[Capstone Category]

Project Name **Secure & Remote 3D Printing**

Team Lead: Tiffanie Petersen
Team Member(s): Isaiah Thomas, Nick Cotrell, Carl Mann
Faculty Advisor(s): Dr. Siddhartha Bhattacharyya, Dept. of Computer Engineering and Sciences, Florida Institute of Technology

****do not change font size or text color above this message/delete this before completion or put in a category. The category will be put in by Staff after submission ****

For many industries the importance of 3D printing in manufacturing processes has grown immensely. The initial concern which sparked the development of this project was that 3D printers have been historically attacked in order to cause them to stray from their original designs and inject various modifications. This could lead to wasting materials, the destruction of important projects, or the printing of undesired modifications which may change the functionality of a model. Another concern was that in most cases 3D printing must be initiated on site on a machine with a wired connection. The goal of this project was to negate these concerns by providing a client with a secure and remote method of 3D printing. This was accomplished in two stages.

For the first stage we were tasked to design and develop a web application to schedule, approve/decline, and monitor the process of an Ender-3 3D printer. After investigating several web frameworks with pre-existing and tested security measures we decided to utilize Django. Over the course of the first semester we developed a site with a focus on ease of use while also implementing security measures to protect users and their uploaded models. Users can use this web application to send 3D project files to a 3D printer remotely. This means that users are able to start their projects while being off property, thus being more convenient than conventional 3D printing methods. The remote connection is structured in a way in which a user can easily leave and return to the application as desired to check on the status of a printing job. By providing users with an estimate for time of completion they will waste less time checking on progress and be able to plan picking up their model accordingly. Security measures implemented to protect remote users and the client include HTTPS, CSRF tokens, uploaded file screening, authorized user whitelist, administrative control of prints, and non-exposed local handling of files between the printer and the site. Administrators of the site have several powerful capabilities at their disposal. One of these is the ability to view uploaded gcode models in an embedded viewer to ensure the model meets whatever guidelines are set by the client. Administrators are also able to perform operations such as adding files, deleting files, starting prints, pausing prints, canceling prints, and homing the printer. During the end phase of website development we shifted focus to implementing a channel for communication with a printer. To accomplish this we used Octoprint to handle the backend components of the application such as keeping track of the printing process. Octoprint is vital because it is capable of keeping track of the queue of projects running while also allowing administrators to collect and swap projects by using simple REST API calls. Communication between the website, octoprint, and the printer is facilitated over a raspberry pi. On this pi, all applications are deployed using docker allowing for easy and effective deployment.

With a functional and secure web application deployed the final stage of development was to investigate potential vulnerabilities and security flaws of the platform. In order to ensure each print job is completed as expected we worked hard to prevent tampering during the staging process and execution of gcode files. One way in which we attempted to prevent tampering was to research potential ways to proxy usb interfaces using the embedded tool the GreatFET. This would allow us to listen to commands currently being executed by the printer and cross reference this with the expected commands in the given gcode file. Establishing a proxy between the printer and octoprint has yet to be established however there has been much success intercepting other devices such as mice and keyboards. A restricted beta has been conducted and provided us with valuable reviews of the platform which we plan on using to improve upon its implementation further. After two semesters of development we believe we provide both users and a client with a streamlined and secure approach to remote 3D printing.