SSH - Lab

We will now practice the following concepts:

- The use of known_hosts files
- SSH connection with password authentication
- RSA version 2 protocol key generation
- Public key copying
- Connecting with private key passphrase using key-based authentication
- Using scp with RSA key authentication
- Some ssh “hacks” without passwords.

*Technically you are still challenged (even if that is a bad pun in English).

SSH - Lab Cont.

The use of known_hosts files

Connect to the machine next to your machine using ssh:

```
ssh admin@pcN.cctld.pacnog2.dnsdojo.net
```

If this is your first connection to this machine you should see (example uses host1 connecting to host2):

```
pc1# ssh admin@pc2.cctld.pacnog2.dnsdojo.net
The authenticity of host 'pc2.cctld.pacnog2.dnsdojo.net (192.216.0.2)'
can't be established.
RSA1 key fingerprint is 60:f7:04:8b:f7:61:c4:41:6e:9a:6f:53:7d:95:cb:29.
Are you sure you want to continue connecting (yes/no)?
```

Go ahead and answer “yes” here, but we'll discuss the implications of this in class. Are there ways around this? Could this be a “man in the middle” attack? What file is created or updated? Why?

SSH - Lab Cont.

ssh connection with password authentication

At the prompt below when you answered yes, you were asked to enter in the admin password for pc2.cctld.pacnog2.dnsdojo.net:

```
host1# ssh admin@pc2.cctld.pacnog2.dnsdojo.net
The authenticity of host 'pc2.cctld.pacnog2.dnsdojo.net (192.216.0.2)' can't
be established.
RSA2 key fingerprint is 60:f7:04:8b:f7:61:c4:41:6e:9a:6f:53:7d:95:cb:29.
Are you sure you want to continue connecting (yes/no)? yes
```

And, this is what you should have seen:

```
Warning: Permanently added 'pc2.cctld.pacnog2.dnsdojo.net' (RSA2) to the list
of known hosts.
[/etc/ssh/ssh_host_key.pub]
admin@pc2.cctld.pacnog2.dnsdojo.net's password:
```

Now you are “securely” connected as admin to pc2.cctld.pacnog2.dnsdojo.net - We will discuss what happened during this connection.

SSH - Lab Cont.

rsa1/rsa2/dsa Key Generation

We will now generate a single RSA SSH protocol 2 key of 2048 bits. To do this, issue the following command. If you are logged in on the other machine, logout first!

```
ssh-keygen -t rsa -b 2048
```

You will be prompted for a file location for the key as well as for a passphrase to encrypt the key file. Be sure to enter a passphrase. Private key files without passphrases are a security hole, or maybe not... We'll discuss this as we complete this excercise. You can use a passphrase other than what was given in class for the *admin* account if you wish.

SSH - Lab Cont.

RSA 2 Key Generation

Here is the output from the command "ssh-keygen -t rsa -b 2048":

```
pc1# ssh-keygen -t rsa -b 2048
Generating public/private rsa key pair.
Enter file in which to save the key
(/admin/.ssh/id_rsa): [enter]
Enter passphrase (empty for no passphrase): [pw]
Enter same passphrase again: [pw]
Your identification has been saved in
/admin/.ssh/id_rsa.
Your public key has been saved in
/admin/.ssh/id_rsa.pub.
The key fingerprint is:
0f:f5:b3:bc:f7:5b:c8:ce:79:d0:b1:ab:2c:67:21:62
admin@pc1.ws.cctld.ke
pc1#
```

SSH - Lab Cont.

Public Key Copying

Now that you have a public and private RSA(2) set of keys you can take advantage of them. We will copy the public key to the same host you connected to previously, save this to the files *known_hosts*, and then reconnect to the host and see the difference:

First you must copy the public key files to the host you used previously (pcn.cctld.pacnog2.dnsdojo.net):

```
cd ~/.ssh
scp id_rsa.pub
admin@pcn.cctld.pacnog2.dnsdojo.net:/tmp/.
```

You will be prompted for the password for the host *and* username you are connecting to. We continue with our example using pc1 connecting to pc2 as admin.

SSH - Lab Cont.

Public Key Copying

The output from the command on the previous page looks like:

```
pc1# scp *.pub admin@pc2.ws.cctld.ke:/tmp/.
admin@pc2.ws.cctld.ke's password:
id_rsa.pub           100% |*****************************|   408       00:00

pc1#
```

You now have the public key file sitting on the host that will need them to use RSA/DSA public/private key authentication with you. You next step is to place these keys in the appropriate files.

You need the RSA keys in *~/.ssh/authorized_keys*

You can try to figure this out, or go to the next slide for steps to do this:

SSH - Lab Cont.

Public Key Copying

To copy the public keys to the correct places do the following:

```
ssh admin@pcn.cctld.pacnog2.dnsdojo.net
cat /tmp/id_rsa.pub >> ~/.ssh/authorized_keys
rm /tmp/id_rsa.pub
exit
```

If you are unsure of what these commands do they will they are explained in class. In addition, you can do this many different ways, and you could issue the commands differently as well. If you understand what these commands do and have a preferred method, then feel free to use it.

Go to the next slide to connect with your public/private keys!

SSH - Lab Cont.

Public/Private Key Connection

To connect using your RSA protocol 2 key simply type:

```
ssh admin@pcn.cctld.pacnog2.dnsdojo.net
```

And, here is the output you should see (pc1 to pc2 example):

```
host1# ssh admin@pc2.cctld.pacnog2.dnsdojo.net
Enter passphrase for RSA key
'admin@pc1.cctld.pacnog2.dnsdojo.net':
```

This is actually pretty neat! You *did not* enter in the admin password for the admin account on pcn.cctld.pacnog2.dnsdojo.net, but rather you used the passphrase that you chose for your private RSA protocol 2 key when you issued the command "ssh-keygen -t rsa -b 2048" - This was used to decode the encoded random number exchanged between the hosts (remember "Magic Phrase?").

Why was the RSA protocol 2 key used? We'll discuss this in class.

SSH - Lab Cont.

SCP Public/Private Key Connection

First disconnect from the ssh session you previously made:

```
exit
```

Now, try copying a file from your machine to the other machine (pick a small file) using SCP (SeCure coPy):

```
scp filename admin@pcn.cctld.pacnog2.dnsdojo.net:/tmp/.
```

What did you notice? You should have noticed that you no longer get a password challenge to this account on this node, but rather you need to provide your RSA protocol 2 private key passphrase.

This is expected. SCP and SSH are from the same package - OpenSSH and both use RSA and DSA keys in the same way.

SSH - Lab Cont.

Another SSH tool - SFTP

In addition to scp, ssh has a secure ftp tool called sftp. Give it a try:

Let's use sftp to get your neighbor's /etc/motd file and place it in your /tmp directory.

```
sftp admin@pcN.cctld.pacnog2.dnsdojo.net
```

Once you are connected:

```
sftp> lcd /tmp      [change local directory to /tmp]
sftp> cd /etc       [change remote directory to /etc]
sftp> get motd      [download /etc/motd to /tmp/motd]
sftp> ?             [view summary help]
sftp> bye           [terminate connection]
ls /tmp/motd        [prove you got the file]
```

SSH - Lab Cont.

Now let's use the power of scp

Multiple file and directory copy:

Let's copy all the files and directories in /usr/ports/palm from your machine to your neighbor's machine using one command (1.4Mb):

```
scp -r /usr/ports/palm/*
    admin@pcN.cctld.pacnog2.dnsdojo.net/tmp/.
```

- "-r" for recursively copy
- "/tmp/." to place files in your neighbor's /tmp directory.

SSH - Lab Cont.

Now let's use the power of scp some more!

(Note: we may skip this exercise...)

Copy a file from one remote machine to another.

Let's move /etc/fstab on your left neighbor's machine to /tmp/fstab.copy on your right neighbor's machine using a single command.

```
scp  admin@pcLEFT.cctld.pacnog2.dnsdojo.net:/etc/fstab \
   admin@pcRIGHT.cctld.pacnog2.dnsdojo.net/tmp/fstab.copy
```

- "\" for newline, not part of the command.
- If admin password is the same on both you only enter it once.
- Did you notice we renamed the file as well?