Lachlan Cox

Software Developer

lcox74@outlook.com | lcox74 | lachlancox.dev | 0421 398 988

Software developer with 6+ years across telecommunications, intelligent transport systems, industrial automation, and self-hosted enterprise infrastructure. Cross-stack from hardware reverse engineering and network protocol code through to cloud deployment and web tooling.

Education

University of Queensland — Brisbane, Australia
Bachelor of Computer Science — *2019 – 2023*

Experience

Senior Software Developer & Security Officer (Full-time Contract) — Nov 2021 – Present
Tillered — *Melbourne, Australia*

- Served as Director from Mar 2022 – Apr 2026.
- Wrote the Go network-acceleration core, including an automated QoS layer that tunes and adapts congestion-control algorithms to link conditions.
- Rebuilt the product as a decentralised, airgapped system using CRDTs and Protobufs, configurable through a custom infrastructure-as-code (IaC) format that the cluster reconciles toward the desired state. Validated in clusters of up to 100 nodes with ephemeral inter-node tunnels.
- Built a custom WireGuard implementation that adds layer-2 bridging on top of the natively layer-3 protocol.
- Reimplemented host firewall and routing using netlink and nftables, with continuous reconciliation between actual and desired state.
- Designed a cryptographic licensing system with key rotation and backwards compatibility, and a custom offline-capable OAuth provider with OpenID Connect compatibility and 2FA used by the cluster frontends.
- Built a custom AlmaLinux image pipeline (stripped minimal base, kickstart-driven first boot) that produces fully provisioned nodes with no operator interaction.
- Built the Terraform deployment for the cloud-hosted product on AWS (EKS, RDS, multi-AZ), with automated patching, database upgrades, restrictive security groups, and ephemeral nano-EC2 Tailscale subnet routers in place of a jump box.
- Maintain user documentation, the status page, infrastructure docs, and the company website and design system; pulled into client PoCs that often require reverse-engineering customer networks from email threads.

Software Engineer (Casual) — Jul 2023 – Jul 2024
Hyperion Technology — *Brisbane, Australia*

- Reverse-engineered low-level sensor hardware where vendor APIs were limited, writing custom software to drive the device and process raw sensor data. Re-used across multiple projects.
- Integrated third-party power-monitoring hardware via Modbus TCP, replacing a cloud-dependent integration with direct local collection. Cut outbound data on slow links and removed a separate authentication surface from the web app.
- Built the backend for a multi-site roadside deployment of sensors, cameras, and signage over poor comms: per-site agents managed local state and forwarded telemetry to a central collector, which also pushed commands back for content updates and on-demand camera pulls.
- Prototyped a machine-vision pipeline for vehicle classification from roadside cameras, fed by a multi-sensor gantry; built a Raylib-based image-labelling tool for training data and adapted the pipeline to multiple deployment sites with different camera configurations.

Software Developer (Part-time) Apr 2019 – Nov 2022

National Advice Solutions *Brisbane, Australia*

- Migrated thousands of records from multiple legacy CRMs into Salesforce across several migration passes, with an automated validation system to confirm no record loss or corruption between source and target.
- Designed a custom client lifecycle pipeline in Salesforce with advisers and paraplanners; built it out using custom objects, relationships, and Lightning Web Components iterated to staff feedback.
- Built a React + Flask internal tool for grouping client documents into review-and-sign bundles, with programmatic PDF signing in Python and a full audit log; used by staff for several months.

Automation Engineer (Casual) Jun 2019 – Mar 2022

Interactive *Brisbane, Australia*

- Sole developer on an internal management portal for technical sales staff (C#.NET backend, React.js frontend, Microsoft auth), wrapping Nokia's NFM-P API via REST and SOAP.
- Exposed client management and provisioning of layer-2 Epipe and layer-3 VPRN services through the portal, replacing manual NFM-P workflows.

Skills

- **Languages:** Go, TypeScript/JavaScript, Python, C/C++, C#, Lua, x86_64 assembly.
- **Networking:** TCP/UDP, NAT traversal and P2P, protocol design and debugging, IP/MPLS provisioning.
- **Systems & embedded:** Linux internals, kernel-adjacent Go, hardware reverse engineering, embedded device integration.
- **Security:** OAuth / OpenID Connect, 2FA, cryptographic key management.
- **Web:** Vue, React, Node.js, HTML5, Salesforce / Lightning Web Components, REST API design.
- **Infrastructure:** Self-hosted Proxmox / LXC, Docker, AWS (EKS, RDS, EC2), Terraform, Tailscale, Git.

Projects

Driftwood | *Go, Lua, Docker*

- Modular Discord bot framework that exposes Lua bindings on top of Go, letting users build commands, components, and event handlers in Lua without touching the Go internals.
- Published as a Docker image; supports interaction components, event scheduling, and persistent state. Powers Bruce, the Aussie BroadWAN community Discord bot, end-to-end.

UDP Hole Punching | *Go, networking, NAT traversal*

- Built a peer-to-peer connection broker using UDP hole punching to establish direct links between hosts behind separate NATs.

Building a Brainfuck Compiler | *Go, x86_64 assembly*

- Wrote a compiler for Brainfuck that emits native x86_64 assembly, covering parsing, IR, and code generation end-to-end.

Booting Go on Bare Metal | *Go, Linux, low-level boot*

- Stripped the runtime and booted a Go binary directly on bare metal without an OS or container.