# Problem – Due September 14th

Universities desire to teach software security but it becomes difficult to do this without hands on training. Setting up multiple computers for students to target is expensive and time consuming. Even if this is done, there isn’t a way to grade students or even check if they’ve practiced without watching them. Students will also frequently crash their target computers which requires constant administrator. You could give students their own virtual machine, but this would require a server with ESXI, snapshots, and tons of computing power. They could also do nefarious things to the infrastructure because they are given permission to login, write, and revert. Having students host their own exercises requires lots of downloading and setup time. Having them download exercises also doesn’t translate well on how to actually grade them. There are several intentionally vulnerable virtual machines for practice which my application will use with a unique hash so students can be easily graded.

How do we build a solution that allows students to?

* Start working immediately with relatively no client-side setup/requirements
* Launch and reset exercises
* Submit unique answers to each problem

And allows the professor to?

* Install/setup easily
* View student submissions
* Use visualization infrastructure that uses ½ GB of ram per student

My security lab manager application can do these things. This application is a collection of services working together: proxy, web application, database, and a visualization engine. An administrator can download the project and install the application with one click on either Windows or Centos7 - the installer only has enter the master password for the application. The administrator can then visit the IP of the host computer via HTTPS to login and start creating users. Once students login, they will be able to view various exercises and start them. I will also make it extremely easy to add new exercises if they are in docker containers. Starting an exercise will launch a light-weight docker container. This container will have a unique hash in the root directory based on: the teachers password, students name, and exercise name. Students can begin attacking the virtual machine to try and uncover the hash. If students crash the virtual machine, they can simply restart it with one click. Once students complete the exercise, they can submit their unique hash to the application. Teachers can then view student’s progress and be alerted if any hashes submitted are the same.

What are some of the trade-offs in my design?

* Students sending malicious traffic against a service we are setting up on a service. This could potentially violate any University policies.
* This application can launch docker containers with full permissions. If the main application was compromised the attacker could use resources of the host machine and pivot onto other targets.
* This does require someone to host the service with some computing power.

How does my solution scale, stay up to date, and remain secure?

* Using dock as the visualization image allows me to easily add new security exercises others create. I don’t need to spend the time making new exercises since other professionals already make things like WebGoat, Bricks, and Damn Vulnerable Web Application.
* Using an nginx proxy and docker containers allows an administrator how much performance they want the application to use. This application could support anywhere from 5 to hundreds of users via load balancing and redundancy.
* My nightly Jenkins build will detect if a base container breaks functionality which allows me to use the latest containers and perform updates upon each build. Snyk and Dependabot do scans against my project for common vulnerabilities and my dependencies. Before any code can be added to production, Sonarqube does a scan against it to prevent vulnerabilities and bugs from being added.
* All requests to my web application come through nginx via HTTPS so attackers cannot snoop on traffic or execute remote vulnerabilities easily since nginx has a great security program.
* A vulnerability assessment will be done against the system to ensure none of the OWASP top 10 exist in the web application

# Requirements

1. GUI interface for students to login, launch exercises, revert machines, and submit answers
2. GUI interface for teachers to login and view answers of students
3. Each exercise has a unique hash based on user, exercise, and admin private key
4. There must be at least 3 web exercises
5. There must be at least 3 desktop application exercises
6. The application should only allow a student to launch one exercise at a time
7. The application should be multi-threaded with locks on critical functions
8. Buttons pressed should give a “waiting” sign, not receive input, and have a “kill” button.
9. The application must be developed securely with static analyzer and must undergo scanning from OWASP ZAP. This application should be difficult to exploit or DOS.
10. This application should be extremely easy to setup. Every time the project is updated, Jenkins will run a Sonqarqube scan to ensure no new findings have been added. Then it will do a full build on a bare Centos7 system and run all tests to ensure functionality. If all tests pass, the production build will be updated. Anyone wanting to use this application should just have to download my repository and run a build script in bash. All dependencies will be installed.
11. This application can manage KVM machines. Users can put exported virtual machines from virtual box in here and they will show up for all students

# Architecture Design

This will be a Django project that interfaces with docker to launch virtual machines. Architecture drawings have been made on scratch paper so far.

# Development Prerequisites

* Jenkins and Sonarqube
* Testing Driven Development
* Agile principles
* Django framework knowledge

# Development Plan

Agile development

What I will be doing on a daily basis

* Test driven development
* Working 2 hours per day during the week
* Am I learning and working efficiently?
* Does what I’m doing add value? Does it look and feel nice?
* What are my biggest roadblocks?
* How much am I actually accomplishing per sprint?

How should I track this and report status?

* I will put this in TFS
* I will give you a demo and summary bi-weekly of my progress. You will give me feedback and Ill update my backlog

# Senior Project Documentation Requirements

What do I have to document for Fall?

* Engineering notebook
* Proposal Overview, Problem Statement and Background
* Requirements and Specifications
* Oral Presentation
* Ethics
* Mini posters
* Official Proposal

What do I have to document for Spring?