Trabajo Práctico N°2

Fecha Límite de Entrega: 1 de abril a las 23:55
Profesor: Dr. Javier Echaiz
Auxiliar: Ing. Fernando Pap

Condiciones de Aprobación:

En líneas generales el trabajo debe dar evidencia del desarrollo realizado. En casos puntuales en los que sea conveniente, incluir un archivo `readme.txt` con notas correspondientes. Entregas individuales subiendo contenido al directorio `ARS2022/Practica/Entregas/[nombre]` del Drive y enviando aviso al email fernandopap@gmail.com o al grupo de whatsapp. Se recomienda utilizar github/ Bitbucket/etc. El trabajo debe ser entregado completo. Pasada la fecha límite de entrega se descontará un punto por cada día transcurrido.

Enunciados

Criptografía asimétrica (clave pública/clave privada)

1. Instalar en el browser la extensión Mailvelope. Generar el par de llaves pública/privada para la dirección de correo de Gmail. Revisar la casilla de entrada y verificar la cuenta desde el mail que envía el servidor de Mailvelope. Coordinar con el docente: cuando todos hayan generado el par de llaves, enviar un mail encriptado a cada uno. Leer los mails recibidos de los compañeros en el browser que tiene la extensión Mailvelope instalada, y desde otro browser que no la tenga.

2. Entorno Linux. Escribir un script de shell que copie el contenido de un directorio de nuestro home, en el home de un usuario de otra computadora. El script no debe revelar ninguna contraseña. Utilizar el anexo.

3. Investigue qué es DKIM y explique cómo funciona y para qué se utiliza.

Anexo

SSH login without password

Your aim
You want to use Linux and OpenSSH to automate your tasks. Therefore you need an automatic login from host A / user a to Host B / user b. You don't want to enter any passwords, because you want to call `ssh` from a within a shell script.
How to do it
First log in on A as user a and generate a pair of authentication keys. Do not enter a passphrase:

```
a@A:~> ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/a/.ssh/id_rsa):
Created directory '/home/a/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
```

```
Your identification has been saved in /home/a/.ssh/id_rsa.
Your public key has been saved in /home/a/.ssh/id_rsa.pub.
The key fingerprint is:
3e:4f:05:79:3a:9f:96:7c:3b:ad:e9:58:37:bc:37:e4 a@A
```

Now use ssh to create a directory ~/.ssh as user b on B. (The directory may already exist, which is fine):

```
a@A:~> ssh b@B mkdir -p .ssh
b@B's password:
```

Finally append a's new public key to b@B:.ssh/authorized_keys and enter b's password one last time:

```
a@A:~> cat .ssh/id_rsa.pub | ssh b@B 'cat >> .ssh/authorized_keys'
b@B's password:
```

From now on you can log into B as b from A as a without password:

```
a@A:~> ssh b@B
```

A note from one of our readers: Depending on your version of SSH you might also have to do the following changes:

- Put the public key in .ssh/authorized_keys2
- Change the permissions of .ssh to 700
- Change the permissions of .ssh/authorized_keys2 to 640

http://www.linuxproblem.org/art_9.html