University of Roehampton London

ASSESSMENT BRIEF

Programme	BSc Computer Science BSc Computer Science (with Foundation Year) BEng Software Engineering BEng Software Engineering (with Foundation Year) BSc Cyber Security BSc Cyber Security (with Foundation Year)
Module Title	Operating Systems
Module Code	CMPN202
Module Level	Undergraduate Level 5
Assessment Type(s)	**Technical Journal (50% of coursework mark)** Published on GitHub Pages documenting your work across all 7 weeks. **Recorded Demonstration (50% of coursework mark)**. An audio and video demonstration of your practical implementation in this coursework. Max 8 minutes
Word Length / Duration	**GitHub Journal:** Max 20 pages Recorded Video Demonstration (Max 8 minutes)
% contribution to module mark	50%
Deadline (date & time) for Submission	**Standard submission:** Week Commencing 15th December 2025 by 14:00 hours **SOA submission:** Week Commencing 12th January 2026 by 14:00 hours
Format/Location of submission	**Required Files to be uploaded to Moodle** 1. *StudentID_GitHub_URL.txt* (Containing your GitHub Pages URL to your weekly journal). 2. *StudentID_OSCoursework_Demonstration.mp4* (Maximum duration: 8 minutes)
Assessment Feedback date:	Week Commencing 2nd February 2026
Learning Outcomes This assessment has been designed to provide you with an opportunity to demonstrate your achievement of the learning outcomes listed below. By successfully completing this assessment, you will be able to…	**LO3:** Assess security vulnerabilities and threats within operating systems and apply appropriate security mechanisms and best practices to protect computer systems from potential exploits. **LO4:** Demonstrate proficiency in using command-line tools and system utilities to perform computer management tasks, including process monitoring, file

	system manipulation, and system configuration. **LO5:** Critically evaluate the limitations and trade-offs inherent in operating system design, synthesising knowledge of hardware constraints, performance considerations, and security requirements to understand the computer as an integrated system.
Employability and Professional Skills This assessment has been designed to provide you with an opportunity to demonstrate your achievement of the following skill(s) which is(are) critical in professional context and jobs. By successfully completing this assessment, you will be able to …	• Deploy and manage a basic Linux server infrastructure • Configure and harden remote systems using SSH and command-line tools. • Develop automation scripts for security verification and system monitoring • Analyse and document technical trade-offs using quantitative data • Communicate complex technical implementations through written reports and recorded demonstrations • Implement industry-standard security controls (firewalls, key-based authentication, mandatory access control, intrusion detection) • Troubleshoot and resolve technical problems in networked environments • Create professional technical documentation with architecture diagrams and performance visualisations

Assessment Requirements

Introduction

This coursework assesses your ability to configure, secure, and evaluate the performance of a Linux operating system. You will implement security controls, develop command-line proficiency through system administration tasks, and critically analyse operating system behaviour under different workloads.

Data centres consume approximately 1% of global electricity, with projections reaching 3-8% by 2030 [1]. Optimised OS configurations can reduce server energy consumption by 15-30% through improved resource utilisation [2]. Understanding how to configure, secure, and evaluate operating systems is fundamental to computing practice.

Assessment Overview

You will configure a Linux server operating system running headless without a graphical interface. All system administration will be performed via command-line interface accessed through SSH from a separate workstation system. This enforces command-line proficiency whilst developing professional remote administration skills.

Two Assessment Components

Technical Journal (50% of the coursework mark). Published on GitHub Pages documenting your work across all 7 weeks. This demonstrates progressive learning, technical implementation, and critical reflection.

Recorded Demonstration (50% of the coursework mark). A video (maximum 8 minutes) demonstrating your work with live command-line demonstration showing your system and explaining your findings.

System Architecture

You will deploy two systems. The server system runs a Linux server distribution such as Ubuntu Server headless without desktop environment. You administer this system only via SSH using command-line tools. The workstation system provides your administrative access point. You may use a second Linux Desktop VM, your host machine with SSH client, or a hybrid approach.

The dual-system architecture is a pedagogical constraint ensuring command-line proficiency. Your learning centres on operating system configuration, security implementation, and performance analysis rather than infrastructure design.

Detailed system setup instructions are provided in Document 2: Technical Setup Guide.

Assessment Phases

Phase 1: System Planning and Distribution Selection (Week 1)

Plan your operating system deployment and justify technical decisions.

Deliverables (for your Journal):

1. Create a System Architecture Diagram showing both systems and network connections
2. Distribution Selection Justification comparing your chosen server distribution with alternatives
3. Workstation configuration decision justifying your choice of workstation option
4. Network configuration documentation covering VirtualBox settings and IP addressing
5. Using a CLI, document system specifications using `**uname**`, `**free**`, `**df -h**`, `**ip addr**`, and `**lsb_release**`

Phase 2: Security Planning and Testing Methodology (Week 2)

Design a security baseline and performance testing methodology.

Deliverables (Journal):

1. Create a performance testing plan describing your remote monitoring methodology and testing approach
2. Security Configuration Checklist covering SSH hardening, firewall configuration, mandatory access control, automatic updates, user privilege management, and network security
3. Threat Model identifying at least 3 specific security threats with mitigation strategies

Phase 3: Application Selection for Performance Testing (Week 3)

Select applications representing different workload types for performance evaluation.

Deliverables (Journal):

1. Select applications representing different workload types (e.g. CPU-intensive, RAM-intensive, I/O-intensive, Network-intensive, and Server applications such as game servers) for performance evaluation and create an Application Selection Matrix listing applications with justifications for choosing them.
2. Installation Documentation with exact commands for SSH-based installation
3. Expected Resource Profiles documenting anticipated resource usage
4. Monitoring Strategy explaining measurement approach for each application

Phase 4: Initial System Configuration & Security Implementation (Week 4)

Deploy your server and implement foundational security controls.

Deliverables (Journal and Video):

1. Configure SSH with key-based authentication
2. Configure a firewall permitting SSH from one specific workstation only
3. Manage users and implement privilege management, creating a non-root administrative user.
4. SSH Access Evidence showing successful connection screenshots
5. Configuration Files with before and after comparisons
6. Firewall Documentation showing complete ruleset
7. Remote Administration Evidence demonstrating commands executed via SSH

Administrative Constraint: All server configurations must be performed via SSH from your workstation.

Phase 5: Advanced Security and Monitoring Infrastructure (Week 5)

Implement advanced security controls and develop monitoring capabilities.

Deliverables (Journal and Video):

1. Implement Access Control using SELinux or AppArmor, with documentation showing how to track and report on access control settings.

2. Configure automatic security updates with evidence of implementation

3. Configure fail2ban for enhanced intrusion detection

4. Create a security baseline verification script (`security-baseline.sh`) that runs on the server (executed via SSH) and verifies all security configurations from Phases 4 and 5.

5. Create a remote monitoring script (`monitor-server.sh`) that runs on your workstation, connects via SSH, and collects performance metrics from the server.

Note:
All scripts should include line-by-line comments explaining their functionality. All implementations must be demonstrated in your video with live command execution and explanation.

Phase 6: Performance Evaluation and Analysis (Week 6)

Execute detailed performance testing and analyse operating system behaviour under different workloads.

Testing Methodology:
For each application or service, you have chosen, select, monitor and compare the following, where appropriate:

1. CPU usage
2. Memory usage
3. Disk I/O performance
4. Network performance
5. System latency
6. Service response times.

Testing Scenarios:
For each application or service, you have chosen, select, monitor and compare the following, where appropriate:

- Baseline performance testing
- Application load testing
- Performance analysis identifying bottlenecks
- Optimisation testing. Aim to implement and evidence at least two improvements.

Deliverables (Journal and Video):
1. Document your approach
2. Create a performance data table with structured measurements for all applications and metrics
3. Create performance visualisations including charts and graphs
4. Capture testing evidence
5. Conduct network performance analysis documenting latency and throughput
6. Capture optimisation analysis results describing improvements with quantitative data

Phase 7: Security Audit and System Evaluation (Week 7)

Conduct a security audit and evaluate overall system configuration.

Mandatory Audit Tasks: Security scanning with Lynis, network security assessment with nmap, access control verification, service audit justifying all running services, and system configuration review.

Deliverables (Journal and Video):

1. Security Audit Report covering infrastructure security assessment, Lynis scores before and after remediation, network security testing results, SSH security verification, service inventory with justifications, and remaining risk assessment

---

Journal Requirements

Your journal must be published on GitHub Pages. Document your work across all 7 weeks demonstrating progressive learning, technical implementation, and critical reflection.

Structure

Your site must include a home page with project overview and table of contents, and seven weekly pages or sections with navigation between weeks.

Guidance on Weekly Content Requirements

Each week of your journal should reflect the phase you worked on during that week. For example, this should include system architecture diagrams, network configuration documentation, command-line evidence with screenshots showing terminal work (ensure command prompts show *username@hostname*), monitoring approaches, and learning reflections.

Demonstration Requirements

Record a video demonstration with clear audio voiceover that must explain what each command does, why you are using it, what the output means, and how it relates to operating system behaviour.

Structure

Summary (1-2 minutes): Operating system configuration approach, key findings with supporting data, conclusions about OS behaviour and configuration trade-offs.

System Demonstration (3-4 minutes): Both systems running, network connectivity demonstration, configuration decisions and rationale, SSH connection from workstation to server, remote execution of security baseline script with explanation, remote monitoring commands, network performance demonstration.

Critical Analysis (1-2 minutes): Security versus performance trade-offs encountered, configuration challenges and solutions, performance optimisation results with quantitative data, key learning about operating system behaviour.

Technical Specifications

Use Loom, OBS Studio, or equivalent to create a suitable video format (e.g., mp4, m4v etc.) format. Maximum duration is 8 minutes. A clear audio voiceover is mandatory throughout.

Mandatory Technical Requirements

System Configuration

Server System (Mandatory): Linux server distribution running headless. Accessible only via SSH. Firewall configured and enabled. Key-based authentication with password authentication disabled.

Workstation System (Mandatory): Choose Option A (Linux Desktop VM), Option B (host machine with SSH client), or Option C (hybrid approach). SSH client configured. Monitoring tools installed.

Network (Mandatory): Systems communicate via network. SSH connection demonstrated functioning. Firewall rules control access. Configuration documented.

GitHub Pages (Mandatory): Journal published on public GitHub Pages. Regular commits throughout 7 weeks. All images embedded properly. Navigation functional.

Prohibited Activities

Installing desktop environment on server without justification. Using direct server console for routine administration. Using graphical tools on server for administration. Implementing insecure configurations without justification. Submitting demonstration without audio voiceover. Uploading all journal content at once. Using private GitHub repository.

Required CLI Competencies

By Week 2: File system navigation, file manipulation, file viewing, text editing, help systems.

By Week 4: Additionally, package management, process management, permissions, system information, service management, SSH basics, Git basics.

By Week 7: Additionally advanced text processing, piping and redirection, networking tools, log analysis, scripting with variables, conditionals, loops, and functions.

Detailed command lists are provided in Document 2: Technical Setup Guide.

Submission Requirements

Submission Platform: Moodle Module VLE.

Required Files:

1. *StudentID_GitHub_URL.txt*
   (Containing your GitHub Pages URL to your weekly journal).

2. *Student_OSCoursework_Demonstration.mp4*
   (Maximum duration, 8 minutes)

Academic Integrity

This is an individual assessment. Your submission must be entirely your own work.

Permitted: Consulting official documentation. Using online resources with proper citation. Discussing general approaches with peers. Attending drop-in sessions. Using GitHub Pages themes with attribution.

Prohibited: Sharing scripts or configuration files. Copying commands without understanding. Using AI tools without full understanding and attribution. Submitting work you cannot explain. Plagiarising without citation. Copying repositories. Having others create your work.

Use IEEE referencing style. Include URL and access date. Mark adapted code with attribution. You may be asked to explain any work or reproduce commands.

Marking Criteria

Full marking criteria and rubric are provided in Document 4: Assessment Rubric, available on the module Moodle page.

Assessment Success Guidance

To Achieve High Marks (70-100%)

For the Technical Journal:

- Document your complete learning journey with clear evidence of problem-solving and skill development
- Include comprehensive CLI evidence: clear screenshots with visible command prompts (username@hostname), detailed command explanations, before/after configuration comparisons
- Show progression in your skills from basic to advanced commands across the 7 weeks
- Provide thorough trade-off analysis with quantitative data supporting your decisions
- Demonstrate critical reflection connecting theory to practice
- Update architecture diagrams weekly to show infrastructure evolution

For the Recorded Demonstration:

- Rehearse thoroughly to ensure smooth delivery within 8 minutes
- Use split-screen or seamless switching to show both server and workstation systems
- Execute commands confidently in real-time whilst explaining their purpose and output
- Provide clear audio narration throughout—explain what you're doing and why

- Structure your demonstration logically: executive summary → architecture demo → live CLI → critical analysis
- Be honest about challenges encountered and how you overcame them
- Ensure terminal text is clearly readable (large font, good contrast)

General Excellence Indicators:

- Implement all 5 mandatory security controls correctly with thorough justification
- Develop functional scripts with error handling, help flags, and comprehensive comments
- Use 25+ distinct CLI commands appropriately across the coursework
- Identify and analyse 6+ specific technical trade-offs with quantitative evidence
- Achieve Lynis security score >80 through systematic vulnerability remediation
- Demonstrate clear understanding of professional cloud infrastructure practices

Resources

Essential Documentation:

- Linux distribution documentation (your chosen distro's wiki)
- Man pages (man <command>)
- Linux Command Line Basics
- Security Hardening Guide
- SSH Essentials
- VirtualBox Network Settings

Tools & Support:

- VirtualBox (virtualisation platform)
- Loom or OBS Studio (demonstration recording)
- OverTheWire: Bandit (CLI practice)
- Module VLE resources (reflective writing guide, demonstration recording guide)

This Assessment Links to Key Themes:

Understand and demonstrate how your coursework links to these key themes...

- **Employability:** Develops practical skills in Linux server administration, remote system management, security hardening, scripting, and technical documentation highly valued in cloud/DevOps roles
- **Sustainability:** Addresses resource efficiency and environmental impact of computing infrastructure through headless server deployment and quantified resource optimisation

- **Professional Practice:** Mirrors industry-standard dual-system architecture used by cloud hosting providers, developing authentic workplace competencies

Assessment Guidance Support and Formative Feedback

Formative Feedback Opportunities:
Attend timetabled sessions for direct support regarding VirtualBox setup, SSH configuration, security implementation, scripting, or any aspect of the coursework, including:

- Sessions where you can discuss your progress and receive guidance
- Peer discussions of general approaches (whilst maintaining individual work)
- Self-assessment using the detailed rubrics provided in the coursework document
- Weekly milestone checklist to track your progress

Moodle Resources:

- Week-by-week guidance materials
- VirtualBox networking configuration guide
- Reflective journal writing guide
- Demonstration recording tutorial
- Example architecture diagrams
- CLI command reference sheets
- Security hardening checklists

Additional Support:

- Module VLE discussion forums for general technical questions
- Library support for referencing and citation

Who you can contact for further information or queries

Module Tutor: Dr Shabih Fatima
Email: shabih.fatima@roehampton.ac.uk

Module Tutor: Tanaya Bowade
Email: Tanaya.Bowade@roehampton.ac.uk

Module Leader: Dr Ogechukwu Okonor
Email: Ogechukwu.Okonor@roehampton.ac.uk
Office: DB414

Ethical Requirements

This assessment involves deploying virtual machines on your personal computer and testing security configurations in an isolated VirtualBox environment. The following ethical guidelines apply:

Network Security Testing:

- You may use nmap and similar security testing tools ONLY within your isolated VirtualBox network for educational purposes
- Never conduct port scanning or security testing on university networks, external systems, or any system you don't own

- All security testing must remain within your local VirtualBox host-only network

Data Protection:

- No personal or sensitive data should be processed in your VMs
- Screenshots and documentation should not contain personal information beyond your student ID
- If you accidentally include personal information, redact it before submission

Responsible Disclosure:

- If you discover security vulnerabilities in software during this coursework, do not exploit them beyond your local test environment
- Report any significant security issues to your module leader

Acceptable Use:

- Follow university Acceptable Use Policy for computing resources
- Use VirtualBox and Linux distributions in accordance with their licences
- Ensure your coursework activities do not impact network performance or security for others

Use of Artificial Intelligence (AI)

The assessment is designed so that the use of AI during the assessment is possible. You must acknowledge any use of AI and appropriately cite all AI generated outputs. Please make sure you read and understand the assessment guidelines and ask your Module Leader if you have any questions. You can find the guidance on the use of AI https://library.roehampton.ac.uk/studyskills/usingai.

Referencing

Use IEEE referencing style. Include a URL and access date. Mark adapted code with attribution. You may be asked to explain any work or reproduce commands.

Mitigating circumstances/late penalties

Sometimes circumstances outside of your control may affect your studies and might prevent you from submitting work on time or attending an exam. The University offers the ability for students to request additional time to complete an assessment or to defer an examination to a later date. If you are finding yourself in such a situation, please speak to your Academic Guidance Tutor, the Roehampton Student Union (RSU) or someone in the **Wellbeing team** first, who can support you. Further details can be found on the mitigating circumstances portal.

If you do not apply for or are not approved for Mitigating Circumstances, late penalties will apply. If work is submitted up to 14 days late, the mark will be capped at 40%; if it is over 14 days late, it will not be marked.

Resubmissions and Reassessment

If you are required to resubmit this assessment or take part in reassessment, you will be notified via Moodle and your student email. Please ensure you check both regularly. Any reassessment tasks will follow the same learning outcomes and criteria.

Submission Checklist

Before you submit, ask yourself:

- Have I fully answered the assessment brief?
- Have I met the formatting requirements?
- Is my referencing complete and accurate?
- Have I declared any AI use honestly?
- Have I proofread my work?
- Am I submitting through the correct platform before the deadline?

Coursework References

[1] E. Masanet, A. Shehabi, N. Lei, S. Smith, and J. Koomey, "Recalibrating global data center energy-use estimates," *Science*, vol. 367, no. 6481, pp. 984-986, 2020. [Online]. Available: https://www.science.org/doi/10.1126/science.aba3758

[2] J. Koomey, S. Berard, M. Sanchez, and H. Wong, "Implications of historical trends in the electrical efficiency of computing," *IEEE Annals of the History of Computing*, vol. 33, no. 3, pp. 46-54, 2011. [Online]. Available: https://ieeexplore.ieee.org/document/5440129