Vaskovskaya Daria

Steps in order to configure the SSH server to perform RSA based authentication.

Keys were created using the command:
`ssh-keygen -t rsa -b 4096 -C "your_email@example.com"`

I downloaded the docker image from here:
https://hub.docker.com/r/rastasheep/ubuntu-sshd/

Created a container with:
`docker run -dit -p 2345:1234 -p 1234:22 rastasheep/ubuntu-sshd`

Written a script:

```
import paramiko
import os
path = './admin' #path to the admin's public keys
another_path = './new_user' #path to a new user's public keys
host = 'localhost'
user = 'root'
secret = 'root'
port = 1234 #22nd port in the container is binded to this host's port
client = paramiko.SSHClient()
client.set_missing_host_key_policy(paramiko.AutoAddPolicy())
client.connect(hostname=host, username=user, password=secret, port=port)

files = []
stdin, stdout, stderr = client.exec_command('cd ~/.ssh/authorized_keys')
error_msg = stderr.read().decode("utf-8")

#Create folder with authorised keys
if "No such file or directory" in error_msg:
   client.exec_command('mkdir ~/.ssh')
   client.exec_command('chmod 700 ~/.ssh')
   client.exec_command('touch ~/.ssh/authorized_keys')
   client.exec_command('chmod 600 ~/.ssh/authorized_keys')

for r, d, f in os.walk(path):
   for file in f:
      if '.pub' in file:
```

```
            files.append(file)
#
for f in files:
   days_file = open(path + "/" + f, 'r')
   key = days_file.read().rstrip()
   command = f'echo "{key}">> ~/.ssh/authorized_keys'
   client.exec_command(command)



# Change config

#Turn off PAM
client.exec_command("sed -i 's/UsePAM yes"
                    "/UsePAM no/g' /etc/ssh/sshd_config")
client.exec_command("sed -i 's/#UsePAM no"
                    "/UsePAM no/g' /etc/ssh/sshd_config")
#Turn off password authentication
client.exec_command("sed -i 's/PasswordAuthentication yes"
                    "/PasswordAuthentication no/g' /etc/ssh/sshd_config")

client.exec_command("sed -i 's/#PasswordAuthentication no"
                    "/PasswordAuthentication no/g' /etc/ssh/sshd_config")

#Switch to an alternate port
client.exec_command("sed -i 's/#Port 22"
                    "/Port 1234/g' /etc/ssh/sshd_config")
#Disable ipv6
client.exec_command("sed -i 's/#ListenAddress 0.0.0.0"
                    "/#ListenAddress 0.0.0.0/g' /etc/ssh/sshd_config")
#Disable X11 forwarding
client.exec_command("sed -i 's/X11Forwarding yes"
                    "/X11Forwarding no/g' /etc/ssh/sshd_config")

#Check if strict modes is the default
client.exec_command("sed -i 's/#StrictModes yes"
                    "/StrictModes yes/g' /etc/ssh/sshd_config")
# Allow only a specific system group to use the service (e.g. root or wheel)
client.exec_command("echo 'AllowGroups root' >> /etc/ssh/sshd_config")
```

```
# Create an account for a teammate and allow him to SSH into your server as user. Show your system logs as acceptance test, once your teammate reached his account.


client.exec_command("mkdir -p /home/mynewuser/.ssh")
client.exec_command("touch /home/mynewuser/.ssh/authorized_keys")
client.exec_command("useradd -d /home/mynewuser mynewuser")
client.exec_command("gpasswd -a mynewuser su")
client.exec_command("usermod -aG root mynewuser")
client.exec_command("chown -Rfv mynewuser:mynewuser /home/mynewuser/.ssh/")
client.exec_command("chmod 700 /home/mynewuser/.ssh")
client.exec_command("chmod 600 /home/mynewuser/.ssh/authorized_keys")


files = []
for r, d, f in os.walk(another_path):
   for file in f:
      if '.pub' in file:
         files.append(file)

for f in files:
   days_file = open(another_path + "/" + f, 'r')
   key = days_file.read().rstrip()
   command = f'echo "{key}">> /home/mynewuser/.ssh/authorized_keys'
   client.exec_command(command)

client.exec_command('cat /etc/ssh/sshd_config')

client.exec_command('/etc/init.d/ssh reload')


# Bonus: restrict the service to the IPv4 subnet or IP addresses of your choice.
client.exec_command("touch /etc/hosts.deny")
client.exec_command("echo 'sshd: 116.31.116.20' /etc/hosts.deny")
# Accounts without password with `UsePAM no` cannot login
# through SSH by default
# changing '!' to '*' in /etc/shadow would probably fix it
client.exec_command('sed -i\'\' -e \'s/newuser:!/newuser:*/\' /etc/shadow')
```

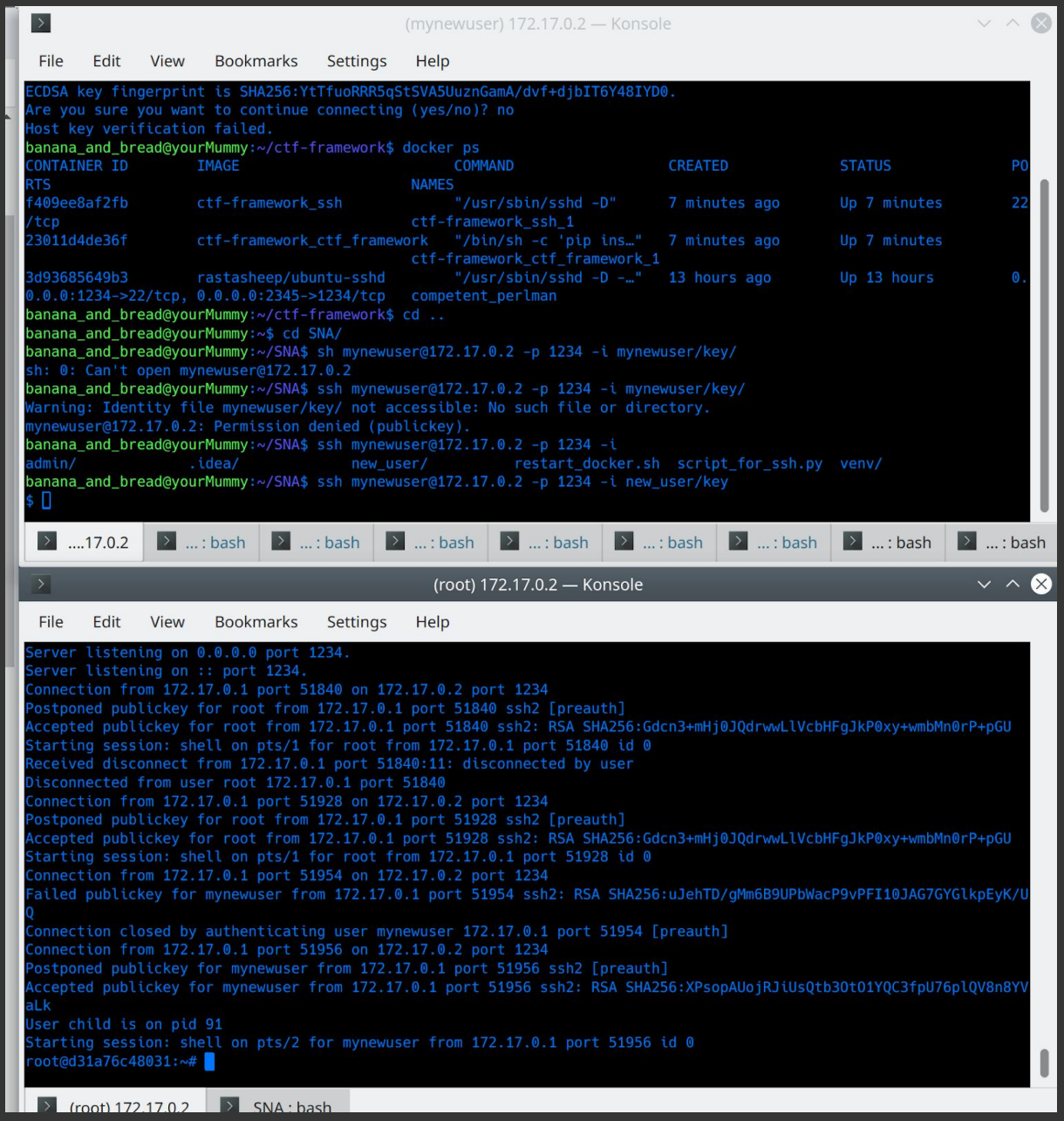
```
client.close()
```

Final config.

After launching the script the overall config looks like this:

```
Port 1234
SyslogFacility AUTHPRIV
LogLevel VERBOSE
PermitRootLogin yes
StrictModes yes
PasswordAuthentication no
ChallengeResponseAuthentication no
UsePAM no
X11Forwarding no
PrintMotd no
AcceptEnv LANG LC_*
Subsystem       sftp    /usr/lib/openssh/sftp-server
AllowGroups root
```

Proof of concept (logs)


(mynewuser) 172.17.0.2 — Konsole
File Edit View Bookmarks Settings Help
ECDSA key fingerprint is SHA256:YtTfuoRRR5qStSVA5UuznGamA/dvf+djbIT6Y48IYD0.
Are you sure you want to continue connecting (yes/no)? no
Host key verification failed.
banana_and_bread@yourMummy:~/ctf-framework$ docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PO
RTS NAMES
f409ee8af2fb ctf-framework_ssh "/usr/sbin/sshd -D" 7 minutes ago Up 7 minutes 22
/tcp ctf-framework_ssh_1
23011d4de36f ctf-framework_ctf_framework "/bin/sh -c 'pip ins…" 7 minutes ago Up 7 minutes
ctf-framework_ctf_framework_1
3d93685649b3 rastasheep/ubuntu-sshd "/usr/sbin/sshd -D -…" 13 hours ago Up 13 hours 0.
0.0.0:1234->22/tcp, 0.0.0.0:2345->1234/tcp competent_perlman
banana_and_bread@yourMummy:~/ctf-framework$ cd ..
banana_and_bread@yourMummy:~$ cd SNA/
banana_and_bread@yourMummy:~/SNA$ sh mynewuser@172.17.0.2 -p 1234 -i mynewuser/key/
sh: 0: Can't open mynewuser@172.17.0.2
banana_and_bread@yourMummy:~/SNA$ ssh mynewuser@172.17.0.2 -p 1234 -i mynewuser/key/
Warning: Identity file mynewuser/key/ not accessible: No such file or directory.
mynewuser@172.17.0.2: Permission denied (publickey).
banana_and_bread@yourMummy:~/SNA$ ssh mynewuser@172.17.0.2 -p 1234 -i
admin/ .idea/ new_user/ restart_docker.sh script_for_ssh.py venv/
banana_and_bread@yourMummy:~/SNA$ ssh mynewuser@172.17.0.2 -p 1234 -i new_user/key
$
....17.0.2
... : bash
... : bash
... : bash
... : bash
... : bash
... : bash
... : bash
(root) 172.17.0.2 — Konsole
File Edit View Bookmarks Settings Help
Server listening on 0.0.0.0 port 1234.
Server listening on :: port 1234.
Connection from 172.17.0.1 port 51840 on 172.17.0.2 port 1234
Postponed publickey for root from 172.17.0.1 port 51840 ssh2 [preauth]
Accepted publickey for root from 172.17.0.1 port 51840 ssh2: RSA SHA256:Gdcn3+mHj0JQdrwwLlVcbHFgJkP0xy+wmbMn0rP+pGU
Starting session: shell on pts/1 for root from 172.17.0.1 port 51840 id 0
Received disconnect from 172.17.0.1 port 51840:11: disconnected by user
Disconnected from user root 172.17.0.1 port 51840
Connection from 172.17.0.1 port 51928 on 172.17.0.2 port 1234
Postponed publickey for root from 172.17.0.1 port 51928 ssh2 [preauth]
Accepted publickey for root from 172.17.0.1 port 51928 ssh2: RSA SHA256:Gdcn3+mHj0JQdrwwLlVcbHFgJkP0xy+wmbMn0rP+pGU
Starting session: shell on pts/1 for root from 172.17.0.1 port 51928 id 0
Connection from 172.17.0.1 port 51954 on 172.17.0.2 port 1234
Failed publickey for mynewuser from 172.17.0.1 port 51954 ssh2: RSA SHA256:uJehTD/gMm6B9UPbWacP9vPFI10JAG7GYGlkpEyK/U
Q
Connection closed by authenticating user mynewuser 172.17.0.1 port 51954 [preauth]
Connection from 172.17.0.1 port 51956 on 172.17.0.2 port 1234
Postponed publickey for mynewuser from 172.17.0.1 port 51956 ssh2 [preauth]
Accepted publickey for mynewuser from 172.17.0.1 port 51956 ssh2: RSA SHA256:XPsopAUojRJiUsQtb3OtO1YQC3fpU76plQV8n8YV
aLk
User child is on pid 91
Starting session: shell on pts/2 for mynewuser from 172.17.0.1 port 51956 id 0
root@d31a76c48031:~#
(root) 172.17.0.2
SNA : bash