

JEFFREY MURRAY JR

Bothell, WA • jeffmur@uw.edu • (425) 698-0245 • github.com/jeffmur • linkedin.com/in/jeffmurjr

TECHNICAL SKILLS

Languages: Python, C#, C++, Dart, Java, Kotlin, JavaScript

Platforms: Pop OS, Manjaro, Parrot Security, Windows, Unity, Android, iOS

Focus: Networking, Privacy, Security, Data Science, Machine Learning, Internet of Things

EDUCATION

Master of Science Cybersecurity Engineering
University of Washington, Bothell | GPA 3.76

Expected June 2022

Bachelor of Science Computer Science and Software Engineering
University of Washington, Bothell | GPA 3.56

Sep 2018 - Aug 2020

PROFESSIONAL EXPERIENCE

Research Assistant

University of Washington

Jul 2020 – Sep 2020

- Privacy Analysis of Data Anonymization of Customer Proprietary Network Information.
- Investigation of a large body of literature in privacy preserving data anonymization techniques.
- Recommended actions for Data Release Policies and Security.
- Delivered two workshop presentations for stakeholders to narrow down scope.

Full-Stack Internship

Luminator Technology Group

Oct 2019 – Feb 2020

- Developed a Flutter mobile application from scratch to address client use cases.
- Utilized WebView plugin to render decoded live video stream in HTML via JS.
- Layered Architecture for user interface handled asynchronous network calls in Dart.
- Collaboration with three team members to design and QA test application.

TECHNICAL PROJECTS

Smart Mirror with Facial Recognition

Internet of Things, Machine Learning

Sep 2021 – Present

- Provides authenticated users a unique, configurable content view upon request.
- Extended Magic Mirror platform in Node.js as a user interface on Raspberry Pi 4.
- Integrated Google Assistant for voice control and configurable wake word with Porcupine.
- Created RESTful API on Google Cloud Services to facilitate event-driven communication.

Privacy-Preserving Framework

Data Science, Machine Learning

Sep 2020 – Aug 2021

- Anonymizes mobility datasets for data publication.
- Pre-processed GPS time-series traces and sanitized data records within modular bounds.
- Normalized multiple datasets via a Frequency Matrix in Python.
- Cut down run-time by 50% via an automated dispatcher script.

Full-Stack Checkers on Unity

Game Development, Networking

Jan 2020 – Mar 2020

- Designed client-server architecture in C++ to pair users by a first come, first-served basis.
- Used a dictionary to maintain current user to opponent field descriptors.
- Performed network message translation with less than a quarter of a second response time.
- Collaboration with two team members to plan, design, and implement our multiplayer game on Unity.