# COMPUTER SYSTEMS & SECURITY

## A Comparative Analysis of User and Access Management

Sergiusz Pieszak

# Contents

# 1. Introduction

I've chosen Windows 10 and Ubuntu as two distinct operating systems due to their widespread popularity. Windows 10 runs on the Windows NT kernel, a core component shared by various Windows versions. Ubuntu is based on the Linux kernel, derived from Debian, prioritizing user-friendliness and community support while maintaining compatibility with Debian packages. Both systems cater to various user preferences, with Windows 10 providing compatibility across hardware architectures and a user-friendly interface. User and access management are essential aspects of operating systems, ensuring security, privacy, and system integrity. They provide mechanisms for establishing and managing user accounts, controlling resource access, and enforcing security policies. By implementing authentication, authorization, and auditing procedures, operating systems regulate user actions, preventing unauthorized access and system misuse. Robust user and access management are essential for maintaining data confidentiality, integrity, and availability, as well as compliance with security regulations and regulations.

# 2. User Creation, Management, and Decommissioning

The primary distinction between the operating systems is their approach to using the terminal. Ubuntu has a more permissive attitude towards terminal usage, a quality particularly advantageous for reducing the amount of PC resources. This flexibility renders Ubuntu a preferred choice for server environments. In contrast, Windows 10 also supports the command prompt; however, it primarily relies on the graphical user interface (GUI) as the main method of interaction. Karan, R. (2024).

## 2.1 Windows 10

### 2.1.1 Creating account

1. Press `Windows Key + R`, type `lusrmgr.msc`, and press Enter to open the Local Users and Groups management console.

2. Expand the `Users` folder in the left pane and right-click an empty area.

3. Select `New User` and fill in the required details, including username, full name, and password settings.

4. Optionally, set additional parameters like password expiration or account disablement.

5. Click `Create` to finish.

6. Close the Local Users and Groups console when done.

Shareef, T. (2024))

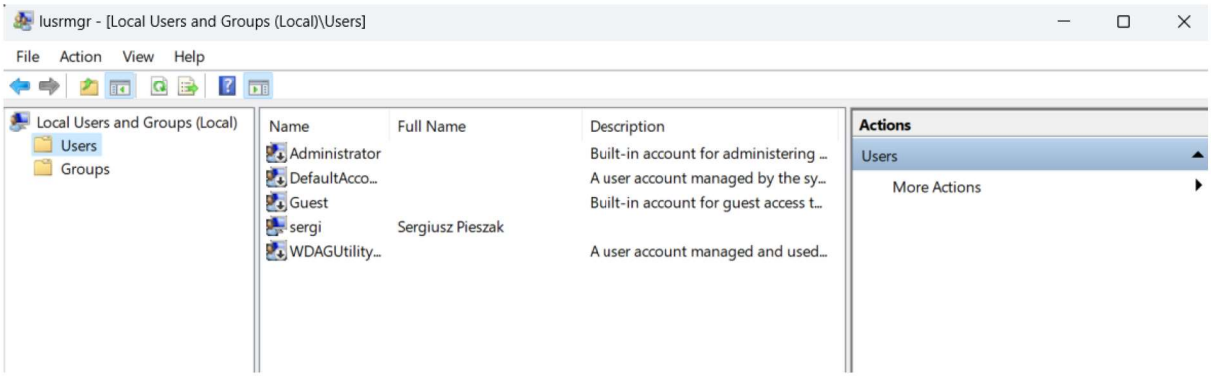


*Figure 1 – List of all accounts available.*

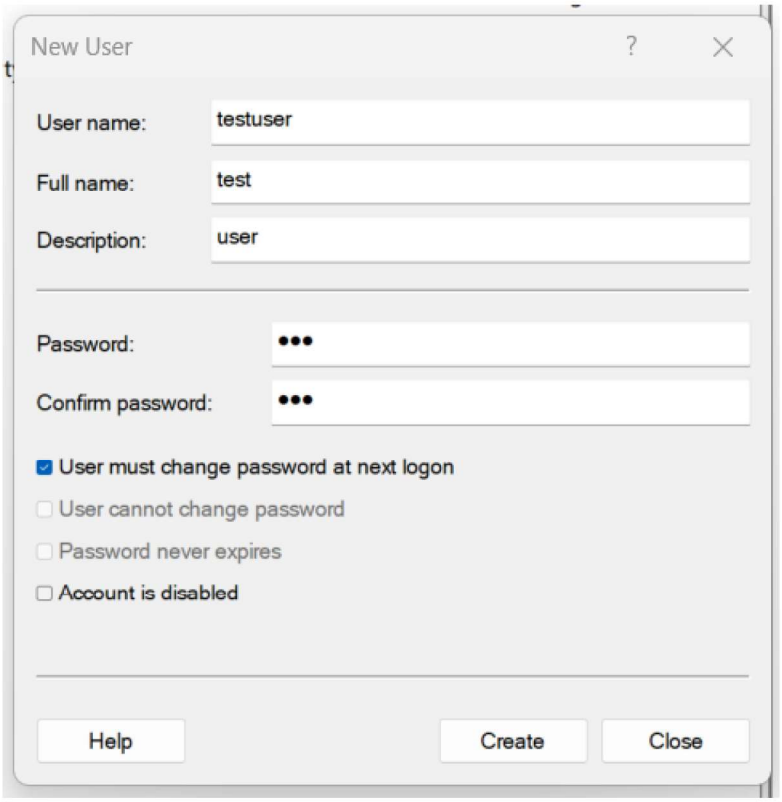


*Figure 2 – Creating new accounts.*

## 2.1.2 Manage account

1. Press `Windows Key + X` and choose `Computer Management`.

2. Expand `Local Users and Groups` in the left pane and click `Users`.

3. Right-click a user for options like `Properties` to modify details, `Disable Account` to prevent login, `Reset Password`, or `Delete` to remove the account.

4. Close Computer Management after making changes.

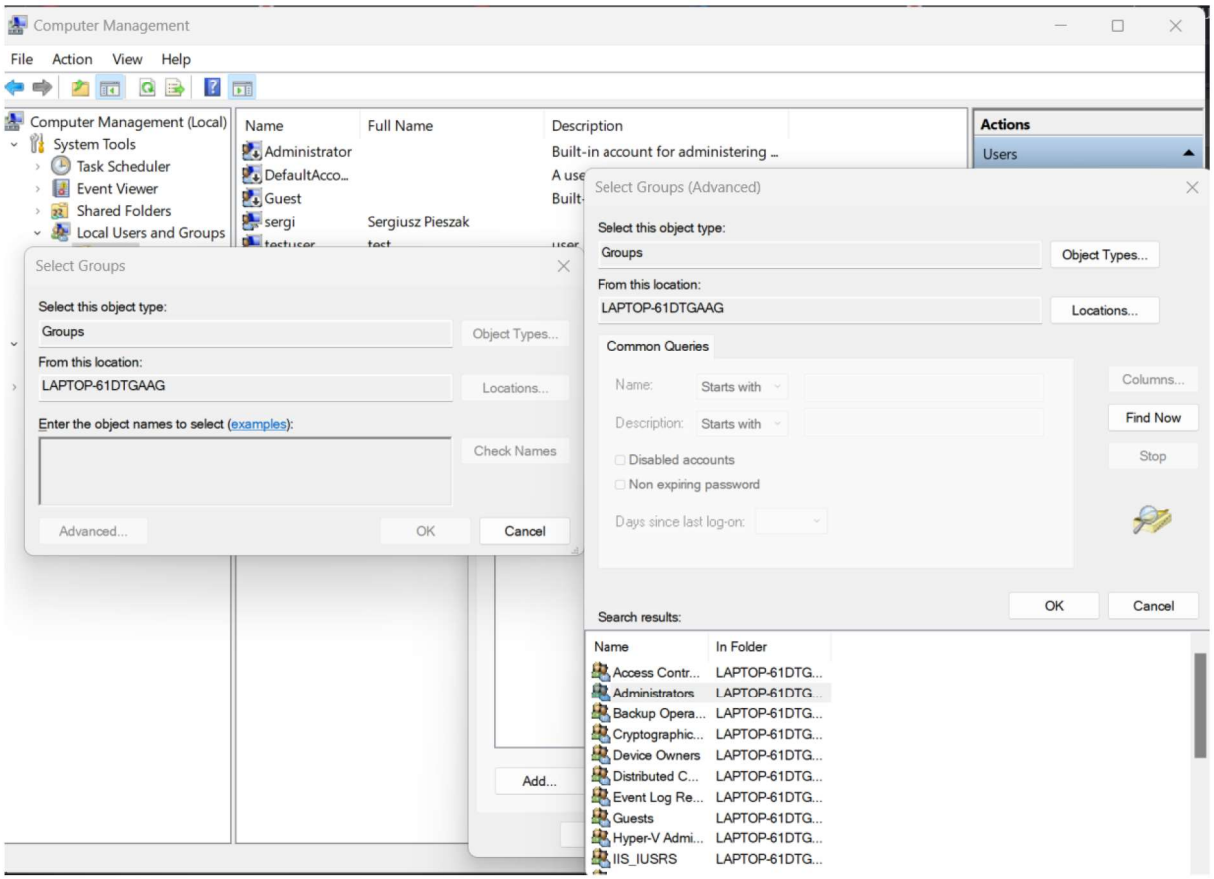


*Figure 3 – Giving new account administrator privileges.*

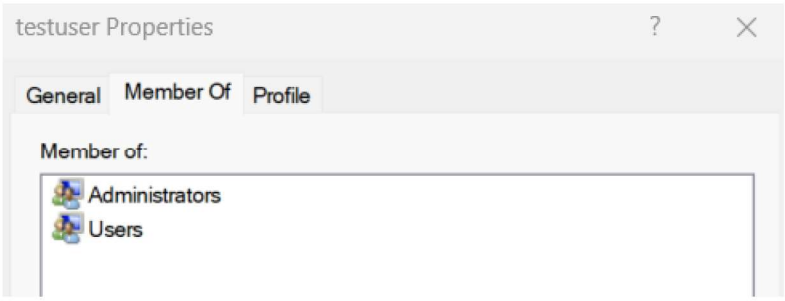


*Figure 4 – New account now has administrator privileges.*

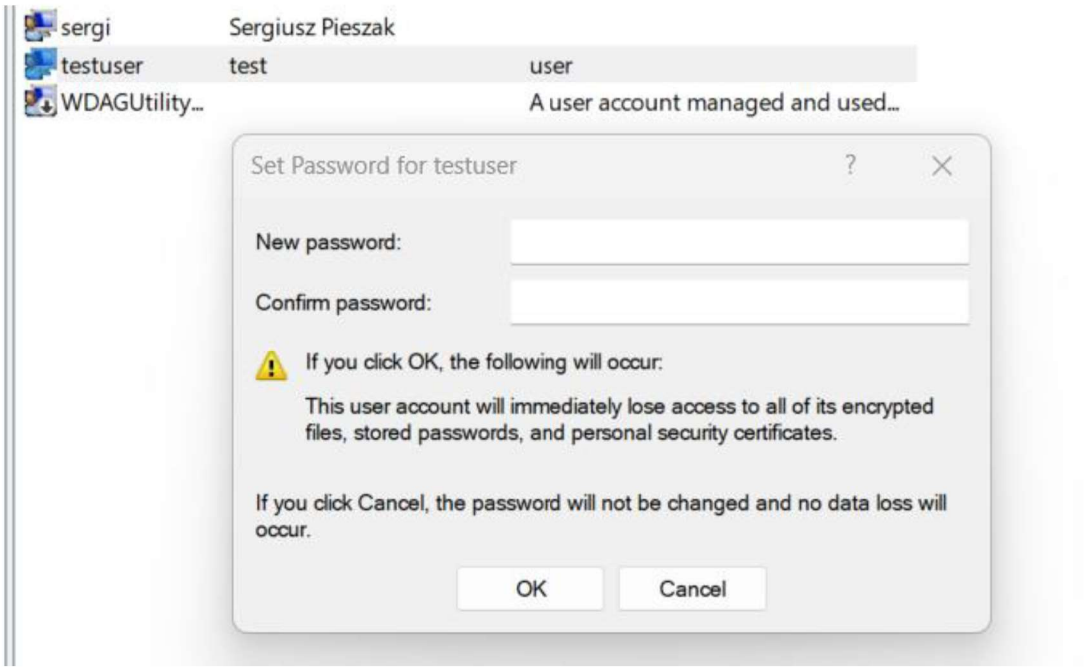


*Figure 5 – Changing password.*

### 2.1.3 Delete account

1. Press `Windows Key + X`, select `Computer Management`.
2. Navigate to `Local Users and Groups` > `Users`.
3. Right-click the user account and choose `Delete`.
4. Confirm the deletion by clicking `Yes`.
5. Remember, deleting a user account permanently removes it, along with associated data. Always back up important data beforehand.

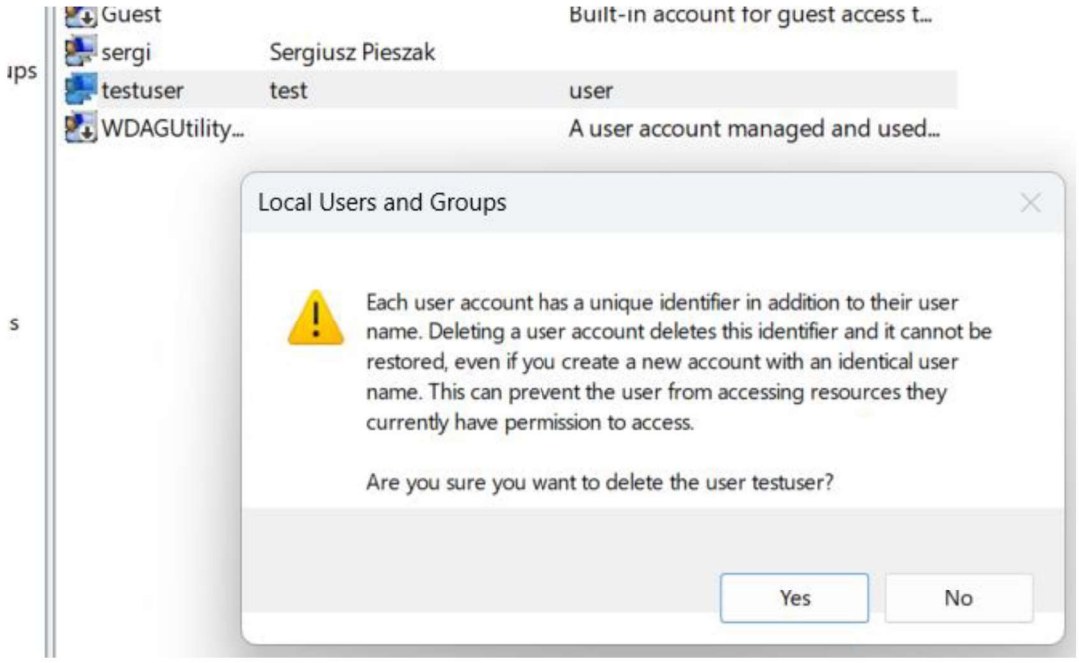


*Figure 6 – Deleting user account. This prompt provides an additional warning.*

## 2.2 Ubuntu

### 2.2.1 Creating account

1. Open a terminal window.
2. Enter `sudo adduser username` (replace `username`).
3. Type a password for the new user and press Enter.
4. Retype the password to confirm and press Enter.
5. Optionally, provide additional user information or leave it blank.
6. Review the information, type `Y`, and press Enter to confirm.
7. The new user account is created, allowing login with the specified username and password.

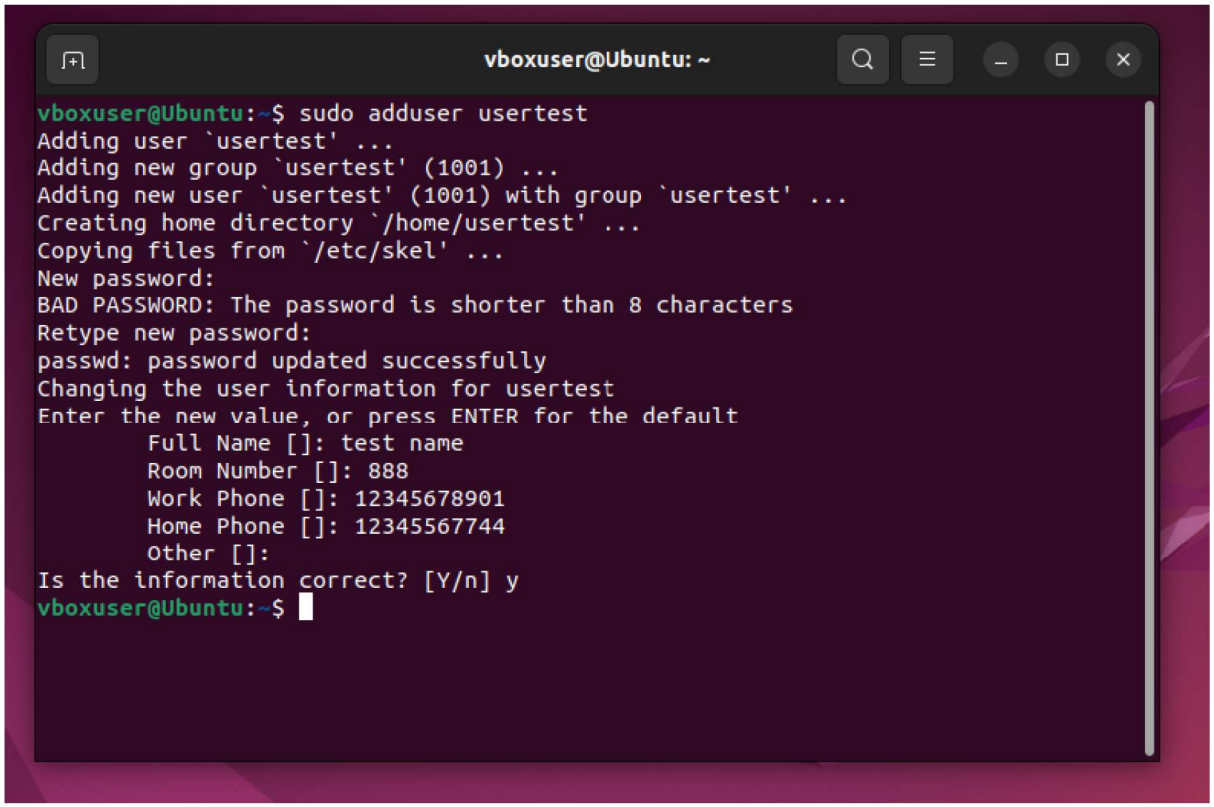


*Figure 7 – Creating new user account.*

### 2.2.2 Managing account

1. Open a terminal window.
2. Enter `sudo visudo` to safely edit the sudoers file.
3. Use `visudo` to navigate to the user privileges section.
4. Add a line granting sudo privileges: `username ALL=(ALL:ALL) ALL`
5. Save and exit the editor (Nano: `Ctrl + X`, `Y`, `Enter`).
6. Validate syntax with `sudo visudo -c`

7. Confirm correct syntax for changes to take effect.

**WARNING!** ***Exercise caution when editing sudoers to avoid system issues or security risks.***

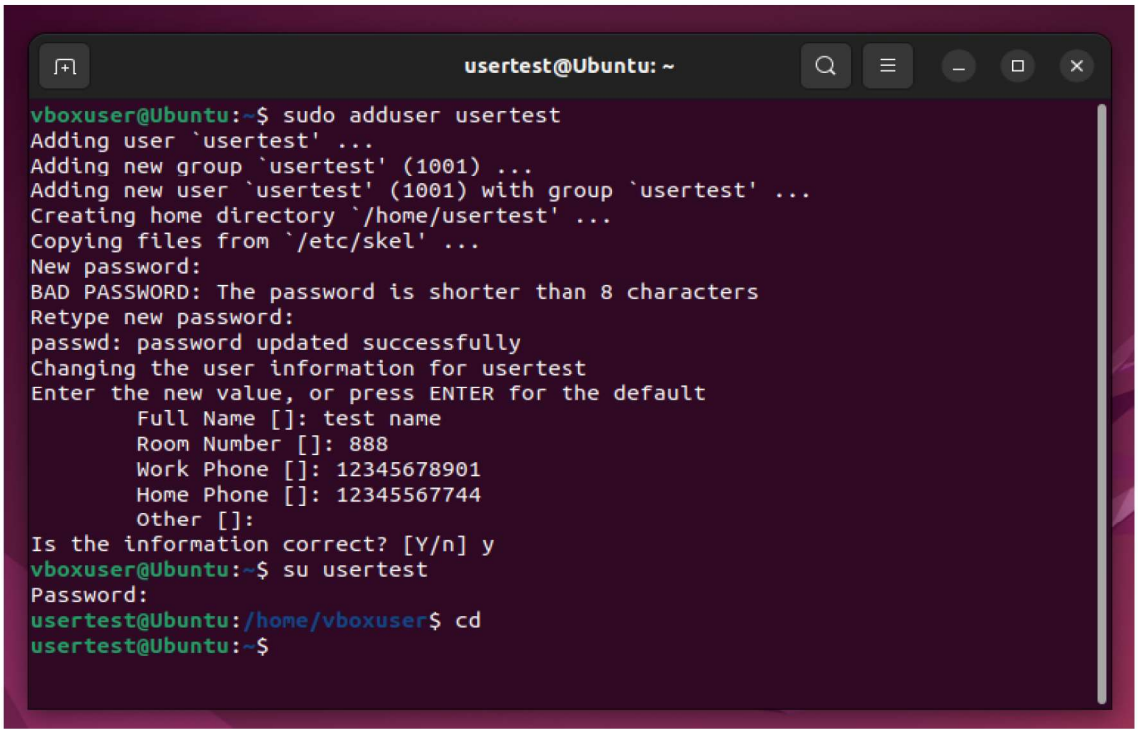


*Figure 8 – Switching users using su command.*

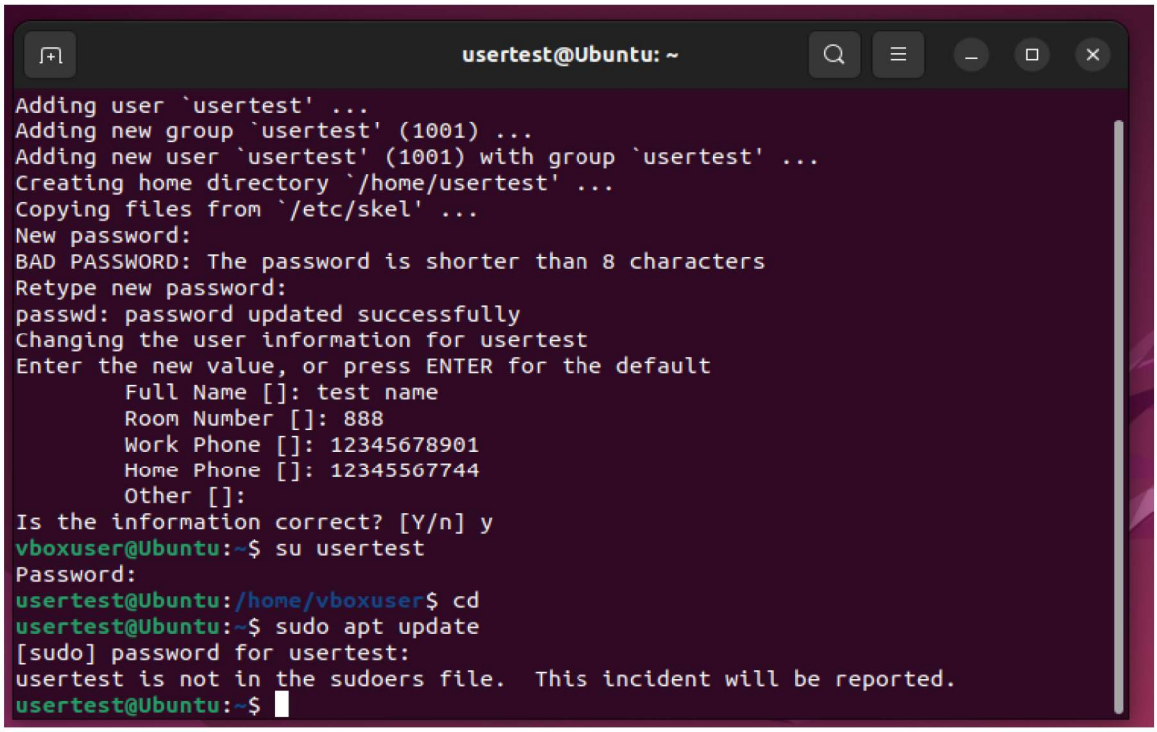


*Figure 9 – User doesn't have administrative privileges as shown by 'usertest is not in the sudoers file'.*

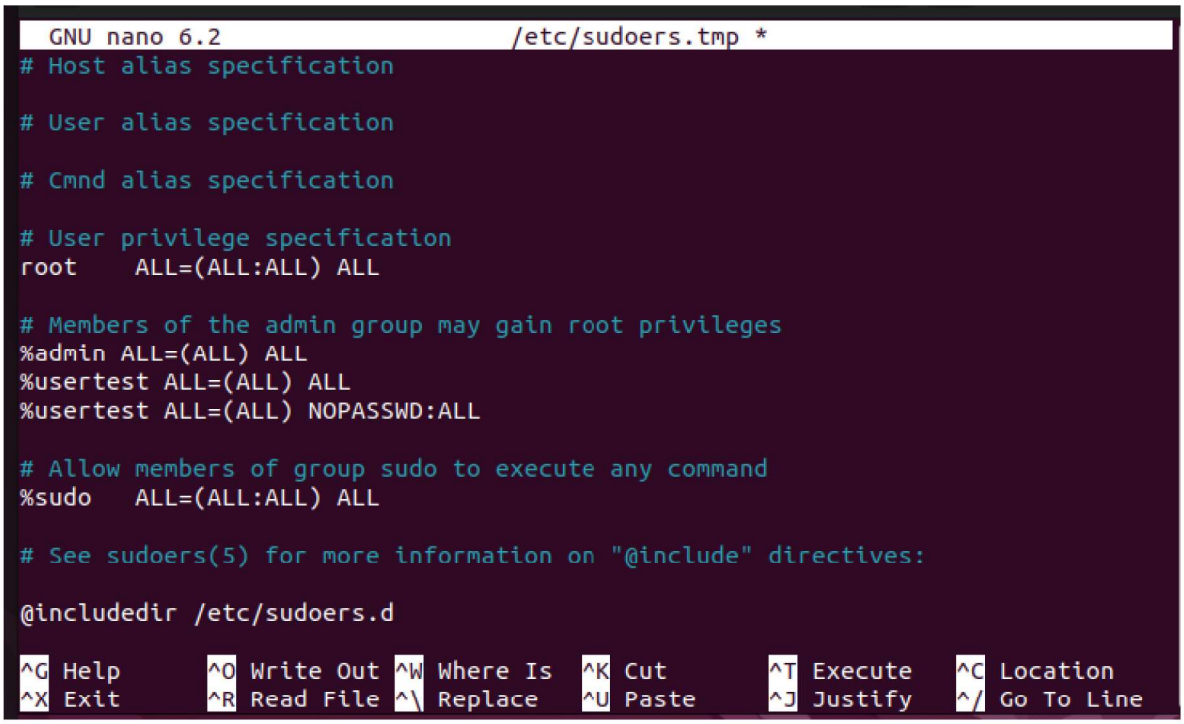


*Figure 10 – Adding 'usertest' into the sudoer file. 'NOPASSWD:ALL' allows this user to bypass password requirements.*

```
vboxuser@Ubuntu:~$ sudo visudo
vboxuser@Ubuntu:~$ su usertest
Password:
usertest@Ubuntu:/home/vboxuser$ cd
usertest@Ubuntu:~$ sudo apt update
[sudo] password for usertest:
Get:1 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Hit:2 http://gb.archive.ubuntu.com/ubuntu jammy InRelease
Get:3 http://gb.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Get:4 http://gb.archive.ubuntu.com/ubuntu jammy-backports InRelease [109 kB]
Get:5 http://gb.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [1,558 kB]
Get:6 http://gb.archive.ubuntu.com/ubuntu jammy-updates/main i386 Packages [609 kB]
```

*Figure 11 – Running 'update' to show that the user had administrative privileges.*

### 2.2.3 Change user password

1. Open a terminal.
2. Switch to root: `sudo su -`
3. Use `passwd username` to change the user's password.
4. Enter and confirm the new password.
5. Exit root shell: `exit`

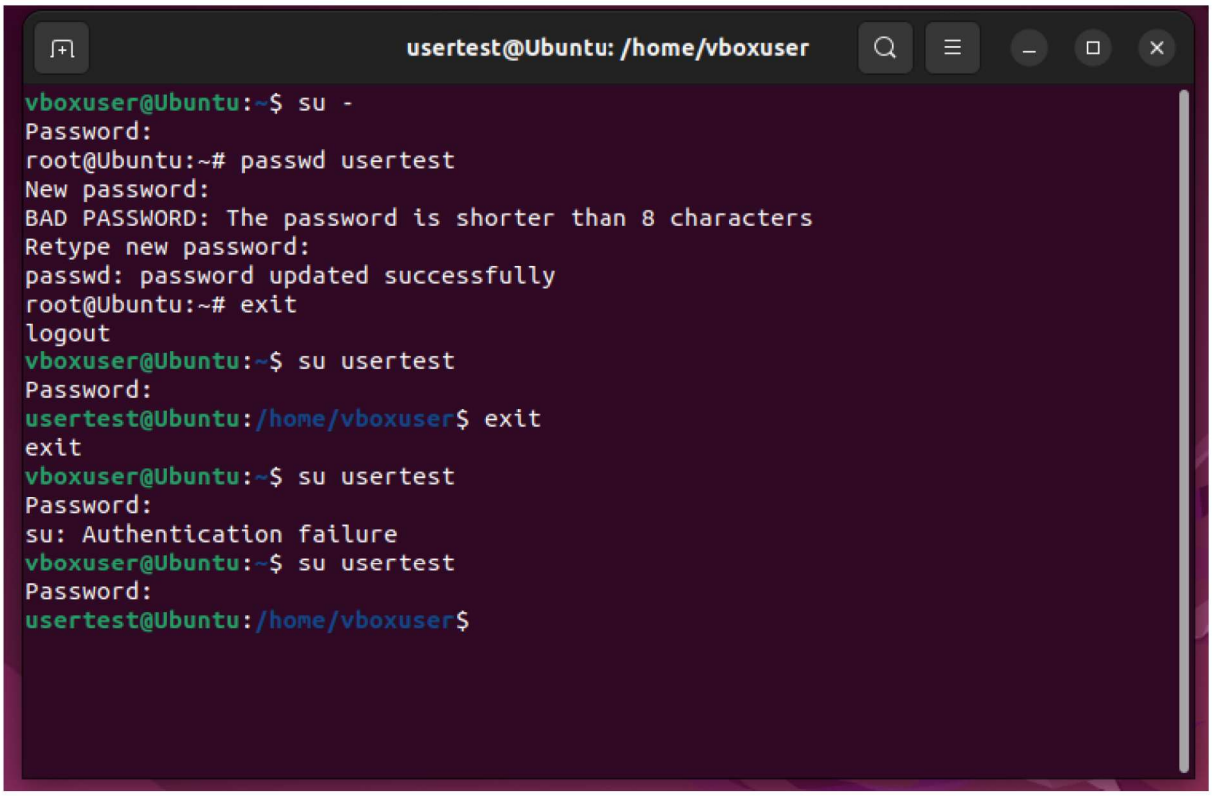


*Figure 12 – Demonstrating changing password.*

### 2.2.4 Delete Account

1. Open a terminal window.
2. Run `sudo deluser username` to delete the user account.
3. Optionally, add `--remove-home` to delete the user's home directory and mail spool.
4. Optionally, add `--remove-all-files` to delete associated groups.
5. Confirm deletion when prompted.

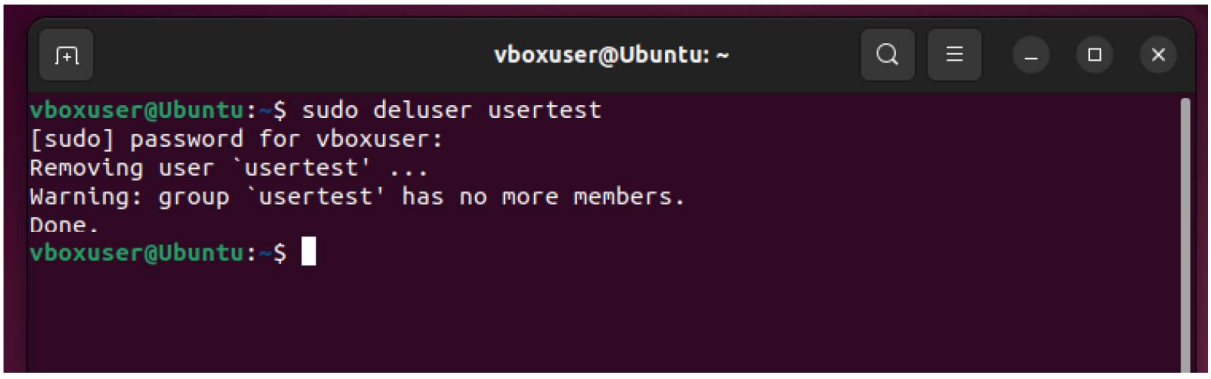


*Figure 13 – Removing user account.*

# 3. Authenticated and Secure Access

## 3.1 Ubuntu

Kili, A. (2024) shows how Linux provides secure user authentication through various mechanisms and processes. User accounts, managed using utilities like `adduser` and `deluser`, provide unique usernames and hashed passwords stored in system files. Password policies, stored in `/etc/shadow`, enforce security measures such as length and complexity. Authentication occurs through methods such as password, SSH keys, and biometrics, which are supported by PAM, which configures authentication policies centrally. As described by Newell, G. (2020) file permissions and ACLs control access to files and directories, while security policies like SELinux, Ionos (2020), and AppArmor enhance system security. Logging and auditing tools such as syslog and auditd monitor user activity for security analysis. By integrating these mechanisms, Linux systems authenticate users securely, manage access, and reinforce system security.

## 3.2 Windows 10

vinaypamnani-msft and paolomatarazzo (2024) discuss how Windows 10 employs various methods and processes to authenticate users and ensure secure access. User accounts, managed by utilities such as Control Panel or Computer Management, provide unique usernames and password hashes stored in the SAM database. Password policies, enforced through Group Policy settings, impose requirements such as length and complexity. In addition, authentication methods include username and password, PINs, biometrics, smart cards, and Windows Hello. As affirmed by Geeks for Geeks (2023) Windows utilises authentication protocols like NTLM and Kerberos secure network communications, with BitLocker Drive Encryption safeguarding data integrity. File permissions and user rights control access to files and system actions, managed through Group Policy settings. Windows Defender Antivirus and Windows Firewall offer built-in protection against threats, supplemented by regular security updates from Microsoft. Through these measures, Windows 10 ensures powerful user authentication, access control, and system security.

## 3.3 Availability in authentication options

As outlined earlier, Windows benefits from its familiarity with the hardware it's designed for, allowing for custom driver development, especially for security authentication methods like biometrics. In opposition, Ubuntu lacks specific hardware requirements, leading to a reliance on third-party drivers, particularly for less common features. This dependency raises security concerns, as Windows employs dedicated teams to developing robust biometric security drivers, unlike the fragmented support available for Ubuntu.

## 3.4 Security features to user authentication

As described above Ubuntu and Windows 10 both offer tough security features that are very similar, however, implementation varies as with the below examples:

- Ubuntu enforces password policies via the `passwd` command or PAM, while Windows 10 uses Group Policy settings.

- Ubuntu supports encryption with LUKS and eCryptfs, while Windows 10 employs BitLocker.
- Both systems utilize SSH, SFTP, HTTPS, and IPsec for secure communication, with Ubuntu focusing on IPsec and Windows 10 on protocols like NTLM, Kerberos, and SMB.

# 4. Access Privileges and Permissions:

Both Windows and Linux utilize access privilege models to regulate user interactions with system resources, using discretionary access control (DAC) and mandatory access control (MAC) mechanisms. Babko, A. (2023).

As stated by Medhi, A. (no date), in Windows, DAC governs resource access through permissions assigned to users or groups, allowing discretionary actions based on the discretion of resource owners. Each file or directory contains an Access Control List (ACL) specifying permissions for users and groups. Users with appropriate privileges can alter these permissions to grant or restrict access to resources based on their discretion. Additionally, Windows implements MAC through technologies like Windows Integrity Levels, HackTricks (no date) and User Account Control (UAC), where system-wide policies restrict access based on predefined rules, expanding the DAC with mandatory restrictions.

Similarly, Barrett-Morrison, N. (2022) explains that Linux utilizes DAC extensively, where file system permissions dictate access rights. Each file or directory contains permission bits specifying read, write, and execute permissions for the owner, group, and others. These permissions are administered through tools like `chmod` and `chown`, providing users discretionary control over resource access. Linux also supports MAC through security frameworks like SELinux (Security-Enhanced Linux) and AppArmor, which enforce mandatory restrictions on resource access based on predefined security policies, overriding DAC permissions when necessary.

# 5. Monitoring Access

## 5.1 Ubuntu

Monitoring individual access to services, software, and hardware in Ubuntu involves a range of tools:

1. System logs track activities such as user logins, software installations, and hardware events, accessible through tools like `journalctl` or directly in `/var/log`. freedesktop (no date).

2. Auditd offers detailed auditing of system calls and file access, with customizable rules for event tracking. Sematext (no date).

3. User and group management utilities like `adduser` and `usermod` allow administrators to configure permissions and group memberships for access control.

4. Resource monitoring tools such as `top`, `htop`, and `iotop` track system resource usage, aiding in identifying unauthorized activity. Zaharia, R. (2022)

5. Access Control Lists (ACLs) enable fine-grained control over file permissions.

6. Remote monitoring via SSH and remote desktop applications facilitates access control and troubleshooting remotely.

7. Security tools like fail2ban, tripwire, and AppArmor enhance security and monitoring capabilities.

8. Compliance monitoring and reporting tools like `auditbeat` and `osquery` ensure adherence to security policies and regulations.

## 5.2 Windows 10

As grodrigues43 (2023) shows, monitoring individual access to services, software, and hardware in Windows 10 involves a range of tools:

1. Event Viewer logs user logins, software installs, and hardware events.

2. Task Manager manages processes and resource usage, identifies security issues.

3. Group Policy enforces security policies and access controls.

4. Device Manager manages hardware devices and drivers.

5. Windows Security provides antivirus, firewall, and threat detection.

6. Resource Monitor tracks system resource usage.

7. Remote Desktop enables remote monitoring and management.

8. Windows Event Forwarding collects and analyses event log data.

9. PowerShell Scripts automates monitoring and management tasks.

# 6. Conclusion

In conclusion, user and access management in both Windows and Linux possess distinct strengths and weaknesses. Windows boasts a user-friendly interface, seamless integration with Active Directory for centralized management, and granular permissions control through ACLs. However, it can be complex to administer, vulnerable to security threats, and may entail significant costs for advanced features. In contrast, Linux offers flexibility, customization options, and robust security features such as DAC and MAC frameworks. Yet, it comes with a learning curve, fragmentation across distributions, and potential compatibility issues. Ultimately, administrators must consider these factors to devise effective user and access management strategies tailored to their specific requirements and objectives.

# 7. References


Babko, A. (2023) *Mandatory Access Control vs Discretionary Access Control*: Which to Choose? Available at: https://www.ekransystem.com/en/blog/mac-vs-dac (Accessed: 9 April 2024)

Barrett-Morrison, N. (2022) *Discretionary Access Control (DAC) Hardening*. Available at: https://www.timesys.com/security/discretionary-access-control-dac-hardening/ (Accessed: 9 April 2024)

freedesktop (no date) *journalctl*. Available at: https://www.freedesktop.org/software/systemd/man/latest/journalctl.html (Accessed: 7 April 2024)

Geeks for Geeks (2023) *Difference between Kerberos and NTLM*. Available at: https://www.geeksforgeeks.org/difference-between-kerberos-and-ntlm/ (Accessed: 9 April 2024)

grodrigues43 (2023) *Exploring Monitoring Tools for Windows Environments: Tracking User Activities and Logins*. Available at: https://techcommunity.microsoft.com/t5/windows-server-for-it-pro/exploring-monitoring-tools-for-windows-environments-tracking/td-p/3926646 (Accessed: 7 April 2024)

HackTricks (no date) *Integrity Levels*. Available at: https://book.hacktricks.xyz/windows-hardening/windows-local-privilege-escalation/integrity-levels (Accessed: 7 April 2024)

Ionos (2020*) What is SELinux?* Available at: https://www.ionos.co.uk/digitalguide/server/security/what-is-selinux/ (Accessed: 10 April 2024)

Karan, R. (2024) *Difference Between Ubuntu And Windows*. Available at: https://www.shiksha.com/online-courses/articles/difference-between-ubuntu-and-windows/ (Accessed: 10 April 2024)

Kili, A. (2024) *20 Useful Security Features and Tools for Linux Admins*. Available at: https://www.tecmint.com/security-features-tools-linux-admins/ (Accessed: 10 April 2024)

Medhi, A. (no date) *Windows Access Control: ACL, DACL, SACL, & ACE*. Available at: https://www.securew2.com/blog/windows-access-control-acl-dacl-sacl-ace (Accessed: 7 April 2024)

Newell, G. (2020) *An introduction to Linux Access Control Lists (ACLs)*. Available at: https://www.redhat.com/sysadmin/linux-access-control-lists (Accessed: 10 April 2024)

Sematext (no date) *auditd*. Available at: https://sematext.com/glossary/auditd/ (Accessed: 7 April 2024)

Shareef, T. (2024) *How to Enable Local Users and Groups Management in Windows 11 and 10 Home*. Available at: https://www.makeuseof.com/windows-home-edition-enable-local-user-group-management/ (Accessed: 10 April 2024)

vinaypamnani-msft and paolomatarazzo (2024) *Windows operating system security*. Available at: https://learn.microsoft.com/en-us/windows/security/operating-system-security/ (Accessed: 10 April 2024)

Zaharia, R. (2022) *Linux htop, iftop and iotop commands*. Available at: https://blog.raduzaharia.com/linux-htop-iftop-and-iotop-commands-a95dcd7d39b2 (Accessed: 9 April 2024)