Internship Overview

- Objective: Apply eBPF in Android for system monitoring, profiling, and security.
- Key Technologies: eBPF, Android Kernel, BCC, libbpf, bpftool
- Final Goal: Deploy eBPF programs to trace syscalls, monitor performance, and enhance observability.

What is eBPF and Why It Matters

- eBPF is a virtual machine in the Linux kernel to run sandboxed programs.
- Benefits: Low overhead, no need to modify kernel source code.
- Use Cases: Trace syscalls, monitor app/network behavior, profile CPU and memory usage.

Initial Work and Setup

- Studying eBPF basics, BPF maps, and hook points.
- Explored BCC and libbpf examples.
- Setting up Android kernel build environment.
- Focusing on attaching BPF programs to kernel hooks and writing minimal tracing tools.

Planned Milestones

- Week 1: Study eBPF basics and Android kernel structure.
- Week 2: Set up development environment with BCC/libbpf.
- Week 3: Deploy sample BPF programs (trace syscalls/network).
- Week 4: Analyze data and demonstrate observability tools.

Challenges and Learning Goals

- Challenges: Cross-compiling for Android, kernel version issues, SELinux restrictions.
- Learning Goals: Write eBPF programs, use BCC/libbpf/bpftool, understand kernel internals via tracing.