

# Jason Ngo

Computer Science Major @ UBC

+1 587-890-5411 | work@jasonn.dev | github.com/Green-Avocado

## Skills Summary

|                              |  |
|------------------------------|--|
| <b>Application Security</b>  | Buffer overflow, Format-string exploits, Return-oriented programming, Use-after-free               |
| <b>Web Security</b>          | SQL injection, Cross-site scripting, Template injection, Local file inclusion, Prototype pollution |
| <b>Binary Analysis</b>       | Ghidra, Radare2, Binary Ninja, GDB, angr, Triton   |
| <b>Software development</b>  | Rust, x86 Assembly, C / C++, Java, NodeJS, NGINX, Google Firebase                                  |
| <b>System administration</b> | Linux, Docker, Bash, SQL, Google Cloud, Oracle Cloud   |

## Work Experience

|                          |  |
|--------------------------|--|
| <b>2022/09 - Present</b> | <b>Vulnerability Researcher and Software Developer</b><br><i>Government of Canada</i> <ul style="list-style-type: none"> <li>Conducted vulnerability research on networking devices using Ghidra and GDB.</li> <li>Developed software in Rust to parse binary data into a human-readable format.</li> <li>Reverse engineered and created a Python script for interacting with an unknown binary protocol.</li> </ul>   |
| <b>2022/05 - 2022/08</b> | <b>Undergraduate Academic Assistant</b><br><i>University of British Columbia</i> <ul style="list-style-type: none"> <li>Discovered and patched vulnerabilities in services involving arbitrary code execution and XML injection.</li> <li>Used Python to automate student evaluation on a variety of topics using randomized questions.</li> </ul>   |
| <b>2020/04 - 2022/02</b> | <b>Freelance Software Development</b><br><i>Commissioned by clients for various projects. Some examples include:</i> <ul style="list-style-type: none"> <li><u>Transactions database</u> — Designed proof-of-concepts for database solutions using Firebase Realtime Database, MySQL, and Google Drive APIs.</li> <li><u>Mosque timetable</u> — Developed a web application to read data from a CSV file and display prayer times using HTML, CSS, and JavaScript.</li> <li><u>covidping.com</u> — Wrote scripts to load current COVID-19 statistics into Google Sheets and send emails to a list of subscribers for notifying users of COVID-19 statistics in their state.</li> </ul> |

## Technical Extracurriculars

|                          |   |
|--------------------------|---|
| <b>2019/09 - Present</b> | <b>Capture The Flag Competitions</b><br><a href="https://github.com/Green-Avocado/CTF">https://github.com/Green-Avocado/CTF</a> <ul style="list-style-type: none"> <li>Reverse engineered binaries using static and dynamic analysis techniques.</li> <li>Identified vulnerabilities in binary applications and web services.</li> <li>Defeated common exploit mitigations such as PIE, ASLR, canaries, and RELRO.</li> <li>Created writeups to explain vulnerabilities and exploit techniques used in each challenge.</li> </ul>   |
| <b>2017/09 - 2020/02</b> | <b>Vex Robotics Club</b><br><a href="https://github.com/Green-Avocado/3388D-vex-robotics-edr-2020">https://github.com/Green-Avocado/3388D-vex-robotics-edr-2020</a> <ul style="list-style-type: none"> <li>Wrote firmware in C++ which used the Vex API to receive instructions from a controller.</li> <li>Used feedback from sensor data to guide autonomous routines and aid user control.</li> <li>Created a user interface for the controller display screen to configure the robot at runtime.</li> <li>Won an award for most efficient autonomous program and was invited to compete in VEX Worlds.</li> </ul> |

## Hackathon Projects

|                |  |
|----------------|--|
| <b>2022/01</b> | <b>Language Exchange</b><br><a href="https://github.com/Green-Avocado/Language-Exchange">https://github.com/Green-Avocado/Language-Exchange</a> <ul style="list-style-type: none"> <li>In a team of 4, created a website using React and NodeJS for helping language students connect.</li> <li>Used Google Firebase to set up authentication, store user data, match users according to their goals.</li> </ul>   |
| <b>2021/11</b> | <b>Speak-able</b><br><a href="https://devpost.com/software/speak-able-inclusive-unconferencing">https://devpost.com/software/speak-able-inclusive-unconferencing</a> <ul style="list-style-type: none"> <li>In a team of 4, created a website for encouraging inclusivity in participant-driven meetings.</li> <li>Used NodeJS and Express to implement a RESTful API to interact with the webpage, allowing users to submit new topics, vote for existing topics, and view the number of votes for each topic.</li> </ul> |
| <b>2020/08</b> | <b>Study Tinder</b><br><a href="https://devpost.com/software/study-tinder">https://devpost.com/software/study-tinder</a> <ul style="list-style-type: none"> <li>In a team of 2, created a website for helping students connect and study while social distancing.</li> <li>Used Google Firebase to set up hosting, a database, and user authentication.</li> <li>Matched students with complementary interests by using JavaScript to query the Firestore database.</li> </ul>   |

2020/08

**BikePath**<https://devpost.com/software/bikepath-dkpstx>

- In a team of 3, created a website to find locations that encourage eco-friendly alternatives to driving.
- Used the Google Maps API to render a map view of selected locations on the webpage.
- Used Python to interact with the Google Places API to retrieve, parse, and interpret data.

2020/08

**COVID Wait**<https://devpost.com/software/covid-wait>

- In a team of 5, created a website to help avoid populated areas to reduce the risk COVID-19 exposure.
- Used Python to develop a server-side application to retrieve data from the Google Maps API.
- Implemented a RESTful API to allow clients to query data from the server.

## Cybersecurity Projects

---

2022/05 - 2022/07

**No Flag 4 U**<https://github.com/Green-Avocado/No-Flag-4-U>

- Created a dynamic shared library using Rust to hook standard library functions.
- Mitigates common vulnerabilities including buffer overflow, format string, and use-after-free.
- Logs function calls by sending data to an external process using a TCP stream.

2021/03 - 2022/04

**pwndocker**<https://github.com/Green-Avocado/pwndocker>

- Wrote a minimal program in C to create symbolic links without standard libraries.
- Used Docker to create an environment for debugging exploits under different versions of Glibc.
- The project became a go-to tool for CTF challenges involving binary exploitation.

2022/02

**BBY Stealer Malware Analysis**<https://github.com/Green-Avocado/bbystealer-malware-analysis>

- Used Wireshark and Windows filesystem auditing to identify connections and filesystem access.
- Reverse engineered JavaScript code that was obfuscated and packaged as a Windows executable.
- Helped victims with incident response by identifying compromised credentials and modified files.

2021/09 - 2021/12

**EasyROP**<https://github.com/Green-Avocado/EasyROP>

- Wrote a Java program using object-oriented design to automate writing scripts for binary exploitation.
- The project began as a command-line application and later included a GUI using Java Swing.
- Return-oriented programming payloads could be saved as a local JSON file and reloaded.

## Personal Projects

---

2020/12 - 2022/01

**website**<https://github.com/Green-Avocado/website-inactive>

- Used NodeJS with Express to serve web pages which are generated using a templating engine.
- Used NGINX to secure connections using TLS and forward HTTP requests to internal services.
- Managed services using Docker and GitHub continuous integration.

2021/11 - 2022/01

**atom-ide-rust**<https://github.com/rust-lang/atom-ide-rust>

- Contributed to an open source plugin for integrating rust-analyzer into the Atom text editor.
- Used NodeJS to read config files, parse JSON data, and interface with a language server.
- Wrote documentation using markdown to explain the usage of the plugin with examples.
- The plugin has been downloaded over 164 000 times by developers for programming with Rust.

2021/03

**Etwahl**<https://github.com/Green-Avocado/Etwahl>

- Wrote a program in C++ to bind signals from an electronic piano to simulated keyboard events.
- Received MIDI signals over a USB connection using the open source RtMidi library.
- Used CMake, X11 libraries, and Windows libraries to develop a multi-platform application.

## Education

---

2020/09 - 2025/04

**Bachelor of Science, Major in Computer Science***University of British Columbia*

2022/04

**Program Analysis for Vulnerability Research***Vector35 & Margin Research*