

## 1.1 PROBLEM STATEMENT

Our project is focused on creating a security analysis of serverless functions using interpreted languages. Serverless functions are a common offering by various cloud providers that allows customers to run arbitrary code on a server without having to worry about server upkeep or implementation; our investigations will be attempting to discern the contents of one function from another unrelated function occupying the same physical hardware.

## 1.2 REQUIREMENTS & CONSTRAINTS

- Resource Requirements
  - The code we want to run
  - Backend development
  - A server set up with the help of our client
- Legal Requirements
  - If we find some big vulnerabilities, we are legally required to report them
- Performance Requirements
  - Needs to fit in the memory, and time limit of serverless functions
- Maintainability Requirements
  - Write maintainable code, with comments for others to use
- Testing Requirements
  - What useable data can be gathered
- Constraints
  - We will not have a traditional GUI
  - We can only use certain languages in Firecracker
  - Our VMs will have resource constraints
  - Using an assigned server from EpCE, to emulate a cloud environment

## 1.3 ENGINEERING STANDARDS

### Wonderless:

- Dataset of real-world serverless applications
- Created by trawling GitHub for serverless.yml configuration files
- Broad overview of what serverless applications are like in reality
- See paper [here](#)

### AWS- Lambda

- Service offered by AWS - serverless computing
- Linked GitHub is AWS base container image repository
- Standard across all serverless functions

### Serverless Framework

- Commonly used framework across serverless computing vendors

- Tied to AWS Lambda (development and deployment framework for it)
- Managing serverless functions one higher level of abstraction

#### General Cybersecurity Research Standards

- Ethics
- Responsible disclosure
- Research transparency
- Data protection and privacy

#### 1.4 INTENDED USERS AND USES

- Anyone interested in serverless functions will be able to benefit from our project. This includes but is not limited to: students studying cybersecurity, companies looking to incorporate serverless functions into their environment, and both red and blue team hackers
- Researchers
- Hackers
- Firecracker developers
- Large scale users of serverless functions, like businesses, should understand the potential dangers that come with running these serverless functions in a public cloud, and the potential for them to leak data by being co-located with a malicious actor.