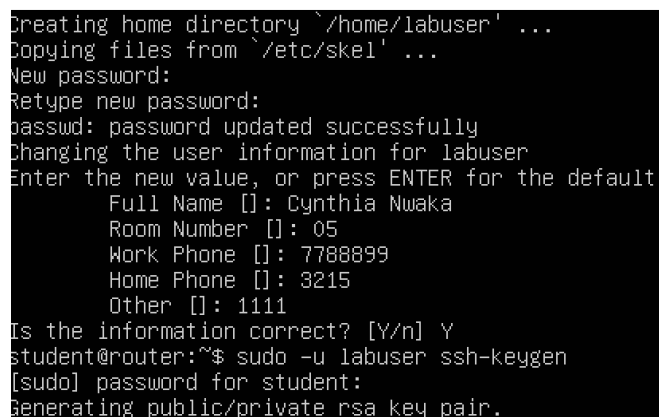


## Identity and Credential Management in Ubuntu 22.04 Activities

Identity, Credential and Access Management refers to the processes and technologies used to manage and secure digital identities and associated credentials within an organization or system. In the context of information technology and cybersecurity, ICAM plays a crucial role in ensuring that the right individuals have the appropriate access to resources and systems while maintaining security and compliance.

### 1. Created account and set the password.



```
Creating home directory '/home/labuser' ...
Copying files from '/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for labuser
Enter the new value, or press ENTER for the default
  Full Name []: Cynthia Nwaka
  Room Number []: 05
  Work Phone []: 7788899
  Home Phone []: 3215
  Other []: 1111
Is the information correct? [Y/n] Y
student@router:~$ sudo -u labuser ssh-keygen
[sudo] password for student:
Generating public/private rsa key pair.
```

*sudo adduser labuser:* This command creates a new user account named "labuser" using the adduser command, which is used to add and manage user accounts in Linux systems. The sudo prefix indicates that the command requires administrative privileges, so one needs to enter the password for your current user account when prompted.

*sudo passwd labuser:* This command sets a password for the newly created "labuser" account. You'll be prompted to enter the password twice to confirm it.

### 2. Generate an SSH key pair for labuser, the key generated is in the picture below.

What is SSH KEY PAIR? This is a pair of cryptographic keys used for secure communication between two parties, typically a client and a server, it is used for securing remote access to systems and secure file transfers.

```
student@router:~$ sudo -u labuser ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/labuser/.ssh/id_rsa):
Created directory '/home/labuser/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/labuser/.ssh/id_rsa
Your public key has been saved in /home/labuser/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:SN0EBIJR06RUAJTNA6YEzvY3tJiXWqmTr+0CRmq7yo labuser@router
The key's randomart image is:
+---[RSA 3072]-----+
|00*X*+=+00..|
|+00.++. . 0|
|.+. .0. . .|
|...+.+.|
|0 + 0. S|
|. + B .|
|+ .|=|
|E .+|
|=+0.++|
+-----[SHA256]-----+
student@router:~$
```

- *sudo -u labuser ssh-keygen*: This command generates an SSH key pair for the "labuser" user. *-u* option specifies that the command should be executed as the "labuser" user. The *ssh-keygen* command prompts the user for a passphrase to protect the private key.
- *sudo cp /home/labuser/.ssh/id\_rsa.pub /home/labuser/.ssh/authorised\_keys*: This command copies the public SSH key file (*id\_rsa.pub*) from the "labuser" user's home directory to the *authorised\_keys* file, which is used to specify allowed SSH keys for the user.
- *sudo chown labuser:labuser /home/labuser/.ssh/authorised\_keys*: This command sets the ownership of the *authorised\_keys* file to the "labuser" user and group.
- *sudo chmod 600 /home/labuser/.ssh/authorised\_keys*: This command sets the permissions of the *authorised\_keys* file to 600, which means that only the owner (labuser) can read and write the file.

### 3. Password Policy Setup:

Install the necessary tools needed.

```

Unpacking libcrack2:amd64 (2.9.6-3.4build4) ...
Selecting previously unselected package cracklib-runtime.
Preparing to unpack .../1-cracklib-runtime_2.9.6-3.4build4_amd64.deb ...
Unpacking cracklib-runtime (2.9.6-3.4build4) ...
Selecting previously unselected package libpwquality-common.
Preparing to unpack .../2-libpwquality-common_1.4.4-1build2_all.deb ...
Unpacking libpwquality-common (1.4.4-1build2) ...
Selecting previously unselected package libpwquality1:amd64.
Preparing to unpack .../3-libpwquality1_1.4.4-1build2_amd64.deb ...
Unpacking libpwquality1:amd64 (1.4.4-1build2) ...
Selecting previously unselected package libpam-pwquality:amd64.
Preparing to unpack .../4-libpam-pwquality_1.4.4-1build2_amd64.deb ...
Unpacking libpam-pwquality:amd64 (1.4.4-1build2) ...
Selecting previously unselected package wamerican.
Preparing to unpack .../5-wamerican_2020.12.07-2_all.deb ...
Unpacking wamerican (2020.12.07-2) ...
Setting up libpwquality-common (1.4.4-1build2) ...
Setting up wamerican (2020.12.07-2) ...
Setting up libcrack2:amd64 (2.9.6-3.4build4) ...
Setting up cracklib-runtime (2.9.6-3.4build4) ...
Setting up libpwquality1:amd64 (1.4.4-1build2) ...
Setting up libpam-pwquality:amd64 (1.4.4-1build2) ...
Processing triggers for libc-bin (2.35-0ubuntu3.1) ...
Processing triggers for man-db (2.10.2-1) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
student@router:/home/labuser$ _

```

The provided script effectively installs the necessary tools for password policy enforcement, edits the relevant configuration files, and sets the minimum password length to 8 characters.

***sudo apt-get install libpam-pwquality*** command installs the libpam-pwquality package, which provides the necessary libraries for enforcing password quality rules. The sudo prefix indicates that the command requires administrative privileges. Also, ***sudo nano /etc/security/pwquality.conf*** opens the password policy configuration file (/etc/security/pwquality.conf) using the Nano text editor.

***sudo nano /etc/pam.d/common-password*** command opens the password policy enforcement file (/etc/pam.d/common-password) using the Nano text editor. The sudo prefix is necessary to modify system-wide configuration files.

```
GNU nano 6.2 /etc/security/pwquality.conf
# Configuration for systemwide password quality limits
# Defaults:
#
# Number of characters in the new password that must not be present in the
# old password.
# difok = 1
#
# Minimum acceptable size for the new password (plus one if
# credits are not disabled which is the default). (See pam_cracklib manual.)
# Cannot be set to lower value than 6.
# minlen = 8
#
# The maximum credit for having digits in the new password. If less than 0
# it is the minimum number of digits in the new password.
# dcredit = 0
#
# The maximum credit for having uppercase characters in the new password.
# If less than 0 it is the minimum number of uppercase characters in the new
# password.
# ucredit = 0
#
# The maximum credit for having lowercase characters in the new password.
# If less than 0 it is the minimum number of lowercase characters in the new
# password.
# lcredit = 0
#
# The maximum credit for having other characters in the new password.
# If less than 0 it is the minimum number of other characters in the new
# password.
# ocredit = 0
#
# The minimum number of required classes of characters for the new
# password (digits, uppercase, lowercase, others).
```

**Add the following lines to pwquality.conf:** suggests adding the following lines to the pwquality.conf file: *minclass = 3* (setting the minimum password class to 3)  
*minlen = 8*(sets the minimum password length to 8 characters.

```
minlen = 8
The maximum cred
it is the minimu
dcredit = 0
The maximum cred
If less than 0 i
password.
ucredit = 0
The maximum cred
If less than 0 i
password.
lcredit = 0
The maximum cred
If less than 0 i
password.
ocredit = 0
The minimum numb
password (digits
minclass = 3
```

# Edit the common-password file to enforce the password policy and # Add the following line to common-password: The instruction suggests adding the following line to the common-password file.

*password requisite pam\_pwquality.so retry=3*: This line enables the pam\_pwquality module for password enforcement and sets the retry limit to 3 meaning that users will have three attempts to enter a password that meets the password policy requirements.

```
# here are the per-package modules (the "Primary" block)
password      requisite                    pam_pwquality.so retry=3
password      [success=1 default=ignore]   pam_unix.so obscure use_authok try_first_pass yesc
# here's the fallback if no module succeeds
password      requisite                    pam_deny.so
# prime the stack with a positive return value if there isn't one already;
# this avoids us returning an error just because nothing sets a success code
# since the modules above will each just jump around
password      required                     pam_permit.so
# and here are more per-package modules (the "Additional" block)
```

#### 4. User Identity Verification:

# Verify the identity of labuser

Whoami (displays the username of the currently logged-in user. In this case, since the script is being executed as the "student " user, the output will be "student".)

# Display information about the currently logged-in users

w

```
tudent@router:~$ whoami
tudent
tudent@router:~$ w
17:38:33 up 1:20, 1 user, load average: 0.04, 0.03, 0.00
SER   TTY      FROM          LOGIN@   IDLE   JCPU   PCPU   WHAT
tudent tty1    -             16:19    1.00s  0.24s  0.00s  w
tudent@router:~$ _
18:06:40 up 56 min, 2 users, load average: 0.03, 0.04, 0.01
USER   TTY      FROM          LOGIN@   IDLE   JCPU   PCPU   WHAT
student tty1    -             17:12    12:48  0.44s  0.12s  -bash
student pts/0    -             17:53    0.00s  0.05s  0.00s  w
student@router:/home/labuser$ sudo passwd -l labuser
passwd: password expiry information changed.
student@router:/home/labuser$ su - labuser
```

#### 5. User Credential Revocation:

# Temporarily disable the account of labuser

*sudo passwd -l labuser* (This command uses the passwd command with the -l flag to lock the "labuser" account, therefore effectively disables the account, preventing the user from logging in.)

# Attempt to log in as labuser (verify access is denied)

*su - labuser*

( the command switches to the "labuser" account using the su command. Since the account

is now disabled, the attempt to log in as "labuser" should fail.)

```
student@router:~$ sudo passwd -l labuser
passwd: password expiry information changed.
student@router:~$ su -labuser
su: invalid option -- 'a'
Try 'su --help' for more information.
student@router:~$ su - labuser
Password:
```

## 6. Process Identity and Credentials:

# Identify processes running under the labuser account

```
student@router:/home/labuser$ ps aux | grep labuser
root      1121  0.0  0.2 11660 5608 tty1    S+   17:53   0:00 sudo su - labuser
root      1122  0.0  0.0 11660  936 pts/0    Ss   17:53   0:00 sudo su - labuser
root      1123  0.0  0.2 10216 4496 pts/0    S    17:53   0:00 su - labuser
labuser   1124  0.0  0.2  8736 5424 pts/0    S    17:53   0:00 -bash
student   1382  0.0  0.1  6476 2300 pts/0    S+   18:11   0:00 grep --color=auto labuser
```

*ps aux | grep labuser* (This command utilizes the *ps* command to list all running processes and filters the output using the *grep* command to only display processes associated with the "labuser" account. This provides an overview of all active processes owned by the "labuser" user.)

# Review process credentials

*ps -eo user,group,cmd | grep labuser* (This command utilizes the *ps* command to display detailed information about running processes, focusing on the user, group, and command (cmd) columns. It also filters the output using the *grep* command to specifically display processes associated with the "labuser" account. This provides a more granular view of the "labuser" user's process activity.)

```
student@router:~$ ps aux | grep labuser
student   1390  0.0  0.1  6476 2196 tty1    S+   17:49   0:00 grep --color=auto labuser
student@router:~$ ps -eo user,group,cmd | grep labuser
student student grep --color=auto labuser
student@router:~$ _
```

## 7. Audit Logging:

Audit logging is a process of recording and tracking events that occur within a computer system or application. These logs provide a detailed record of user activities, system changes, and security events, enabling organizations to monitor and analyze their systems for potential security breaches, compliance violations, and operational issues.

# Install auditd *sudo apt-get install auditd*(command installs the auditd package, which is responsible for generating and managing audit logs in Linux systems.)

```

    audispd-plugins
The following NEW packages will be installed:
    auditd libauparse0
0 upgraded, 2 newly installed, 0 to remove and 5 not upgraded.
Need to get 270 kB of archives.
After this operation, 876 kB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://gb.archive.ubuntu.com/ubuntu jammy/main amd64 libauparse0 amd64 1:3.0.7-1build1 [58.0 kB]
Get:2 http://gb.archive.ubuntu.com/ubuntu jammy/main amd64 auditd amd64 1:3.0.7-1build1 [212 kB]
Fetched 270 kB in 0s (8,446 kB/s)
Selecting previously unselected package libauparse0:amd64.
(Reading database ... 109538 files and directories currently installed.)
Preparing to unpack .../libauparse0_1%3a3.0.7-1build1_amd64.deb ...
Unpacking libauparse0:amd64 (1:3.0.7-1build1) ...
Selecting previously unselected package auditd.
Preparing to unpack .../auditd_1%3a3.0.7-1build1_amd64.deb ...
Unpacking auditd (1:3.0.7-1build1) ...
Setting up libauparse0:amd64 (1:3.0.7-1build1) ...
Setting up auditd (1:3.0.7-1build1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/auditd.service → /lib/systemd/system/auditd.service.
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for libc-bin (2.35-0ubuntu3.1) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

```

# Enable audit logging for user-related events

`sudo nano /etc/audit/audit.rules`(command opens the auditd rules file using the Nano text editor)

# Add the following line to audit.rules

`-w /etc/passwd -p wa -k identity_changes` (line is added to the file. This line instructs auditd to log write (w) actions to the /etc/passwd file, with all attributes (a), and to associate the events with the identity\_changes key.)

```

-w /etc/passwd -p wa -k identity_changes
-D
-b 8192
-f 1
--backlog_wait_time 60000

```

# Restart the auditd service

`sudo systemctl restart auditd` (restarts the auditd service to apply the new configuration and start generating audit logs.)

```

student@router:~$ sudo systemctl restart auditd
student@router:~$ sudo systemctl restart auditd
student@router:~$ sudo su - labuser
labuser@router:~$ sudo passwd labuser
[sudo] password for labuser:

```

# Perform actions and review audit logs

`sudo su - labuser`

`sudo passwd labuser`

# Change password

The `sudo su - labuser` command switches to the "labuser" account. The `sudo passwd labuser` command changes the "labuser" password. The `sudo passwd -l labuser` command locks the "labuser" account. These actions will trigger audit log entries due to the configured rules. Review the audit logs: The `sudo ausearch -k identity_changes` command searches the audit logs for events associated with the `identity_changes` key. This will display the audit log entries generated by the actions performed earlier.

#### PROBLEMS ENCOUNTERED:

**Initially, when I got to this point, I was not able to change the password for Labuser as it kept giving the error output, I tried several times by starting from the beginning but it didn't work out.**

I had to give privileges to the super user using `sudo -i`: (used to perform administrative tasks that require root privileges. For example, you might use it to install software, configure system settings, or troubleshoot problems.)

After granting this privilege, I used the command `usermod -aG sudo labuser` to give the labuser the ability to execute commands with the privileges of the superuser (root) using the `sudo` command and was now able to change the password successfully and it worked!!!

```
Last login: Mon Nov 27 19:29:58 UTC 2023 on tty1
student@router:~$ sudo -i
[sudo] password for student:
root@router:~# usermod -aG sudo labuser
root@router:~# sudo su - student
student@router:~$ sudo passwd labuser
[sudo] password for student:
New password:
Retype new password:
passwd: password updated successfully
student@router:~$ sudo su - labuser
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

labuser@router:~$ sudo su - labuser
[sudo] password for labuser:
labuser@router:~$ sudo passwd labuser
[sudo] password for labuser:
New password:
Retype new password:
passwd: password updated successfully
labuser@router:~$
```

#### WHAT I WOULD HAVE DONE BETTER?

When I changed the password for labuser I would have granted root privileges to him so as to be able to use the password.

This can be useful for delegating administrative tasks to specific users without giving them full root access. However, it's important to use `sudo` responsibly and only when necessary, as it grants a significant level of power to the user as attackers can pose as labuser and infiltrate the system.



## 8. Cleanup:

# Re-enable the account of labuser

`sudo passwd -u labuser` (This command unlocks the "labuser" account using the `passwd` command with the `-u` flag. This allows the "labuser" user to log in again.)

# Remove the SSH key pair for labuser

`sudo rm -r /home/labuser/.ssh` (This command recursively removes the entire `~/.ssh` directory from the "labuser" user's home directory. This effectively deletes the "labuser" SSH key pair, preventing the user from logging in using SSH key-based authentication.)

# Disable audit logging

`sudo systemctl stop auditd` (This command stops the `auditd` service using the `systemctl` command. This disables audit logging, preventing further audit log entries from being generated.)

```
labuser@router:~$ sudo passwd -l labuser
passwd: password expiry information changed.
labuser@router:~$ sudo ausearch -k identity_changes
<no matches>
labuser@router:~$ sudo passwd -u labuser
passwd: password expiry information changed.
labuser@router:~$ sudo rm -r /home/labuser/.ssh
labuser@router:~$ sudo systemctl stop auditd
labuser@router:~$ _
```

REFLECTION: IMPORTANCE OF THESE PRACTICES IN ENHANCING SECURITY AND MANAGING IDENTITIES AND CREDENTIALS EFFECTIVELY

- **Enforcing Strong Password Policies:** Strong passwords should be complex, unpredictable, and unique to each account. Additionally, mandating regular password changes further enhances security by minimizing the potential for password compromise. This is crucial to avoid unauthorized access attempts.
- **Implementing SSH Key-Based Authentication:** Replacing traditional password authentication with SSH key-based authentication provides a more secure alternative. It eliminates password interception risks providing a more secure login method.
- **Leveraging Two-Factor Authentication (2FA):** Employing 2FA adds an extra layer of security by requiring additional verification beyond just the password.
- **Account Disabling:** Disabling unused or compromised accounts reduces the attack surface and mitigates potential security breaches.

- **Enabling Comprehensive Audit Logging:** This provides a detailed record of user activities and system

### ***Importance of Security Practices in Real-World Scenarios***

**Strong Password Management:** In today's interconnected world, strong password management is paramount. Weak passwords are easily compromised, leading to unauthorized access and data breaches. Eg. In healthcare organizations where all employees may have access to patient medical records, Enforcing strong password policies and implementing two-factor authentication (2FA) are essential measures to protect against unauthorized access. Also in financial institutions, employees have access to customers' data, and with strict password policies, there is protection against unauthorized access and fraud.

**Secure SSH Key Authentication:** SSH key-based authentication provides a more secure alternative to traditional password authentication, especially in real-world scenarios where remote access is common. SSH keys eliminate the risk of password interception or phishing attacks, which are prevalent in online environments. For example, A managed IT services provider employs SSH key-based authentication for all technicians accessing customer networks to perform maintenance and troubleshooting.

**Account Disablement:** Promptly disabling unused or compromised accounts is a critical practice in real-world scenarios to reduce the attack surface and minimize the potential damage of a security breach. Inactive accounts present potential entry points for unauthorized individuals, and disabling them helps safeguard sensitive information.

**Comprehensive Audit Logging:** Audit logging provides valuable insights into user activities and system events, aiding in security incident detection and investigation. For example, A government agency maintains comprehensive audit logs for all access to sensitive government data to ensure compliance with regulatory requirements and track user activities related to critical information.

**Principle of Least Privilege:** Adhering to the principle of least privilege ensures that users only have access to the resources and permissions necessary for their designated roles.

For example, A university restricts access to student financial aid records to authorized financial aid officers, preventing unauthorized access to sensitive financial data.

## REFERENCES

1. Canadian Centre for Cyber Security "Identity, Credential, and Access Management (ICAM) - ITSAP.30.018."Government Of Canada. Accessed: Nov 26, 2023. [Online]. Available:<https://www.cyber.gc.ca/en/identity-credential-and-access-management-icam-itsap30018>
2. M. Harris, "The Importance of Identity and Access Management in Today's Digital World,"EM360, <https://em360tech.com/tech-article/importance-identity-and-access-management-todays-digital-world> (accessed Nov 26, 2023).
3. Manage Engine."Privileged access management (PAM) for Linux and Unix. Accessed: Nov 28, 2023. [Online]. Available:<https://www.manageengine.com/privileged-access-management/unix-linux-privilege-management-pam.html>