SSH Security

Introduction and/or Background

It is useful to have increased secuirty when dealing with SSH (Secure SHell) sessions. As an added bonus we the end user can also gain passwordless access to the remote server account assigned to user on the server.

To set up the parameters for SSH requires two items, the server to be accessed and the client to access the server. To accomplish this is a two step process. First is to generate a key set public and private on the server to be accessed. Second to pass the public key component to the client device.

Reminder: One must have an active account on the server. The key pair for one client connection will only be good for the account associated on the server. Fact if someone is foolish enough to pass around public keys really does not need SSH to begin with!

There are four file sets associated with rsa based ssh keys --

id_rsa
id_rsa.pub
known_hosts
authorized_keys

id_rsa and id_rsa.pub are the private and public key pair for any given rsa key set client to host. Known_hosts is a list of the fingerprints of known devices logged into the device. authorized_keys is a list of rsa keys provided to the server from the client. authorized_keys is distinct from known_hosts in this regard.

Two tools are used ssh-keygen, and ssh-copy-id. ssh-keygen will create a set of unique keys. One has the option to either use a passphrase or not. It is recommended however to include the passphrase to prevent unauthorized use of the pub key. ssh-copy-id is used to copy the copy the public key to the target device. If a passphrase is used the operator will be challenged to provide it before the copy proceeds.

Objectives
In this project/lab the student will:

- Gain familiarity with SSH
- Understand RSA keys

Equipment/Supplies Needed
- As specified in Lab 0.0.1.
- Linux Installation File: Debian server (VMSVR1) and client (VMPC1)

Procedure
Perform the steps in this lab in the order they are presented to you. Answer all questions and record the requested information. Use the Linux Virtual Machine to perform lab activities as directed. Unless otherwise stated, all tasks done as a non-root user. If root access is needed use the sudo command.

Beginning

Start with clearing out all the fluff. On your client (VMPC1) open the termianal and perform the following:

1. ls -la.
2. cd .ssh
3. rm *.* or via the individual file names

On your server (VMSVR1) login as your user and open a terminal session and repeat the steps 1,2,3 as you just did on VMPC1.

You now have a clean slate on both devices.

Server Side

At this point you should still be on VMSVR1, in the .ssh directory of the user you logged in as. You will use a tool called ssh-keygen to create a public-private key pair. Execute:

4. ssh-keygen -t rsa -b 2048 -C “<acct>@<myserver ip>”

Where -

t determines the encryption type, rsa in this case.
b determines the byte size. It can be any power of 2, in this case 2048
C determines the contents of a comment for human readable reference

Example output --

```
drdog@drdog-Dell:~/.ssh$ ssh-keygen -t rsa -b 2048 -C "my first"
Generating public/private rsa key pair.
Enter file in which to save the key (/home/drdog/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/drdog/.ssh/id_rsa
Your public key has been saved in /home/drdog/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:DJdq4s5eVyuzee3h4oOFbfVYQ/Wn7CDPv5t98YPt3OQ my first
The key's randomart image is:
+---[RSA 2048]----+
|               ..|
|         .    . .|
|      . o    .  o|
|       =    ..o..|
|    . o Soo..+o. |
|   . o  ..+=.o.. |
|    . . ++..+ + +|
|   o . ..=+..+.B+|
|   .+   oo.+o *=E|
+----[SHA256]-----+
drdog@drdog-Dell:~/.ssh$
```

Note that you are asked where to put the file with a default suggested location. Accept that suggestion. Next you are asked for a passphrase. Do so. This is optional and can be left empty by pressing enter, and again. If you want the passphrase you need to enter it and repeat it again to complete.

Now take a look at your .ssh directory.

```
drdog@drdog-Dell:~/.ssh$ ls -la
total 16
drwx------  2 drdog drdog 4096 Aug  1 18:11 .
drwxr-x--- 33 drdog drdog 4096 Aug  1 17:44 ..
-rw-------  1 drdog drdog 1856 Aug  1 18:11 id_rsa
-rw-r--r--  1 drdog drdog  390 Aug  1 18:11 id_rsa.pub
drdog@drdog-Dell:~/.ssh$
```

Two entries associated now --

id_rsa is the private key.
id_rsa.pub is the public key.

Provide a screen shot of the result.

Here is the contents of the public key -

```
drdog@drdog-Dell:~/.ssh$ cat id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDUGy5/V3KTp99J6yBqkNkPOkvyvLVQq11BUBkfdwv
/eEZdETsBCENCWlQw7ma/psyP+n0hNhygAGu8MpgO7zI6xQ/oFKzZaxaCjoOw5Dncnlc9cI4nzc7rka
0FYFAvpCZ3Fxwl+2Jsqm9fkRWzF9VITUgF8sNuz0lcsPGagSREk0/GZGNjODkVJdCYE+NZrDjKQZm7P
qc5g3anAXTG+3GzR3RV2fR9+/P22L7YhtyrcgG3ChbiPbMUYxWJA2Wd2WoM7SVYjC+hLgJXTsXCyFJb
6KpdUaOmZn9uz33jdOEIPR9tWk/oDtFCLaNLv57YZaDoPiiCfRU+n2X7ZbRf my first
drdog@drdog-Dell:~/.ssh$
```

Provide a screen shot of the result of your public key.

The public key is what will be shared. The private key is intended to remain private, associated with the user account and stays on the server, VMSVR1 in this case. So we have a pair of keys one of which needs to be shared. But how to securely move the public key?

Well there is a tool for that too, ssh-copy-id. So still on the server execute --

5. ssh-copy-id <your user id>@<your ip address of the client>
6. Provide your password

At this point you will receive a challenge like the one below.

7. Provide your passphrase that you used when you created the key.
8. Ip ad.

Provide a screen shot of the result.

9. Logout

Log in again

10. ssh <your user id>@<your ip address of the client>

Client Side

Now login to your client device. Execute --

11. cd .ssh
12. cat id_rsa
13. cat id_rsa.pub

Provide a screenshot for submission.

So you generated rsa keys on the server using ssh-key-gen, and pushed the public key

to client using ssh-copy-id. Steps 11,12,13 validated that this occured.

For continuity purposes you executed the steps backwards; generated keys from the server to the client then validated access from the server to the client. Most situations would be reversed client --> server. That is the next part of the exercise.

Repeat steps 1-3 to clear out the keys on both the client (VMPC1) and the server(VMSVR1). Then from the client (VMPC1) repeat steps 4-13 again.

Provide a screenshot for submission of the results of steps 11,12,13 from the server side (VMSVR1).

Reflection Questions

1. How can the use of file encryption help protect an organization's sensitive files?
2. What is the purpose of the passphrase when ssh-keygen was utilized?
3. What is my security exposure if I do not use a passphrase?

Lab Submissions Proof: Provide screenshots as indicated in the lab; upload your proof to Canvas for grading.

Rubric

Checklist/Single Point Mastery

Concerns Working Towards Proficiency	Criteria Standards for This Competency	Accomplished Evidence of Mastering Competency
	Criteria #1: Screenshot of key entries (14 points)	
	Criteria #2: Screenshot of ssh public key (14 points)	
	Criteria #3: Screenshot of ssh-copy-id (14 points)	
	Criteria #4: Screenshot of client rsa keys (14 points)	
	Criteria #5: Screenshot of server rsa keys (14 points)	
	Criteria #6: Reflection Ques 1 (10 points)	

	Criteria #7: Reflection Ques 2 (10 points)	
	Criteria #8: Reflection Ques 3 (10 points)	