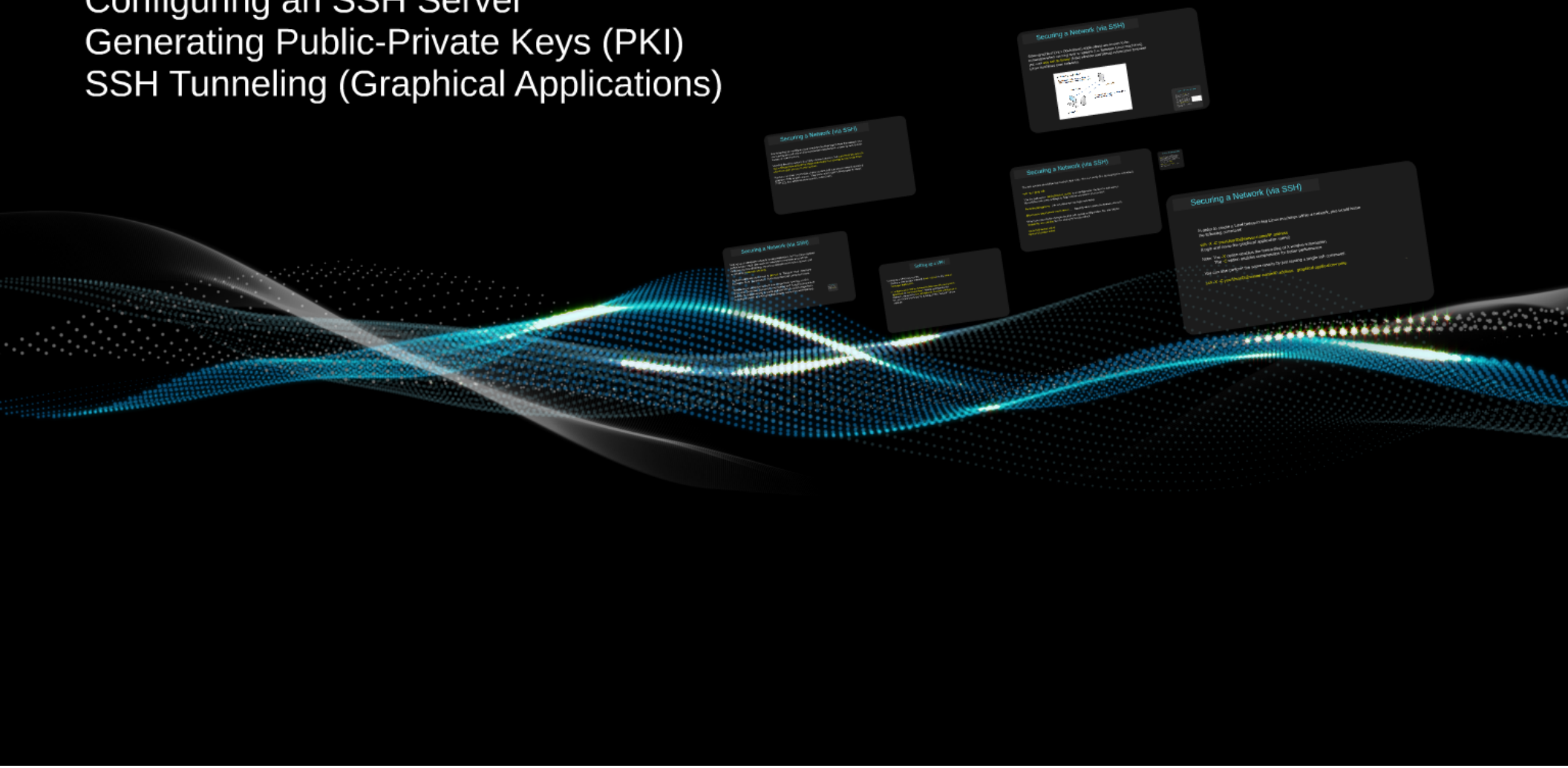
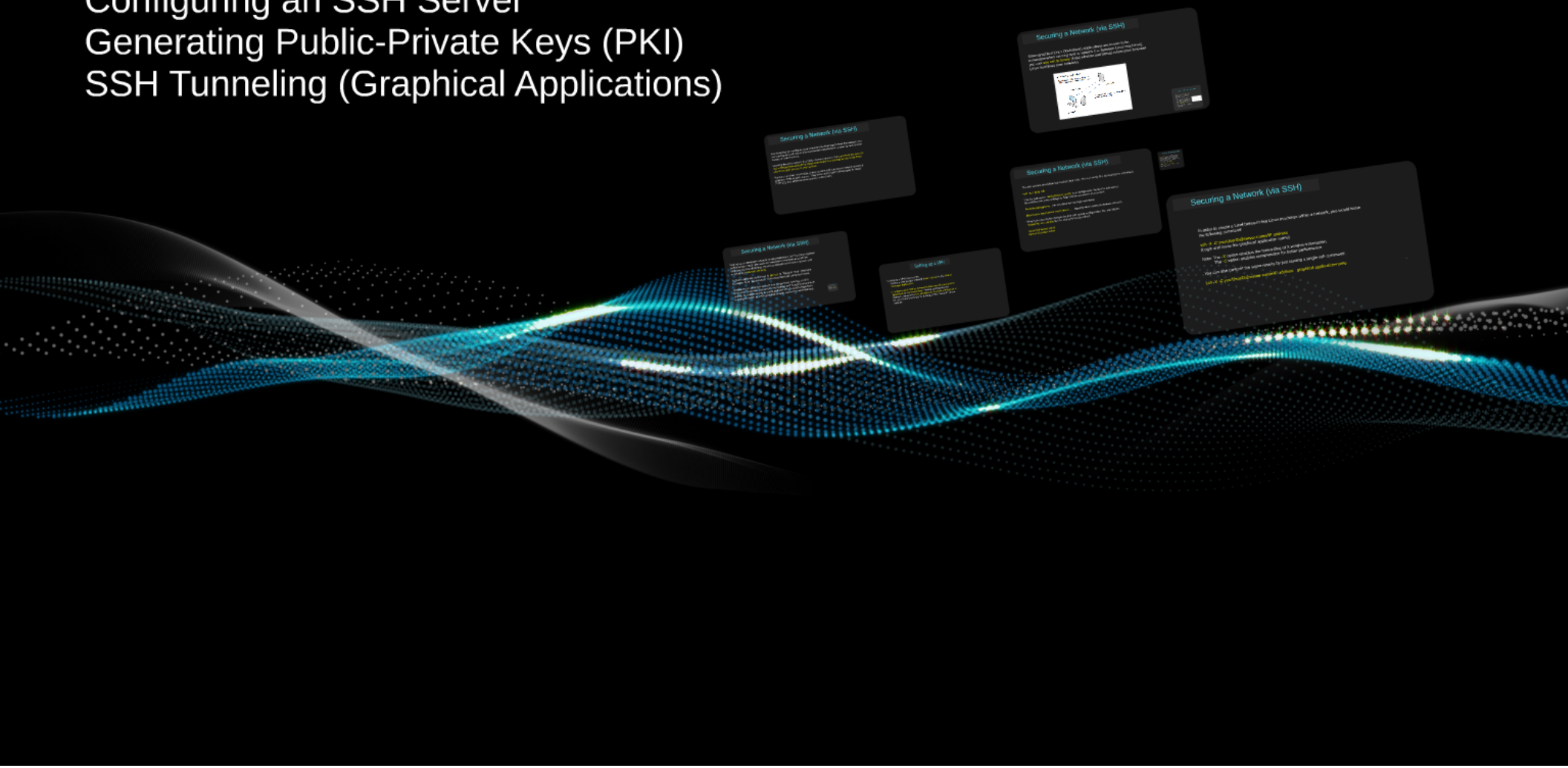
Securing a Network (via SSH)

Configuring an SSH Server
Generating Public-Private Keys (PKI)
SSH Tunneling (Graphical Applications)



Securing a Network (via SSH)

Configuring an SSH Server
Generating Public-Private Keys (PKI)
SSH Tunneling (Graphical Applications)



Securing a Network (via SSH)

Setting up a computer network is very important, but the Linux system administrator must also perform networking maintenance which includes trouble-shooting, repairing network connection issues and maintaining network security.

System administrators need to protect or "harden" their computer networks from "penetration" from unauthorized computer users.

Hardening a computer system can range from running an IDS (Intrusion Detection System) to monitoring and flagging suspicious activity to implementing security policies which could range from running firewalls to setting locked screen savers on workstations.

Securing a Network (via SSH)

We don't have the time to discuss in details how to harden your computer network - that would take at least another class (eg. SEC520) or could take an entire program (eg. IFS).

In lab7 we will focus on using SSH in order to help secure your Linux network. The topics will include how to configure and run an SSH server, generate public/private keys and use shared public keys to login to remote Linux servers, and use SSH tunneling to run Linux applications from remote servers, but display them on your local Linux machine.

Securing a Network (via SSH)

Any time that you configure your computer to allow logins from the network you are leaving yourself vulnerable to potential unauthorized access by penetration testers or even hackers.

Running the sshd service is a fairly common practice but care must be taken to make things more difficult for those individuals that attempt to use brute force attacks to gain access to your system.

Hackers use their knowledge of your system and can use password guessing programs help to gain access. They know which port is likely open to attack (TCP:22), the administrative account name (root).

Securing a Network (via SSH)

The ssh service should be installed on your VMs. You can verify this by issuing the command:

rpm -qa | grep ssh

The file pathname /etc/ssh/sshd_config is a configuration file for the ssh server.
You will be using two settings to help harden access to your server:

PermitRootLogin=no (do not allow root to login remotely)

AllowUsers username1 username 2 ... (specify which users to connect via ssh)

Whenever you make changes to your ssh server configuration file, you MUST
restart the ssh service for the changes to take effect:

systemctl restart sshd
systemctl status sshd

Securing a Network (via SSH)

As a system administrator, you have the ability to generate or create public and private keys to ensure safe and secure ssh connections. This will require a user to prove who they say they are in order to access a Linux server via SSH (i.e. authentication). The system administer can generate these keys for the first time, or if the system administrator suspects that a hacker has compromised or trying to penetrate the server, they can remove the existing keys and generate new keys.

Generate public/Private key pair: ssh-keygen

Copy public key to remote server:
ssh-copy-id -i ~/.ssh/id_rsa.pub userid@servername

Note: it is important to know which local and which remote userid will use the share public key!

Securing a Network (via SSH)

A common type of attack, Arp Poisoning (a.k.a: DNS spoofing or Man in the Middle Attack), can be used to redirect packets to a third party while maintaining the illusion that the connection is secure.

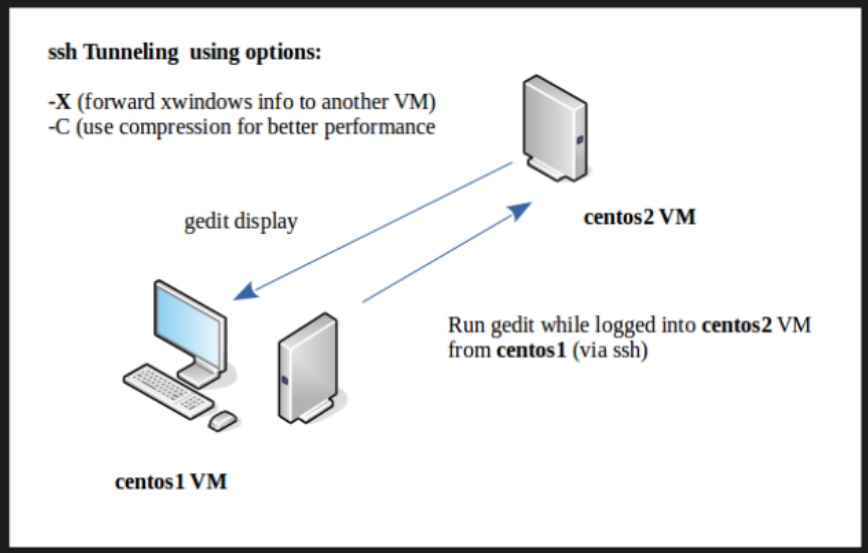
If you ever receive a message like the one displayed on the right, you should investigate why it is happening as it could indicate a serious security issue, or it could just mean that something on the host has changed(i.e. the OS was reinstalled). You can always generate a new set of public/private keys to be on the same side...

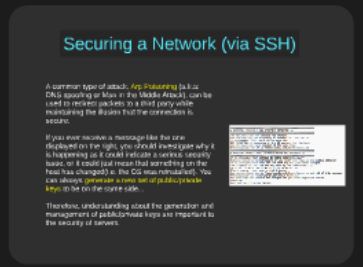
Therefore, understanding about the generation and management of public/private keys are important to the security of servers

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@ WARNING: POSSIBLE DNS SPOOFING DETECTED! @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
The RSA host key for centos3 has changed,
and the key for the according IP address 192.168.235.13
is unchanged. This could either mean that
DNS SPOOFING is happening or the IP address for the host
and its host key have changed at the same time.
Offending key for IP in /home/user1/.ssh/known_hosts:10
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@ 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 the RSA host key has just been changed.
The fingerprint for the RSA key sent by the remote host is
96:92:62:15:90:ec:40:12:47:08:00:b8:f8:4b:df:5b.
Please contact your system administrator.
Add correct host key in /home/user1/.ssh/known_hosts to get rid of this message.
Offending key in /home/user1/.ssh/known_hosts:53
RSA host key for centos3 has changed and you have requested strict
checking.
Host key verification failed.
```

Securing a Network (via SSH)

Since graphical Linux (Xwindows) applications are known to be vulnerable when running over a network (i.e. between Linux machines), you can use ssh to tunnel (hide) window and bitmap information between Linux machines over networks.





Securing a Network (via SSH)

In order to create a tunel between two Linux machines within a network, you would issue the following command:

ssh -X -C yourUserID@server-name/IP address
(Login and issue the graphical application name)

Note: The -X option enables the forwarding of X window information
The -C option enables compression for better performance

You can also perform the same results by just issuing a single ssh command:

ssh -X -C yourUserID@server-name/IP address graphical-application-name

Securing a Network (via SSH)

Configuring an SSH Server
Generating Public-Private Keys (PKI)
SSH Tunneling (Graphical Applications)