Using Linux Lunch and Learn #3 Using SSH

What we’ll cover

- Introductions
- What is SSH
- SSH client and server software
- Logging in with SSH
- SSH Config file
- Public key authentication(PKA)
- Jump hosts
- Port forwarding and X11 forwarding
- SSH PKA and jump hosts

About me

- Erik Meitner
- Systems Administrator for the Math Dept.
- Room 507
- What I do
- How I can help you
- How you can help me

About you

- Name
- How much have you used SSH?
- Do you like winter? Why or why not?

Questions are allowed!

Linux in the Math Dept.

- Systems running Linux:
  - Workstations
  - Research servers
  - WURC cluster
  - (Infrastructure servers)
- Your account:
  - “Math account”
  - (Future) NetID sign-on

Research Servers

- Five servers: https://kb.wisc.edu/math/internal/114567
- Various CPU, Memory, and GPU configurations
- Log in with your Math Account
- Naming: server name and “magma” alias
- Your home directory is available on all servers
- Which one is the best to use?
  https://dashboard.math.wisc.edu/

Software Available

intel.
1
oneAPI
HPC TOOLKIT

DEDALUS

MAGMA

Wolfram
Mathematica

GCC

python

Maple

MATLAB

OpenMPI

GAP

sage

What is SSH?

- SSH = **S**ecure **SH**ell
- A client-server protocol for two way encrypted communications.
- It provides:
  - Server authentication
  - User authentication
  - Data confidentiality
  - Data integrity
  - Multiplexing

SSH Server Software

- There are around 10 SSH server systems
- On Linux: OpenSSH
- On Mac: OpenSSH
- Windows: OpenSSH
- Others

SSH Client Software

- Linux
  - CLI
    - OpenSSH
  - SFTP
    - SSHFS, GVFS
- Mac
  - CLI
    - OpenSSH
  - SFTP
    - FileZilla
    - CyberDuck

- Windows
  - CLI
    - OpenSSH
  - SFTP
    - FileZilla
    - WinSCP
- Others

SSH Clients

- Shell mode
  - Provides a connection to a program running on a remote server. Typically a shell like Bash.
- SFTP mode
  - Allows file listing, two-way file transfer, and file attribute modification with the remote server.

Logging In With SSH

- ID: Username
- Password and/or SSH key
- Command:

```
ssh emeitner@login.math.wisc.edu
```

- SSH will use the username of the local user if none is provided.

Logging in with SSH

- If you've never connected to the server before you'll see:

```
The authenticity of host 'login.math.wisc.edu (144.92.166.43)' can't be
established.

ED25519 key fingerprint is
SHA256:B+X0AcC5fVHYgq1MsUjWaqxfEZcQniGKICrMSV4Jec4.

This key is not known by any other names

Are you sure you want to continue connecting (yes/no/[fingerprint])?
```

- The underlined text may not be present.

Logging in with SSH

- Fingerprints are stored in "~/.ssh/known_hosts"
- They look like this:

```
vv101c.math.wisc.edu ecdsa-sha2-nistp256
AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBCesi/R502ado1RQkwMLYwpP3LPLAOzFGvbPzuqf6C8fR
h2MeHkr0gEBkIMn2V89dcdv0otZISLR0qr/zBhPMAo=
```

- Or like this:

```
|1|j/KvPv6o8pkKk6/VHsjCYknIktA=|pG3MQKC2XRbQaQetrJ9Pa0E6sC4= ecdsa-sha2-nistp256
AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBCesi/R502ado1RQkwMLYwpP3LPLAOzFGvbPzuqf6C8fR
h2MeHkr0gEBkIMn2V89dcdv0otZISLR0qr/zBhPMAo=
```

- In the second one the host name is hashed. This is a security feature that you can enable/disable.

Logging in with SSH

- If the server fingerprint does not match the one you saved when you first connected you will see:

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@    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 ECDSA key sent by the remote host is
SHA256:XxS6DkytHZ2kxKsVuzLuI0TVnVMTn/xBsHE0Dr9QOfE.
Please contact your system administrator.
Add correct host key in /home/emeitner/.ssh/known_hosts to get rid of this
message.
Offending ECDSA key in /home/emeitner/.ssh/known_hosts:20
  remove with:
  ssh-keygen -f "/home/emeitner/.ssh/known_hosts" -R "login.math.wisc.edu"
Host key for login.math.wisc.edu has changed and you have requested strict
checking.
Host key verification failed.
```

Logging in with SSH

- The remote host fingerprint will only change when:
  - The server SSH key has changed due to server upgrade or re-installation. This is something for me to look into. I should let you know when this is expected to happen.
  - The server has been renamed in DNS and is using a name that another SSH server once used and your SSH client has the old fingerprint for the name.
  - There is a DNS problem.
  - Your computer is compromised and malicious actor interfering
  - A malicious actor is intercepting and redirecting your SSH traffic
- All but the first are reasons to report this to IT staff

Logging in to SSH

- After logging in successfully, SSH will run your default shell. Probably Bash.

The SSH Config File

- You can permanently set numerous per-host settings to save you time each time you connect.
- Your SSH settings folder is in the ".ssh/" folder of your home directory.
- Create a file in ".ssh/" named "config"
- In it, add these lines:

```
Host *.math.wisc.edu
    User YOUR_MATH_USER_NAME
```

- This tells SSH to use the same user name for All math.wisc.edu hosts
- Now you can connect like this:
  `ssh rossby.math.wisc.edu`

The SSH Config File

- You can also create short names. Add this to the file:

```
Host greenbay
    HostName greenbay.math.wisc.edu
```

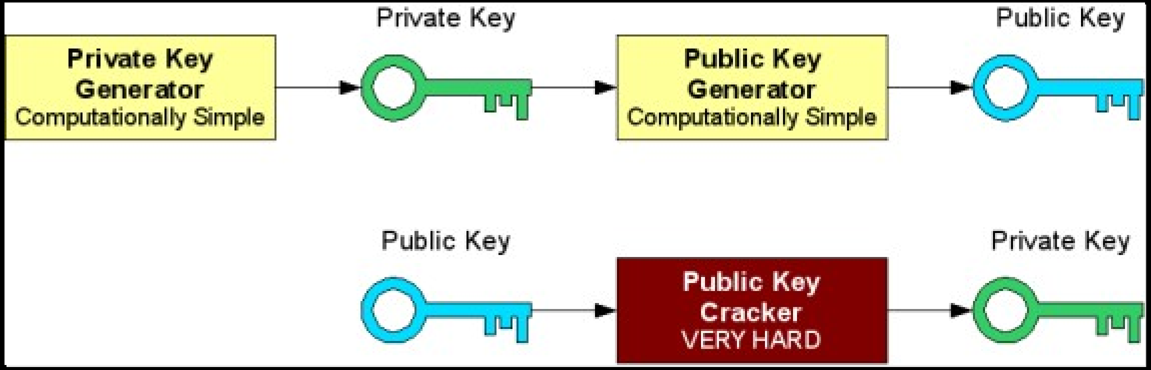
- Then you only need to do this to connect:

```
    ssh greenbay
```

- Note that the previous entry for *.math.wisc.edu applies also.

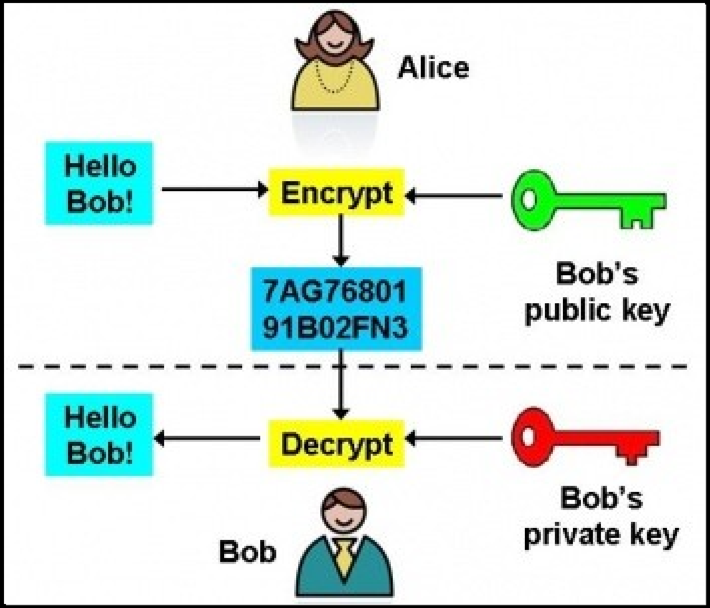
SSH Public-Key Authentication

- SSH can use public-key cryptography for authentication.
- Each SSH server and SSH client needs a key pair

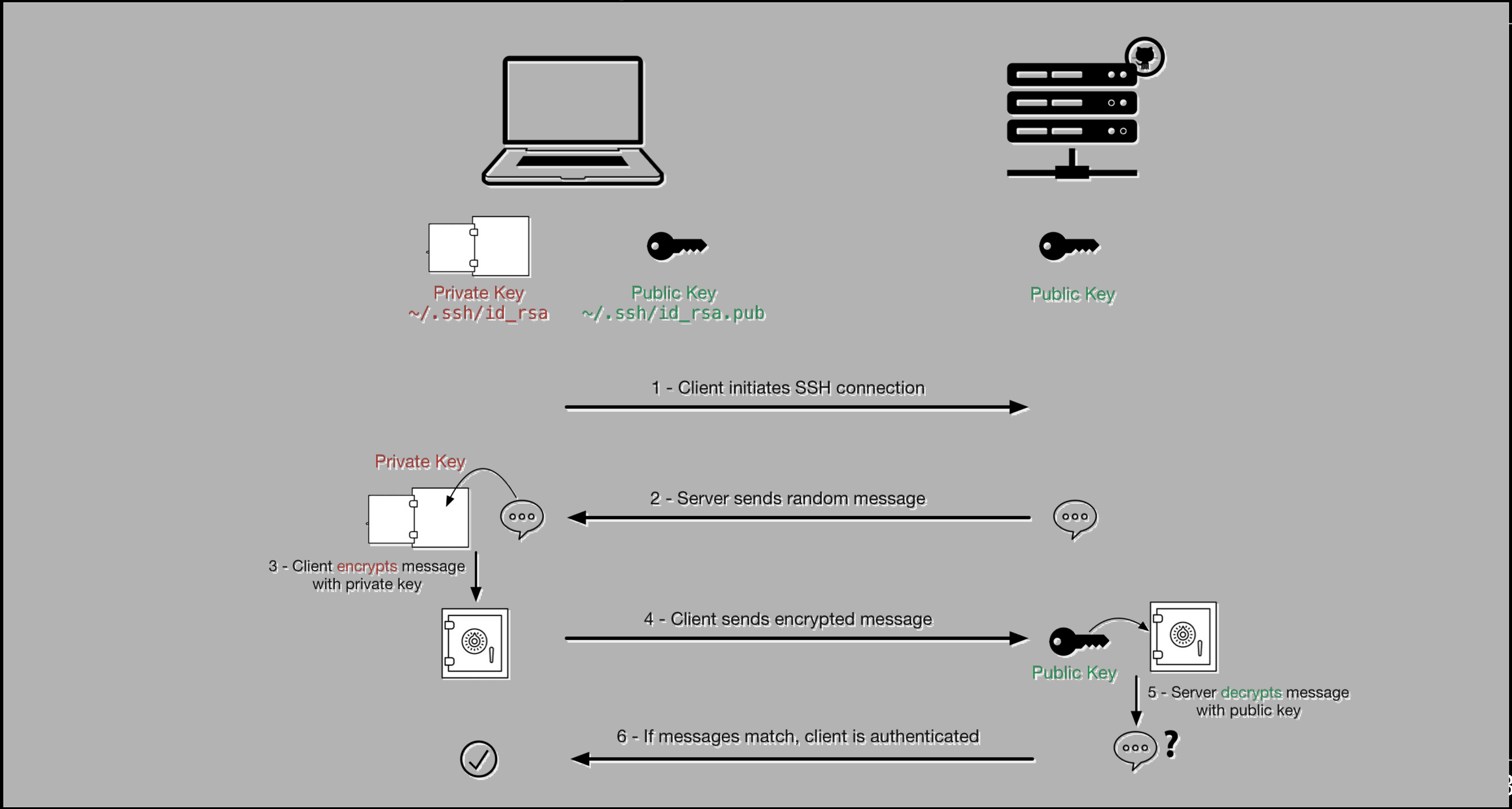


SSH Public-Key Authentication

- Just as PK cryptography can be used to authenticate users as well as encrypt messages
- SSH servers may require SSH key authentication instead of or in addition to regular passwords



SSH Public-Key Authentication



1 - Client initiates SSH connection

2 - Server sends random message

3 - Client encrypts message with private key

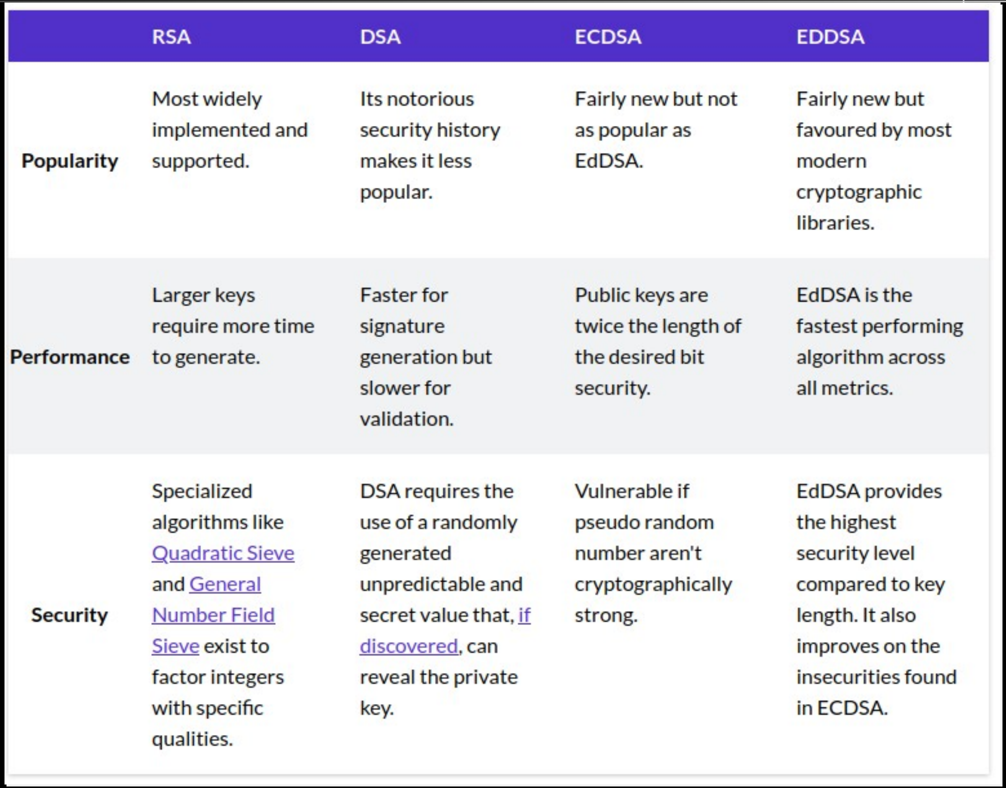
4 - Client sends encrypted message

5 - Server decrypts message with public key

6 - If messages match, client is authenticated

SSH PK Authentication: Create your keys

- Check your ".ssh" folder.
  Key file names look like this:
  id_ecdsa
  id_ecdsa.pub
  id_ed25519
  id_ed25519.pub
  id_rsa
  id_rsa.pub
- Which is best?
  Ed25519, then RSA, DSA
- Open a terminal and generate a key:
  ssh-keygen -t ed25519
  or
  ssh-keygen -t rsa -b 4096



	RSA	DSA	ECDSA	EDDSA
Popularity	Most widely implemented and supported.	Its notorious security history makes it less popular.	Fairly new but not as popular as EdDSA.	Fairly new but favoured by most modern cryptographic libraries.
Performance	Larger keys require more time to generate.	Faster for signature generation but slower for validation.	Public keys are twice the length of the desired bit security.	EdDSA is the fastest performing algorithm across all metrics.
Security	Specialized algorithms like Quadratic Sieve and General Number Field Sieve exist to factor integers with specific qualities.	DSA requires the use of a randomly generated unpredictable and secret value that, if discovered, can reveal the private key.	Vulnerable if pseudo random number aren't cryptographically strong.	EdDSA provides the highest security level compared to key length. It also improves on the insecurities found in ECDSA.

SSH PK Authentication: Create your keys

```
emeitner@krug:~$ ssh-keygen -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/emeitner/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/emeitner/.ssh/id_ed25519
Your public key has been saved in /home/emeitner/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:jBm2wlrxMMRnW65ulXk88fEyW1Rr+LxyAYFmk0llZMs emeitner@krug
The key's randomart image is:
+--[ED25519 256]--+
|   ..     ..*=   |
|   .. o .  B+.. .|
|    +oo+  o .Eo..|
|   . *.*. . .o.o |
|    + =.S+ o +=  |
|   o .. + + + o+ |
|  .  . . . . =  o|
|      o     .. o |
|     .        o  |
+----[SHA256]-----+
```

SSH PK Authentication: Create your keys

- Always save it in the location offered.
- If you decide to add a passphrase to your private key, you will need to enter it every time SSH needs access to the key when connecting.

SSH PK Authentication: The public keys

- If there is a file named "authorized_keys" in the users ".ssh" folder on an SSH server it will be checked for SSH public keys that can be used for authentication.
- Keys are added to this file manually. One line per key.
- An example authorized_keys file:

```
ssh-ed25519
AAAAC3NzaC1lZDI1NTE5AAAAIBVgKx4kd84jRPWS2MHBqR6rxehzFasnlu2Imbj8iS5e joe@chbs
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABgQDIfA6RmfDUnn/lKdLMrt5aJrpedsfhg5DcBkdxsd0XM/
zmlPMuHTGEhwgH+c/ftJttEOqCXAIN+dOepfJk+0YuXaSUshFWLUx2fXJFhw/
hmY+hcnyHV5jjhyjkycb6SRHsc8GgiLU2Z7l5VtUoXRksYc84vP7lSBxag9C5MUJxpZi/
DMhMumxGFdykNRYAFlzZuruMereNnsC4mDF0ytdSPmkDlE5YE63YtxK7nFyOOAi7INYnw9fnO4ZWU
PTdYZ9QF9hlqh1nO1pQT6kV0ZNgOJ81WcsHXVbruYodSs= fran@shelf.library.wisc.edu
```

SSH PK Authentication: The public keys

- SSH to login.math.wisc.edu then do
  ```
  mkdir -p ~/.ssh
  chmod 700 ~/.ssh
  ```
- Then from your computer where you created the SSH key pair:
  ```
  cat .ssh/id_ed2559.pub
  ```
- Copy the public key text
- On login.math.wisc.edu append this text to the authorized_keys file. Run this command:
  ```
  cat >> ~/.ssh/authorized_keys
  ```
- Now paste the content of the public key
- Press Enter, the CTRL-D to stop appending.

SSH PK Authentication: Testing

- Now your account will authenticate using the keypair on your computer.
- If you have a second computer you can create a key pair for it and add the public key to the authorized_keys file as well.
- The login process will be passwordless IF you did not set a passphrase on the private key

A Note on Private Keys

- THE SECURITY OF YOUR ACCOUNTS DEPENDS ON THE SECURITY OF THE SYSTEM YOUR PRIVATE KEY IS STORED ON. KEEP. IT. SAFE.
- Do not copy the key onto untrusted systems.
- Be careful where you store backups.
- Do not share your private key.
- In the event that your private key is compromised simply remove(or have IT remove) your public keys from your server account(s).

Jump Hosts

- Depending on the network you are on(VPN, Wireless, residence halls) you may need first connect to login.math.wisc.edu before connecting to our research servers.
- That involves:
  ```
  you@yourcomputer:~$ ssh login.math.wisc.edu
  ```
  then:
  ```
  you@login:~$ ssh rossby
  ```
- How about in one command?
  ```
  ssh -J login rossby
  ```
- The -J option tells SSH to use login as a "jump host" to get to rossby
- This can also be added to the SSH config file.

SSH config File Example

```
Host *.j
    ProxyJump login.math.wisc.edu

Host login
    User yourusername
    HostName login.math.wisc.edu
Host greenbay greenbay.j
    User yourusername
    HostName greenbay.math.wisc.edu
Host freetown freetown.j
    User yourusername
    HostName freetown.math.wisc.edu
Host edmonton edmonton.j
    User yourusername
    HostName edmonton.math.wisc.edu
Host rossby rossby.j
    User yourusername
    HostName rossby.math.wisc.edu
Host hongkong hongkong.j
    User yourusername
    HostName hongkong.math.wisc.edu
```

X11 Forwarding

- The Linux desktop environment(based on X Window system, aka X11) allows for a local computer to act as a display for a remote one.
- You can run a graphical application on one of the servers and interact with it on one of the Linux workstations.
- Add the "-X" option to the SSH command. This enables X11 forwarding to the local machine.
- Example:

  ```
  you@yourlaptop:~$ ssh -X rossby
  you@rossby:~$ mathematica
  ```

- Note: This requires a **LOT** of network bandwidth. If you try this from home over a VPN you may experience a lot of screen update delays and input lag.
- The config file option is: ForwardX11

  ```
  Host myhost
      ForwardX11
  ```

Port Forwarding

- Scenario: you want to run Jupyter Notebook on server Rossby. When you do this it tells you to go to this URL in your web browse: http://127.0.0.1:8888/?token=8ec4143e2d235
- If you open the URL in your computer's web browser you get an error. That's because the network address 127.0.0.1/localhost is local to the server and is accessible only by the server.
- How to connect to this then? Port forwarding.
- We want to connect to port 8888 on the server from our computer that has SSH.
- We are going to create a local port(local to our computer) that gets forwarded to a port on the server(remote port).
- The option syntax is: -L LOCAL_PORT:REMOTE_HOST:REMOTE_PORT
- In our case we will forward port 8888 on our laptop to port 8888 on the server:

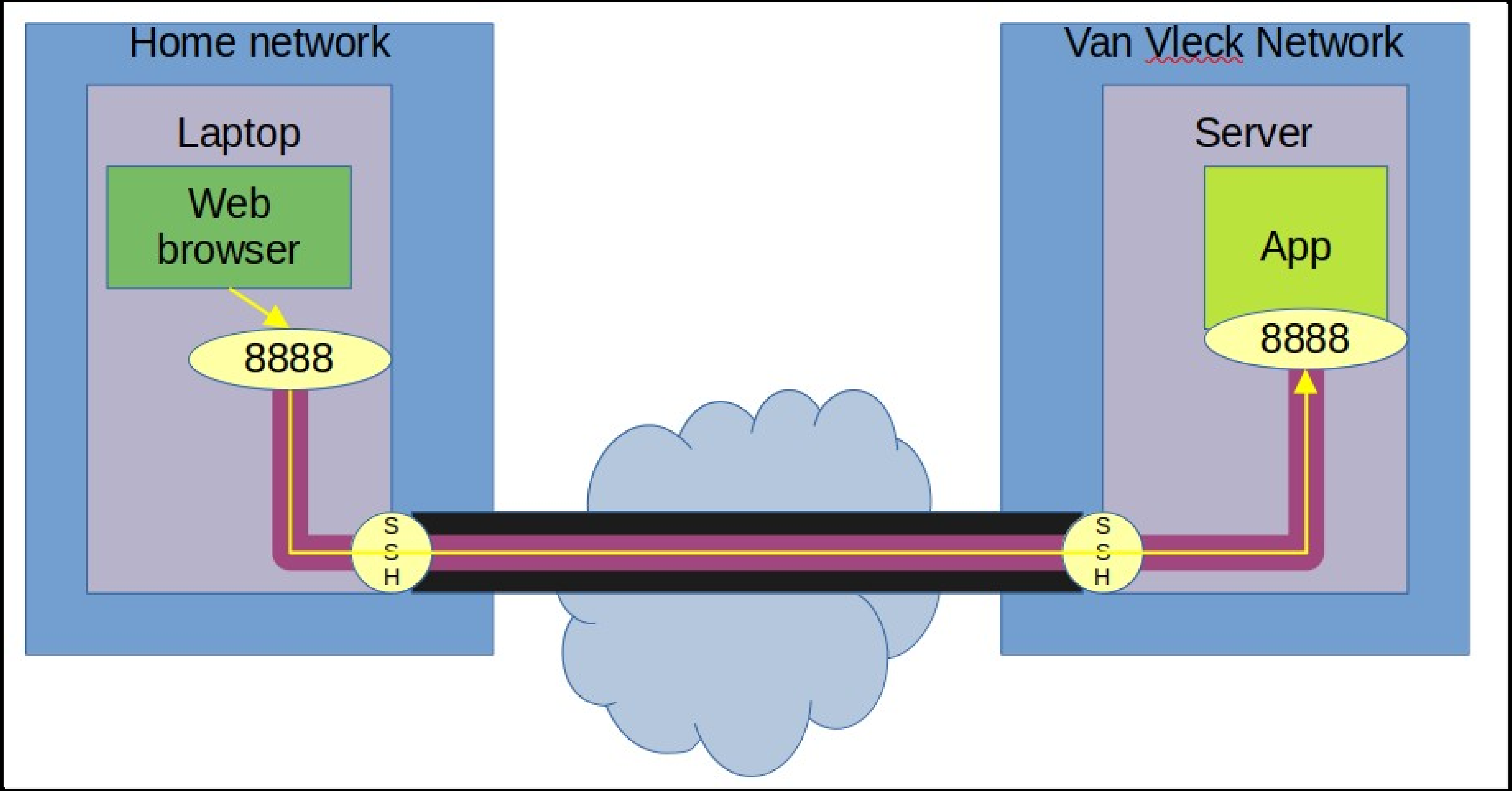
  ```
  ssh -L 8888:localhost:8888 rossby
  ```

  or

  ```
  ssh -L 8888:rossby:8888 rossby
  ```

- Note that "localhost" is telling the remote computer to connect forward to itself.

Port Forwarding



SSH PK Authentication and Jump Hosts

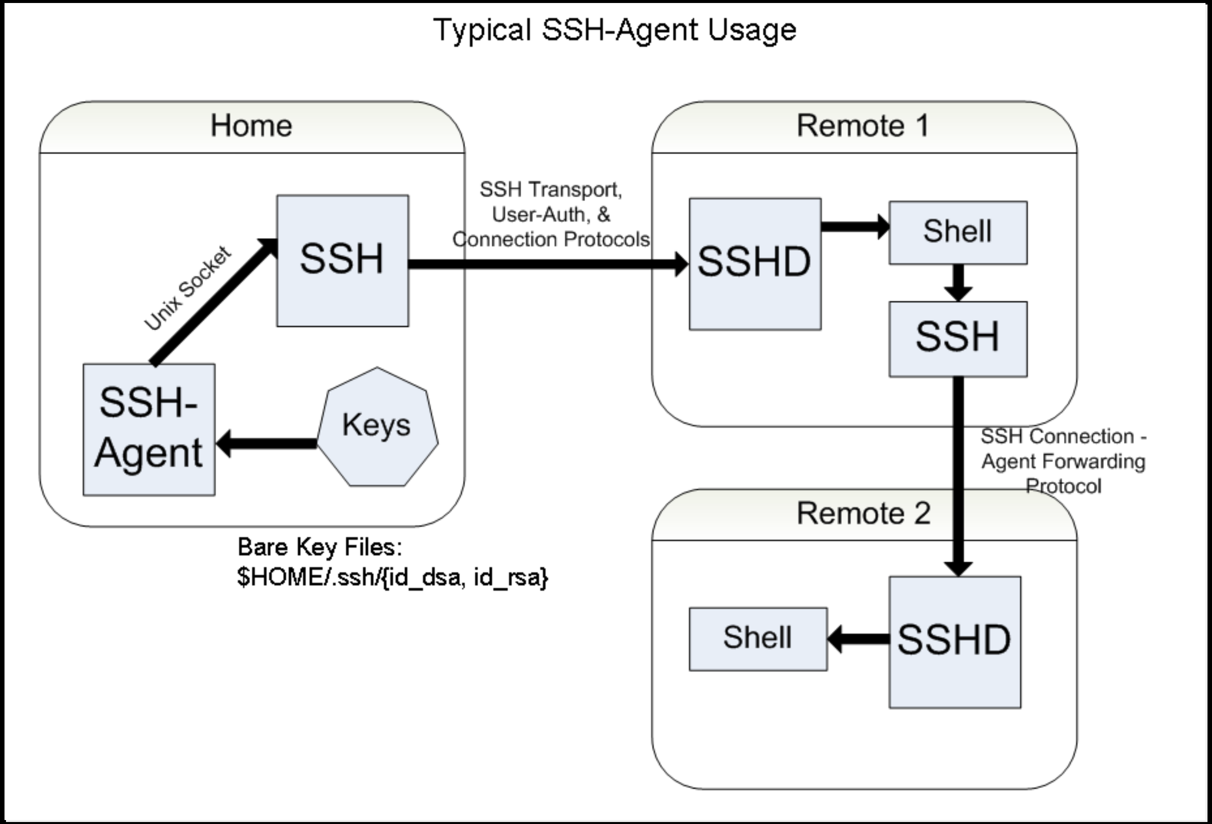
- When your laptop uses PKA when connecting to a server it can do so because it has the SSH private key on it.
- What about when you SSH to one server(Login.math.wisc.edu for example) and from there to Edmonton? Unless you copied your private key to Login(not recommended) you will not be able to use PKA.
- SSH has a program called ssh-agent that runs on your computer. It's job is to hold private keys used for PKA.
- To set up forwarding so that when you run SSH on Login it has access to the private key via ssh-agent:

  ```
      ssh -A login
  ```

  or the config file option:

  ```
      ForwardAgent yes
  ```

SSH PK Authentication and Jump Hosts

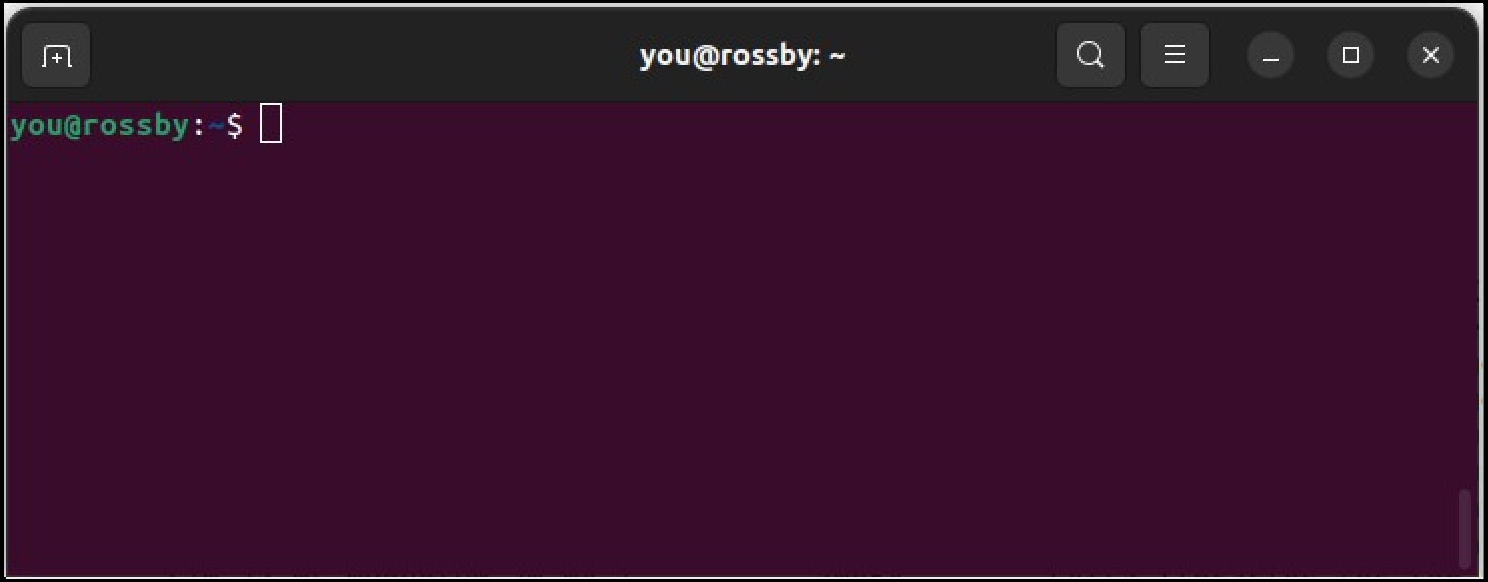

Typical SSH-Agent Usage
Home
SSH
Unix Socket
SSH-Agent
Keys
SSH Transport, User-Auth, & Connection Protocols
Remote 1
SSHD
Shell
SSH
SSH Connection - Agent Forwarding Protocol
Remote 2
Shell
SSHD
Bare Key Files:
$HOME/.ssh/{id_dsa, id_rsa}

SSH config File Example

```
Host *.j
    ProxyJump login.math.wisc.edu

Host login
    User yourusername
    HostName login.math.wisc.edu
    ForwardAgent yes
Host greenbay greenbay.j
    User yourusername
    HostName greenbay.math.wisc.edu
Host freetown freetown.j
    User yourusername
    HostName freetown.math.wisc.edu
Host edmonton edmonton.j
    User yourusername
    HostName edmonton.math.wisc.edu
Host rossby rossby.j
    User yourusername
    HostName rossby.math.wisc.edu
Host hongkong hongkong.j
    User yourusername
    HostName hongkong.math.wisc.edu
    LocalForward 8888 localhost:8888
```

Question and Answer Session



```
you@rossby:~$ 
```

Notes from today

- Will be posted on the Math Dept. wiki: https://wiki.math.wisc.edu/
- Search for “lunch and learn”

Next time

- Time, date, and topic to be announced on the mailing list
- To join the list send an email to: math-linux-help+join@g-groups.wisc.edu

Contacting me

- You can always contact me directly with questions:
  emeitner@math.wisc.edu
  608-263-4189(office)
- Or stop by my office:
  Van Vleck room 507

Thank you