Process Management and Initial Security Configuration

Learning Objectives

- Understand process lifecycle and states
- Use command-line tools for process monitoring
- Manage system processes effectively
- Implement coursework security controls

Pre-Lab Preparation

Watch the following video lectures before attending the lab session:

- Process Management Fundamentals
- Understanding System Monitoring Tools
- Process States and Lifecycle
- Operating System Structure

Lab and Coursework Activity: Process Management and Security Implementation

Coursework Phase: Initial System Configuration & Security Implementation

What you will do today:

- Learn to monitor and manage processes using command-line tools
- Configure SSH with key-based authentication on your server
- Implement firewall rules to restrict access
- Create and manage users with appropriate privileges
- Document all configurations for your journal with evidence

Part 1: Process Fundamentals and Management

Task 1.1: Exploring Process States

1. Open a terminal on your Linux system
2. View all running processes:

   ```
   ps aux
   ```

3. Identify the following columns:
   - USER: which user owns the process

- PID: process identification number
- %CPU: CPU usage percentage
- %MEM: memory usage percentage
- STAT: process state
- COMMAND: the command that started the process

4. Compare different process listing formats:

```
ps -ef
```

5. View real-time process monitoring:

```
top
```

Observe CPU and memory usage, press 'q' to quit

6. If available, install and use htop for enhanced monitoring:

```
sudo apt install htop
htop
```

For your journal: Take screenshots showing process listings with visible command prompts. Explain what the different process states (R, S, D, Z, T) represent.

Task 1.2: Process Relationships and Control

1. View process hierarchy:

```
pstree
pstree -p
```

2. Start a background process:

```
sleep 300 &
jobs
```

3. Practice process control:

```
sleep 500
```

Press `Ctrl+Z` to suspend, then:

```
jobs
bg
fg
```

4. Practice process termination:

```
sleep 600 &
kill [PID]

sleep 600 &
sleep 600 &
killall sleep
```

5. Experiment with process priority:

   ```
   nice -n 10 sleep 400 &
   top
   ```

   Observe the NI (nice) value

For your journal: Document the process lifecycle with examples. Explain when you would use foreground vs background processes, and the difference between `kill` and `kill -9`.

Part 2: SSH Key-Based Authentication Setup

Task 2.1: Generating SSH Keys

1. On your workstation, generate an SSH key pair:

   ```
   ssh-keygen -t ed25519 -C "your_email@example.com"
   ```

2. Accept the default location when prompted
3. Set a secure passphrase
4. View your public key:

   ```
   cat ~/.ssh/id_ed25519.pub
   ```

For your journal: Take a screenshot showing key generation. Explain why ed25519 is recommended over RSA for new keys.

Task 2.2: Copying Key to Server

1. Copy your public key to the server:

   ```
   ssh-copy-id username@server_ip
   ```

   Enter your password when prompted

2. Test passwordless login:

   ```
   ssh username@server_ip
   ```

3. Verify you connected without entering a password

For your journal: Screenshot showing successful passwordless SSH connection with visible command prompt showing username@hostname on both systems.

Task 2.3: Hardening SSH Configuration

1. On the server (via SSH), backup the original configuration:

   ```
   sudo cp /etc/ssh/sshd_config /etc/ssh/sshd_config.backup
   ```

2. Edit the SSH configuration:

   ```
   sudo nano /etc/ssh/sshd_config
   ```

3. Find and modify these lines (remove # if commented):

   ```
   PasswordAuthentication no
   PubkeyAuthentication yes
   PermitRootLogin no
   ```

4. Save the file (Ctrl+O, Enter, Ctrl+X)

5. Restart SSH service:

   ```
   sudo systemctl restart sshd
   ```

6. From a different terminal on your workstation, verify password authentication is disabled

For your journal: Include before and after screenshots of the sshd_config file showing the critical security changes. Explain why each setting improves security.

Part 3: Firewall Configuration and User Management

Task 3.1: Implementing Firewall Rules

1. Check if UFW is installed:

   ```
   sudo ufw status
   ```

2. If not installed:

   ```
   sudo apt update
   sudo apt install ufw
   ```

3. Set default policies:

   ```
   sudo ufw default deny incoming
   sudo ufw default allow outgoing
   ```

4. Allow SSH from your workstation only (replace with your workstation IP):

   ```
   sudo ufw allow from workstation_ip to any port 22
   ```

5. Enable the firewall:

   ```
   sudo ufw enable
   ```

Confirm when prompted

6. Verify the rules:

```
sudo ufw status numbered
sudo ufw status verbose
```

7. Test SSH connection from your workstation to confirm access still works

For your journal: Document your complete firewall ruleset with screenshots. Create a table showing each rule, its purpose, and security justification.

Task 3.2: User and Privilege Management

1. Create a non-root administrative user:

```
sudo adduser adminuser
```

Set a strong password when prompted

2. Add the user to the sudo group:

```
sudo usermod -aG sudo adminuser
```

3. Verify group membership:

```
groups adminuser
id adminuser
```

4. Test sudo access:

```
su - adminuser
sudo apt update
whoami
```

5. List all users with sudo privileges:

```
getent group sudo
```

For your journal: Document the user creation process with screenshots. Explain the principle of least privilege and why using a non-root administrative user is important.

Task 3.3: Remote Administration Evidence

1. From your workstation, execute various administrative commands via SSH:

```
ssh username@server_ip 'uname -a'
ssh username@server_ip 'free -h'
ssh username@server_ip 'df -h'
ssh username@server_ip 'sudo ufw status'
ssh username@server_ip 'systemctl status sshd'
```

2. Demonstrate interactive SSH session:

```
ssh username@server_ip
pwd
hostname
ip addr show
exit
```

For your journal: Screenshots demonstrating that all server administration is performed via SSH from the workstation, not from the server console.

Journal Entry Requirements for This Week

Your journal entry must include:

Process Management Section:

- Screenshots of `ps aux`, `top`, and `pstree` with explanations
- Examples of foreground/background process control
- Explanation of process states and signals
- Reflection on process management concepts

SSH Configuration Section:

- SSH key generation evidence
- Before and after comparison of sshd_config file
- Screenshots showing successful passwordless authentication
- Explanation of security improvements

Firewall Configuration Section:

- Complete firewall ruleset with `sudo ufw status numbered`
- Table documenting each rule and its justification
- Evidence that SSH is accessible only from your workstation
- Discussion of defence-in-depth strategy

User Management Section:

- User creation and privilege assignment process
- List of users with sudo privileges
- Explanation of principle of least privilege

Remote Administration Section:

- Multiple screenshots showing commands executed via SSH
- Evidence of workstation-to-server architecture
- Command prompts must show username@hostname for both systems

Reflection:

- Challenges encountered and how you resolved them
- Security trade-offs you considered
- Connection between theory (lectures) and practice (lab work)

Technical Requirements:

- All screenshots must show visible command prompts with username@hostname
- Include the command, the output, and your explanation
- Update your system architecture diagram to show security controls
- Commit and push to GitHub: `git add . && git commit -m "Security implementation" && git push`

Moodle Resources

Video Lectures:

- Process Management Fundamentals
- Understanding System Monitoring Tools
- Process States and Lifecycle
- Operating System Structure
