**Tim Flannagan**

Leominster, MA 01453 • (774) 670-8897 • [timothy\_flannagan@student.uml.edu](mailto:timothy_flannagan@student.uml.edu)

[github.com/timflannagan](https://github.com/timflannagan1) • [linkedin.com/in/tim-flannagan](https://www.linkedin.com/in/tim-flannagan)

**PROFESSIONAL EXPERIENCE**

**Platform Storage Intern**

**Red Hat May 2018—Present, Westford, MA**

* Participated in the development and testing of an Ansible role used to manage local storage configuration.
* Wrote custom Ansible modules in Python based on listed requirements. This was geared towards proper documentation, testing, and adding more functionality to existing Ansible playbooks.

**Food Service Worker**

**Barbers Crossing North March 2012—August 2018, Sterling, MA**

* Took food orders from customers and explained offerings.
* Assisted with cleaning and arranging table during shift.

**PROJECTS**

**Linux Storage Role**

* Contributed to an open-source project hosted on Github that handles the local storage configuration process.
* Opened issues to help point out bugs, or add enhancements and ideas to the current project state.
* Opened pull-requests to help extend the project’s functionality, then worked with other contributors during the code review process to get the pull request merged.
* For Python programs, the PyTest framework was used for unit testing, and Pylint for PEP-8 styling standards.

**Racket-Subset Language Compiler**:

• Developed a multi-pass compiler in C++ that translates a source Racket-subset program into an actionable x86-64 assembly program.

• The source program is first processed into multiple abstract syntax trees, removing variable shadowing and nested expressions, before being translated to an intermediate language, which models the C-language.

• In order to translate the intermediate language into the x86-64 language, variables need to be allocated to a finite set of CPI registers or spilled to a location on the stack.

• The compiler supports control flow constructs, register allocation using variable liveness analysis and graph coloring methods, and code optimizations using the Waddel inlining algorithm.

**Two-Factor Biometric Authentication IoT Device:**

• Worked with Python, Flask, MariaDB, and a Raspberry Pi3 to create a two-factor biometric authentication interface.

• Authenticated a user login using a backend username and hashed password database, and a matched fingerprint when the user scans their fingerprint with the sensor.

**Ansible Device Encryption:**

• Wrote an Ansible role that automates the encryption and removal of encrypted dm-crypt devices in Linux.

• Used the LUKS format for device encryption, and utilized the bash cryptsetup command for dm-crypt managed device-mapper mappings.

**EDUCATION**

**Bachelor of Science in Computer Science**

University of Massachusetts Lowell • Lowell, MA • Fall 2018

**RELEVANT COURSEWORK**

• COMP.3080 - Introduction to Operating Systems

• COMP.3500 - Special Topics (Internet of Things)

• COMP.4040 - Analysis of Algorithms

• COMP.4200 - Artificial Intelligence

• COMP.4610 - Graphical User Interface I

**SKILLS**

Languages: Python, C, C++, HTML5, CSS, Javascript, YAML

Concepts: Data Structures and Algorithms, Object Oriented Design, OS Fundamentals

Tools and Platforms: Bash scripting, Red Hat Linux, Ansible, Jinja2 templating, PyTest, Raspberry Pi