**ROUGH PLAN FOR PROJECT**

**Month 1: Familiarization**

1. **Understand CHERI Architecture:**
   * Read the CHERI technical reports and white papers.
   * Study CHERI's capabilities for memory safety and protection.
   * Set up a CHERI-compatible development environment (e.g., QEMU, Morello board, or simulators).
   * Experiment with simple programs to see how CHERI capabilities work.
2. **Understand AFL:**
   * Learn how AFL works, including instrumentation, mutation strategies, and coverage-guided fuzzing.
   * Practice fuzzing basic programs to get familiar with AFL.

**Month 2: Define Objectives and Setup Environment**

1. **Refine Objectives:**
   * Clearly outline what you want to achieve with CHERI and AFL. For example, test how CHERI's bounds-checking impacts memory vulnerabilities like buffer overflows.
   * Define key research questions (e.g., "How does CHERI mitigate memory vulnerabilities compared to traditional architectures?").
2. **Environment Setup:**
   * Configure AFL to work in a CHERI environment (may require porting or adjustments).
   * Ensure the toolchain for CHERI supports AFL instrumentation (e.g., LLVM-based toolchain).

**Month 3: Experimentation with Simple Programs**

1. **Design Simple Test Cases:**
   * Write small programs with intentional memory issues (e.g., buffer overflows, use-after-free).
   * Instrument these programs for AFL fuzzing.
2. **Run Initial Tests:**
   * Observe how AFL identifies issues in a traditional environment.
   * Compare results when running on CHERI, focusing on how vulnerabilities are prevented or handled.

**Months 4-5: Develop Advanced Test Suite**

1. **Expand Test Cases:**
   * Develop more complex programs mimicking real-world scenarios.
   * Include various types of memory vulnerabilities (stack/heap overflows, dangling pointers, etc.).
2. **Integrate with CHERI:**
   * Instrument these programs to utilize CHERI capabilities.
   * Test AFL with and without CHERI, analyzing performance and bug detection.

**Month 6: Analyze Results**

1. **Data Collection:**
   * Document differences in vulnerabilities detected on traditional vs. CHERI architectures.
   * Measure the performance of AFL with CHERI-enabled programs (execution time, coverage).
2. **Preliminary Conclusions:**
   * Identify patterns or strengths/weaknesses in CHERI's memory protection.
   * Note any limitations in applying AFL to the CHERI architecture.

**Months 7-8: Refine and Expand**

1. **Enhance AFL for CHERI (if needed):**
   * Modify AFL to better support CHERI-specific features (e.g., handling capabilities in the input mutation phase).
   * Test new functionality on CHERI-enabled binaries.
2. **Expand Test Scope:**
   * Include third-party open-source software, if feasible, to test in real-world scenarios.
   * Compare results to baseline tests on traditional architectures.

**Month 9: Document Findings**

1. **Write Report:**
   * Summarize all results, including methodology, experiments, and key findings.
   * Highlight how CHERI architecture impacts memory protection and AFL's effectiveness in this context.
2. **Prepare Presentation:**
   * Create slides and visualizations of your results.
   * Practice explaining your work clearly and concisely.

**Month 10: Finalize and Submit**

1. **Polish Deliverables:**
   * Review and refine your report, ensuring it meets all university requirements.
   * Finalize the presentation and prepare for any viva or oral defense.
2. **Submission:**
   * Submit your project and practice explaining your findings to peers and supervisors.