Task 2 - Operating System Security Fundamentals

1. Introduction to Operating System

1.1 What is an Operating System (OS)?

An operating system is the core software that manages a computer's hardware and software resources. It acts as an interface between the user and the system, controlling how applications run, how files are stored, and how hardware such as memory, processor, and storage is used. Common examples of operating systems include Windows, Linux, and macOS.

In simple terms, the operating system decides **who can do what**, **when**, and **how** on a system.

1.2 What is Operating System Security?

Operating system security refers to the measures and controls used to protect the operating system from unauthorized access, misuse, and attacks. It focuses on securing users, files, processes, and system services to ensure that the system operates safely and reliably.

OS security ensures that only authorized users can access system resources, critical files are protected from modification, and malicious activities are detected or prevented.

1.3 Types of Operating System Security

Operating system security is implemented through multiple mechanisms, including:

- **User and Access Control Security** – managing user accounts and privilege levels
- **File and Permission Security** – restricting access to files and directories
- **Process and Service Security** – controlling which background services are allowed to run
- **Network Security** – filtering traffic using firewalls and network rules
- **System Hardening** – reducing attack surface by disabling unnecessary features

These mechanisms work together to protect the system as a whole.

1.4 Why Do We Need Operating System Security?

Operating systems are a primary target for attackers because they control the entire system. If an attacker compromises the OS, they can bypass application-level security, access sensitive data, and take full control of the system.

Without proper OS security:

- Malware and Rootkits can gain high-level privileges
- Unauthorized users can access or modify system files
- Network services may be exposed to attacks
- The system becomes unstable or unreliable

OS security helps prevent these risks by enforcing strict controls at the system level.

1.5 Why is OS Security Foundational to Cybersecurity?

Operating system security forms the foundation of cybersecurity because all applications and services rely on the OS to function securely. Even the most secure application cannot protect itself if the underlying operating system is compromised.

The OS enforces authentication, authorization, isolation, and monitoring, making it the first layer of defense against cyber threats. Weak OS security can render higher-level security controls ineffective.

1.6 Role of Operating System Security in Cybersecurity

OS Security Function	How It Contributes to Cybersecurity
User Authentication	Ensures only authorized users can access the system and its resources.
Privilege Management	Limits user actions based on roles, reducing damage from misuse or malware.
File and Access Control	Protects critical files from unauthorized reading, modification, or deletion.
Process and Service Control	Prevents malicious or unnecessary services from running in the background.
Network Control	Filters incoming and outgoing traffic using firewalls to block unauthorized access.
System Monitoring	Helps detect suspicious activities through logs and running process monitoring.
Attack Surface Reduction	Disabling unused features and services minimizes entry points for attackers.

2. User Accounts & Privilege Management

2.1 What are User Accounts?

A user account is an identity created within an operating system that allows an individual or process to access system resources. User accounts help the operating system identify **who is using the system** and **what actions they are allowed to perform**.

Operating systems use user accounts to separate different users and their activities. This separation ensures that one user's actions do not unintentionally affect other users or the system itself. Each account is associated with specific permissions, settings, and access rights, which play a critical role in maintaining system security.

By managing user accounts properly, operating systems can enforce access control, track activity, and reduce the risk of unauthorized system usage.

2.2 Types of User Accounts

Different user accounts are designed to serve different purposes while maintaining system security.
Types of User Accounts in Operating Systems:

User Account Type	Description	Security Purpose
Standard User	Has limited permissions for daily tasks such as running applications and accessing personal files.	Prevents accidental or malicious changes to system settings.
Administrator / Root User	Has full control over the system, including installing software and modifying system configurations.	Allows system management but poses high risk if misused.
Guest User	Temporary account with very limited access and no ability to make permanent changes.	Provides short-term access while minimizing security risks.
Service Account	Used by system services or applications to run background processes.	Restricts services to only the permissions they require.

This separation of user account types helps limit system exposure and ensures controlled access to sensitive resources.

2.3 Privilege Levels & Access Rights

Privilege levels define how much control a user or process has over an operating system. Access rights determine **what actions are permitted**, such as reading files, modifying system settings, or installing software.

Privilege Levels and Access Rights – Comparative View

a) **Privilege Level:** Low Privilege (Standard User)
Level of Control: Minimal system control
Allowed Actions: Run applications, access personal files
Security Impact: Limits damage from malware and user errors.

b) **Privilege Level:** Medium Privilege (Elevated User)
Level of Control: Partial system control
Allowed Actions: Modify selected settings, install approved software
Security Impact: Requires careful management to prevent misuse.

c) **Privilege Level:** High Privilege (Administrator / Root)
Level of Control: Full system control
Allowed Actions: Modify system files, manage users, change security settings
Security Impact: High risk if compromised; can lead to full system takeover.

d) **Privilege Level:** System / Service Privilege
Level of Control: Task-specific control
Allowed Actions: Run background services with restricted permissions
Security Impact: Reduces risk through controlled execution.

Security Insight:
As privilege levels increase, the potential security impact of misuse or compromise also increases.

2.4 Principle of Least Privilege

The principle of least privilege states that users and processes should be granted **only the minimum level of access required** to perform their tasks.

Applying least privilege improves security by:

- Reducing the attack surface
- Limiting damage caused by mistakes
- Containing the impact of malware

Examples

- A standard user account cannot install system-wide software, preventing accidental configuration changes.

- Malware running under a low-privilege account is restricted from modifying system files or disabling security controls.
- Limiting administrative access reduces the risk of insider misuse and unauthorized system changes.

2.5 Security Risks of Excessive Privileges

Granting excessive privileges significantly weakens operating system security.

- **Expanded Attack Surface:**
  When all users have administrative access, attackers gain immediate access to critical system components.
- **Insider Threat Amplification:**
  Insiders with unnecessary high-level access can intentionally or unintentionally compromise system security.
- **Malware Escalation:**
  Malware executed with administrator or root privileges can disable security mechanisms and install persistent threats.
- **Configuration Abuse:**
  Excessive privileges allow unauthorized changes to firewall rules, services, and security settings.
- **Stealth and Persistence:**
  High-privilege malware can hide deeply within the system, making detection and removal difficult.

2.6 Practical Observation (Windows & Linux)

User privilege management was observed on both Windows and Linux systems. On Windows, the system uses administrator and standard user roles to control access to system-level settings. On Linux, commands such as “whoami” help identify the current user context, highlighting whether actions are performed with root privileges. These observations demonstrate how privilege separation is implemented in real operating systems to reduce security risks.

Figure 2.6.1: *User privilege management was observed on a* Windows system*. The user account type was identified as an administrator account, demonstrating how operating systems differentiate access levels to control system-level actions. This distinction highlights the importance of restricting administrative access to*

reduce security risks.

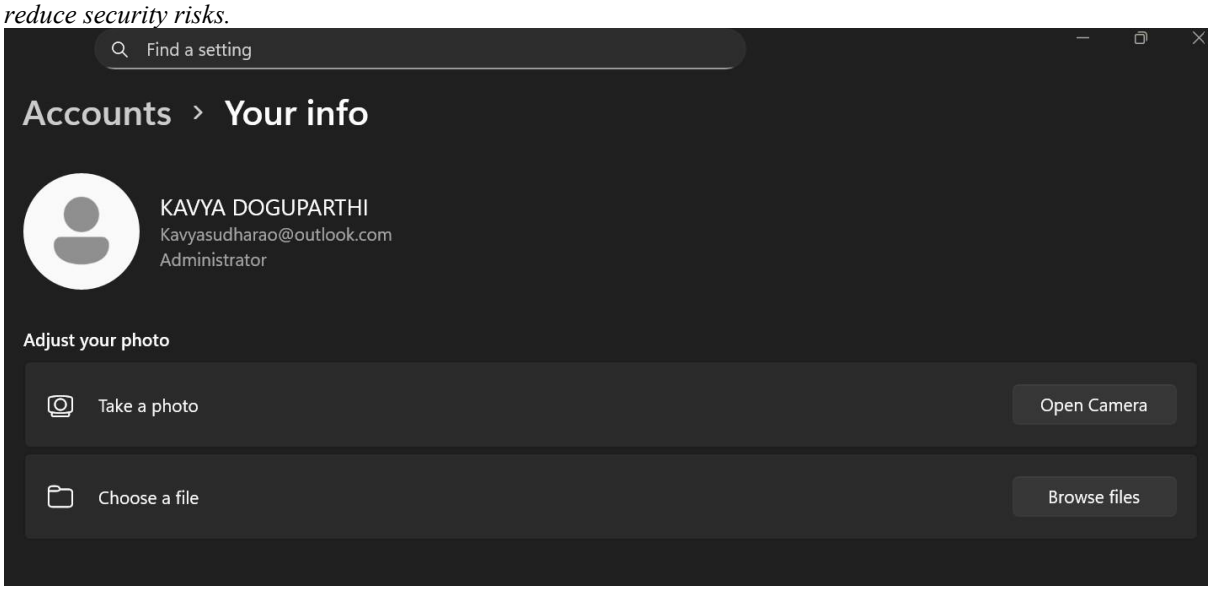
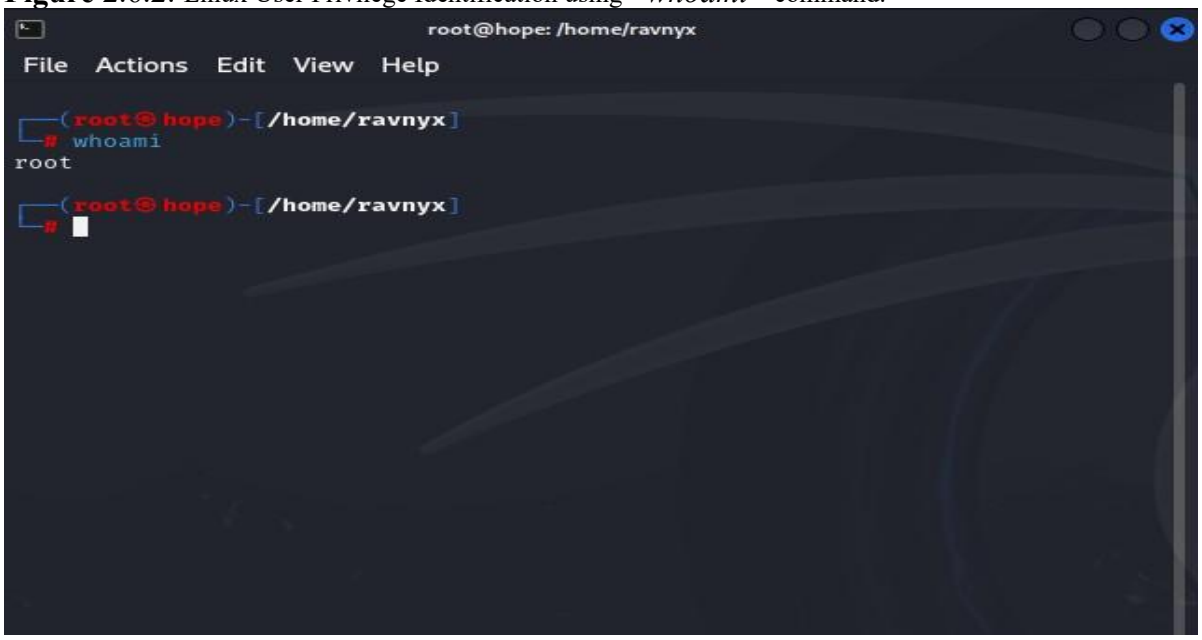


Figure 2.6.2: Linux User Privilege Identification using *"whoami"* command.



2.7 How Privilege Management Supports OS Security

Proper privilege management strengthens OS security by ensuring that users and processes operate within defined boundaries. By restricting access to sensitive system resources, privilege separation helps maintain confidentiality, integrity, and system stability.

Effective privilege management ensures that security incidents are contained and do not escalate into full system compromises.

3. File Permissions & Access Control

3.1 What are File Permissions?

File permissions define how files and directories can be accessed and used within an operating system. They control who is allowed to read, modify, or execute files, helping protect sensitive data and critical system resources from unauthorized access.

Files play a direct role in system security because they store system configurations, application data, and executable programs. Properly configured file permissions ensure that only authorized users and processes can interact with these files, reducing the risk of misuse or compromise.

Relationship Between Files and System Security

File Aspect	Role in the System	Security Impact
System Files	**Control core OS behaviour and functionality**	**Unauthorized modification can compromise the entire system**
User Files	**Store personal and application data**	**Improper access can lead to data leakage**
Executable Files	**Run programs and scripts**	**Malicious execution can cause malware infections**
Configuration Files	**Define system and service settings**	**Tampering can weaken security controls**
Log Files	**Record system and user activities**	**Restricted access prevents attackers from hiding traces**

3.2 Types of File Permissions (Read, Write, Execute)

Operating systems use three basic permission types to control file access:

- ✓ Read (r): Allows viewing the contents of a file or listing a directory.
- ✓ Write (w): Allows modifying or deleting a file or directory.
- ✓ Execute (x): Allows running a file as a program or accessing a directory.

Among these, execute permission is particularly sensitive because it allows code to run on the system. Improper execute permissions can enable malicious programs to run without authorization.

3.3 Ownership and Permission Groups

Every file in an operating system is associated with:

- An owner – usually the user who created the file
- A group – a set of users with shared access
- Other users – all remaining users on the system

Ownership helps the operating system decide which permissions apply to which users. By separating owners, groups, and others, the system ensures controlled access and prevents unauthorized users from modifying sensitive files.

3.4 Linux Permission Representation (rwx Model)

In Linux systems, file permissions are represented using the rwx model, which groups permissions into three categories:

rwxr-x---

- The first group applies to the owner
- The second group applies to the group
- The third group applies to others

Each group defines whether read, write, or execute access is allowed. This structured representation makes it easier to understand and manage access control for files and directories.

3.5 Permission Management Commands

Linux provides commands to view and manage file permissions:

- ✓ ls -l displays file permissions, ownership, and group details.
- ✓ chmod is used to modify read, write, and execute permissions.
- ✓ chown is used to change file ownership.

These commands allow administrators to enforce access control policies. In this task, permissions were observed conceptually without modifying system settings to avoid unintended system changes.

3.6 Security Importance of File Permissions

File permissions play a critical role in operating system security by:

- Preventing unauthorized access to sensitive files
- Protecting system and configuration files from modification
- Limiting the impact of malware by restricting file access
- Supporting accountability through controlled ownership

Proper permission management ensures system stability and reduces the likelihood of security incidents**.**

3.7 Risks of Misconfigured File Permissions

Incorrect file permissions can introduce serious security risks, including:

- **Unauthorized Access**: Sensitive files may be read or modified by unintended users.
- **Privilege Escalation**: Attackers may exploit writable system files to gain higher privileges.
- **Malware Propagation**: Executable permissions on unsafe files can enable malware execution.
- **Data Leakage**: Poor access control can expose confidential data.

These risks highlight the importance of regularly reviewing and maintaining secure permission settings.

3.8 Practical Observation (Linux Environment)

File permissions were observed using a Kali Linux virtual machine environment. The ***ls -l*** command was used to examine file ownership and permission settings, demonstrating how Linux enforces access control at the file system level without modifying system configurations**.**

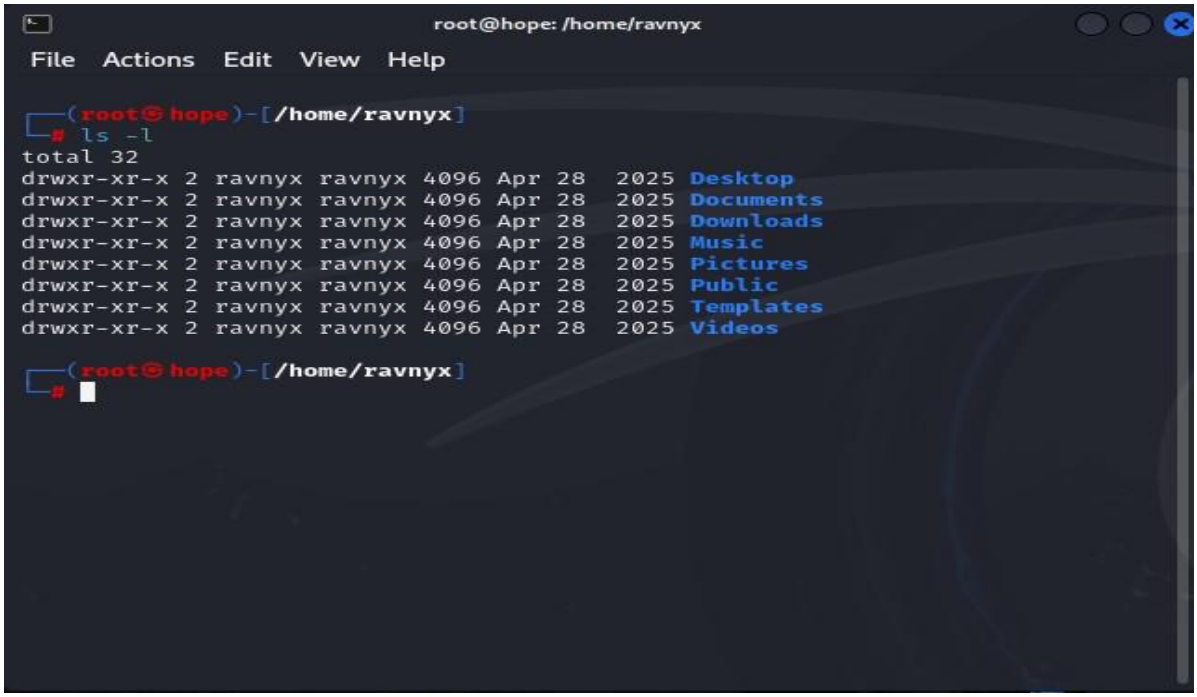


Figure 3.8*: File permissions were observed using the* ls -l *command in a Kali Linux virtual machine, demonstrating ownership and permission settings enforced by the operating system.*

4. Firewall Configuration

4.1 What is a Firewall?

A firewall is a security mechanism that monitors and controls network traffic entering or leaving a system. At the operating system level, a firewall acts as a protective barrier between the system and external networks by deciding which connections are allowed and which are blocked.

Firewalls work by applying predefined rules to network traffic. These rules help ensure that only legitimate and authorized communication is permitted, while suspicious or unauthorized access attempts are denied.

4.2 Types of Firewalls (Operating System Perspective)

From an operating system security standpoint, firewalls can be categorized as:

- **Host-Based Firewalls:**
  Installed and managed directly on the operating system. Examples include Windows Firewall and Linux UFW. These firewalls protect individual systems by controlling traffic at the device level.
- **Network Firewalls:**
  Deployed at the network perimeter to protect multiple systems. While important, they do not replace the need for OS-level firewalls.

OS-level firewalls are essential because they provide protection even when a system is connected to untrusted networks.

4.3 Role of Firewall in Operating System Security

Firewalls play a critical role in OS security by:

- Blocking unauthorized inbound network connections
- Restricting unnecessary outbound communication
- Protecting system services from direct exposure
- Reducing the system's overall attack surface

By filtering traffic based on defined rules, firewalls prevent attackers from directly interacting with vulnerable services running on the system.

4.4 Firewall Rules and Traffic Control

Firewall rules define how network traffic is handled by the operating system. These rules specify:

- Which connections are allowed or denied
- Whether the rule applies to inbound or outbound traffic

- The network profiles under which the rule is enforced

A common security practice is to follow a "**default-deny approach,"** where traffic is blocked unless explicitly permitted. This approach minimizes exposure and limits unexpected communication.

4.5 Practical Observation (Windows Firewall)

Windows Firewall was observed to be enabled across public, private, and domain network profiles. This ensures that the system remains protected regardless of the network environment it is connected to.

No configuration changes were made during this task, as the focus was on understanding firewall functionality and its role in OS security.

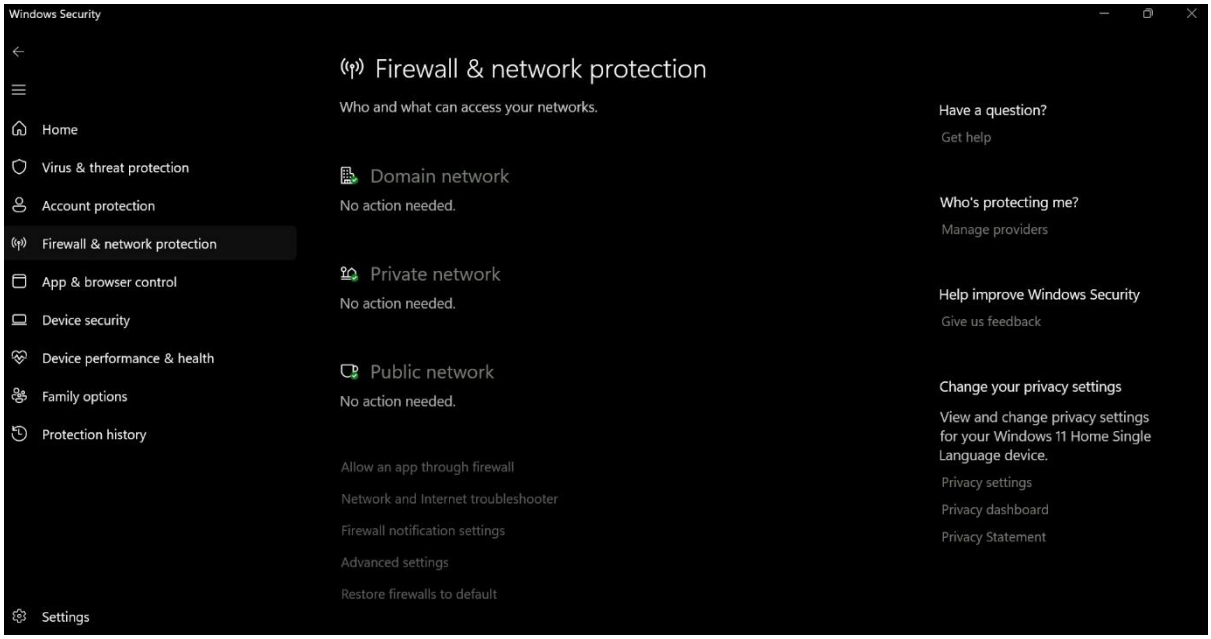


Figure 4.5: *Windows firewall was observed to be enabled across public, private, and domain network profiles, ensuring protection against unauthorized network traffic.*

4.6 Risks of Disabled or Misconfigured Firewalls

Disabling or misconfiguring a firewall can lead to serious security risks, including:

- **Exposed Services:**
  System services may become accessible to attackers over the network.
- **Increased Attack Surface:**
  Open ports and unrestricted traffic create additional entry points.
- **Remote Exploitation:**
  Attackers can exploit vulnerabilities in exposed services.

- **Malware Communication:**
  Malicious software may communicate freely with external servers if outbound traffic is unrestricted.
- These risks highlight the importance of maintaining proper firewall configurations.

5. Running Processes & Services

5.1 What are Processes and Services?

Processes and services are programs that run on an operating system to perform specific tasks. A **process** is an instance of a running program, while a **service** is a background process designed to run continuously and provide functionality to the system or applications.

Operating systems rely on processes and services to manage system operations such as networking, user authentication, and application execution. While many of these are essential, some services may be unnecessary depending on system usage.

5.2 Difference Between User Processes and System Services

- **User Processes:**
  Initiated by users when applications are opened. These processes usually stop when the application is closed.
- **System Services:**
  Run in the background and often start automatically when the system boots. These services support core OS functions and application dependencies.

System services typically operate with higher privileges, making them a critical focus area for operating system security.

5.3 Role of Processes and Services in OS Security

Processes and services directly impact OS security because they:

- Control how system resources are used
- Listen for network connections
- Interact with system files and memory

Every running service increases the system's attack surface. Securing processes involves ensuring that only required services are active and that they operate with appropriate privileges.

5.4 Why Unnecessary Services are a Security Risk

Unnecessary or misconfigured services pose security risks such as:

- ✓ **Expanded Attack Surface:**
  Each running service creates a potential entry point for attackers.
- ✓ **Privilege Abuse:**
  Services often run with elevated privileges, increasing impact if compromised.
- ✓ **Exploitation of Vulnerabilities:**
  Outdated or unused services may contain unpatched vulnerabilities.

- ✓ **Persistence Mechanisms:**
  Attackers can hide malicious activities within background services.

Disabling unused services helps reduce exposure and improve system security.

5.5 Monitoring Running Processes and Services

Monitoring running processes helps identify:

- Suspicious or unknown processes
- Resource misuse
- Unauthorized background activities

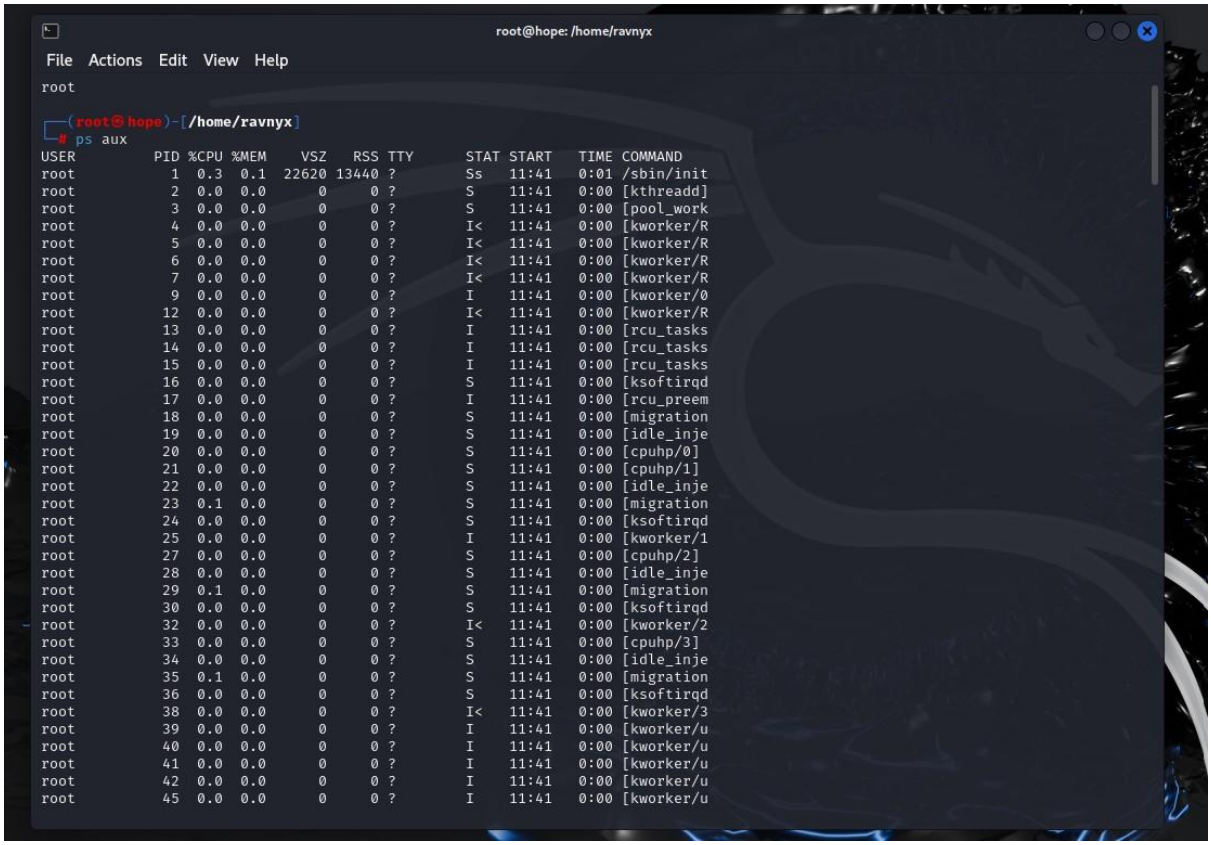
On Linux systems, commands such as ps aux can be used to view active processes. Regular monitoring enables early detection of abnormal behaviour and supports incident response.

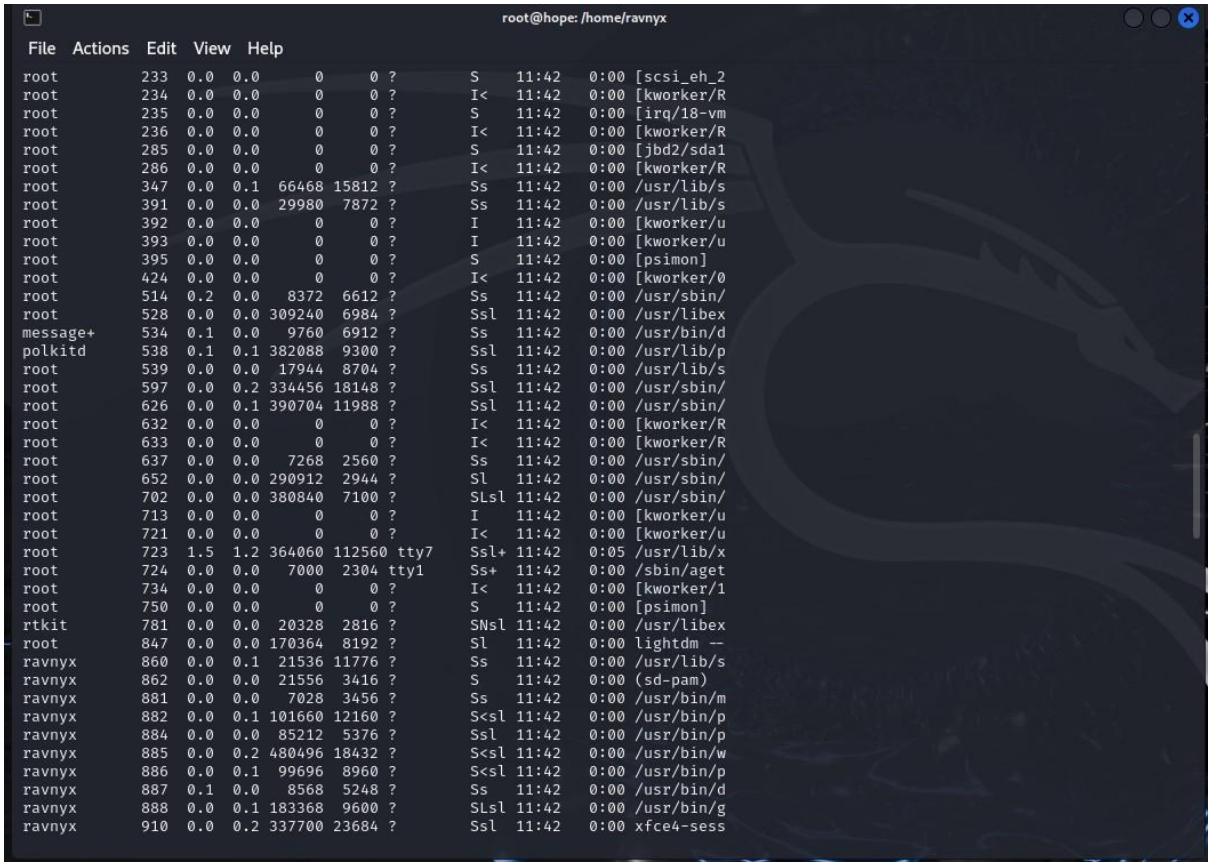
5.6 Practical Observation (Linux Environment)

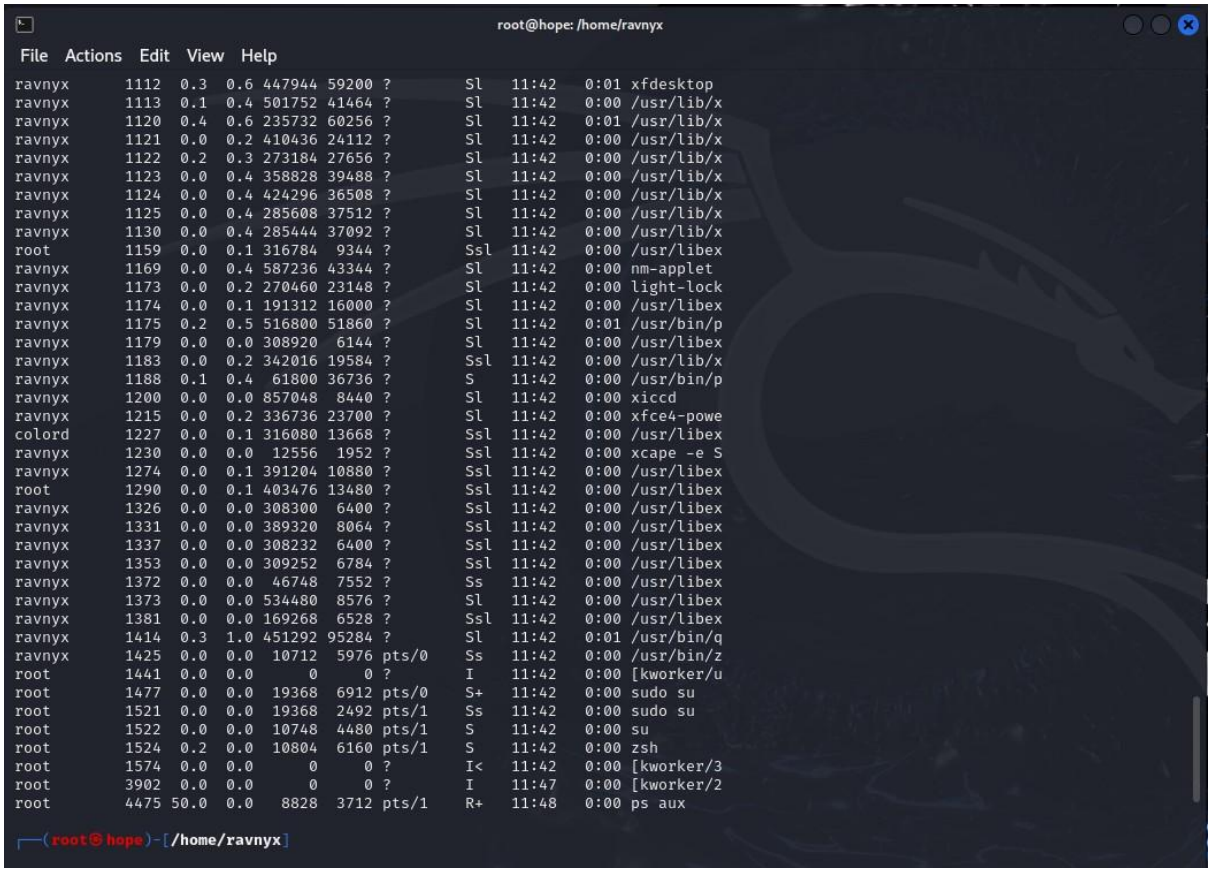
Running processes were observed using a Kali Linux virtual machine environment. The ps aux command was used to examine active system processes and services, demonstrating how operating systems manage background tasks and how service visibility contributes to security awareness.

No changes were made to running services during this task.

Figure 5.6: *Active processes and services were examined using the* **ps aux** *command in Kali Linux environment to understand how background services contribute to system functionality and attack surfaces.*







5.7 Security Best Practices for Managing Services

Good security practices for managing processes and services include:

- Running only essential services
- Disabling unused or legacy services
- Keeping system services updated
- Monitoring processes regularly
- Restricting service privileges where possible

These practices help minimize security risks and improve system stability.

6. Operating System Hardening Checklist

6.1 What is OS Hardening?

Operating system hardening refers to the process of securing an OS by reducing vulnerabilities and minimizing its attack surface. This is achieved by disabling unnecessary features, applying security controls, and enforcing strict access policies.

OS hardening ensures that even if an attacker gains limited access, the impact of the attack remains controlled and contained.

6.2 Importance of OS Hardening

OS hardening is essential because:

- Operating systems are common attack targets
- Default configurations may expose unnecessary services
- Weak configurations can be exploited easily
- Hardening strengthens the system's first line of defense

Hardening helps convert a general-purpose OS into a more secure and controlled environment.

6.3 OS Hardening Checklist

The following checklist outlines essential OS hardening practices:

✓ Use strong and unique passwords
✓ Enforce the principle of least privilege
✓ Enable and configure firewalls
✓ Disable unnecessary services and features
✓ Apply regular operating system updates
✓ Secure file permissions and ownership
✓ Monitor running processes and services
✓ Enable logging and review system logs
✓ Avoid using administrator/root access for daily tasks

This checklist helps ensure that the operating system remains secure, stable, and resilient against common attacks.

6.4 How OS Hardening Reduces Attack Surface

OS hardening reduces attack surface by:

- Limiting the number of exposed services
- Restricting access to critical system resources

- Preventing unauthorized network communication
- Containing malware through permission and privilege controls

By reducing available entry points, attackers have fewer opportunities to exploit vulnerabilities.

6.5 Relationship Between OS Hardening and Cybersecurity

OS hardening supports broader cybersecurity goals by:

- Protecting confidentiality through access control
- Preserving integrity by preventing unauthorized modifications
- Ensuring availability by maintaining system stability

A hardened operating system strengthens all higher-level security mechanisms, including applications and network defenses.

Conclusion / Learning Summary:

This task helped me understand how operating system security forms the base of cybersecurity. By studying user privileges, file permissions, firewalls, running services, and OS hardening practices, I learned how security controls at the system level reduce risks and limit attack surfaces.

Observing these mechanisms in both Windows and Linux environments made it easier to connect theoretical concepts with real system behaviour. Overall, this task reinforced the importance of securing the operating system to protect data, maintain system integrity, and prevent security incidents**.**