Greater Melbourne Area
Australia
omribor@gmail.com
applegamer22.github.io
omri-bornstein
AppleGamer22
Updated on **2022-09-27**

Omri Bornstein

Software Engineer

Education

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

2020 – Present **Bachelor of Computer Science**, *Monash University*, Melbourne

Skills

Computer Programming Languages

Go	Expert	*server-side and command-line interfaces (CLIs)*
TypeScript	Expert	*full-stack*
JavaScript	Expert	*client-side on web browsers*
Python	Proficient	*data analysis and research*
Kotlin/Java	Intermediate	*Minecraft plugins*
C/C++	Intermediate	*research*

Document Markup Languages

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

Tools

Git	Expert	*source code version control (see open-source projects for examples)*
GitHub/GitLab	Expert	*collaboration and CI/CD*
MongoDB	Proficient	*document non-relational database*
SQL	Intermediate	*relational database querying*
Docker	Proficient	*container-style packaging*
Kubernetes	Intermediate	*container orchestration*
Vagrant	Proficient	*virtual machine (VM) management*

Other

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

Leadership Experience

June 2022 – Present **Vice President**, *Monash University's 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.
- 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 University's 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 a binary-level reverse engineering workshop (a recording is available available at `https://youtu.be/893L13SxDUg`).
- 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 **General Representative**, *Monash University's Cyber Security Club (MonSec)*, Melbourne

- Helped to organise and ran a workshop about brute-forcing tools used for penetration testing.
- Participated in ångstromCTF

Projects

Open-Source

May 2022 – Present **`raker`**, ***`https://github.com/AppleGamer22/raker`***

- 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`**, ***`https://github.com/AppleGamer22/stalk`***

- A cross-platform file-watcher that can run a command after each file-system operation on a given files or simply wait once until a file is changed.
- Built with Go, Cobra and FSnotify.

January 2022 – Present **`cocainate`**, ***`https://github.com/AppleGamer22/cocainate`***

- 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.

December 2021 **CTFtime Discord Bot**, ***`https://github.com/monsec/ctftime-discord-bot`***

- A Discord bot for MonSec's Discord server, that fetches statistics about competing Capture the Flag (CTF) teams from CTFtime, and displays them in the Discord interface.
- Built with Go.

June 2020 – January 2021 **sp**, ***`https://github.com/AppleGamer22/sp`***

- 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**, ***`https://github.com/AppleGamer22/scr-cli`*** *&* ***`https://github.com/AppleGamer22/scr-web`***

- 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 packages 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.

Freelancing

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

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