

Louis Jackman

<https://volatilethink.com>

I'm a security engineer in London who specialises in cybersecurity and general software development. My experience covers appsec, infosec, and netsec; this includes internal pentests, appsec-focused code reviews, maintaining and hardening software-defined networks in the cloud, and helping to maintain an ISMS.

Experience

REST Services	Implemented a passphrase hashing system with runtime switchable algorithms, constant-time operations to avoid timing-attacks, and a monadic validation system to tighten input data to prevent injections.
Compliance	I do in-house pentests and have assured clients of our company's security policies and controls, our persual of ISO27001 and GDPR.
Infrastructure	Container-based infrastructure atop Kubernetes, infrastructure-as-code with Terraform, tooling with Python 3, Ansible for configuration management, and maintaining a highly available infrastructure atop AWS with custom tooling. Enough Go to write small tools and understand the internals of tools like Kubernetes plugins.
IoT-style Devices	Python 3 on Raspberry Pi utilising a serial-over-USB interface, abstracting byte-over-wire serial protocols into pluggable modules to handle different pager systems, handles program updates in-place, and uses AWS SQS to queue pending messages.
Internal Tooling	An internal service for quickly spinning up and testing network services, aiding the agile methodology and working with various branching models. Backed by Java and fronted with a HTML5/JavaScript frontend, and using Docker and Akka actors.

Specific Skills

Secure Coding	Well versed on guidelines offered by OWASP, NIST, and CIS. An understanding of memory-safety, pedantically finding logic-holes, data-tainting, and resource exhaustion attacks against availability.
Pentesting	I keep abreast of CVEs and other general cybersecurity issues. Can effectively

use tools like netcat, nmap, OWASP ZAP, sqlmap, msf3, aircrack-ng, siege, and others.

Java	Java 11 SE and EE, Spring Framework and WebFlux, Java 9's Pub/Sub with Reactor.io, Hibernate/JPA, JUnit 4 & 5 with Jupiter API, JSR305, Maven, JDBC, Google Guice, Apache Commons, and Google Guava.
Python	Python 3, boto3, Linux automation, mailers, HTML/XML/JSON manipulation, Ansible modules. The usual Python environment tools like pip, setuptools, and virtualenvs.
Linux	iptables, SELinux, ACLs, cgroups, namespaces, IDS; security features like W^X stacks and PIE, and post-compromise procedures. Bash, its POSIX shell subset, git, coreutils, vim, KVM/QEMU, and the usual Unix programs.
Operations	AWS experience, both setting things up manually and automating it via Terraform. EC2, VPCs, RDS, SQS, GuardDuty, CloudWatch, CloudTrail, Elasticache, IAM, and most recently EKS. Private VPCs, autoscaling setups, and tight security groups to control inter-network communication. A small amount of Heroku experience too.
Agile Workflow	I'm used to the agile workflow of using issue ticketing systems, providing estimates, writing up postmortems, providing retrospectives, and taking ownership of code from the developer environment to the production server.
Systems Development	C89/C99/C11 and Unix/POSIX APIs. I can do complex data-structures without segfaulting, and write application using lower level APIs like sockets without riddling code with security holes. Rust, which I have used it since university back in the 0.x days. Now using stable versions for personal projects. See projects page for more details.
The Web	Can do basic layouts like this page (knowing how to style both screen and printed PDF-style content), semantic markup with useful metadata, graceful degradation, and the usual build tools (minifiers, transpilers, and analytics). JavaScript (ECMAScript 6), TypeScript, standard browser APIs such as the DOM, React, redux, react-redux, redux-observable, Webpack 4, and various other Node.js-based build tools.