2-1-1 ssh
Secure SHell

Using Public Key Cryptography

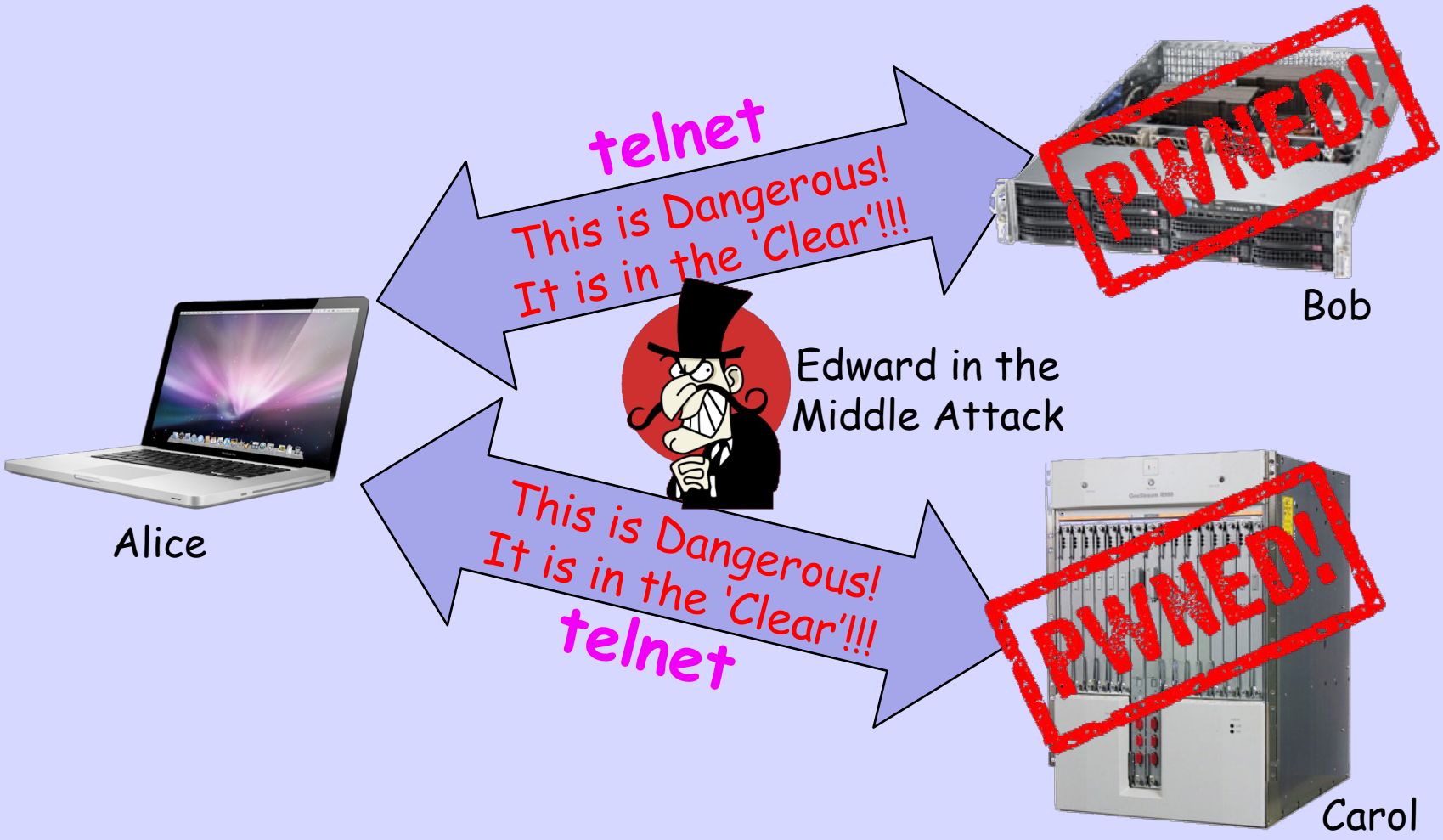
Keying, Key Exchange, and Session Setup

Communicate Safely with Remote Systems

What is "Safely"

- Authentication – I am Assured of Which Host I am Talking With
- Authentication - The Host Knows Who I Am
- Privacy - The Traffic is Encrypted
- Integrity – The Traffic is Unmodified

Traditional



Encrypted

Alice

ssh

xehlhxn j;ijf ure hq
ewioihbyugi'owef;glgu

Edward is Unhappy

;qfhhlqwcwiefxweix iw
xfenowixfuewg2384

ssh

Bob

Carol

Secure SHell

- Provides authenticated and encrypted shell access to a remote host
- But it is much more
- It is used by other protocols, sftp, scp, rsync, …
- You can use it to build custom tunnels

Think of SSH as a Bit Like PGP where the Other End is a Computer, Not a Human

But PGP is

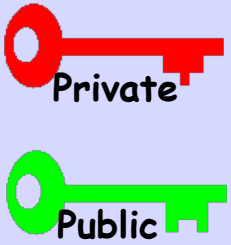
Object Security

SSH is

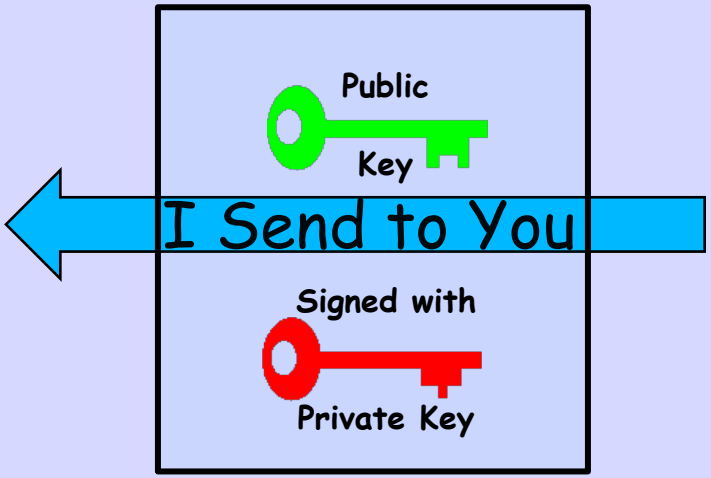
Channel/Transport Security

Proof of Possession

If I Have a Key Pair



How Do I Convince You That I Have Both Private and Public Keys Over The Public Net?



You Verify Signature Using The Public Key

If It Verifies, Then You Know That
I Must Have The Private Key

And You Know You Have the
Corresponding Public Key

ssh - Keying Setup

ssh-keygen -t rsa

/home/usr/.ssh

Private

Public

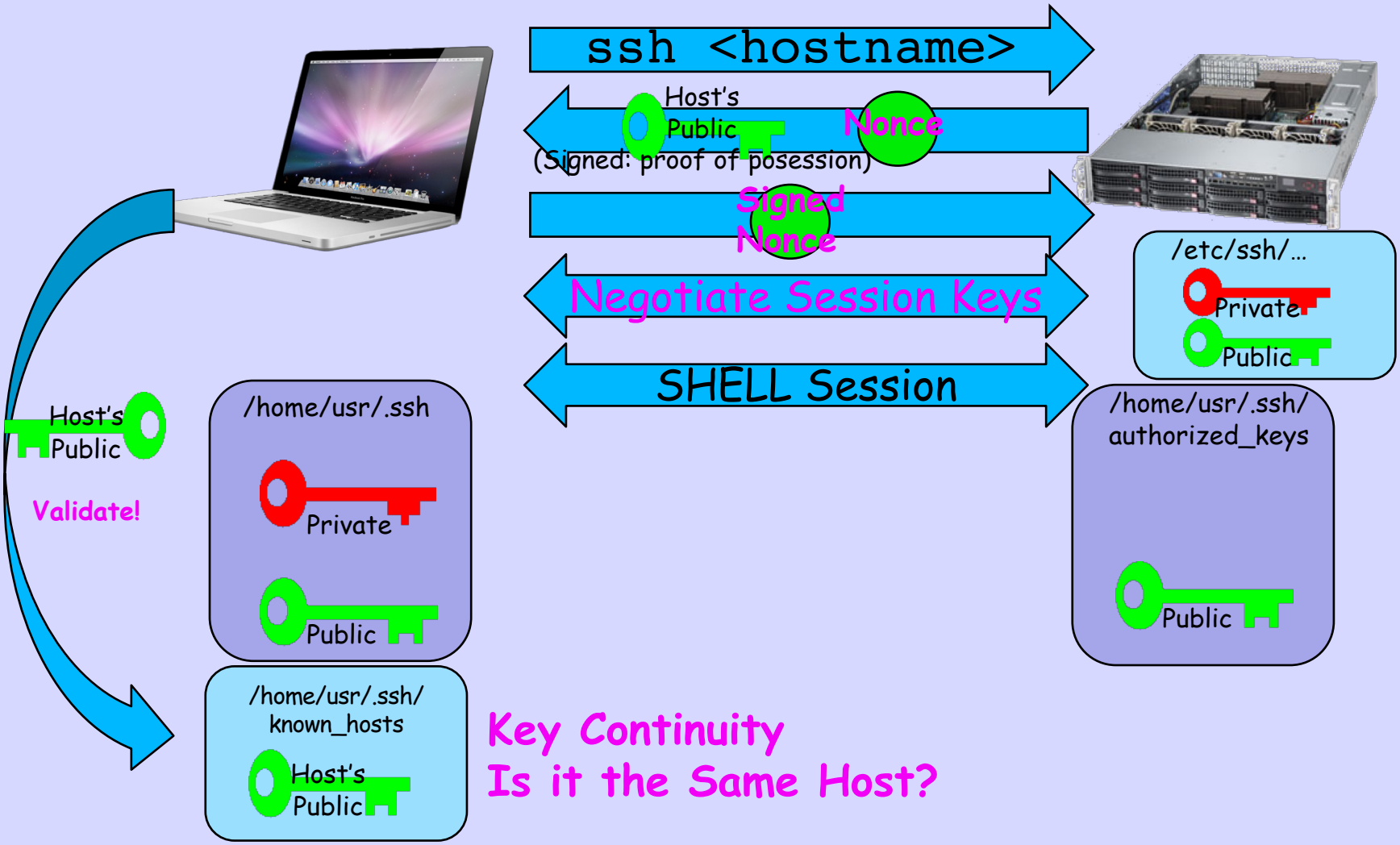
/etc/ssh/...

Private

Public

/home/usr/.ssh/
authorized_keys

2-Way Authentication



Checking Host's Keys

```
$ ssh -o VisualHostKey=yes psg.com
Host key fingerprint is
d2:2b:f1:17:75:0d:c9:86:74:71:e2:00:62:0f:22:02
+--[ RSA 1024]----+
|E.. . . + .ooo=o.|
|    . . o +  .++= |
|          . ..o .|
|       .   . .   |
|      o S .      |
|       + . .     |
|      . o .      |
|       . .       |
|                 |
+-----------------+
```

And you check it against what you got out of band

ssh-keygen RSA Key

```
/usr/home/foo> ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/usr/home/foo/.ssh/id_rsa):
Created directory '/usr/home/foo/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /usr/home/foo/.ssh/id_rsa.
Your public key has been saved in /usr/home/foo/.ssh/id_rsa.pub.
The key fingerprint is:
27:99:35:e4:ab:9b:d8:50:6a:8b:27:08:2f:44:d4:20 foo@psg.com
The key's randomart image is:
+--[ RSA 2048]----+
|E.o       .      |
|.. .     o       |
|.         +      |
| .       + o     |
|.       S o      |
|..     o +       |
|.o .  + .        |
|. o .o.= o       |
| .  .oo +        |
+-----------------+
```

Use Keys Not Passwords

- In /etc/ssh/sshd_config
  ```
  PermitRootLogin without-password
  PasswordAuthentication no
  UsePAM no
  ```
- Never Store Private Key on a Multi-User Host
- Store Private Key ONLY on Your Laptop and Protect Your Laptop (Encrypt Disk!)
- It is OK to Use SSH_AGENT to Remember your Key ONLY if your Laptop Locks Very Quickly

The Only Compromise I Have Had to My Infrastructure was a Researcher who Stored Their Private Key on a Shared University Host

Private Key Protection

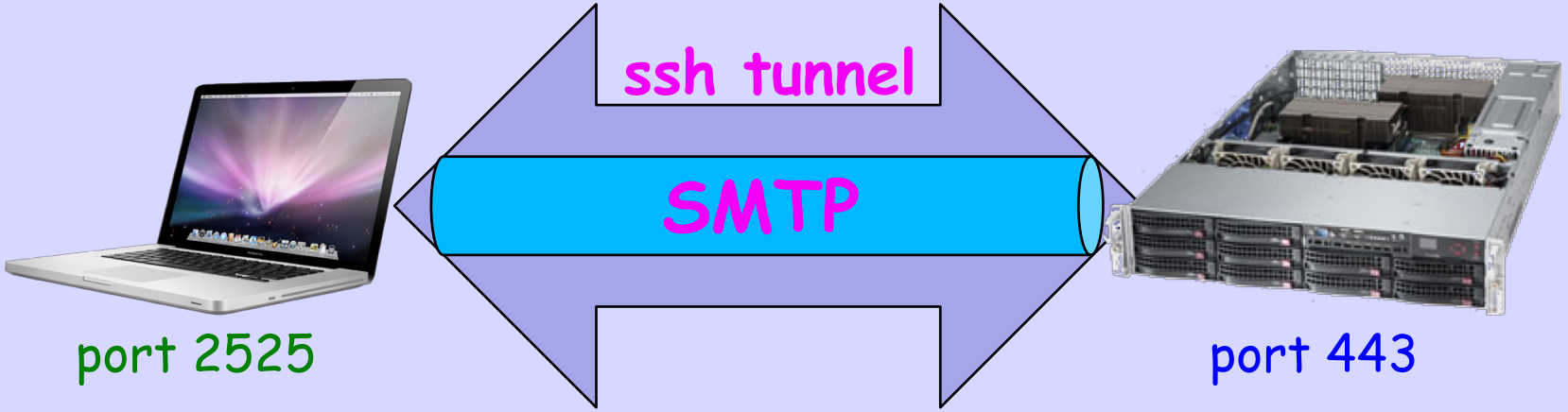
- FreeBSD Repository Compromise Six Years Ago

  "The compromise is believed to have occurred due to the leak of an SSH key from a developer who legitimately had access to the machines in question, and was not due to any vulnerability or code exploit within FreeBSD."

General Purpose Tunnel

- I am in my hotel room and want to send mail from my laptop
- I do not want unencrypted mail going over the net
- So I want the SMTP traffic to be encrypted to my SMTP server
- I own the SMTP server

General Purpose Tunnel




```
$ ssh -N ssh.psg.com -p 443 -L 2525:127.0.0.1:25
```

Target Host

Tunnel Port

Port on MacBook

Tunnel EndPoint

SSH is Built In

UNIX

Linux

MacOS X

Windows 10