Milestone 3: Progress Report
(Due November 21st) (50 points)

The progress report document (at least 2 pages, font Times New Roman size 12) must have the following format:

1 Project Overview (∼0.5 pages)

- **Project Title and Team Members**
- **Project Type:** Specify whether your final deliverable will be:
  - Option A: Paper only (5 pages max)
  - Option B: Program only
  - Option C: Combination (3-page paper + program)
- **Brief Description:** 2-3 sentences summarizing your OS security project
- **Security Problem Addressed:** What specific security issue, vulnerability, or threat does your project tackle?

2 Current Progress (∼1 page)

2.1 Completed Work

- **Research/Literature Review:** What papers, documentation, or existing tools have you studied?
- **Design/Code/Prototype/Writing Completed:** Quantify your progress
  - For program: What architectural or implementation choices have you made?
  - For program: What components are functional? Lines of code written? Any prototype?
  - For papers: What sections are drafted? How many pages have been completed?
  - For both: Any initial security tests or experiments conducted?

2.2 Technical Details

- **Tools and Technologies:** What programming languages, libraries, frameworks, or OS features are you using?
  - Examples: C/Linux kernel modules, Python/FUSE, SELinux policies, cryptographic libraries
- **Development Environment:** OS version, virtualization setup, testing environment
- **Key Security Mechanisms:** What OS security primitives or techniques are you implementing/analyzing?
  - Examples: Access control lists, capability-based security, encryption, sandboxing, audit logging

2.3 Challenges Encountered

- **Technical Challenges:** What unexpected difficulties have you faced?
  - Examples: Kernel API changes, debugging kernel panics, understanding SELinux policies, cryptographic implementation issues
- **Security Challenges:** Any difficulties in threat modeling, secure design, or vulnerability testing?
- **Resource Limitations:** Issues with hardware, software access, or documentation?
- **Solutions/Workarounds:** How did you address these challenges? What approaches worked or didn't work?

3 Remaining Work and Timeline (0.5 pages)

3.1 Program

- Features/modules yet to implement
- Security testing planned (penetration testing, fuzzing, vulnerability analysis)
- Code cleanup and documentation
- Performance benchmarking

3.2 Paper

- Sections to write (e.g., security analysis, evaluation, related work)
- Experiments or measurements needed
- Figures, diagrams, or tables to create
- References to add

3.3 Combo

- Discuss both paper and program tasks separately

Submission: Upload your report on Moodle (1 per team) and pload your sources (papers, articles, documents, ...) to GitHub. If you write any code, the code must be committed and pushed to GitHub as well.