393
```

Look at the last three lines there. I fortuitously had DISPLAY set, and had those two nice-looking lines from /usr/bin/xauth.

From there it was child's play to move aside my /usr/bin/xauth to /usr/bin/xauth.old, disconnect from ssh and stop the sshd, then launch sshd and ssh back in to localhost.

When /usr/bin/xauth was gone, I didn't see DISPLAY reflected in my environment.

There's nothing brilliant going on here. Mostly I got lucky in choosing a sane approach to try reproducing this on my local machine.

answered Aug 30 '12 at 19:03

Belden
491 4 3

Wow, thank you so much for your anwer. I was doing everything good except the `export DISPLAY=:10` . I never guessed that number of display. – erm3nda Oct 31 '15 at 22:34

The fix is to add this line to your `/etc/ssh/sshd_config` :

```
X11UseLocalhost no
```

https://joshua.hoblitt.com/rtfm/2013/04/how_to_fix_x11_forwarding_request_failed_on_channel_0/

edited Aug 3 '13 at 11:06

Anthon
44k 14 58 116

answered Aug 3 '13 at 10:49

Ace
61 1 1

I have 2 Ubuntu server. On the one I needed it set to yes, on the other it had to be no. I am sure there is a explanation, but it's worth to try both. – alfonx Aug 19 '15 at 13:08

For me the problem was in **nodev** mount option for /tmp filesystem. X11 needs a special file to be created in there.

So check what are the mount options for /tmp filesystem if you use a separate partition or disk for that.

answered Jun 19 '14 at 10:24

yakovpol
1

I think you might want to see other answers to the original question and take a moment to think how your own answer improves on them. – Sami Laine Jun 19 '14 at 11:28

Add `X11UseLocalhost no` to `/etc/ssh/sshd_config` and restart the SSH server.

If you get no DISPLAY, check if xauth is installed correctly and then try

it again.

RHE/CEntos doesn't have this issue, this is a Ubuntu thing!

edited Aug 1 '14 at 9:31

polym
4,193 1 14 49

answered Aug 1 '14 at 9:28

stephen cooke
1

`X11Forwarding` must be set on the SSH server (in your case the Ubuntu box) in its `sshd_config` , and you must allow X11 to be forwarded for the SSH client (your Fedora box) by passing the `-x` option or editing the `ssh_config` file to add the `ForwardX11` default.

edited Sep 24 '15 at 20:46

underscore_d
138 11

answered May 6 '11 at 18:05

Caleb
37.1k 5 96 136

1 You also need `xauth` installed on the remote machine, otherwise the x authority stuff won't work. – Faheem Mitha May 6 '11 at 18:22

What about setting `DISPLAY` ? – Mr. Shickadance May 6 '11 at 18:36

1 ssh will automatically set `$DISPLAY` if `X11Forwarding` is enabled and `xauth` is present on the client system. – Shadur May 6 '11 at 18:58