# Week 10

## Task 1. Essential Eight Mitigation Strategies

For each of the selected strategies, a description of how it is applied for your Project.

• Reasons why you selected these strategies (and not the other of the Essential Eight).

1. Essential → Application Control from ‘Prevent Malware Delivery and Execution’

   Reason : In a corporate environment, there is an undeniable need and usage of working with multiple options, there may be alternatives to applications, scammed applications etc. Point is that the applications used for specific utility also serve as great attack vectors.

   Solution strategy? Having control over execution. To control applications that are untested, unapproved and have unjustifiable execution of scripts/dll etc.

   Example & Description: Our Project is based on Protection and Detection. And having a thorough application control will aid our plan. For example, the Test network System will have minimal access in downloading and installation of applications, It will be password or protected with either mechanisms and would require Admin’s consent and notice before installation of any application.

2. Essential → Restricted Administrative Privileges from ‘Limit the Extent of Cyber Security Incidents’

   Reason : Limited Privileges has always been a great approach . We do not need high privileges to run every day tasks. This could help in reduction of possible attack scenarios. It may also ensure that even if an employee account gets compromised through thoroughly sorted Social engineering attacks, the risks will be comparatively low.

   Example & Description: Unnecessary extra privileges may lead to multiple accounts being at risk. Social Engineering attacks are nothing unheard of, a recent example being of UBER. Let’s limit the privileges or construct accounts with dual consent for operation.

3. Excellent→ Network Segmentation from ‘Limit the Extent of Cyber Security Incidents’

   Reason : A common strategy of Malwares or any attack vector to propagate in a compromised system is to check for linked networks. The wise strategy is to separate networks and restrict traffic between computers unless required. Limit access to network drives, databases and other configuration files based on user defined duties.

Example & Description: The test network will be kept and maintained separately. It will have internal firewalls or security controls to allow specific or no internetwork traffic to reduce the compromise scale.

4. Excellent➞ Continuous Incident detection and response from ‘Detect Cyber Security Incidents and Respond’

   Reason : Attackers are never off the clock, thus it is necessary to implement strategies that could keep a track of malicious activities at all the time. This calls for a continuous MONITORING and real time detection schemes/softwares or a dedicated team of professionals .

   Example & Description: To put some good examples, on our test network or test assets, SIEM tools and a dedicated SOC team shall be deployed. Objective is to provide continuous monitoring and real time detection of malicious activities.

## Task 2. Explore and Select NIST Controls

For each of the selected controls, an explanation of its relevance and description of how it is

applied for your Project.

AU-14 Session Audit : This control is crucial because it makes sure that every user action within a system is tracked and logged. Organizations can recognise suspicious behavior and spot potential security breaches by keeping track of user behaviors. This control might be put into place on a secure network by installing audit settings on all pertinent systems and making sure that logs are kept for a long enough time. An organization may, for instance, set the retention duration to 90 days and arrange their Windows servers to audit user behavior.

AT-3 Role Based Training : This control is crucial because it guarantees that users possess the knowledge and abilities required to securely carry out their given tasks. Organizations can lower the risk of human error and lessen the chance of security incidents by offering training that is specific to each user's position within the company. This control could be accomplished in a safe network by creating role-based training courses and making sure that every user receives the right instruction for their position. For instance, a company might create a security awareness training course including modules for executives, IT personnel, and non-technical staff that are based on roles.

AT-1 Policies and Procedures : This control is crucial since it guarantees that everyone using the system is aware of the organization's security policies and practices. Organizations may foster a culture of security and reduce the risk of unintentional or purposeful security breaches by clearly defining and communicating security requirements. This control could be put into place in a secure network by creating thorough security policies and procedures and making sure that all users are trained in them. In order to ensure that all users are aware of the policy, training could be given once an organization develops a password policy that mandates users set secure passwords and change them frequently.

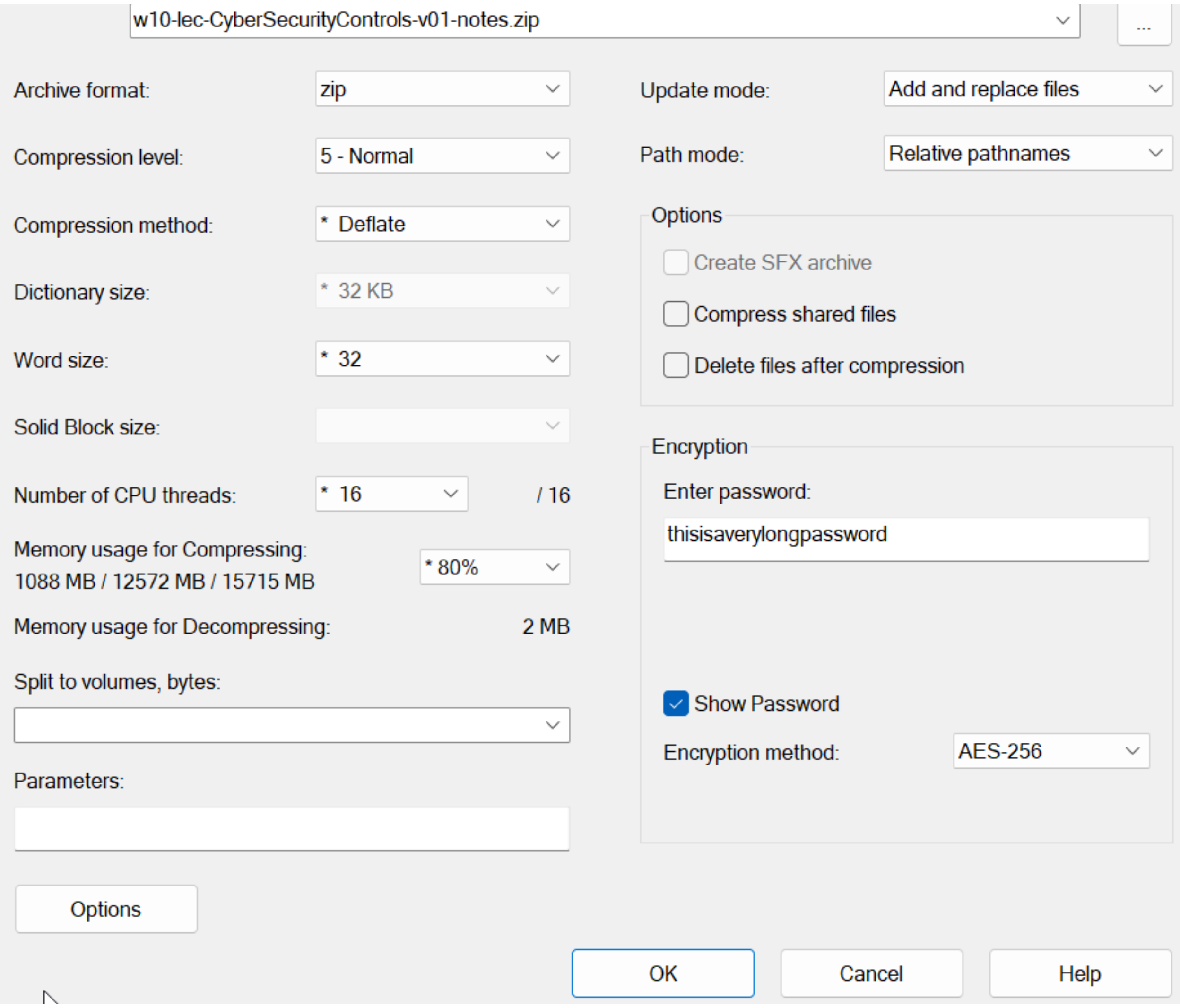
CM-7 Least Functionality : This control is crucial because it guarantees that systems are set up to grant only the minimal amount of access required for them to function as intended. Organizations can restrict the attack surface and lower the risk of unauthorized access or data loss by limiting the functionality of systems and apps. This control might be put into place in a secure network by doing routine vulnerability assessments and penetration tests to find pointless services and functionality, and then turning them off or eliminating them. For instance, a company could examine the web application's vulnerabilities and find that the FTP service is being used needlessly. The FTP service might then be turned off to lessen the attack surface.

CA -7 Continuous Monitoring : This control is crucial because it guarantees that networks and systems are continuously checked for security-related accidents and events. Organizations may notice and respond to security breaches more rapidly by continually monitoring their systems, which lowers the impact and lowers the chance of data loss or theft. This control could be applied in a secure network by utilizing security monitoring solutions that offer real-time alerts and reporting. An organization might, for instance, install an intrusion detection system (IDS) that keeps track of network traffic and notifies security personnel of any unusual behavior.

CA-8 Penetration Testing : This control is crucial because it guarantees that networks and systems are examined for flaws and vulnerabilities by mimicking actual attacks. Organizations can find security flaws and fix them before attackers take advantage of them by conducting frequent penetration tests. This control could be put into place in a safe network by conducting routine penetration tests on important networks and systems. An organization might, for instance, contract with an outside security company to conduct a penetration test on its online application in order to find vulnerabilities and offer suggestions for fixing them.

## Task 3. Encrypt a File

Screenshot of the settings used to encrypt the file.



w10-lec-CyberSecurityControls-v01-notes.zip

Archive format: zip

Compression level: 5 - Normal

Compression method: * Deflate

Dictionary size: * 32 KB

Word size: * 32

Solid Block size:

Number of CPU threads: * 16 / 16

Memory usage for Compressing: 1088 MB / 12572 MB / 15715 MB * 80%

Memory usage for Decompressing: 2 MB

Split to volumes, bytes:

Parameters:

Update mode: Add and replace files

Path mode: Relative pathnames

Options

☐ Create SFX archive

☐ Compress shared files

☐ Delete files after compression

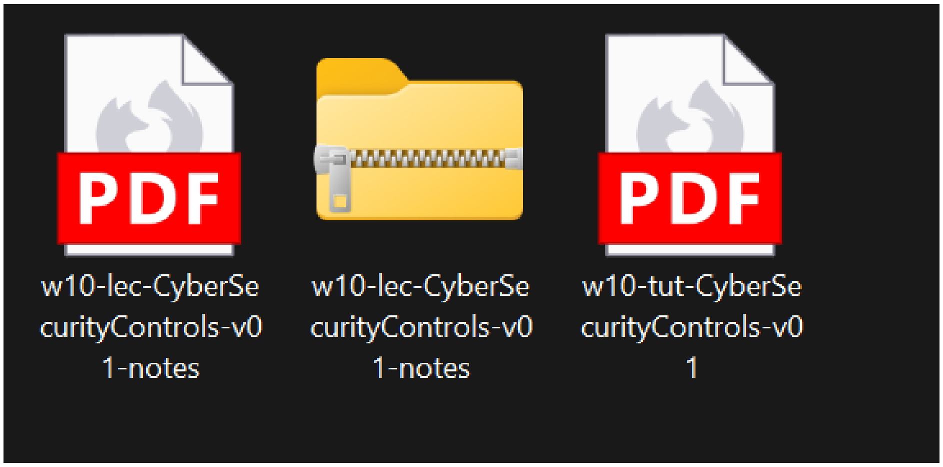
Encryption

Enter password: thisisaverylongpassword

☑ Show Password

Encryption method: AES-256

After completion, we got our zip file.


w10-lec-CyberSecurityControls-v01-notes
w10-lec-CyberSecurityControls-v01-notes
w10-tut-CyberSecurityControls-v01

• Discuss how you shared the secret key, the limitations of that approach, and recommendations for more secure ways to share a secret key.

I shared the secret pass key via an email. It could also be sent through a chat application or in Person.

Limitation to this approach and the possible solutions :

- If the transmission method is insecure (for instance, if you share the password over email), the password may be intercepted or compromised while in transit.
- The recipient could easily forget the password, in which case it would be difficult to decrypt the file.

Some other useful measures than the general ones.

Using a secure file sharing service: You can use a secure file sharing service that enables you to encrypt the file and share the password in a secure manner rather than sharing the encrypted file and password individually. OneDrive, Google Drive, and Dropbox are a few examples of these services.

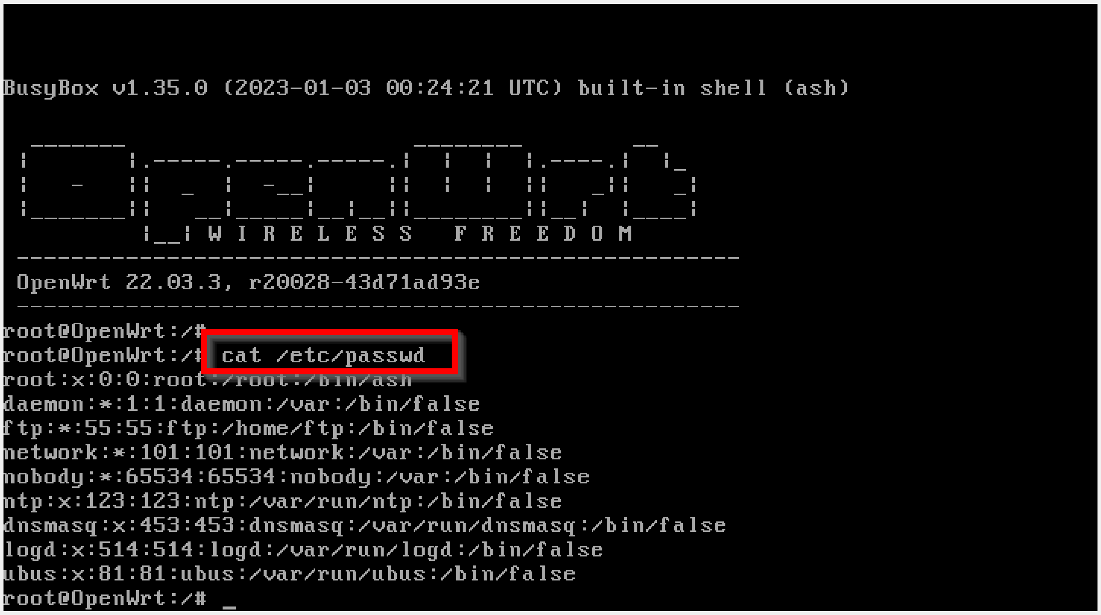
Using a secure messaging app: You can use a secure messaging app that enables end-to-end encryption, such as Signal or WhatsApp, in place of sending the password by email or text message.

Using a key management system: You can use a key management system to securely store and share the password with authorized users rather than disclosing it to them directly. LastPass and KeePass are a couple of examples of these solutions.

## Task 4. View Password Information Stored in Linux

Screenshot or copy-and-paste of the /etc/shadow file entries that show your new user and

password information.

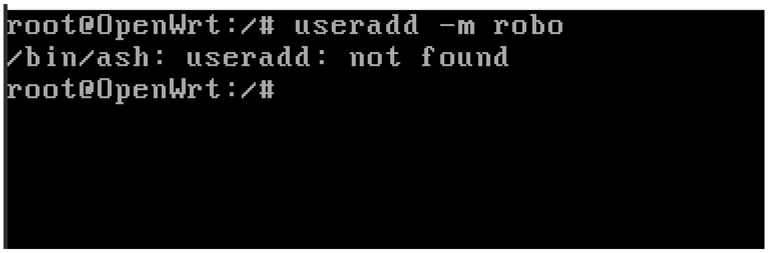


```
BusyBox v1.35.0 (2023-01-03 00:24:21 UTC) built-in shell (ash)

  _______                     ________        __
 |       |.-----.-----.-----.|  |  |  |.----.|  |_
 |   -   ||  _  |  -__|     ||  |  |  ||   _||   _|
 |_______||   __|_____|__|__||________||__|  |____|
          |__| W I R E L E S S   F R E E D O M
 -----------------------------------------------------
 OpenWrt 22.03.3, r20028-43d71ad93e
 -----------------------------------------------------
root@OpenWrt:/#
root@OpenWrt:/# cat /etc/passwd
root:x:0:0:root:/root:/bin/ash
daemon:*:1:1:daemon:/var:/bin/false
ftp:*:55:55:ftp:/home/ftp:/bin/false
network:*:101:101:network:/var:/bin/false
nobody:*:65534:65534:nobody:/var:/bin/false
ntp:x:123:123:ntp:/var/run/ntp:/bin/false
dnsmasq:x:453:453:dnsmasq:/var/run/dnsmasq:/bin/false
logd:x:514:514:logd:/var/run/logd:/bin/false
ubus:x:81:81:ubus:/var/run/ubus:/bin/false
root@OpenWrt:/# _
```

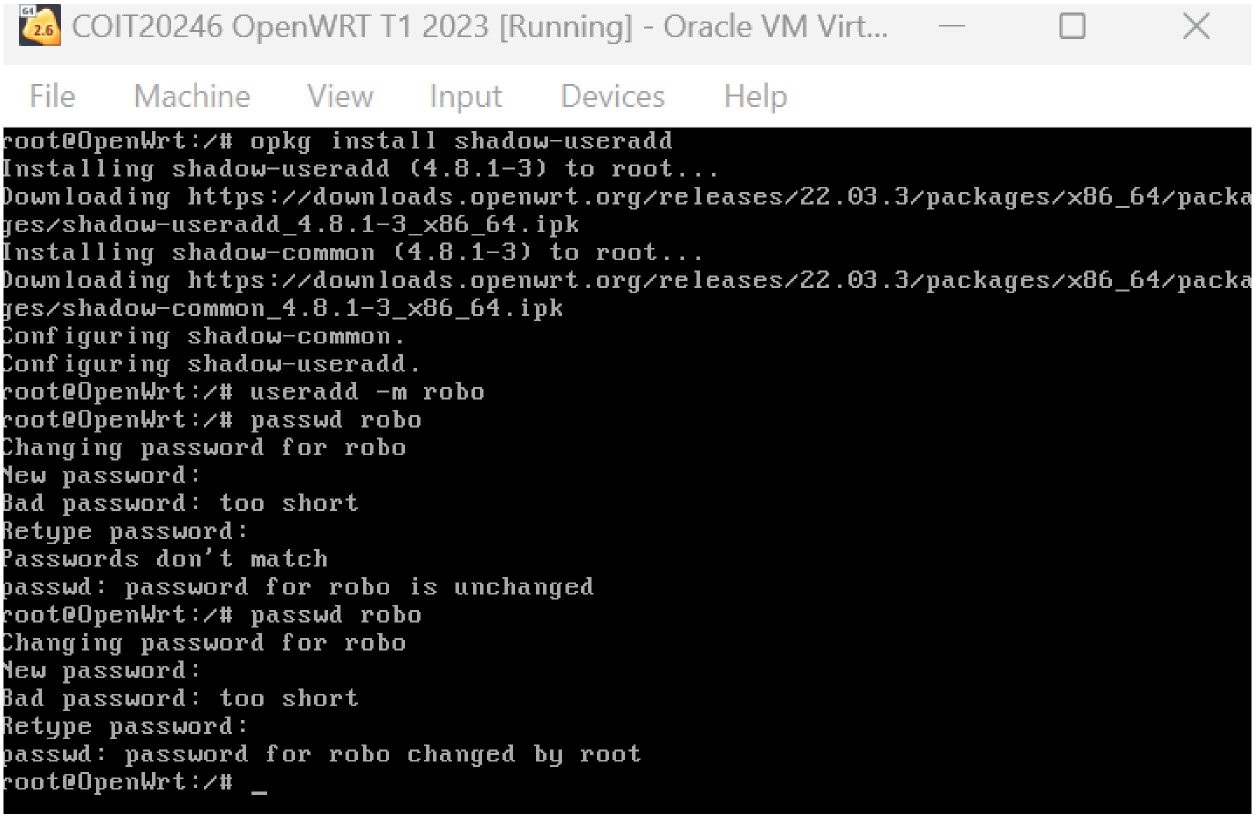
To add as new user, I tried the standard linux command **useradd -m <username>**

But I got error,



```
root@OpenWrt:/# useradd -m robo
/bin/ash: useradd: not found
root@OpenWrt:/#
```

After checking with the video tutorial, I came back and updated my system, **opkg update, → opkg install shadow-useradd**



```
root@OpenWrt:/# opkg install shadow-useradd
Installing shadow-useradd (4.8.1-3) to root...
Downloading https://downloads.openwrt.org/releases/22.03.3/packages/x86_64/packa
ges/shadow-useradd_4.8.1-3_x86_64.ipk
Installing shadow-common (4.8.1-3) to root...
Downloading https://downloads.openwrt.org/releases/22.03.3/packages/x86_64/packa
ges/shadow-common_4.8.1-3_x86_64.ipk
Configuring shadow-common.
Configuring shadow-useradd.
root@OpenWrt:/# useradd -m robo
root@OpenWrt:/# passwd robo
Changing password for robo
New password:
Bad password: too short
Retype password:
Passwords don't match
passwd: password for robo is unchanged
root@OpenWrt:/# passwd robo
Changing password for robo
New password:
Bad password: too short
Retype password:
passwd: password for robo changed by root
root@OpenWrt:/# _
```

Commands used :

opkg update

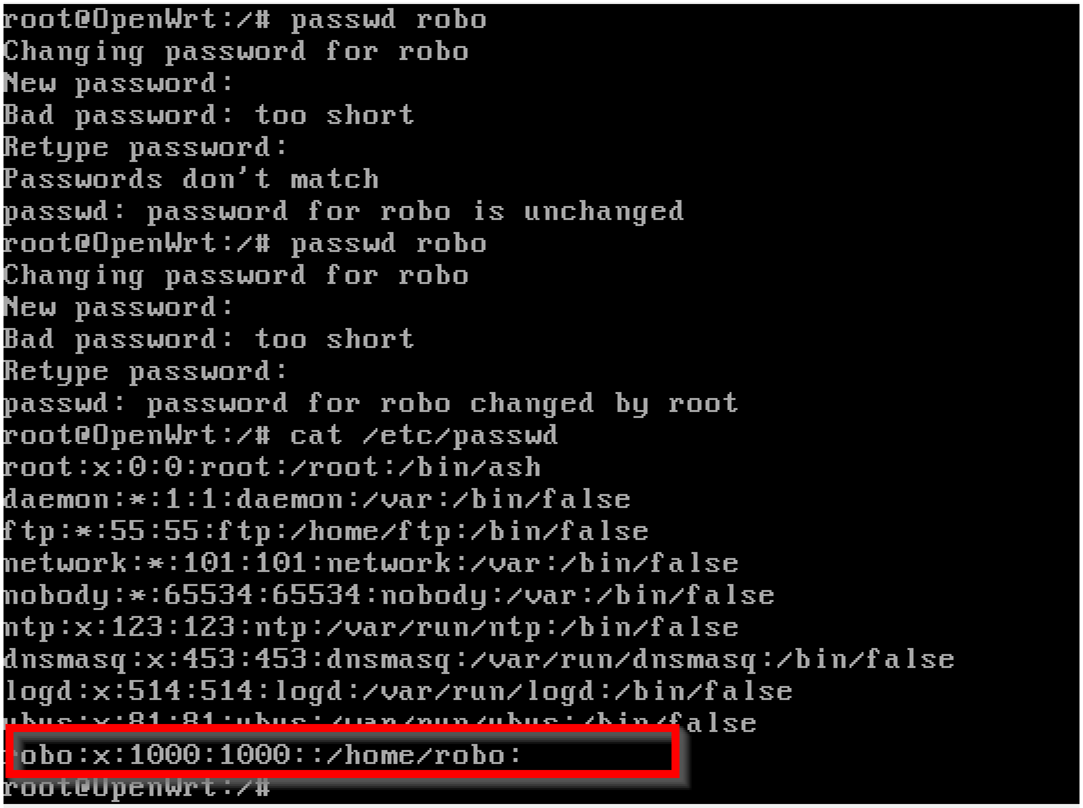
opkg install shadow-useradd

useradd -m <name_of_user>

passwd <name_of_user>

To confirm new user added, we can once again go back and check the passwd file byu command

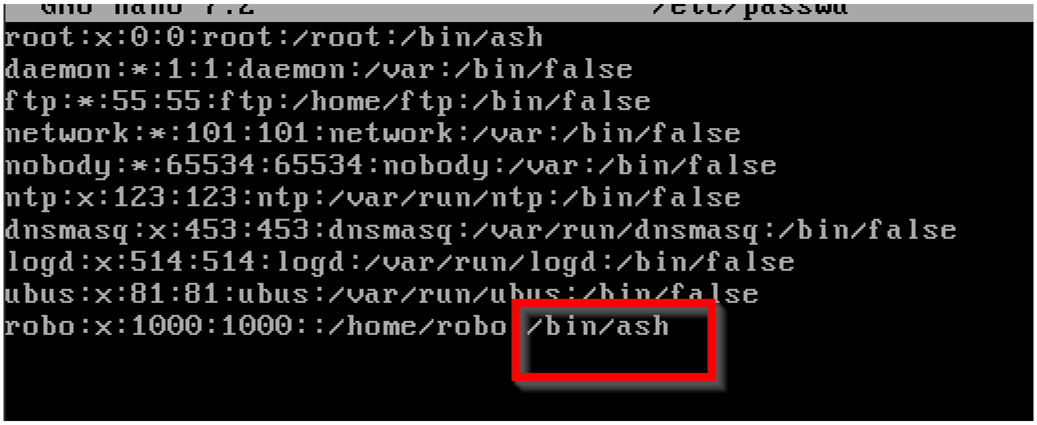
cat /etc/passwd



```
root@OpenWrt:/# passwd robo
Changing password for robo
New password:
Bad password: too short
Retype password:
Passwords don't match
passwd: password for robo is unchanged
root@OpenWrt:/# passwd robo
Changing password for robo
New password:
Bad password: too short
Retype password:
passwd: password for robo changed by root
root@OpenWrt:/# cat /etc/passwd
root:x:0:0:root:/root:/bin/ash
daemon:*:1:1:daemon:/var:/bin/false
ftp:*:55:55:ftp:/home/ftp:/bin/false
network:*:101:101:network:/var:/bin/false
nobody:*:65534:65534:nobody:/var:/bin/false
ntp:x:123:123:ntp:/var/run/ntp:/bin/false
dnsmasq:x:453:453:dnsmasq:/var/run/dnsmasq:/bin/false
logd:x:514:514:logd:/var/run/logd:/bin/false
ubus:x:81:81:ubus:/var/run/ubus:/bin/false
robo:x:1000:1000::/home/robo:
root@OpenWrt:/#
```

We can see details of the new user **robo** now.

We will do a couple of more things, we will edit our /etc/passwd file to allow putty login. We can do this by adding **/bin/ash** at the end of the new line created for the new user. Save and exit the nano editor.



```
  GNU nano 7.2                          /etc/passwd
root:x:0:0:root:/root:/bin/ash
daemon:*:1:1:daemon:/var:/bin/false
ftp:*:55:55:ftp:/home/ftp:/bin/false
network:*:101:101:network:/var:/bin/false
nobody:*:65534:65534:nobody:/var:/bin/false
ntp:x:123:123:ntp:/var/run/ntp:/bin/false
dnsmasq:x:453:453:dnsmasq:/var/run/dnsmasq:/bin/false
logd:x:514:514:logd:/var/run/logd:/bin/false
ubus:x:81:81:ubus:/var/run/ubus:/bin/false
robo:x:1000:1000::/home/robo:/bin/ash
```

Explanation of the password information stored in /etc/shadow, and why the actual password is not stored.

Only ‘Root’ user account or the one’s listed in the Sudoers List can view/do changes to the etc/passwd file.

The multiple field present in the /etc/passwd file are

Username: The name of the user.

Password: The encrypted password of the user.

Last password change: The number of days since the password was last changed.

Minimum password age: The number of days that must pass before the password can be changed again.

Maximum password age: The maximum number of days that a password can be used before it must be changed.

Warning period: The number of days before the password expires that the user is warned.

Inactivity period: The number of days after the password expires that the account is disabled.

Expiration date: The date on which the password will expire.

Reserved field: This field is not currently used.

The reason why the actual password is not stored in the /etc/shadow file is for security reasons. Storing the actual password in plain text would make it much easier for an attacker to gain unauthorized access to the system, as they could simply read the password from the /etc/shadow file. However, by storing only the encrypted hash, an attacker would need to crack the encryption to discover the password. This is a much more difficult and time-consuming process, particularly if the password is strong and complex.

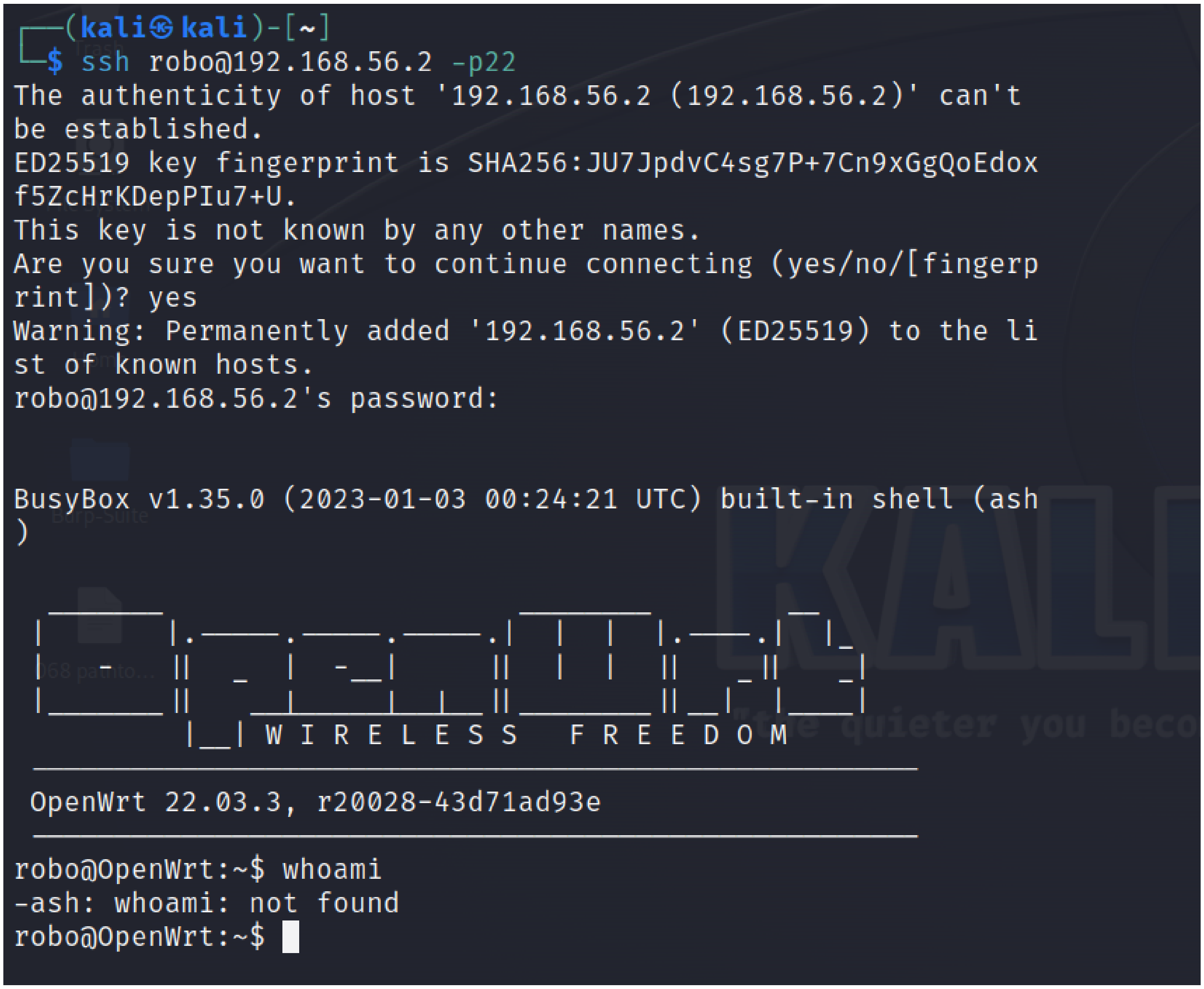
## Task 5. Setup Key-Based Authentication

Choosing task (a) Key-based SSH login for OpenWRT Linux VM using PuTTy

Screenshots or copy-and-paste of the steps/commands you used.

After creating the newuser, I quickly wanted to test out the ssh login so I just tried simple ssh logging into the OpenWRT linux VM , on IP 192.168.56.2.

Command used ssh <username>@<ip_address> -p<port_number> [port number is 22 for ssh by default].



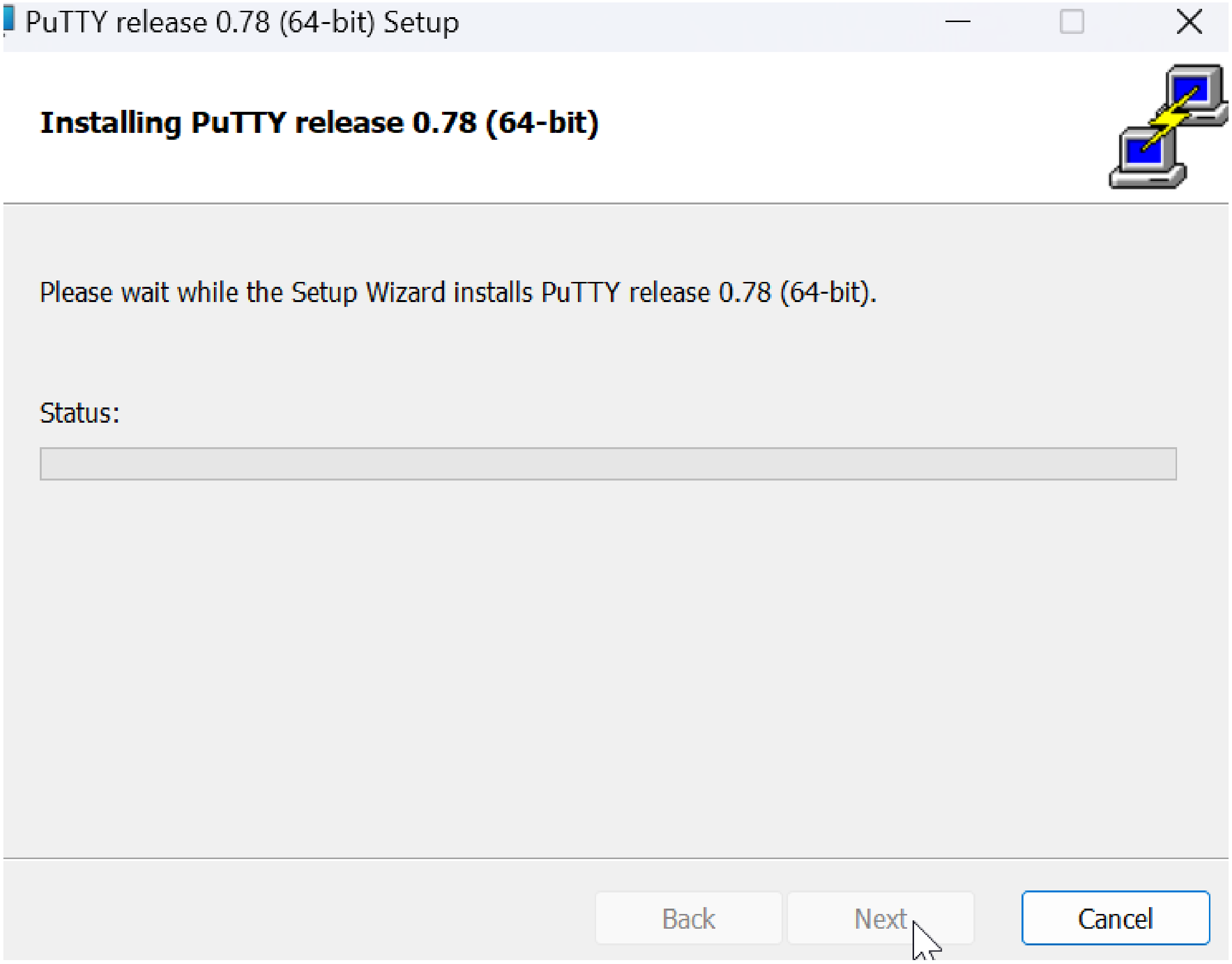
```
┌──(kali㉿kali)-[~]
└─$ ssh robo@192.168.56.2 -p22
The authenticity of host '192.168.56.2 (192.168.56.2)' can't
be established.
ED25519 key fingerprint is SHA256:JU7JpdvC4sg7P+7Cn9xGgQoEdox
f5ZcHrKDepPIu7+U.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerp
rint])? yes
Warning: Permanently added '192.168.56.2' (ED25519) to the li
st of known hosts.
robo@192.168.56.2's password:


BusyBox v1.35.0 (2023-01-03 00:24:21 UTC) built-in shell (ash
)

  _______                     ________        __
 |       |.-----.-----.-----.|  |  |  |.----.|  |_
 |   -   ||  _  |  -__|     ||  |  |  ||   _||   _|
 |_______||   __|_____|__|__||________||__|  |____|
          |__| W I R E L E S S   F R E E D O M
 -----------------------------------------------------
 OpenWrt 22.03.3, r20028-43d71ad93e
 -----------------------------------------------------
robo@OpenWrt:~$ whoami
-ash: whoami: not found
robo@OpenWrt:~$ 
```

SSH login using PuTTy:

Installation :

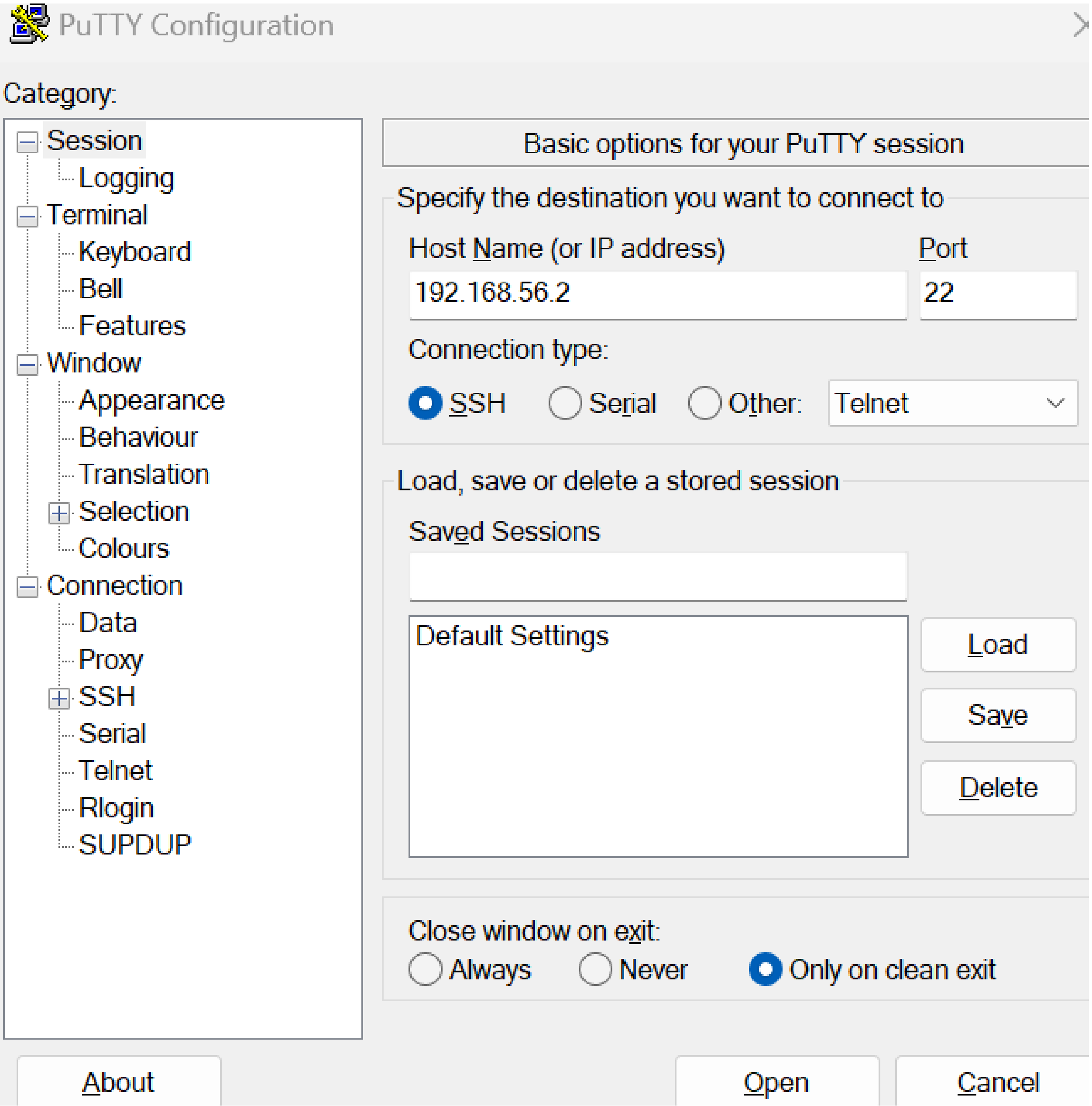


PuTTY release 0.78 (64-bit) Setup

Installing PuTTY release 0.78 (64-bit)

Please wait while the Setup Wizard installs PuTTY release 0.78 (64-bit).

Status:



Specify the destination you want to connect to

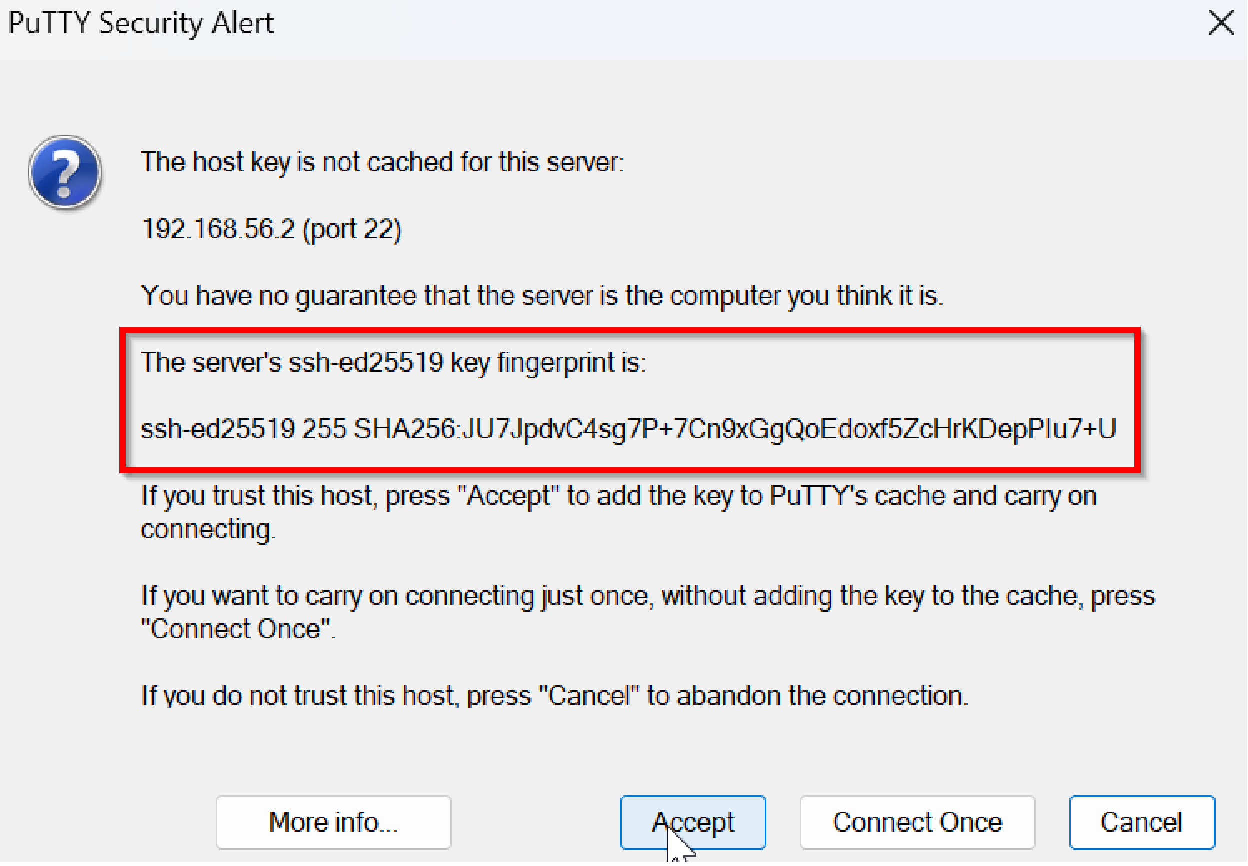
Host Name (or IP address): 192.168.56.2

Port: 22

Connection type: ● SSH ○ Serial ○ Other: Telnet

Close window on exit: ○ Always ○ Never ● Only on clean exit

Let’s click on open the ssh connection, and read and accept the alert generated by the PuTTY client. Here OpenWRT is serving as the server.


PuTTY Security Alert

The host key is not cached for this server:

192.168.56.2 (port 22)

You have no guarantee that the server is the computer you think it is.

The server's ssh-ed25519 key fingerprint is:

ssh-ed25519 255 SHA256:JU7JpdvC4sg7P+7Cn9xGgQoEdoxf5ZcHrKDepPlu7+U

If you trust this host, press "Accept" to add the key to PuTTY's cache and carry on connecting.

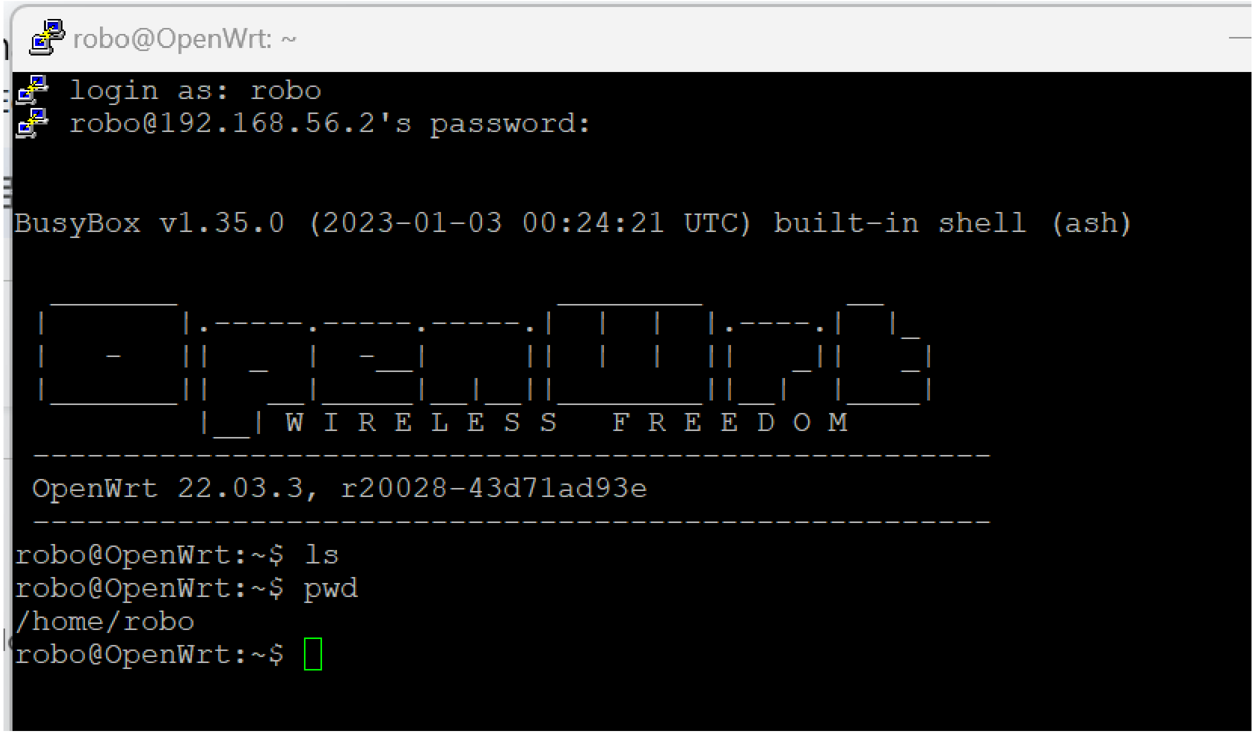
If you want to carry on connecting just once, without adding the key to the cache, press "Connect Once".

If you do not trust this host, press "Cancel" to abandon the connection.

After accepting the connection, we are prompted with a login screen.

Here enter the credentials for the user you want to login.

robo is the user in our case.



```
login as: robo
robo@192.168.56.2's password:


BusyBox v1.35.0 (2023-01-03 00:24:21 UTC) built-in shell (ash)

  _______                     ________        __
 |       |.-----.-----.-----.|  |  |  |.----.|  |_
 |   -   ||  _  |  -__|     ||  |  |  ||   _||   _|
 |_______||   __|_____|__|__||________||__|  |____|
          |__| W I R E L E S S   F R E E D O M
 -----------------------------------------------------
 OpenWrt 22.03.3, r20028-43d71ad93e
 -----------------------------------------------------
robo@OpenWrt:~$ ls
robo@OpenWrt:~$ pwd
/home/robo
robo@OpenWrt:~$
```

- **Explain why key-based authentication can be more secure than password-based authentication when connecting to a SSH server (e.g. on OpenWRT, GitHub or Azure).**

Key-based authentication method is more secure than password based authentication in situations of connecting to an SSH server because:

1. It has a stronger and even Complex password.
2. No concept of Password Sharing or reuse.
3. They are tough against brute force attacks.
4. Have Enhanced Security features like passphrase protection and key revocation.