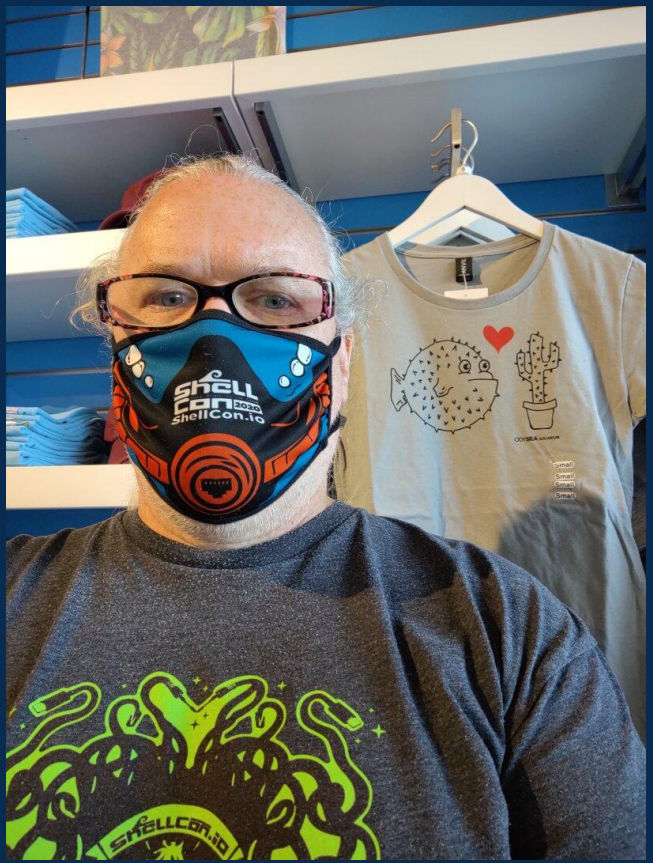
SSH Tunnels and More

Michigan!/usr/group
2021Aug10 @ 19:00 EDT
jitsi

der.hans
CDE
Object Rocket, a rackspace company
https://www.ObjectRocket.com/

Research Trip



Presumed Knowledge

Basic SSH usage

Filesystem permissions required by SSH

Basic SSH key and fingerprint usage

SSH

SSH == Secure SHell

OpenSSH is an OpenBSD project

Essential tool for system administrators and DevOps

Creates secure, authenticated, encrypted connections between computers

Allows passing data across encrypted TCP connections

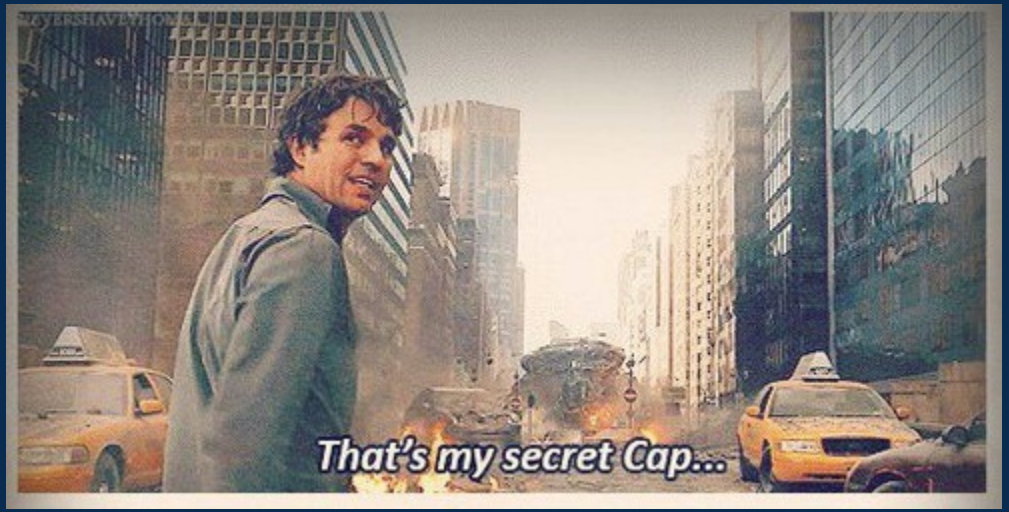
Requires an account on the remote computer

Creates secure, authenticated, encrypted connections across hostile networks

Yes

Yes, I said "hostile networks"

There’s a Secret



The Internet is a hostile network

Basic Connection

```
ssh -p 22 remote.example.com
```

Profit

```
// include::security_dissertation.adoc[]
```

Basic Tunnel

```
ssh firewall.example.com -L 2222:firewall.example.com:22
```

Entering the Tunnel From a Different Shell

```
ssh -p 2222 localhost
```

```
scp -P 2222 -pr local_dir_to_sync localhost:
```

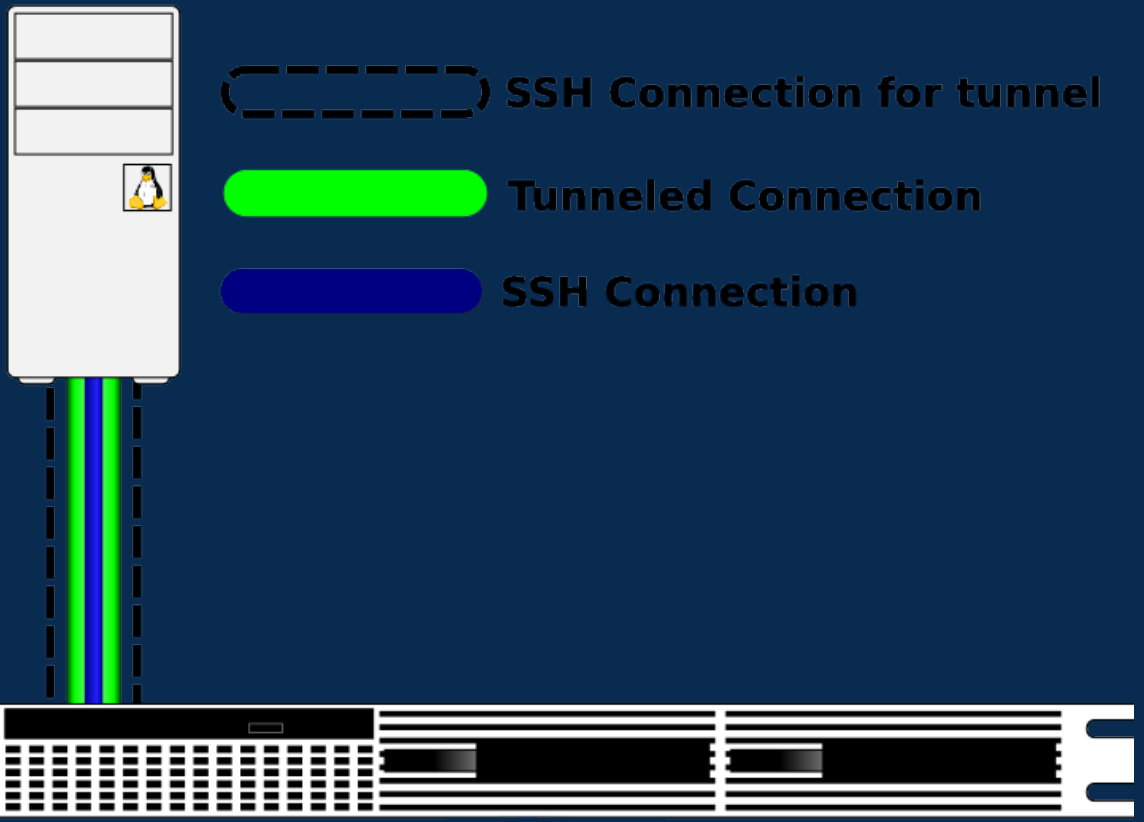
Tunnel Diagram

```
ssh -L 2222:firewall.example.com:22 firewall.example.com
```

```
ssh -p 2222 localhost
```

```
scp -P 2222 -pr rem_dir_to_sync localhost:
```

Your Machine



Gateway

Sock it to me?

`ssh -L 2222:127.0.0.1:22 demo ssh -p 2222 localhost`

```
root@demo:~# ss -tp state established '( dport = :ssh or sport = :ssh )'

Recv-Q     Send-Q          Local Address:Port            Peer Address:Port      Process

0          0                   192.168.100.130:ssh         192.168.100.1:56470
users:(("sshd",pid=7641,fd=4),("sshd",pid=7606,fd=4))

0          0                   127.0.0.1:37070              127.0.0.1:ssh
users:(("sshd",pid=7641,fd=11))

0          0                   127.0.0.1:ssh                127.0.0.1:37070
users:(("sshd",pid=7677,fd=4),("sshd",pid=7654,fd=4))

root@demo:~# hostname -I

192.168.100.130

root@demo:~#
```

Going in Reverse

```
ssh firewall.example.com -R 2222:firewall.example.com:22
```

Entering the Tunnel From the Remote System

```
ssh -p 2222 localhost
```

Where’s localhost?

hostname

```
ssh firewall.example.com -R 2222:firewall.example.com:22
```

localhost

```
ssh firewall.example.com -R 2222:localhost:22
```

Tunneling Via 3rd Party

Create Reverse Tunnel from host2

```
host2$ ssh -R 2222:firewall.example.com:22 firewall.example.com
```

Create Local Tunnel from host1

```
host1$ ssh -L 2222:firewall.example.com:2222 firewall.example.com
```

Simplified Bastion Tunnel

```
host2$ ssh -R 2222:localhost:22 firewall.example.com
```

```
host1$ ssh -L 2222:localhost:2222 firewall.example.com
```

Double Reverse

```
host1$ ssh -L 2222:localhost:2222 -R 3333:localhost:22 firewall.example.com
```

```
host2$ ssh -L 3333:localhost:3333 -R 2222:localhost:22 firewall.example.com
```

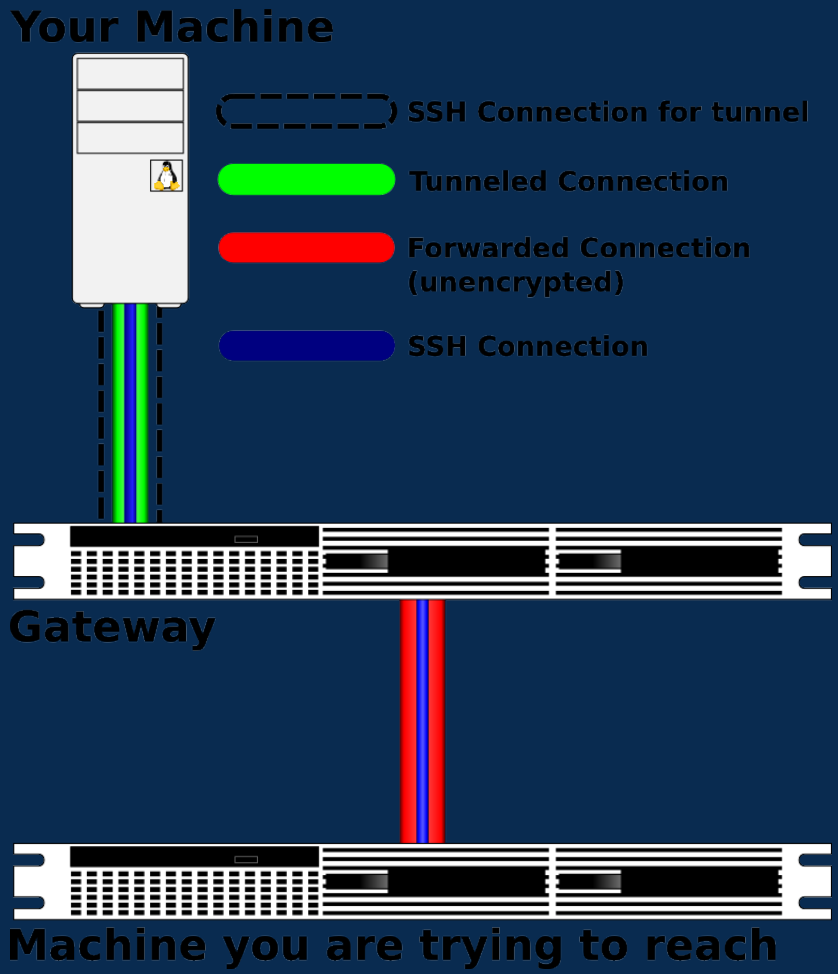
Throwing the Connection

```
ssh -N -f -L 3306:mysql.example.com:3306 firewall.example.com
```

```
$ grep 3306 /etc/services
mysql           3306/tcp
mysql           3306/udp
```

Careful of the Unencrypted Leg

```
desktop <====encrypted====> firewall <----NOT encrypted----> internalserver
```



SOCKS

```
ssh -D 1080 firewall.example.com
```

TIP | FoxyProxy

Getting Graphical

laptop$ `ssh -Y desktop.example.com`

desktop$ `firefox -new-instance -ProfileManager presentation`

Sandboxing Via VM or Container

desktop$ `ssh -Y vm.example.com`

vm$ `firefox -new-instance -ProfileManager javascriptIsDangerous`

Keys

```
ssh-keygen -f .ssh/id_new
```

authorized_keys

ssh-copy-id

Service Examples: MySQL

```
ssh -N -f -L 3306:db.example.com:3306 firewall.example.com
```

```
mysql -h localhost -p 3306 --protocol=TCP
```

Use 127.0.0.1

```
mysql -h 127.0.0.1
```

Service Examples: Email

```
ssh -L 2143:imap.example.com:143 -L 2993:imap.example.com:993 -L
2025:smtp.example.com:25 firewall.example.com
```

Service Examples: Web

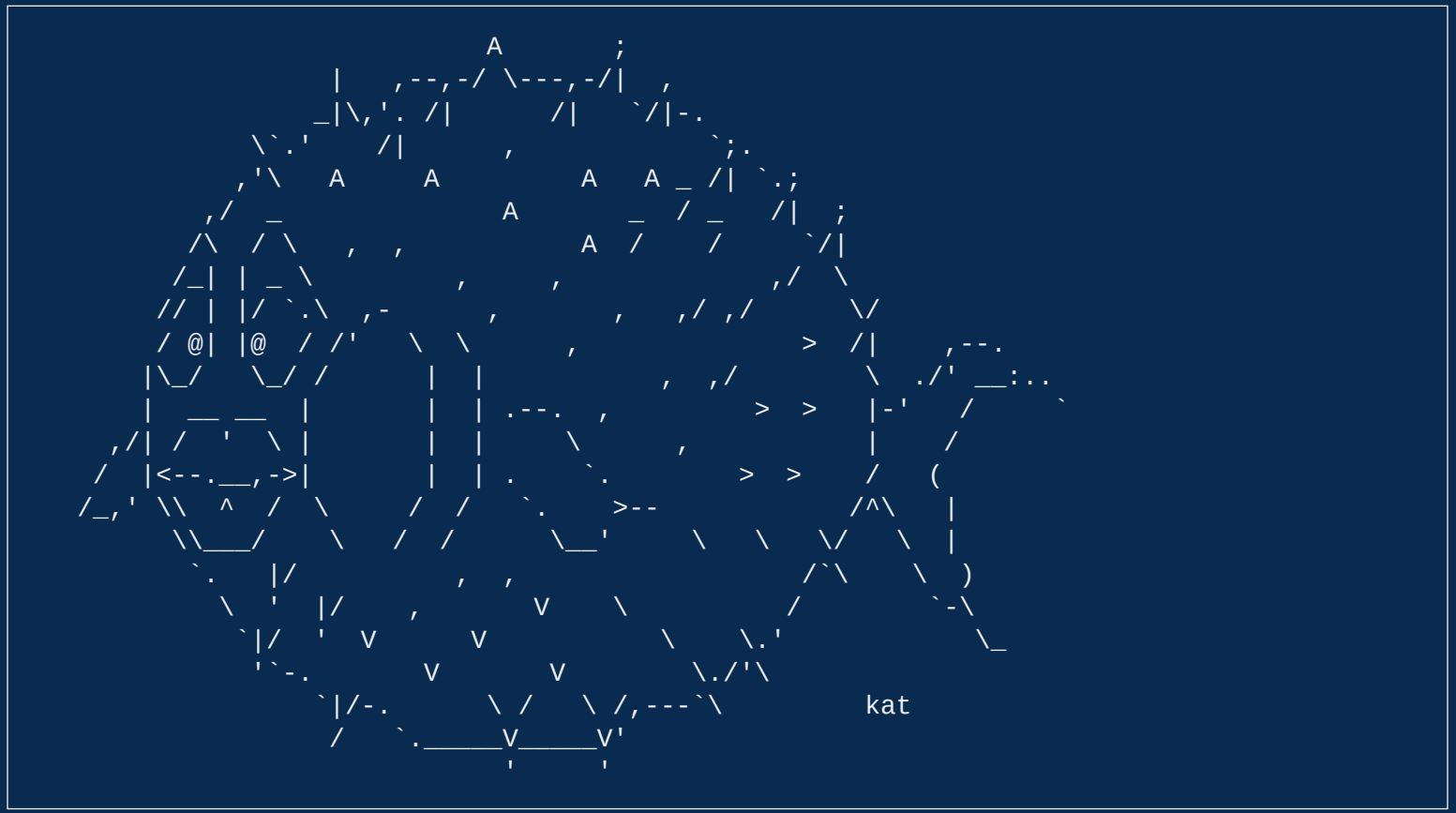
```
ssh -L 8080:www.example.com:80 firewall.example.com
```

```
links -http.extra-header "Host: www.example.com" http://localhost:8080/
```

```
ssh -D 1080 firewall.example.com
```

FoxyProxy

ASCII Puffy



~/.ssh/config Tips

Multiple known_hosts files
For instance, one for normal use, one imported regularly from orchestration
`UserKnownHostsFile ~/.ssh/known_hosts ~/.ssh/known_hosts_automagic`

~/.ssh/authorized_keys Tips

Per key restictions in authorized_keys entries
Important for keys for automated tasks
`no-port-forwarding,no-X11-forwarding,no-agent-forwarding,no-pty`

Force a per key specific command (aka forced command)
Runs the specified command rather than whatever was requested by the client
`command="hostname"`
`command="echo ${SSH_ORIGINAL_COMMAND} >>/tmp/what_did_they_run.log`

Specify per key network access restrictions
`from="localhost,127.0.0.1"`

Command Line Tips

-N == no remote command
-f == background after authentication

```
ssh -N -f -L 2222:localhost:22 firewall.example.com
```

-v == verbosity, maximum of 3

-G == show configuration that would be used

-t == force pseudo-terminal

```
ssh -p 2222 localhost screen -x myScreen
```

-o == specify any configuration file option on the command line

```
ssh -o FingerprintHash=md5 firewall.example.com
```

Shell Variables

PS1 : make sure remote prompt has enough information

SSH_AUTH_SOCK

Tools

ssh-copy-id

sshfs

rsync

scp now considered dangerous

sftp

autossh

Do not use ssh-keyscan. It does't verify keys!

from the ssh-keyscan man page

```
     If an ssh_known_hosts file is constructed using ssh-keyscan without
     verifying the keys, users will be vulnerable to man in the middle attacks.
```

Extra Stuff

```
ssh firewall.example.com "sudo tar -C /etc cfz -" | tar -C /tmp xfz -

ssh firewall.example.com "ps auxw" | tee /tmp/firewall_ps.txt | less

rsync -e ssh -avHS photos/ mybackupserver:photos/
```

Set remote shell environment

```
$ cat sshrc_wrapper.sh
#!/bin/bash
rhost=$1 && shift # grab the first arg as the host and pop it off the stack

# set useful MYSQL_PS1 and TERM, redirect a .bashrc file from STDIN, setting
# PS1, start a shell, make the shell interactive and set editing mode as vi
export ssh_bashrc='export MYSQL_PS1="(\u@\${HOSTNAME%.*.*}.\d)> "; export TERM=xterm;
bash --rcfile <(echo export PS1=\"\\\u@\${HOSTNAME%.*.*}\\\w\\\$ \") -i -o vi'

# exec to drop the wrapper process, -t to allocate a pseudo-terminal,
# destination host, the above shell creation, give the rest of the original
# command line to the shell as a command to execute
exec ssh -t $rhost $ssh_bashrc ${1:+-c "$@"}
```

SSH Home Game

Go forth securely!

Thanks you!

Finding Hans

- micro-blog on Mastodon
  - https://floss.social/@FLOX_advocate
- IRC
  - FLOX_Advocate on Freenode, usually in #PLUGAZ and #SeaGL
- blog on Plume
  - https://fediverse.blog/~/LuftHans
- List of some upcoming and previous talks and publications
  - https://www.LuftHans.com/talks/

Credits

- SSH Home Game by Brian C, used with permission
- SSH Thrown Tunnel by Brian C, used with permission
- SSH Single Tunnel modified version of Brian's Thrown Tunnel, used with permission
- Bruce Banner images: https://weheartit.com/entry/185262694
- ShellCon and Puffy (c) 2021, der.hans, CC-BY-SA
    - https://pixelfed.social/p/FLOX_Advocate/329715337455079424

Resources

- OpenSSH site
  - https://www.OpenSSH.com/
- FoxyProxy
  - https://getfoxyproxy.org/
- My Linux Journal article on SSH tunneling
  - https://www.LinuxJournal.com/magazine/use-ssh-cross-suspect-host-securely
- OpenSSH book
  - https://en.wikibooks.org/wiki/OpenSSH
- OpenBSD site (creators of OpenSSH)
  - https://www.openbsd.org/
- scp has issues, consider rsync
- Nixie Pixel's "Protect your Packets with SSH" video (good review of FoxyProxy)
  - https://www.youtube.com/watch?v=5mCNO_aL4BA
- Creating SSH tunnels with systemd
  - https://blog.kylemanna.com/linux/ssh-reverse-tunnel-on-linux-with-systemd/

- https://gist.github.com/drmalex07/c0f9304deea566842490