Proxy

Sometimes you will be on a machine without access to the internet and you need to get a file downloaded. You could first download the file somewhere else and then transfer it to the machine on the private network. This get tedious... hence the reason for this article.

Some notes before we begin:

1. Anywhere CS infrastructure is specified it may be possible to substitute any similar UNIX host.
2. I assume you use the 'bash' shell.

Initial setup

If you have not done so already you will need to setup passwordless authentication to the server you wish to connect to.

SOCKS

Read up on SOCKS proxies here: https://en.wikipedia.org/wiki/SOCKS

Where the port number is set to '20000', you can set this to any number between 1025-65535. You may find you will actually need to change this number if some other process is already using that port.

Now we can create a SOCKS proxy without a password:

```
user@computer:~$ ssh -f -C -q -N -D20000 user@linux.cs.uchicago.edu
```

Notice how the ssh session is now running in the background:

```
user@computer:~$ ps aux | grep linux1
user 11995  0.0  0.0  58240  1016 ?        Ss   11:04   0:00 ssh -f -C -q -N
-D20000 user@linux
user 11997  0.0  0.0  11740   916 pts/2    S+   11:04   0:00 grep linux
```

Set the ALL_PROXY environment variable:

```
user@computer:~$ export ALL_PROXY='socks5://127.0.0.1:20000'
```

Any tool that respects this environment variable will tunnel its traffic through the proxy we just setup.

```
user@computer:~$ git clone https://github.com/foo/bar.git
Cloning into 'bar'...
remote: Counting objects: 120662, done.
remote: Compressing objects: 100% (48/48), done.
```

```
remote: Total 120662 (delta 23), reused 0 (delta 0), pack-reused 120608
Receiving objects: 100% (120662/120662), 36.26 MiB | 5.24 MiB/s, done.
Resolving deltas: 100% (74544/74544), done.
Checking connectivity... done.
Checking out files: 100% (1456/1456), done.
```

Automatically setup a SOCKS proxy

Now to make it automatic on login. Add the following to ~/.profile or ~/.bash_profile Remove the 'echo' commands if you wish.

```
if [[ $(nc -z 127.0.0.1 20000; echo $?) -eq 0 ]]; then
  echo 'proxy already running'
else
  echo 'setting up proxy'
  ssh -f -C -q -N -D20000 user@linux.cs.uchicago.edu
  export ALL_PROXY='socks5://127.0.0.1:20000'
fi
```

Git

First you will need to setup passwordless authentication.

Add the following to your `~/.ssh/config` file:

- Replace `CNETID` with your actual CnetID.
- Replace `id_rsa` with the private key you use to ssh to 'linux.cs.uchicago.edu' without a password.

```
Host github.com
  User CNETID
  ProxyCommand ssh -i ~/.ssh/id_rsa CNETID@linux.cs.uchicago.edu nc %h %p
  IdentityFile ~/.ssh/id_rsa
```

HTTP(S)

You can also use our HTTP proxy server.

Setup

In the shell type the following:

```
export http_proxy=http://webproxy.cs.uchicago.edu:3128
export https_proxy=https://webproxy.cs.uchicago.edu:3128
```

SSH Port Forwarding

This has many uses, but one use case we see a lot is to proxy a database connection.

You can also do some port forwarding magic with ssh. This is known to work on macOS and Linux based OS's.

The following command will open an ssh session to one of the linux machines. Once you've signed in, you can minimize this terminal window. Port 5432 on your local machine has been forwarded through the SSH connection to port 5432 on $dbserver.

```
user@mymachine:~$ ssh -L 5432:$dbserver:5432 <cnetid>@linux.cs.uchicago.edu
```

Now by connecting to localhost:5432 we are actually connecting to the remote server. We can omit '-p 5432' because it is the default. e.g. `user@mymachine:~$ psql -p 5432 -h localhost -U <username> <database-name>`

```
user@mymachine:~$ psql -h localhost -U $username $dbname
```

SSHFS

- Community tutorial

This tutorial assumes mac or linux. See the community tutorial if you wish to use windows.

Install SSHFS

Debian/Ubuntu

```
apt-get install sshfs
```

MacOS: Download and install both packages from this site: https://osxfuse.github.io/

Create Local directory

Create a directory on your local machine to which you wish to mount your CS home directory to.

```
mkdir $HOME/cshome
```

Mount

```
sshfs -o idmap=user,reconnect,allow_other,default_permissions
cnet@linux?.cs.uchicago.edu:/home/cnet/ $HOME/cshome/
```

From:
https://howto.cs.uchicago.edu/ - **How do I?**

Permanent link:
https://howto.cs.uchicago.edu/nix:proxy?rev=1610655017

Last update: **2021/01/14 14:10**