Daniel Lee

daniel@devshell.net
(360)-909-6604
Camas, WA

Technical Expertise

Languages Python, Golang, C#, C/x86 Assembly, SQL, Javascript, HTML/CSS

Technologies Flask, Git, Docker Swarm, .NET Core, MySQL, Redis, Kafka, LaTeX, Bash, Vim

Education

Washington State University Vancouver Vancouver, WA
Bachelor of Science in Computer Science *Expected: May 2021*

Experience

OfficeAlly Vancouver, WA
Web Developer *May 2017 - Present*

- Developed 12 factor apps and services to run in docker swarm.
- Worked in full stack role to quickly develop and demo new projects and technologies.
- Refactored badly written sections to improve overall code health and increase performance.
- Collaborated with developers to create optimal designs and workflows for back-end services.
- Reviewed colleague's code and change lists ensuring code quality and best practices
- Implemented agile and scrum methodology to improve velocity and increase team communication.
- Added comprehensive unit tests and practiced TDD to ensure code stability and reliability.
- Introduced gRPC and the micro-service architecture to support communication between different languages and environments across the company's applications and services.
- Added a caching mechanism to a taxing web-API route which improved performance and decreased load on other services.
- Supported the adoption of Git from SVN by suggesting a series of workflows to ease transition.
- Contributed to discussions and documentation to improve and standardize company git practices such as branch naming, commit message formatting, and re-basing history.
- Discovered and reported critical security vulnerabilities that could cause massive damage to company systems and infrastructure.

Projects

Patient Visualization Tool Web Application used to see concentrations of patients across the United States. Users can filter the view by geolocation, diagnosis codes, and patient age ranges. The client application is written using Vuejs, Leaflet, and Material Design principles. The back-end application is written in Go. The client and server communicate using gRPC.

Distributed Management Tool Software used to experiment with concurrent programming and distributed service design. The stack includes a Command and Control (C2) server, a package download server, and an Advanced Persistent Threat (APT) application. The APT has a tiny footprint and runs management functionality modularly upon request. The APT establishes a communication channel with the C2 and intelligently requests executables from local download servers. The system is also capable of switching to a decentralized C2 system using APTs and a version of the RAFT consensus algorithm.

ESRA CubeSat Internals Software for the internals of a 3u sized experimental cube satellite. Written in Go and designed to manage physical components of the satellite on a simple distributed platform across multiple Raspberry Pi microcomputers. Accompanied by a ground-station application which communicates with the cube satellite over radio. Software was used for an annual competition hosted by the Experimental Sounding Rocket Association.