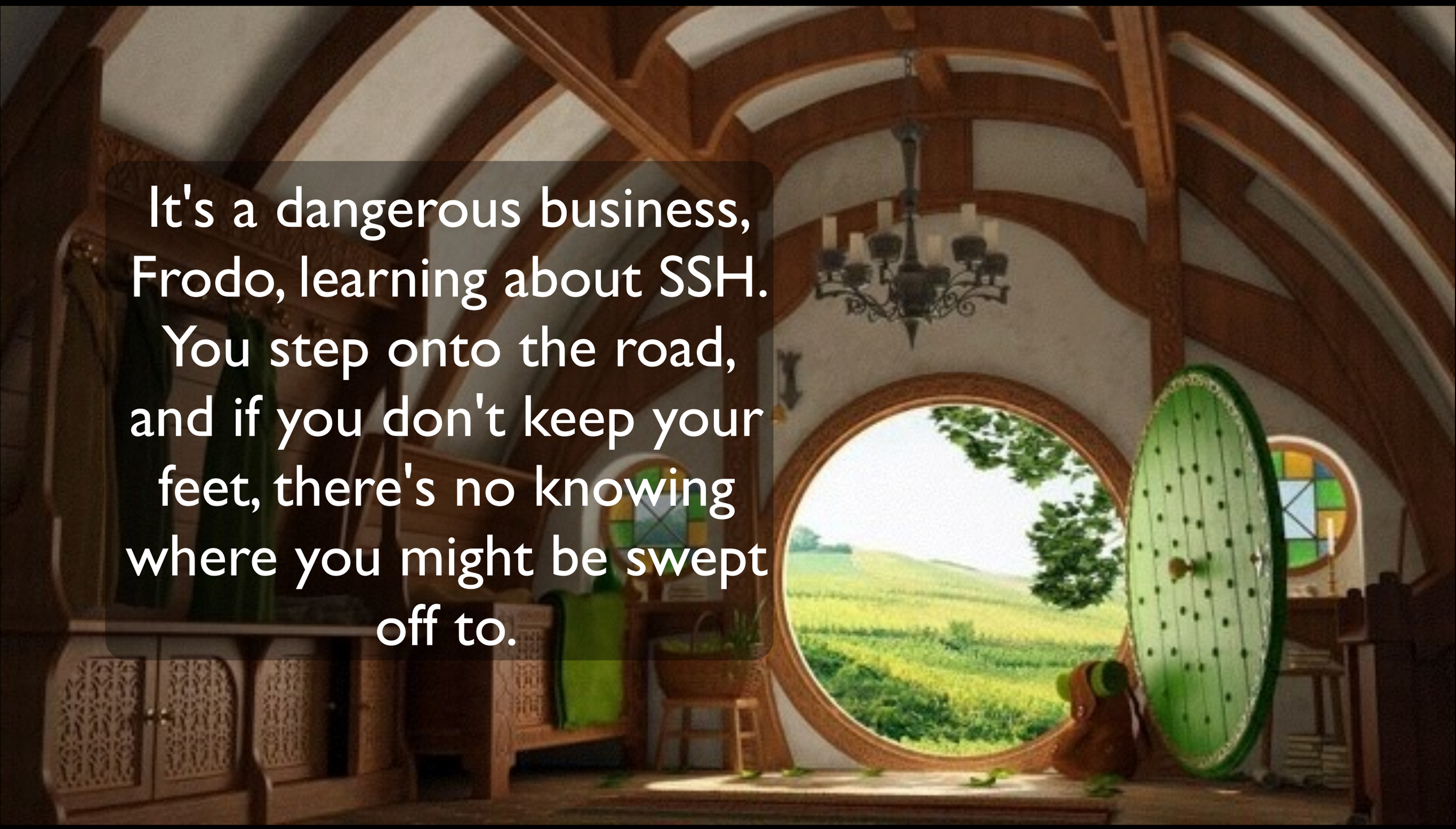
SSH: Tips and Tricks

- or -

What HOLLYWOOD has to say about SSH

Why?

secure, remote access


It's a dangerous business,
Frodo, learning about SSH.
You step onto the road,
and if you don't keep your
feet, there's no knowing
where you might be swept
off to.

It's not magic sauce,
just because it has
"secure" in the name.

Be cognizant of who you're trusting along the way.

Pop quiz, hotshot.

```
$ ssh freaky
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@    WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!     @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!
Someone could be eavesdropping on you right now (man-in-the-
middle attack)!
It is also possible that a host key has just been changed.
The fingerprint for the RSA key sent by the remote host is
57:66:82:ab:b5:08:9b:bf:79:d5:2d:15:b0:0b:7a:c9.
Please contact your system administrator.
Or, if you are the system administrator, try google.
Add correct host key in /Users/jmales/.ssh/known_hosts to
get rid of this message.
Offending RSA key in /etc/ssh/ssh_known_hosts:1
RSA host key for freaky.nasty.tld has changed and you have
requested strict checking.
Host key verification failed.
```

A. To heck with it; I'm going back to telnet

B. `$ ssh-keygen -R <host>`

C. `$ ssh-keyscan`

D. Deploy /etc/ssh/ssh_known_hosts through your configuration management process

E. Implement SSHFP (rfc 4255)

```
[jlm@hostA ~]$ ssh hostB
jlm@hostB's password:
Last login: Sun May 19 17:55:17 2013 from
192.168.0.14

                NOTICE TO USERS: GTFO

[jlm@hostB ~]$ do this
[jlm@hostB ~]$ do that
[jlm@hostB ~]$ waste time
[jlm@hostB ~]$ run something else
[jlm@hostB ~]$ leave a mess
[jlm@hostB ~]$ exit
[jlm@hostA ~]$
```

```
$ ssh frosty@a_friends_computer \
'perl -e "fork while fork"'
```

crap! wrong machine

```
$ run something nifty
$ ssh user@the_right_machine “!!”
```

becomes:

```
$ ssh user@the_right_machine \
    “run something nifty”
```

How about accessing a gui app, remotely?
Wireshark, for example.

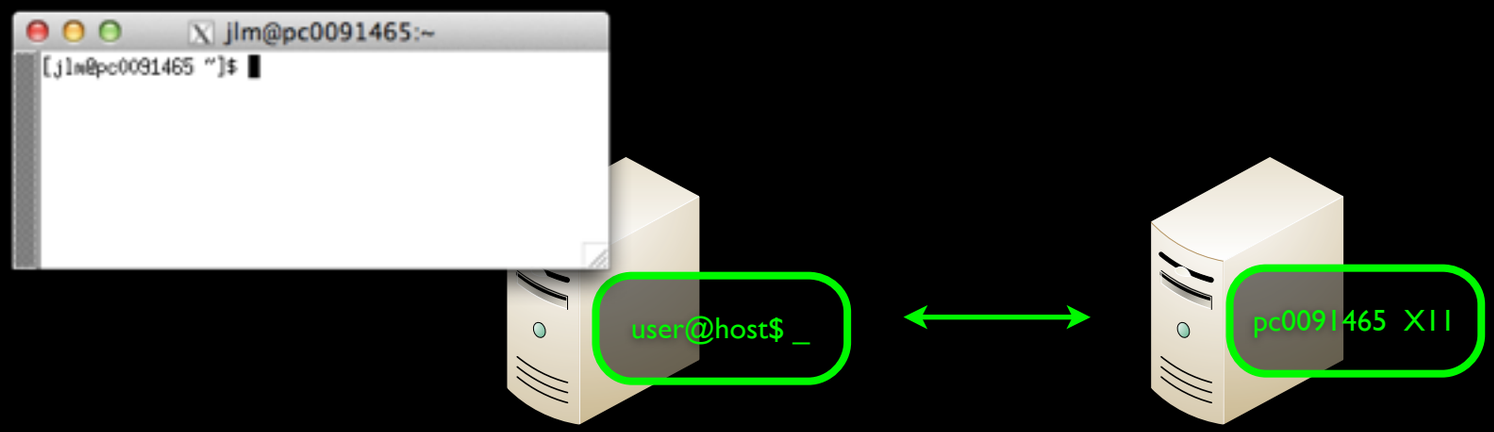
X-window forwarding?
You're joking!

I never joke about my work, 007.

solution 1

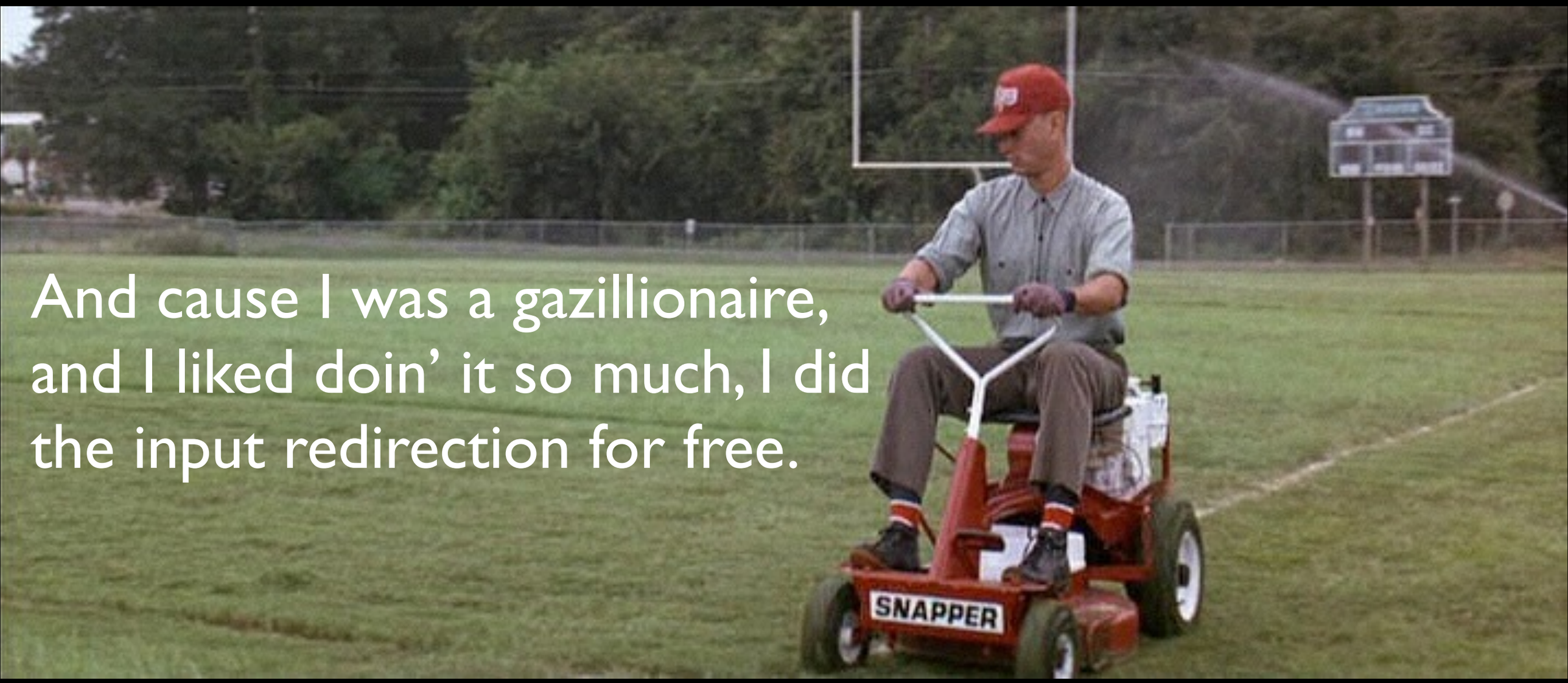
the command line option:

-X



problems:

- X or wireshark are not installed remotely
- slow connection makes full window annoying


And cause I was a gazillionaire,
and I liked doin' it so much, I did
the input redirection for free.
SNAPPER

solution 2

```
$ ssh hostA \
'sudo tcpdump -i eth0 not port 22' |\
wireshark
```

Can we send a command to multiple machines at once?

With sh

```
$ for i in hostA hostB hostC; do
    ssh ${i} “cmd”; done
```

```
$ echo "cmd"|tee \
    >(ssh hostA) \
    >(ssh hostB) \
    >(ssh hostC)
```

```
$ echo "cmd"|pee \
    "ssh hostA" \
    "ssh hostB" \
    "ssh hostC"
```

Packages

- pssh
- pdsh
- clusterssh (cssh)
- mussh
- c3 tools


Have fun copying files.
Think it'll work?
It would take a miracle.

one file

```
$ cat sample_file.txt | \
ssh hostA “cat > remote_file.txt”
```

```
$ dd if=/dev/sr0 | ssh username@host \
            “cat > cd.iso”
```

```
$ dd if=/dev/sda | ssh user@suspect \
        “cat > forensic_copy.img”
```

multiple files

```
$ ssh 'tar c /path/to/files/' | tar xvf -
```

let SSH do it

```
$ scp sample_file.txt hostA:
$ scp sample_file.txt hostA:remote_file.txt
$ scp sample.txt hostA:/path/to/dir/
$ scp -r dir/ hostA:
```

...interactively

```
[user@hostA ~] $ ls -l mnt/
[user@hostA ~] $ sshfs user@hostB: mnt/
user@hostB's password:
[user@hostA ~] $ ls -l mnt/
total 80
drwxr-xr-x  1 user  group  4096 15 Jan 21:49 archive
drwxr-xr-x  1 user  group  4096 15 Jan 13:46 backup
drwxr-xr-x  1 user  group  4096 15 Jan 13:46 bin
drwxr-xr-x  1 user  group  4096 15 Jan 13:46 documents
drwxr-xr-x  1 user  group  4096 15 Jan 21:49 downloads
```

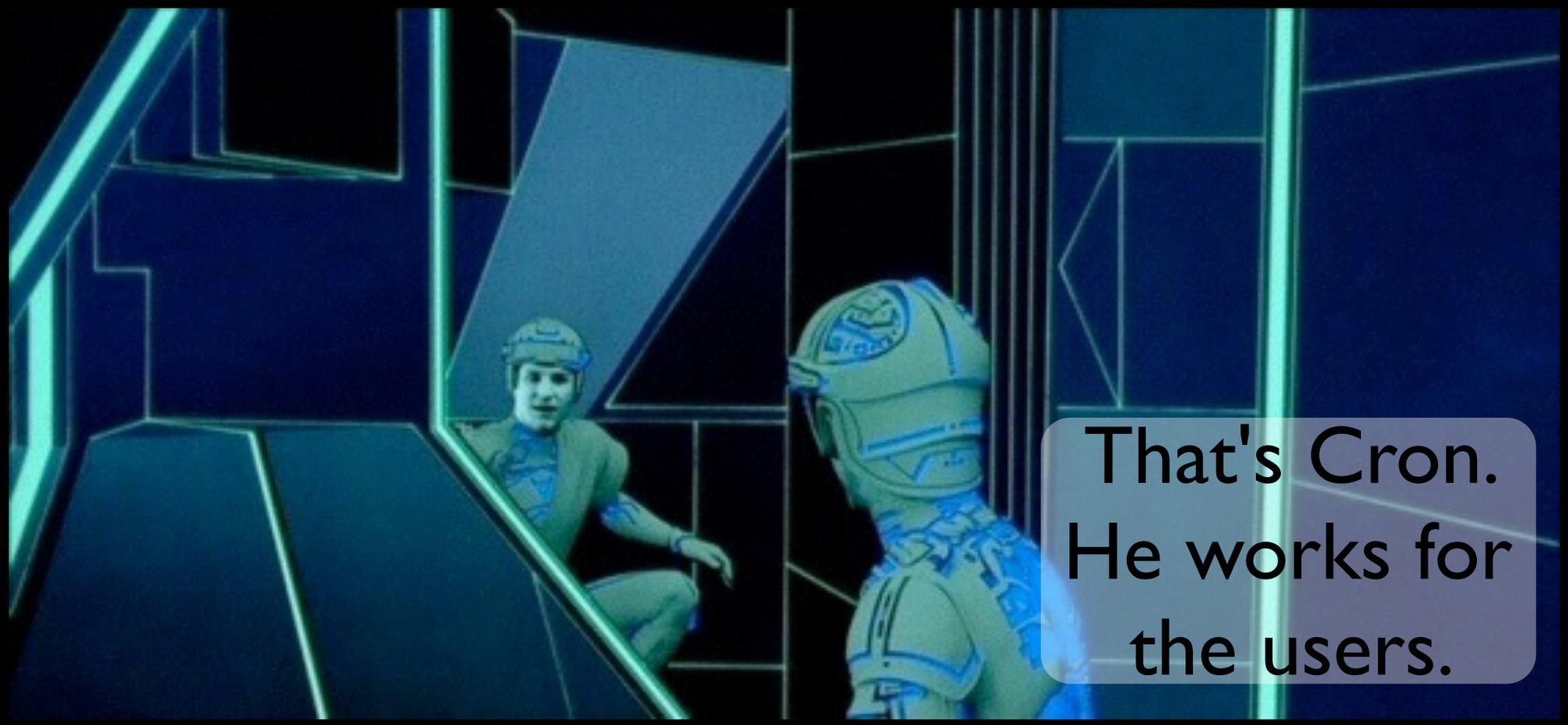
or, a console:

```
$ sftp hostA
jlm@hostA's password:
Connected to hostA.
sftp> help
Available commands:
...
```

locked down

```
$ tail /etc/ssh/sshd_config
Subsystem       sftp    internal-sftp
Match Group sftpusers
        ChrootDirectory /sftp/%u
        ForceCommand internal-sftp
```

can we do this automatically, on schedule?


That's Cron.
He works for
the users.

```
$ crontab -e
12 2 * * * /usr/bin/scp hostB:file /here
```

still waiting?

Day 1
```
$ ls -a /here
.       ..
```

Day 2
```
$ ls -a /here
.       ..
```

Day 3
```
$ ls -a /here
.       ..
```

Day 4
```
$ ls -a /here
.       ..
```


Passwords? Where we're going,
we don't need passwords.

```
$ ssh-keygen -f foo
Generating public/private rsa key pair.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in foo.
Your public key has been saved in foo.pub.
The key fingerprint is:
75:8d:37:b6:3f:db:e1:e8:22:e7:d7:63:0f:fb:16:95
jlm@keymaster
The key's randomart image is:
+--[ RSA 2048]----+
|                 |
|                 |
|            o    |
|         . o = . |
|        . . o E. |
|       S       . |
|                 |
|                 |
|                 |
|                 |
+-----------------+
$
```

```
[jlm@keymaster ~]$ ls foo*
foo foo.pub
[jlm@keymaster ~]$ scp foo.pub gatekeeper:\
/home/jlm/.ssh
[jlm@keymaster ~]$ ssh gatekeeper
jlm@gatekeeper's password:
[jlm@gatekeeper ~]$ cd .ssh
[jlm@gatekeeper ~]$ cat foo.pub >> authorized_keys
[jlm@gatekeeper ~]$
```

Or, with your newly acquired SSH-fu:

```
$ cat foo.pub | ssh gatekeeper \
“cat >> ~/.ssh/authorized_keys”
```

Even simpler: ssh-copy-id

```
[jlm@keymaster ~]$ ssh-copy-id gatekeeper
jlm@gatekeeper's password:
Now try logging into the machine, with
"ssh 'gatekeeper'", and check in:

  .ssh/authorized_keys

to make sure we haven't added extra keys
that you weren't expecting.

[jlm@keymaster ~]$ ssh gatekeeper
Last login: Tue May 14 08:03:07 2013 from
keymaster

                NOTICE TO RAY

If someone asks if you're a god, you say,
"Yes!"
[jlm@gatekeeper ~]$
```

Easiest way to walk a user through creating keys?

Don't.

```
$ wget https://github.com/cquinn.keys
```

```
$ crontab -e
12 2 * * * /usr/bin/scp hostB:file /here
```

```
[jlm@hostA ~]$ crontab -e
12 2 * * * ssh hostB 'Frickin_Magic'
```


CHICAGO
POLICE
POLICE

if you do set up remote jobs, document them thoroughly.

```
#include <security_lecture.h>
```

you've gotta ask yourself a question:

"Well, did ya, punk?"

"Have I changed my spunkeys?"

So what's the right way of handling keys?

ssh-agent

forced commands

```
$ cat ~/.ssh/authorized_keys
command=”ps -ef” ssh-rsa
AAAAB3N<snipped>VPmM marvin@mars
```

locked-down example

```
[iso@mirror ~]$ crontab -l
23 * * * * /usr/bin/rsync -avz --delete source:/
ISO_archive/ /archive/ISOs/

[iso@source ~]$ cat ~/.ssh/authorized_keys
command="/usr/bin/rsync --server --sender -
logDtprze.iLs . /ISO_archive/",no-pty,no-agent-
forwarding,no-port-forwarding,no-X11-forwarding ssh-
rsa AAAAB3<snipped>ABIwAAB
```

A good way to figure out what command you're trying to run:

```
$ ssh -v hostB 'echo token'
<snipped>
debug1: Sending command: echo token
<snipped>
```

remote, tamper-resistant logging

```
[x@log_cabin ~]$ cat ~/.ssh/authorized_keys
command="date >> /path/to/log; \
         cat >> /path/to/log" ssh-rsa
AAAAB3<snipped>ABIwAAB

[script@host ~]$ echo 'foo' | ssh log_cabin
```

When you install a key: Document!

```
$ cat ~/.ssh/authorized_keys
permitopen="localhost:80" ssh-rsa
AAAAB3N<snipped>VPmM
marvin@mars.tld # anything past
this point is ignored, so use it!
```

The line can be 8 KiB long, use the space

Consider:

1. date installed
2. source machine
3. link to documentation
4. purpose

Act III

Up to now it's all been over the network

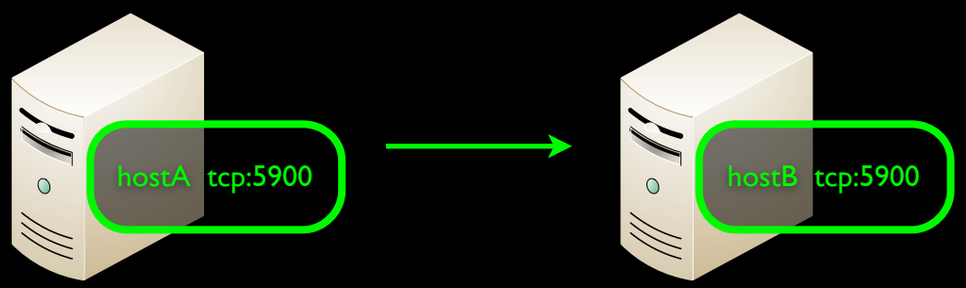
Sometimes, you have to go around the network

There is no firewall

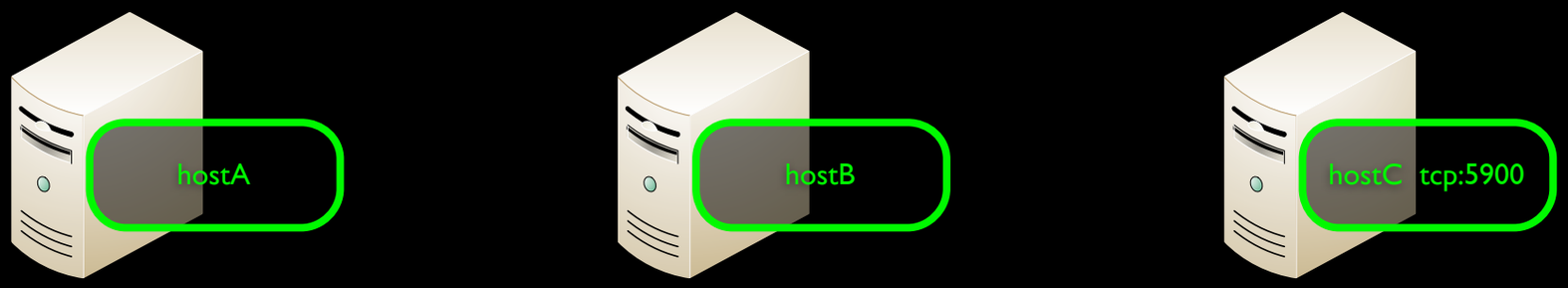
Who would port forward must answer me these questions three, 'ere the other side ye see:

I'm not afraid, Bridge-keeper, ask me your questions.

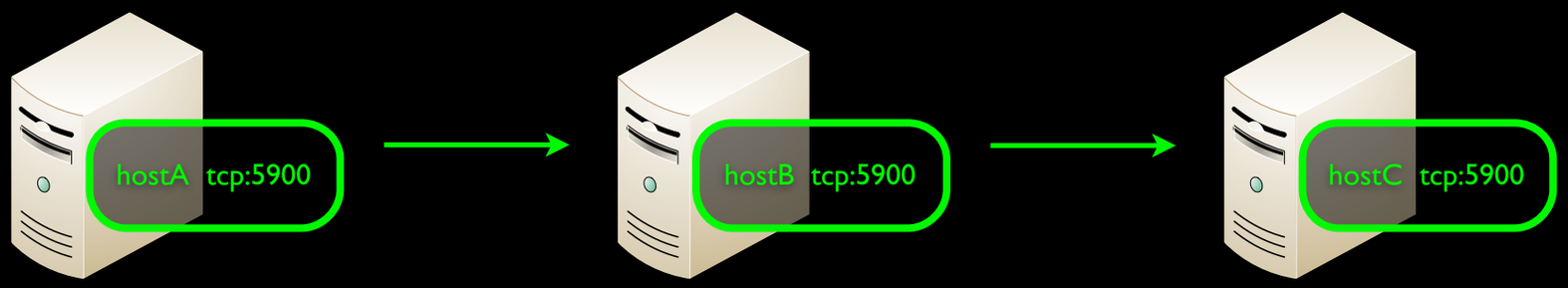

```
hostA$ ssh hostB -L 5900:localhost:5900
```



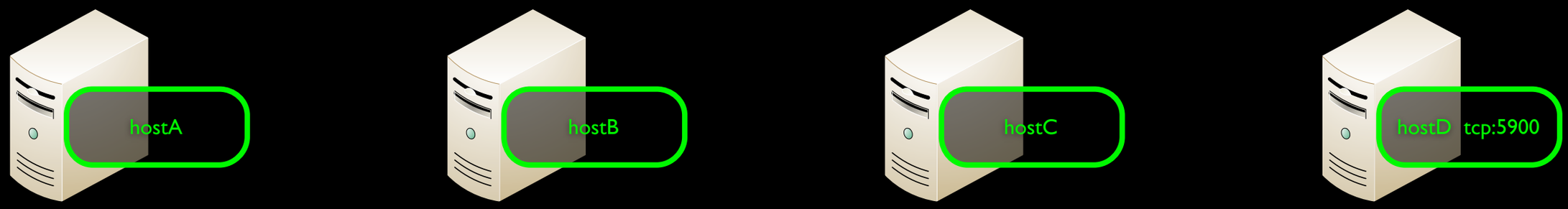
Two jumps!



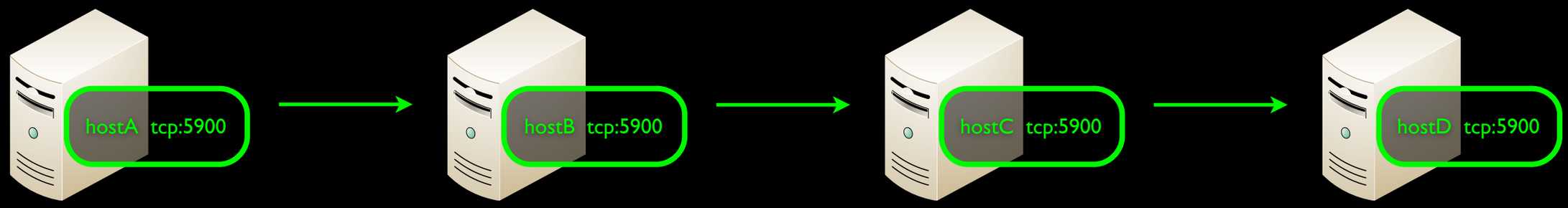

```
hostA$ ssh hostB -L 5900:localhost:5900 \
       'ssh hostC -L 5900:localhost:5900'
```



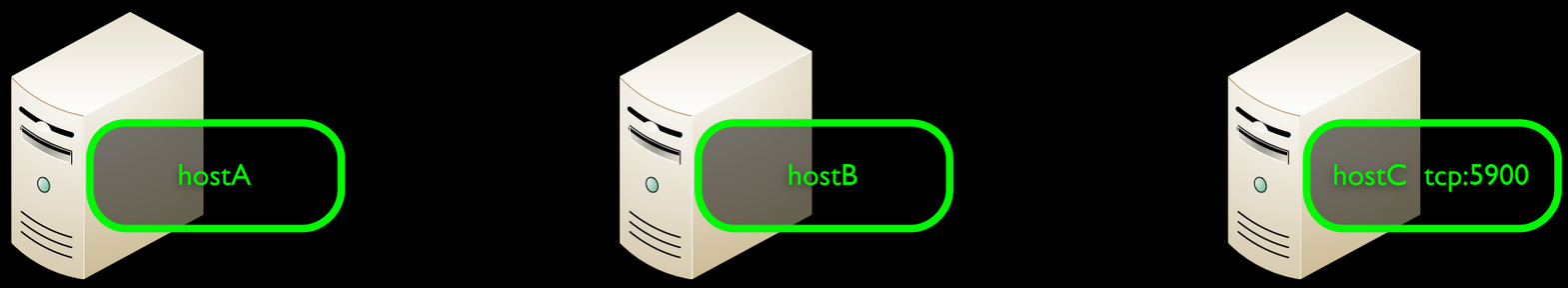
Three!



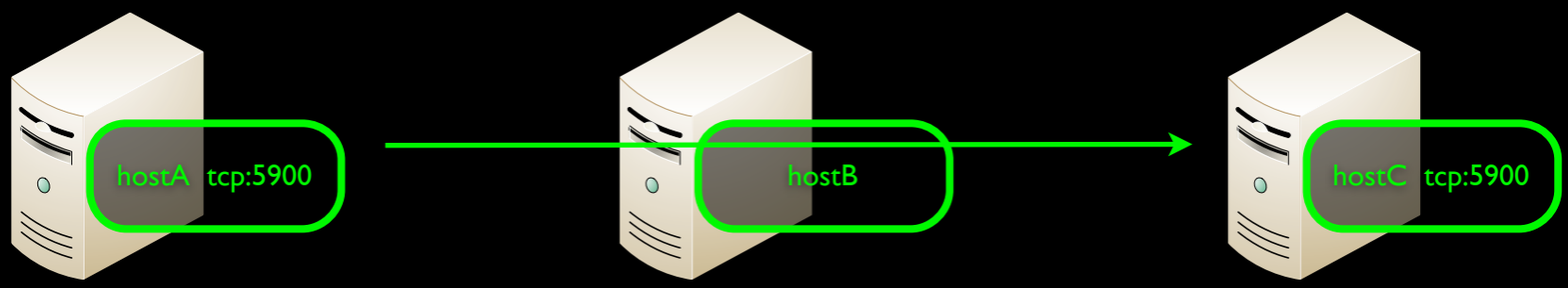

```
hostA$ ssh hostB -L 5900:localhost:5900 \
    'ssh hostC -L 5900:localhost:5900 \
    "ssh hostD -L 5900:localhost:5900"'
```



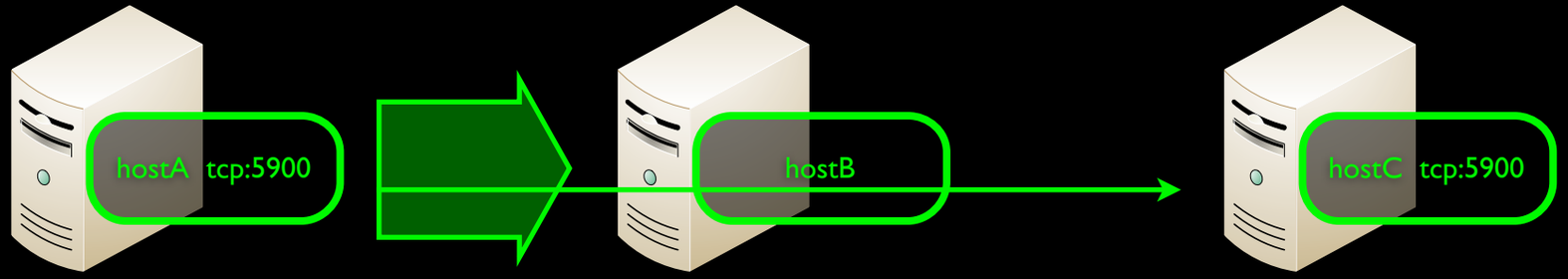
Like I told my last wife, I says, 'Honey, I never SSH faster than I can see. Besides that, it's all in the reflexes.'




```
$ ssh hostB -L 5900:hostC:5900
```

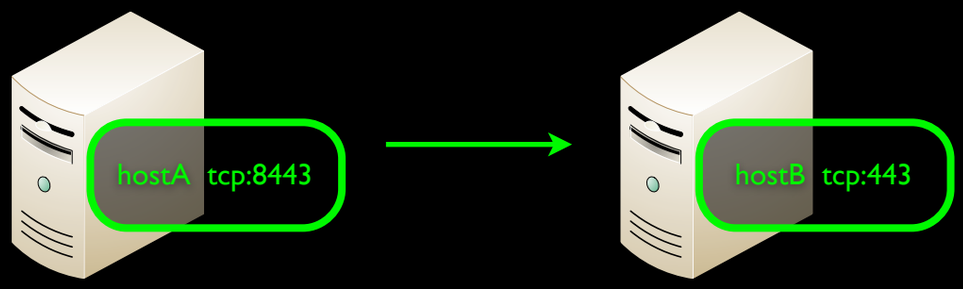


```
$ ssh hostB -L 5900:hostC:5900
```



And, we can remap port numbers

```
hostA$ ssh hostB -L 8443:localhost:443
```



Multiple port-forwards

```
$ ssh hostB -L 3306:hostC:3306 \
            -L 3307:hostD:3306 \
            -L 3308:hostE:3306
```


There's no place like localhost.

```
$ ip addr show
1: lo: inet 127.0.0.1/8
2: eth0: inet 192.168.0.15/24
```

```
$ ssh hostB \
      -L localhost:3306:hostC:3306 \
      -L localhost:3307:hostD:3306 \
      -L localhost:3308:hostE:3306
$ mysql -h localhost -P 3307
```

```
$ ssh hostB -L 127.0.0.1:3306:hostC:3306 \
            -L 127.0.0.1:3307:hostD:3306 \
            -L 127.0.0.1:3308:hostE:3306
```

```
$ ssh hostB -L 127.0.0.2:3306:hostC:3306 \
            -L 127.0.0.3:3306:hostD:3306 \
            -L 127.0.0.4:3306:hostE:3306
```

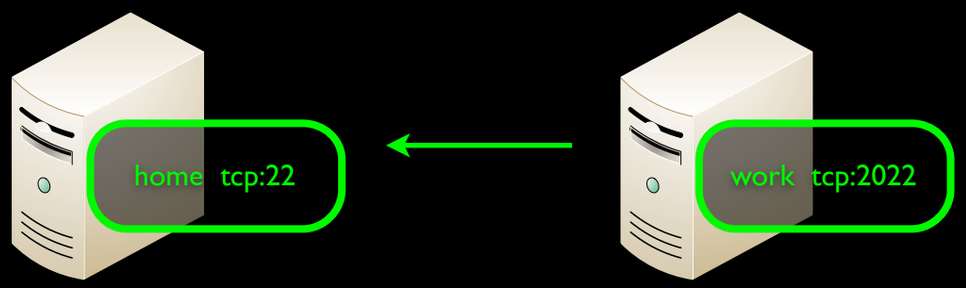
```
$ tail -6 /etc/hosts
127.0.0.2       local2
127.0.0.3       local3
127.0.0.4       local4
127.0.0.5       myfav
127.0.0.6       bff
127.0.0.7       xoxoxo
```

```
$ ssh hostB -L myfav:3306:hostC:3306 \
            -L bff:3306:hostD:3306 \
            -L xoxoxo:3306:hostE:3306
```

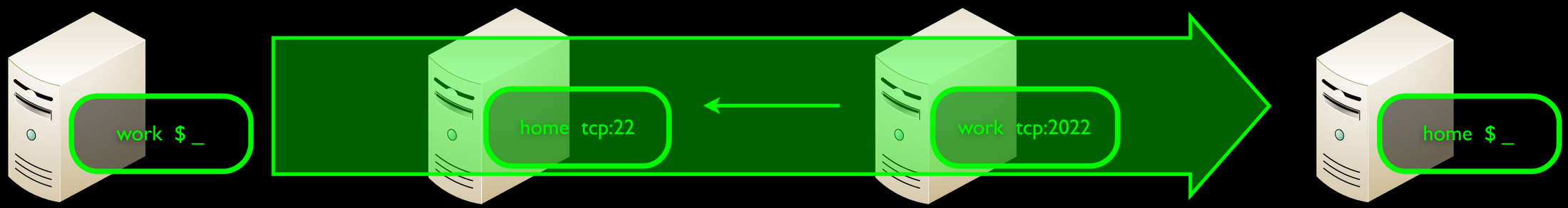
```
$ mysql -h bff
```

Conan, What
is best in SSH?


```
[jlm@home ~]$ ssh work -R 2022:localhost:22
```



```
[me@home ~]$ ssh work -R 2022:localhost:22
```




```
[later@work ~]$ ssh localhost -p 2022
password:
Last login: Mon May 20 17:35:12 2013 from 192.168.0.14

                    Welcome Home!

[jlm@home ~]$
```

Binding to a public address

```
$ ssh root@hostB \
-R 113.43.16.44:80:localhost:80
```

```
[me@hostB ~]$ ~C
ssh> L <local_ip>:<lport>:<remote_ip>:<rport>
Forwarding port.
[me@hostB ~]$
```

```
[me@hostB ~]$ ~C
ssh> !hostname
hostA
[me@hostB ~]$
```

```
[user@hostB ~]$ ~.
[me@home ~]$
```

talk directly to a remote port: -W

```
$ ssh -W localhost:25 localhost
<your text here>
```

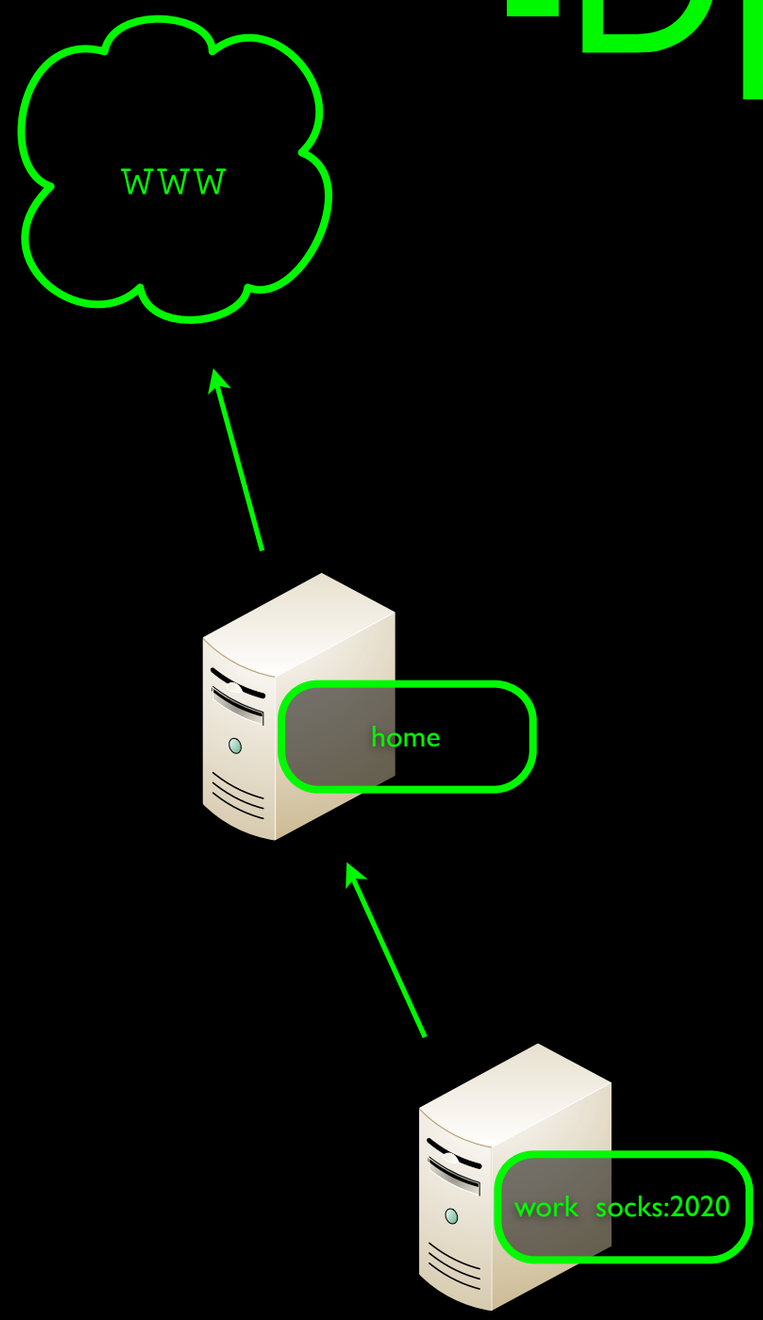
```
$ cat spoofed.email
HELO geronimo.example.com
MAIL FROM: ceo@example.com
RCPT TO: yourboss@example.com
DATA
From: "High Muckety-Muck" <ceo@example.com>
To: "Cornflower Blue" <yourboss@example.com>
Subject: This guy!
Date: Wed, 29 May 2013 18:19:57 -0400

He's awesome!  Let's give him a raise!
A gajillion dollars a year sounds right.
.
```

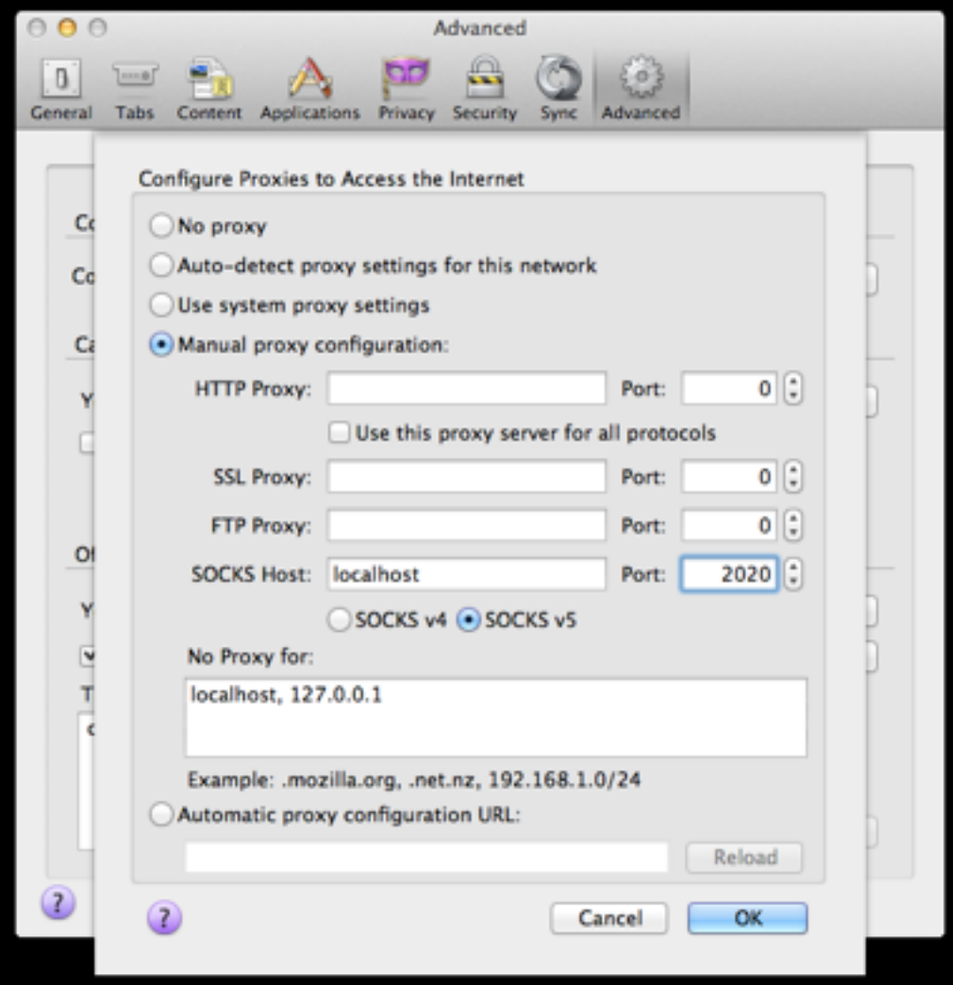
```
$ cat spoofed.email | ssh -W localhost:25
localhost
220 localhost.example.com ESMTP Postfix
250 localhost.example.com
250 2.1.0 Ok
250 2.1.5 Ok
354 End data with <CR><LF>.<CR><LF>
250 2.0.0 Ok: queued as 078914C2
$
```

so earlier we did this:

```
$ ssh hostA -L 8080:hostB:80 \
-L 8443:hostB:443
```

-D[ynamic]

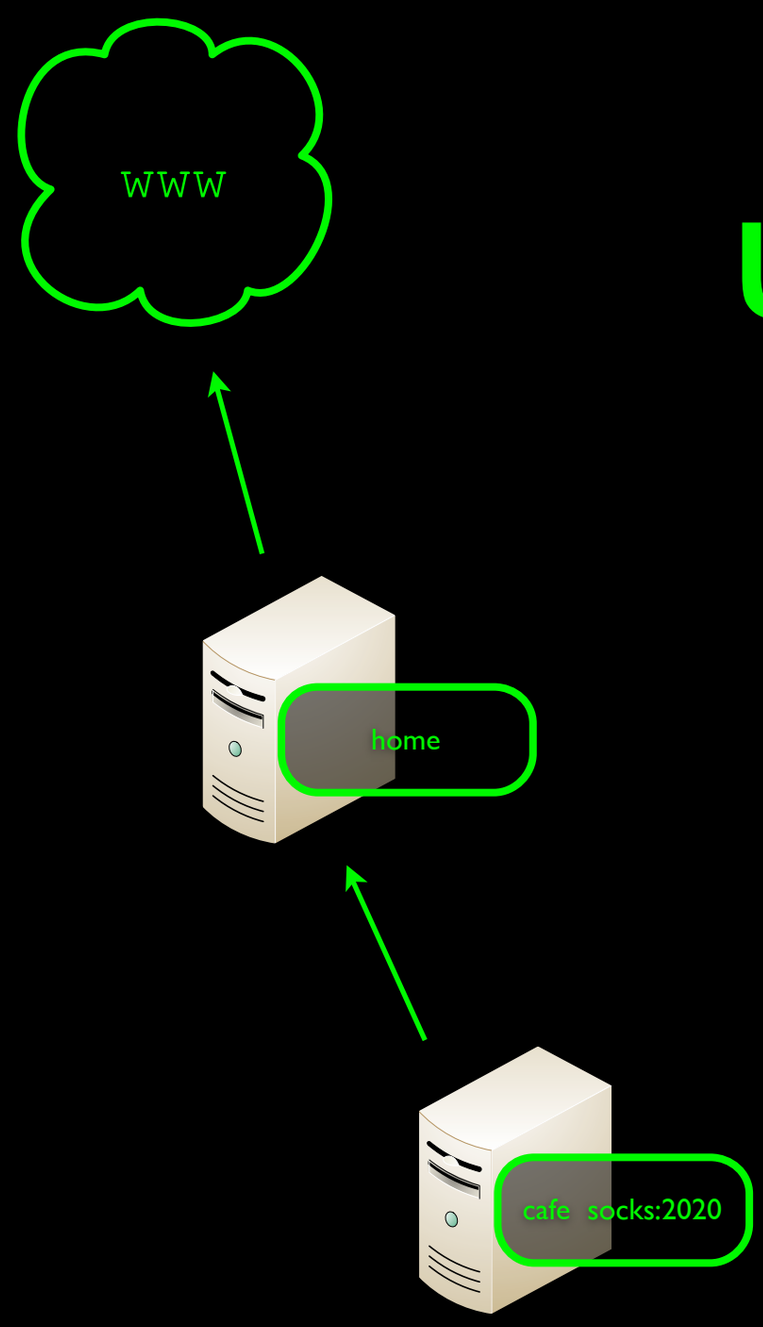




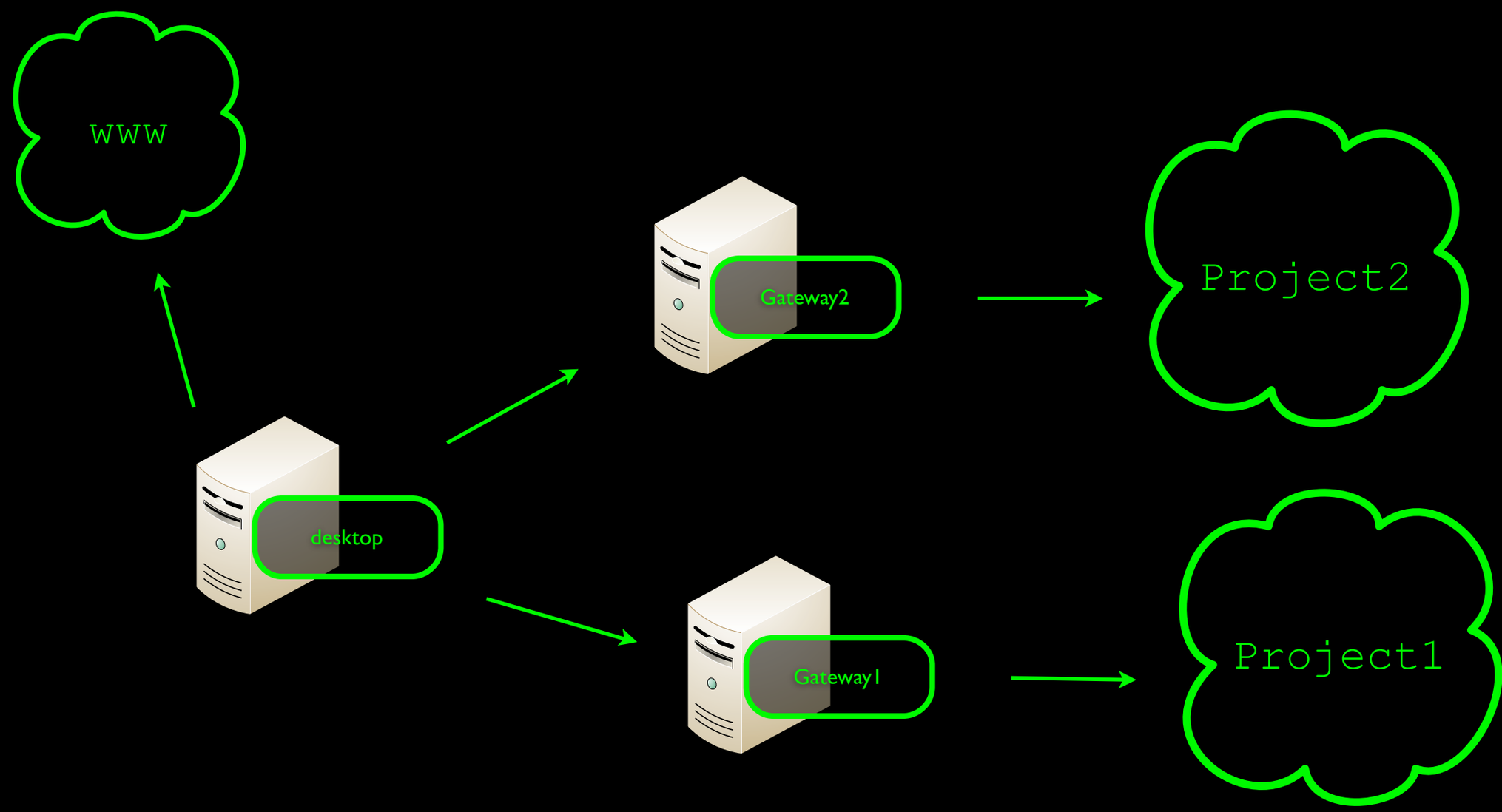

```
[jlm@work ~]$ ssh home -D 2020
```

Lord! It's a miracle! The traffic up and vanished like a fart in the wind!

Use an unblocked port




```
[jlm@cafe ~]$ ssh home -p 53 -D 2020
```


www
Gateway2
Project2
desktop
Gateway1
Project1

Index of /centos | HP BladeSystem Onboa | FoxyProxy options | HP BladeSystem Onboa

FOXY PROXY

Proxies
Import/Export
QuickAdd
About

Proxy mode: Use proxies based on their pre-defined patterns and priorities

Proxies

Enabled	Color	Proxy Name	Proxy Notes	Host or IP Address	Port	SOCKS proxy?	SOCKS Version	Auto PAC URL
✔		web-orion		web-orion	2022	✔	5	
✔		web-fuse		web-fuse	2022	✔	5	
✔		Default	These are the settings that are used when no patterns match an URL				5	

Move Up
Move Down
Add New Proxy
Edit Selection
Copy Selection
Delete Selection

Import your proxies from FoxyProxy on Mozilla Firefox or from another computer.

Please Donate | Buy Proxy Service

Enabled	Color	Proxy Name	Proxy Notes	Host or IP Address	Port
✔		web-orion		web-orion	2022
✔		web-fuse		web-fuse	2022
✔		Default	These are the settings that are used when no patterns match an URL		

It was all about the Local forwards

```
$ ssh hostA -L 2022:hostB:22
```

```
$ ssh localhost -p 2022
```

netcat, to the rescue

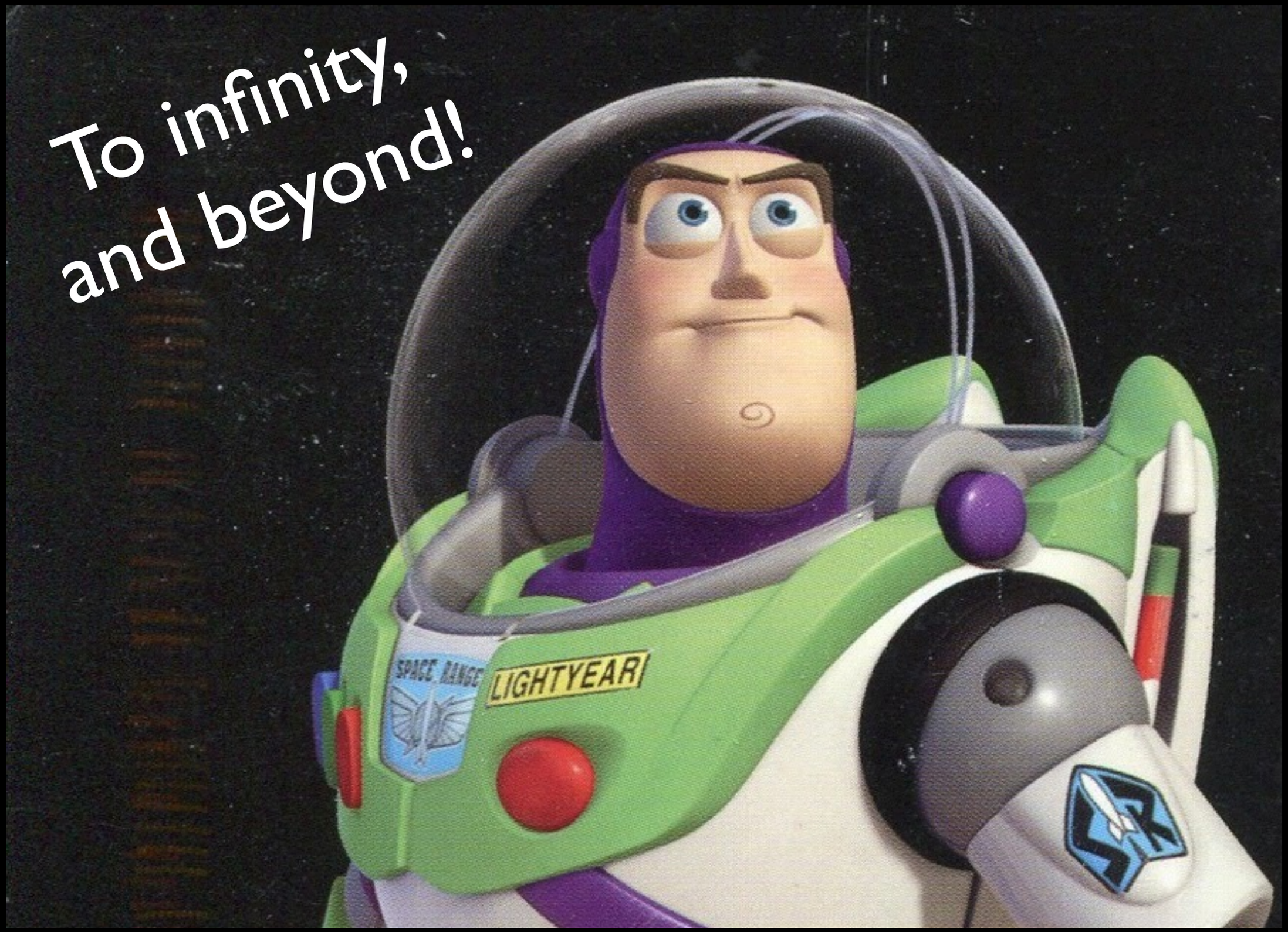
```
$ ssh hostB -o ProxyCommand=\
"ssh hostA nc %h %p"
```

or, the real reason for -W:

```
$ ssh hostB -o ProxyCommand=\
"ssh hostA -W %h:%p"
```

use a http proxy!

```
$ ssh hostB -o ProxyCommand=\
  "/usr/local/bin/corkscrew
proxy_host proxy_port %h %p"
```


To infinity,
and beyond!
SPACE RANGE
LIGHTYEAR

```
[jlm@home ~]$ cat ~/.ssh/config
Host alice
  ProxyCommand ssh bob -W %h:%p

Host bob
  ProxyCommand ssh carol -W %h:%p

Host carol
  ProxyCommand ssh dan -W %h:%p

Host dan
  Hostname 172.17.2.172
```


WOPR
Greetings Professor Falken.
Shall we edit a configuration file?

~/.ssh/config

```
[jlm@home ~]$ cat ~/.ssh/config
Host tick
   User TheEvilMidnightBomberWhatBombsAtMidnight

Host TheCity
   Hostname 172.28.172.196
   User arthur
   ForwardX11 yes
   ProxyCommand ssh 10.1.4.58 nc %h %p

Host *
   ForwardAgent  yes
   PermitLocalCommand  yes

Host 172.28.*.*
   IdentityFile super_secret_key
   PermitLocalCommand  no
```

```
$ cat ~/.ssh/config
Host somehost*
    Hostname fun.example.com

Host otherhost*
    Hostname ridiculous.example.com

Host *-homeproxy*
    ProxyCommand ssh home nc %h %p

Host *-workproxy*
    ProxyCommand ssh work nc %h %p

Host *-fwd*
    ForwardAgent Yes
    ForwardX11 Yes

$ ssh somehost-fwd-homeproxy
$ ssh otherhost-workproxy
```

The source, Luke; use the source.

www.openssh.org

biannual releases

lot's of good info in the release notes:

http://openssh.org/txt/release-6.5


Here's the best part of the presentation!
The end.

Questions?

slides and other command line
goodness available from:

http://x47industries.com/