Omri Bornstein

Software Engineer

Greater Melbourne Area
Australia
omribor@gmail.com
applegamer22.github.io
omri-bornstein
AppleGamer22

Updated on ***2023-01-26***

Education

2020 – In Progress **Bachelor of Computer Science**, *Monash University*, Melbourne

2017 – 2019 **South Australian Certificate of Education**, *Australian Science & Mathematics School (ASMS)*, Adelaide

Skills

Computer Programming Languages

Go	`raker`, `stalk` & `cocainate`	*server-side & command-line interfaces (CLIs)*
TypeScript	scr-cli/scr-web	*full-stack*
Python	FIT2082 research project	*data analysis*
Kotlin/Java	`sp`	*Minecraft plugins*
C/C++	FIT2082 research project	

Document Markup Languages

HTML & CSS	*client-side UI on web browsers*
TEX/LATEX	*PDF document typesetting*
Markdown	*technical documentation and communication*

Tools

Git	open-source projects	*source code version control*
GitHub/GitLab	open-source projects	*collaboration & CI/CD*
MongoDB	`raker` & scr-cli/scr-web	*document non-relational database*
SQL		*relational database querying*
Docker	`raker` & scr-cli/scr-web	*container-style packaging*
Kubernetes		*container orchestration*
Vagrant		*virtual machine (VM) management*
Terraform		*infrastracture definitions as code*

Other

- **Platforms**: Linux, Cloud Native, web servers/browsers, macOS, Windows
- **Soft Skills**: technical writing, critical/analytical thinking, presenting/public speaking, research, troubleshooting/debugging, explaining, collaboration/teamwork

Projects

Open-Source

May 2022 – Present **`raker`**, *`AppleGamer22/raker` on GitHub*
- A social media scraper that is interfaced via a server-side rendered HTML user interface (or a CLI), and is managed by a REST API and a NoSQL database.
- Server-side is built with Go, MongoDB, JSON Web Tokens (JWTs) and Docker.
- Client-side is built with HTML/CSS (Bootstrap).
- The companion CLI utility and configuration are built with Cobra and Viper.

May 2022 – Present **`stalk`**, *`AppleGamer22/stalk` on GitHub*
- A cross-platform file-watcher that can run a command after each file-system operation on a given set of files or simply wait once until a file is changed.
- Built with Go, Cobra and FSnotify.

January 2022 – Present **`cocainate`**, *`AppleGamer22/cocainate` on GitHub*
- A cross-platform re-implementation of the macOS utility `caffeinate` that keeps the screen turned on either until stopped, for a set duration of time or while another process still runs.
- Built with Go and Cobra.

June 2020 – January 2021 **sp**, *AppleGamer22/sp on GitHub*

- My first attempt at building a Minecraft server plugin. This plugin adds the requirement that the player supplies the password (via a server command) before proper server interaction is allowed, and as long as the password isn't provided, the currently-unauthorized player is blinded and immobile.
- Built with Kotlin.

April 2019 – May 2022 **scr-cli/scr-web**, *AppleGamer22/scr-cli on GitHub*/*AppleGamer22/scr-web on GitHub*

- My previous attempt at building a full-stack (and a CLI) social media scraper with a single-page website framework and a RESTful server.
- Server-side is built with TypeScript & Nest (with a Node.js runtime) MongoDB, JSON Web Tokens (JWTs) and Docker.
- Client-side is built with Angular and Ionic.
- The full-stack package is bundled with Nx.
- The CLI is built with OCLIF

Research

August 2021 – December 2021 **Software Contributor**, *Monash University's FIT2082 unit*, Melbourne

- I contributed to an existing codebase, based on prior research by (Gange, Harabor and Stuckey, 2021) about *Lazy CBS*, their Multi-Agent Path Finding (MAPF) algorithm.
  - I modified the *Lazy CBS* codebase such that the algorithm also outputs the final set of constraints that is used to rule out paths, such that *Lazy CBS* is formally an **Explainable** Multi-Agent Path Finding (XMAPF) algorithm.
  - I learned how to enable *Python-to-C++* bindings, such that the compiled *Lazy CBS* codebase can be used as a Python-facing library for future projects.
- Built with C/C++ and Python on top of Linux.

Experience

Volunteering

2023 – Present **President**, *Monash Cyber Security Club (MonSec)*, Melbourne

June 2022 – Present **Vice President**, *Monash Cyber Security Club (MonSec)*, Melbourne

- Coordinated collaboration with the university's Faculty of Information Technology for purposes of events and advertising.
- Club website:
  - Updated the theme to its latest version, and resolved new layout bugs in collaboration with other club committee members.
  - Improved the Kali Linux virtual machine set-up guide such that it includes more details on alternative installation methods.
- Changed our CTFd's theme. In addition, I disabled the theme's default CRT-like flicker effect by editing its CSS files, such that people with epilepsy would not get an inferior experience.
- Organised and ran an introductory-level workshop about steganography (a recording is available on YouTube).
- Club representation:
  - Faculty of IT's Take CTRL (Cryptography & Web Hacking Workshop)
  - Faculty of IT's Munch & Mingle
  - Faculty of IT's open day
  - Orientation week of 2022's 2nd semester.
- Capture the Flag (CTF) participation:
  - The University of Adelaide's CTF
  - SHELL CTF
  - DownUnderCTF

January 2022 – June 2022 **Secretary**, *Monash Cyber Security Club (MonSec)*, Melbourne

- Organised and recorded official committee and club meetings.
- Represented the club during the orientation week of 2022's 1st semester.
- Organised and ran an introductory-level binary reverse-engineering workshop (a recording is available on YouTube).
- Started a section on the resources page of the club's website, with a detailed section with a guide on how to easily install and set-up a Kali Linux virtual machine.

May 2021 – January 2022 **Assistant Member Training Officer**, *Monash Cyber Security Club (MonSec)*, Melbourne

- Helped to organise and ran a workshop about brute-forcing tools used for penetration testing.
- Assisted in the club's management and operations.
- Participated in ångstromCTF.

Freelancing

June 2021 – December 2021 **Software Engineer**, *Contract*, Melbourne

- Implemented a fault-tolerant file back-up system that enables the continuation of file transferring from a variably-approximate point in time before the back-up disruption.
- Built with Go.