# SSH Access

One of the services under constant threat from people trying to get unauthorised access to University systems is SSH. Although a secure protocol in the sense that network traffic is encrypted, finding a way in through a crackable password or a badly configured server remains common, A decision was therefore taken to restrict outside access to SSH servers by blocking it at the University firewall.

## University VPN

The University provides a VPN service in order to connect to internal services from outside of the University network, this includes SSH.

Details of the VPN service can be found here:

https://www.nottingham.ac.uk/it-services/connect/working/off-campus.aspx#vpn

## CS Bastion Hosts

> **Warning: Use non-default SSH port 2222 on the Bastions**
>
> After 21/7/2021 access to the Bastions on default SSH port 22 will not work. Instead please use port 2222 on the Bastions, only (far-end servers will still use port 22 by default for SSH).

Computer Science also provides Bastion Hosts for SSH as an alternative to the VPN service where it's more appropriate. The remainder of this document describes this service allowing you to 'hop' through one of these SSH hosts with valid University login credentials.

Computer Science Bastion Hosts:

- valley.cs.nott.ac.uk (Staff or researcher login required)
- canal.cs.nott.ac.uk (Taught student login required)

## Command Line (including recent Windows 10 releases)

If you have a recent version of ssh installed this will support the **-J** flag to jump through one of our bastion hosts.

*The WIndows 10 version of ssh may contain a bug preventing connecting using the -J flag, please use ProxyCommand in this instance.*

```
ssh -J <username>@<jump host>:2222 <username>@<destination host>
```

To connect to severn from outside the University staff and researchers would therefore use:

```
ssh -J <username>@valley.cs.nott.ac.uk:2222 <username>@severn
```

And taught students would be able to reach avon with:

```
ssh -J <username>@canal.cs.nott.ac.uk:2222 <username>@avon
```

Should you find the -J flag is not supported you can specify a ProxyCommand:

```
ssh -o ProxyCommand="ssh -W %h:%p -p 2222 <username>@valley.cs.nott.ac.uk" <username>@severn
```

or

```
ssh -o ProxyCommand="ssh -W %h:%p -p 2222 <username>@canal.cs.nott.ac.uk" <username>@avon
```

## Command Line - Advanced

If you have OpenSSH 7.3 or later installed, you may wish to edit your ~/.ssh/config file to avoid specifying the jump host each time, a member of staff /researcher could use the following:

```
Host severn.cs.nott.ac.uk
    ProxyJump <username>@valley.cs.nott.ac.uk:2222
```

Now, typing **ssh** <*username*>**@severn.cs.nott.ac.uk** will automatically use the Jump Host.

Another useful trick, if you will be making use of multiple connections to the final destination host you can multiplex them. Doing this you will be only prompted to login once:

```
mkdir -m 0700  ~/.ssh/cm
```

And add Control lines to your ~/.ssh/config file:

```
Host severn.cs.nott.ac.uk
    ProxyJump <username>@valley.cs.nott.ac.uk:2222
    ControlPath ~/.ssh/cm/%r@%h:%p
    ControlMaster auto
    ControlPersist 10m
```

You should now not be prompted for your usename and password for subsequent connections after an initial authenticated one for as long as that connection stays open.

## Windows PuTTY/plink

*If you have a recent version of WIndows 10 you may find it easier to use ssh from the command line as documented above rather than using PuTTY.*

If you have PuTTY installed on Windows along with plink (which is installed as part of the full installation package) you can use plink to create a tunnel through the bastion hosts.

Open up a command prompt and as a member of staff/researcher type:

```
plink  -N -L 2201:severn.cs.nott.ac.uk:22 -P 2222 <username>@valley.cs.nott.ac.uk
```

or a taught student:

```
plink  -N -L 2201:avon.cs.nott.ac.uk:22 -P 2222 <username>@canal.cs.nott.ac.uk
```

You will be prompted for your username and password to the bastion host. Press return to start the session and you now have a tunnel to the school server.

You now need to just open putty and connect to:

- Hostname:localhost
- Port: 2201
- Connection type: SSH