

Omri Bornstein

Software Engineer

Greater Melbourne Area
Australia

✉ omribor@gmail.com
🌐 applegamer22.github.io
in [omri-bornstein](#)
🎮 [AppleGamer22](#)

Updated on **2022-09-27**

Education

2017 **South Australian Certificate of Education**, [Australian Science & Mathematics School](#) (ASMS),
2019 Adelaide
2020 **Bachelor of Computer Science**, [Monash University](#), Melbourne
Present

Skills

Computer Programming Languages

| | |
|-----------------------------|---|
| Go | <i>server-side and command-line interfaces (CLIs)</i> |
| TypeScript | <i>full-stack</i> |
| JavaScript | <i>client-side on web browsers</i> |
| Python | <i>data analysis and research</i> |
| Kotlin/Java | <i>Minecraft plugins</i> |
| C/C++ | <i>research</i> |

Document Markup Languages

| | |
|--|--|
| HTML/CSS | <i>client-side on web browsers</i> |
| T_EX/L^AT_EX | <i>PDF document typesetting</i> |
| Markdown | <i>technical documentation and communication</i> |

Tools

| | |
|-------------------------------|---|
| Git | <i>source code version control</i> |
| GitHub/GitLab | <i>collaboration and CI/CD</i> |
| MongoDB | <i>document non-relational database</i> |
| SQL | <i>relational database querying</i> |
| Docker | <i>container-style packaging</i> |
| Kubernetes | <i>container orchestration</i> |
| Vagrant | <i>virtual machine (VM) management</i> |

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 **Vice President**, [Monash University's Cyber Security Club](#) (MonSec), Melbourne
Present

- 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 **Secretary**, *Monash University's Cyber Security Club (MonSec)*, Melbourne
- June 2022
- Organised and recorded official committee and club meetings.
 - Represented the club during the orientation week of 2022 1st semester.
 - Organised and ran a binary-level reverse engineering workshop (a recording is 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 **General Representative**, *Monash University's Cyber Security Club (MonSec)*, Melbourne
- January 2022
- Helped to organise and ran a workshop about brute-forcing tools used for penetration testing.
 - Participated in [ångstromCTF](#)

Projects

Open-Source

- May 2022 **raker**, <https://github.com/AppleGamer22/raker>
- Present
- 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 **stalk**, <https://github.com/AppleGamer22/stalk>
- Present
- 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 **cocainate**, <https://github.com/AppleGamer22/cocainate>
- Present
- 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 **sp**, <https://github.com/AppleGamer22/sp>
- January 2021
- 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 **scr-cli** & **scr-web**, <https://github.com/AppleGamer22/scr-cli> & <https://github.com/AppleGamer22/scr-web>
- May 2022
- 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 **Software Contributor**, *Monash University's FIT2082 unit*, Melbourne
- December 2021
- 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 **Software Engineer**, *Contract*, Melbourne
- December 2021
- 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](#).