Using SSH Key Authentication On The Unified Linux Systems

The SICE Linux systems have started requiring DUO 2-factor authentication for all SSH logins using password authentication. If you are not able to use DUO or you require unattended SSH connections then you will have to use SSH key authentication instead of your normal account passphrase. This KB page documents the procedures for setting this up for various combinations of Operating System and SSH Client including:

- Windows + PuTTY
- Windows + WinSCP
- Linux + OpenSSH
- MacOS + OpenSSH
- MacOS + Cyberduck

Note that you may also want to use SSH authentication for access to group accounts where you may not even know the group account passphrase. This provides a convenient way to grant and revoke login access to group accounts in a way that provides more accountability than giving out the passphrase to multiple people where accountability is lost.

Windows + PuTTY

Note that these instructions assume you want to log into your own account on the linux system. If, instead, you want to use your ssh key to log into some other account (like a shared group account) then you will follow the same steps except you will alter the username you use (step 8) and add your public key to the authorized_keys file in the group account instead of your own account.

1. If not already installed, you must first install PuTTY. If you use the normal windows installer for PuTTY you will also get PuTTYgen but if you just downloaded PuTTY.exe then also get PuTTYgen.exe. On the SICE-managed Windows systems you should be able to install it from the Software Center, if not already installed.
2. Open a command prompt window (cmd) on your Windows PC and run puttygen
3. Click the 'Generate' button to generate a new keypair and move the mouse around, as directed, to generate randomness.
4. Type in the passphrase you will use with the key in the 'Key passphrase' and 'Confirm passphrase' boxes.
5. Click the 'Save private key' button and save the key in .ppk format. You can save it anywhere and give it any name you want but you will need to know where the key is in the following steps.
6. Click the 'Save public key' button and also save that. I recommend giving it the same name as the private key put with the .pub extension.
7. Go ahead and 'cut' the text from the OpenSSH key box of puttygen and leave that window open. We will need that in the following steps.
8. Fire up PuTTY and enter the Host Name (eg. silo.sice.indiana.edu). Note that you can also use the username@hostname syntax (eg. janedoe@silo.sice.indiana.edu) to also specify your username so you aren't prompted for that each time.
9. In PuTTY, then go to Connection > SSH > Auth and click the 'Browse' button under 'Private key file for authentication' and select the private key .ppk file saved in step 5 above.
10. Go back to the main Session screen in PuTTY and enter a name for this login session under 'Saved Sessions' and click Save.
11. Click Open to open a connection to the specified host. If this is the first time trying to log into this server you will have to accept the host key.
12. Log in with your username and IU passphrase (not your ssh key passphrase). The first time you do this you may see a message 'Server refused our key' but ignore that.
13. Once logged into a SICE Linux system, you will create a file named "authorized_keys" in your ~/.ssh folder. That is just a text file that contains the public keys (one per line) that you want to be able to use for logins. So, just paste the contents of the OpenSSH public key as shown in the puttygen window (the really long line in the box at the top that starts with something like ssh-rsa). When you are done, you should have a single very long line in your ~/.ssh/authorized_keys file on the linux system.
14. Log out and then log in again. Use the same saved session you created above by just double clicking the session saved name.
15. This time, it should prompt you for the passphrase for your saved SSH key so use the passphrase you used to create the key in puttygen. Since you are using an SSH key you will not have to use DUO as your 2nd factor in the authentication since you have 2-factor authentication this way with the SSH key plus the passphrase.

Windows + WinSCP

1. You must first set up your keys as noted in the above Windows + PuTTY section. This includes added the public key to the remote linux server authorized_keys file.
2. Fire up WinSCP. On the SICE-managed Windows systems you should be able to install it from the Software Center, if not already installed.
3. Click Advanded and go to SSH > Authentication > Private key file: and browse to the private key file you created in step 1.
4. Click OK to go back to the main configuration screen
5. Make sure the File protocol is set to SFTP
6. Set the hostname (eg. silo.sice.indiana.edu)
7. Set the username to your normal IU username
8. Click Save. Note that you can set the ssh key passphrase on the configuration page and then click 'Save password' on the Save screen if you want to save the password but, as noted on this screen, this is not recommended for security reasons.
9. Once the config is saved, just double click on it from the main WinSCP screen to start it up. The first time you do this you will have to accept the host key.
10. When prompted for the passphrase for your ssh key (assuming you didn't ignore the security warning and save it.

Linux + OpenSSH

1. Generate your SSH keypair by running "ssh-keygen" on the system you are wanting to use to log into the SICE linux systems. For logging into SICE linux systems from other SICE linux systems this can be run on any of the SICE Linux systems. If you are logging in from a personal linux computer, run this on that system.
2. When prompted, it is recommended that you accept the default location for the key file.

3. Enter a passphrase for the key that you will use when logging in. You are advised to use a secure passphrase for this and that you not use an empty passphrase (unless you know EXACTLY what you are doing).
4. This will generate two files, the private key files (probably something like id_rsa by default) and the public key file (probably something like id_rsa.pub by default). The private key file MUST be kept secure and not shared. The public key is safe to share and will be used for access to the remote systems.
5. In your SICE linux account, you will create a file named "authorized_keys" in your ~/.ssh folder. That is just a text file that contains the public keys (one per line) that you want to be able to use for logins. So, just cut/paste the contents of the public key file (eg. id_rsa.pub) into your authorized_keys file. Note that these public key lines are very long so be sure you don't inadvertently introduce line breaks when doing the cut /paste.

Once you have this set up, you should be able to log into the SICE linux system and see something like:

```
$ ssh janedoe@silo.sice.indiana.edu
Enter passphrase for key '/home/janedoe/.ssh/id_rsa':
```

Note that this is prompting for the passphrase you used when you generated your keypair (above). If, instead, you see something like this then it is NOT using your SSH key:

```
$ ssh janedoe@silo.sice.indiana.edu
janedoe@silo.sice.indiana.edu's password:
```

This is prompting for your normal account password instead of your ssh key passphrase so something went wrong. Here are a few things to check if this happens:

1. Make sure the keypair was created on the system where you are running the ssh.
2. Make sure the keypair was saved to the default location when you ran ssh-keygen. If you saved it to some non-standard location the ssh won't find it (unless you explicitly tell it where the key file lives)
3. Make sure the permissions are all correct. You must make sure the private key file is not readable by anyone but yourself and that your .ssh directory and authorized_keys files on the SICE system are not writable by anyone other than yourself.
4. You can use the -v (or -vv or -vvv) flag to enable various levels of debugging which may be helpful if you are still having trouble.

MacOS + OpenSSH

The instructions on the Mac are essentially identical to those for OpenSSH on Linux so refer to the previous section.

MacOS + Cyberduck

Note that Cyberduck is available to install on the SICE managed Macs in the Self Service application.

1. You must first set up openssh keys as noted in the previous MacOS + OpenSSH section. This includes added the public key to the remote linux server authorized_keys file.
2. Fire up Cyberduck. Note that Cyberduck is available to install on the SICE managed Macs in the Self Service application.
3. Click the '+' icon in the lower left corner
4. Set the protocol to SFTP and fill out the form as follows
    a. Nickname: any name you want to use
    b. Server: the DNS name of the remote linux server (eg. silo.sice.indiana.edu)
    c. Username: your IU username
    d. SSH Private Key: Select the key created in step 1 (eg ~/.ssh/id_rsa)
    e. Password: The passphrase you used when creating your key via ssh-keygen
5. Once this is filled out, close the window
6. You should now see the newly created bookmark so double click to start it
7. The first time you run it, it will ask you to accept the remote host key. Check always and accept the key.
8. If that all works, you will then get logged in without having to use Duo.