FELIX ELADI

Nairobi, Kenya | +254 795 478 854 | felixeladi877@gmail.com

Portfolio | GitHub | LinkedIn

PROFESSIONAL SUMMARY

Systems Software Engineer focused on the hardware–software interface, operating system internals, firmware architecture, and low-level networking. Transitioned from product-level development into systems programming with a first-principles understanding of the full execution stack—from pre-boot firmware through kernel and user space. Hands-on experience with UEFI-level security experimentation, production environment hardening, and building low-level abstractions for security-critical systems.

TECHNICAL CORE

- **Systems Programming:**
  C, C++, x86-64 Assembly, UEFI EDK II, POSIX Threads, IPC, Memory Mapping.

- **Kernel & OS:**
  Linux Kernel Hardening (SELinux/AppArmor), System Call Interception, IRQs, xv6.

- **Networking & Sockets:**
  TCP/UDP Socket Programming, Raw Sockets, Protocol Design (L4-L7), Wireshark.

- **Security & Reliability:**
  Zero Trust Architecture, RBAC, SRE Principles (SLIs/SLOs), Cryptographic Primitives.

PROFESSIONAL EXPERIENCE

Systems Reliability Intern | *Business Systems Architect (BSA)* | **June 2025 – Sept 2025**

- **Distributed Monitoring:** Engineered deployment strategies for enterprise IoT monitoring (AKCP), optimizing data ingestion pipelines to maintain **99.9% availability** across distributed nodes.
- **Infrastructure Automation:** Developed Python-based automated health checks and observability pipelines, reducing manual intervention by **30%** and improving Mean Time To Recovery (**MTTR**).
- **Access Control:** Architected and enforced **Least Privilege (PoLP)** and RBAC policies at the OS level for production servers, mitigating lateral movement risks without impacting system throughput.

Systems & Security Intern | *Daystar University Data Center* | **Aug 2021 – Jan 2022**

- **Endpoint Hardening:** Audited and hardened **250+ heterogeneous endpoints**; implemented GPOs and shell scripts to enforce security baselines, reducing unauthorized access vectors by **15%**.
- **Network Resilience:** Diagnosed and resolved L2/L3 bottlenecks in campus-wide infrastructure, utilizing packet analysis to optimize routing tables and achieving a **20% increase in network uptime**.
- **VLAN Isolation:** Scaled and maintained a 100+ endpoint IP-surveillance network, implementing network-level segmentation to isolate sensitive traffic from general campus broadcast domains.

DEEP-TECHNICAL PROJECTS

I. Low-Level & Compilers

Regex Engine & Lexical Analyzer | *C++, Python*

- Implemented **Thompson's Construction Algorithm** to transform regular expressions into Non-deterministic Finite Automata (NFA).
- Engineered a subset construction algorithm to convert NFAs into **Deterministic Finite Automata (DFA)**, optimizing for **$O(n)$ time complexity** relative to input length.

- Managed manual memory allocation in C++ to handle state-transition tables, ensuring deterministic behavior in resource-constrained environments.

UEFI Firmware Security Suite | *C, EDK II*

- Developing pre-boot security protocols using **EDK II** to intercept the boot process before the OS loader (GRUB/Windows Boot Manager) initializes.
- Researching **Runtime Services** and **Boot Services** to implement hardware-level lockdown and cryptographic integrity checks for sensitive device environments.

II. Systems Engineering

Secure Socket Management Server | *C#, Network Sockets*

- Designed a multi-threaded server utilizing **Asynchronous Sockets** to manage concurrent remote administrative sessions.
- Implemented a custom application-layer protocol with end-to-end encryption to facilitate secure remote command execution and file system telemetry.

III. Product Engineering

Job Portal & Magazine Platforms | *Laravel, Flutter, Docker*

- Built full-stack and mobile systems with role-based access control and real-time workflows
- Applied Clean Architecture (MVVM) to manage **state consistency, separation of concerns, and long-term maintainability**
- Treated product systems as **distributed systems**, reasoning about latency, concurrency, and failure at the application layer

EDUCATION & RESEARCH

- **BSc. Computer Science** | Daystar University | *Awaiting Graduation*
- **Self-Directed Research:**
    - Operating System Internals (xv6)
    - Socket Programming & Network Protocols
    - Compiler Design & Formal Languages

    - Firmware & Boot Process Architecture
    - Reading and implementation-based learning using **Linux source code, xv6, and RFC documentation**

- **Certifications:** Zero Trust Architecture, Vulnerability Analysis, Computer Forensics, Flutter MVVM.