

Setting Up SSH Passthrough

Category: Security & Logging In

Summary: By investing some time to set up SSH passthrough, you can make your future NAS logins and inbound file transfers easier and faster.

The SSH passthrough feature allows you to log into any system in the enclave by typing just one SSH command. *Without* passthrough, you have to log into an SFE, and then log into pfe[20-27], bridge[1-4], cfe2, or Lou[1-2]. *With* SSH passthrough you "pass through" the SFEs directly to a NAS high-end computing system where you will do most of your work.

After following the steps below, you can use SSH from your local host to log into a NAS high-end computing system and be prompted for only your SecurID passcode (your PIN plus fob tokencode) and password. The public key authentication is handled automatically, so you will not be prompted for the passphrase of your private/public keys.

Before You Start

Before you are ready to set up SSH passthrough, you must already have gone through the initial steps of logging into the NAS HECC enclave, as outlined in [Security Overview](#). That is, you need to have done the following steps, in the order listed below.

- [Enabled your fob, and created your password and pin](#)
- Mastered [two-step login with password](#)
- Set up [public key authentication](#) (creating SSH Public/Private key pair, and copying SSH public key to SFEs)
- Mastered [two-step login with passcode](#)

How To Set Up SSH Passthrough

At this point, you are only three steps away from streamlining all your future logins and inbound file transfers. You will be able to log in quickly to any host in the NAS HECC enclave, and you will be able to copy files from your local host to any NAS host without first copying the files to an SFE.

Step 1: Copy OpenSSH Public Key to Hosts Inside the Enclave

Hosts inside the enclave use [OpenSSH](#), so you will need to copy the OpenSSH version of your public key to the hosts inside the enclave and place the key in your

.ssh/authorized_keys file. *This must be done for every system inside the enclave to which you want to connect using SSH passthrough.*

The following example uses lou2.nas.nasa.gov as the enclave host and sfe1.nas.nasa.gov as the secure front end.

Substitute your NAS username for *username@*. If your local host username and your NAS username are the same, you can skip the *username@*.

Copy your OpenSSH Public Key

If you have done the Pre-Steps above, you already have your public key on one or more of the bastion front ends, sfe[1-4].

TIP: If you don't have an **.ssh** directory on the NAS host (in this example Lou2), make sure that you create one (log in to Lou2 and issue the command **mkdir .ssh**) before issuing the **scp** command below. Otherwise, the command will copy the file **id_rsa.pub** to Lou2 with the filename **.ssh**.

On your local host, type:

```
your_localhost% ssh username@sfe1.nas.nasa.gov
```

On sfe1, type:

```
sfe1% scp .ssh2/id_rsa.pub username@lou2:~/.ssh
```

Add Your OpenSSH Public Key to Your .ssh/authorized_keys File on the Enclave Host

On sfe1, type:

```
sfe1% ssh username@lou2
```

On lou2.nas.nasa.gov, type:

```
lou2% cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
```

If you get the error **"/u/username/.ssh/authorized_keys: No such file or directory"** after issuing the above command, you likely have **set noclobber**, which prevents you from overwriting files. You can use the command **unset noclobber** first to avoid this problem.

WARNING: The permission for the **authorized_keys** file must be set to 600 (issue the command **chmod 600 authorized_keys**). Group/others write permissions on **/u/username** and **/u/username/.ssh** are not allowed for public key authentication. Repeat this step for the other NAS hosts you plan to use, such as bridge[1-4] or a PFE. That is, add your public key to the **.ssh/authorized_keys** file on the other hosts.

Step 2: Modify the .ssh/config File on Your Localhost

In your `~/ .ssh/config` file on your localhost, add the entries for the hosts inside the enclave you want to access. If you do not have a `~/ .ssh/config` file, create a new file called `config` in your `~/ .ssh` directory and add the entries. For your convenience, we provide a template.

Download the template `.ssh/config` (a text file named `config_nas.txt`). Move this template to your `.ssh` directory on your localhost and rename it to `config`.

The contents of this file are also shown below. In this template, `sfe1` is used; you can switch to `sfe2`, `sfe3`, or `sfe4` if you want to use a different secure front end for SSH passthrough. Replace `<NAS_login_name>` with your NAS username.

If you will be using this file to access additional hosts outside of NAS, see the note "Customizing Your Own `.ssh/config`" below.

```
Host sfe
# Replace sfel by sfe2, sfe3, or sfe4 if sfel is unavailable
HostName                 sfel.nas.nasa.gov

Host sfe sfe?.nas.nasa.gov
Ciphers                   aes128-cbc,aes192-cbc,aes256-cbc,3des-cbc
ForwardAgent             no
MACs                     hmac-sha1

Host sfe sfe?.nas.nasa.gov dmzfs?.nas.nasa.gov sup*.nas.nasa.gov
LogLevel                 info
ProxyCommand             none

Host pfe pfe-last pfe.nas.nasa.gov pfe-last.nas.nasa.gov
HostKeyAlias              pfe20.nas.nasa.gov
ProxyCommand             ssh -oCompression=no sfe /usr/local/bin/ssh-balance %h

# Add additional hosts to the list below as needed
Host *.nas.nasa.gov lou lou? cfe? pfe? bridge? sfe pfe pfe-last
ForwardAgent             yes
HostbasedAuthentication no
Protocol                 2
ProxyCommand             ssh -oCompression=no sfe /usr/local/bin/ssh-proxy %h
ServerAliveInterval      10m

# Replace <NAS_login_name> with your NAS username
User                     <NAS_login_name>

# Enabling compression may improve performance for slow connections
#Compression             yes

# Uncomment this line if you are using OpenSSH 4.7 or later
#MACs                    umac-64@openssh.com,hmac-md5,hmac-sha1
```

Customizing Your Own .ssh/config File

If you use your `.ssh/config` file for accessing both NAS systems and systems at other sites, you can add entries at the top of the above template. The entries take the form:

```
Host hostname
ProxyCommand ssh username@hostname.nas.nasa.gov /usr/local/bin/ssh-proxy hostname
```

Hostname is the name of the host you want to access. It can be abbreviated as *hostname* (such as `pfe20`) or can be the fully-qualified domain name (such as `pfe20.nas.nasa.gov`). Note that using `bbftp` requires the fully qualified domain name, so it is a good idea to include both.

Step 3: Set Up SSH Agent

The `ssh-agent` program holds and manages the private key on your local host and responds to key challenges from remote hosts. The private key is not initially stored in the agent and is added through the `ssh-add` program.

Typically, `ssh-agent` is started at the beginning of an X session or a login session, and you provide your passphrase to unlock your private key for this originating session. For any SSH connection to a remote host (for example, `sfe1`) made from this original session, the `ssh-agent` remembers your private key and will respond to challenges automatically without prompting you to type in your passphrase again.

How SSH passthrough works: If you want to use SSH to connect from the first remote host (for example, `sfe1`) to a second remote host (for example, `pfe20`) and possibly from the second remote host to a third remote host, a feature called **agent forwarding** allows a chain of SSH connections to forward all the key challenges back to the original agent, thus eliminating the need for using a password or public/private keys for these connections.

Note: In order for agent forwarding to work, your public key must be installed in all the remote hosts that you intend to connect to. See Step 1, above.

Setting up SSH Agent on UNIX or LINUX Systems

If you use `csh` or `tcsh`, to launch `ssh-agent`, type the following command:

```
your_localhost% eval 'ssh-agent -c'
```

Or, if you use `sh` or `bash`, to launch `ssh-agent`, type the following command:

```
your_localhost% eval 'ssh-agent -s'
```

To add your private key to **ssh-agent**, type the following command:

```
your_localhost% ssh-add private_key
```

Example:

```
your_localhost% ssh-add ~/.ssh/id_rsa  
Enter passphrase for /Users/username/.ssh/id_rsa: type your passphrase
```

Setting up SSH Passthrough may be complicated, but it is worth doing to save time in the future.

Article ID: 232

Last updated: 17 Dec, 2012

The HEC Environment -> Security & Logging In -> Setting Up SSH Passthrough

<http://www.nas.nasa.gov/hecc/support/kb/entry/232/?ajax=1>