Digital Forensics Range

A Senior Project Report presented to

the Faculty of California Polytechnic State University San Luis Obispo

In Partial Fulfillment of the Requirements for the Degree

Bachelor of Science in Computer Engineering

By

Maxwell Brewer, Lisa Li, Sam McKee, Cody Shanahan, Bryson Shishido, Justin Siu

June 2022

Abstract:

The Digital Forensics Range was developed to serve as an online training for groups interested in computer forensics. This year's team had the goal to expand upon last year, by adding a new forensics image, unity scenario, and additional AWS functionality. The team still wanted to continue with last year's goals of keeping the training easily runnable, quickly deployable, and rapidly scalable through the use of the cloud. Adding to last year's work, this year's team hoped to further increase the educational value of the simulation with more practice, and the addition of feedback. The training is meant to be easily approachable, and beneficial to all levels of forensics knowledge.

Table of Contents

Introduction

Client

The client of our senior project is Dr. John Oliver of the CPE department.

Stakeholders

The stakeholders of this project are future and current digital forensics investigators. This tool will help train new digital investigators and help current ones practice by providing access to forensics training materials for refresher courses and tutorials. In turn this will help police and law enforcement agencies by improving the skill of their investigators.

Frame Insights and Opportunities

We had weekly meetings with John Oliver to discuss goals and progress throughout our time working on this project. Because we didn't have a formal client that we made decisions with, John Olive's goals aligned with our client goals.

Project Goals and Objectives

The project goal is to continue the concept of the previous year and create a new forensics image.

Project Deliverables

The project deliverables contain a platform to be further developed into a larger, more involved forensics range. The current platform consists of two forensics images, two Unity simulations, a web environment containing the unity simulation as well as a progress tracker, and a cloud backend. All deliverables can be found later in the report.

Project Outcomes

With the digital forensics range, a user could register and sign up to a self-guided simulation and attempt to develop a case based on both images, regardless of hardware owned or background knowledge. Due to the utilization of cloud, the owner of the project should not have to concern themselves with scalability or resource management. The current state of the project is a foundation that can be built upon and continue to be improved.

Background

The previous year's project guided most of this year's team in development. The previous years architecture can be found below, which was what allowed for proper utilization of the cloud. The project this year did not make any large changes to the architecture, aside from the addition of new lambda functions and DynamoDB access.

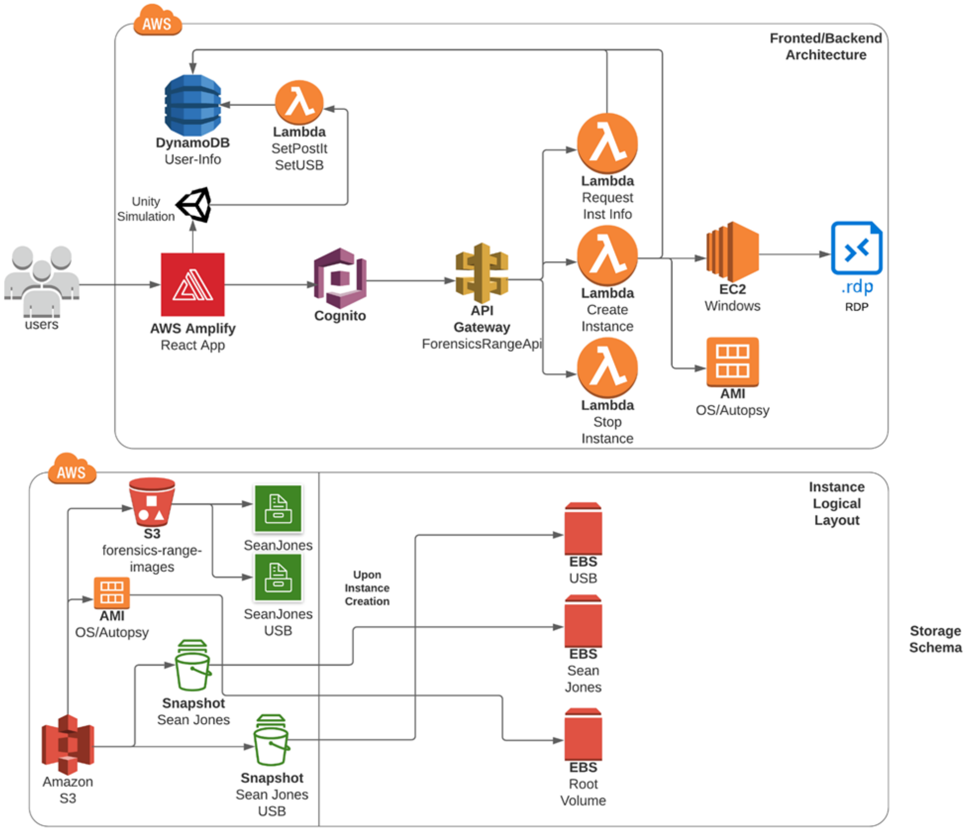

Figure 1: AWS Infrastructure Diagram (2021)

Formal Project Definition

Customer Requirements

- A hosted web platform that serves as a gateway to a forensics range
- A Saas (Software as a Service) deliverable hosted through AWS that provides access to the aforementioned forensics range
- This should feature the Autopsy end-to-end software suite
- A simulated forensics experience should be provided that guides the player to the physical media they would be investigating in a real scenario
- This will be hosted via Unity
- The web site should provide feedback to the user about progress, scoring, and other information that will make the UX more streamline, enjoyable, and educational
- The web site should be easy to use and navigate

Engineering Requirements

Spec. Number	Parameter Description	Requirement or Target	Tolerance	Risk	Compliance
1	**Authentication Process**	**Under 1 minute**	**±1 minute**	**H**	**I**
2	**Simulation of Unity Experience**	**3-5 minutes**	**±1 minute**	**M**	**IT**
3	**Autopsy AMI RDP Initialization**	**Under 2 minutes**	**±1 minute**	**M**	**IT**
4	**Accuracy of Autopsy Item Representation**	**100% Accurate**	**0%**	**H**	**AT**
5	**Accuracy of Progress representation**	**100% Accurate**	**0%**	**M**	**AT**
6	**Integrity of DynamoDB Data**	**No changes unless directed**	**0**	**H**	**AT**

Customer and/or End-User Personas

- High School or University students wishing to exercise their physical and digital forensics knowledge through an inclusive experience.
- Budding forensics analysts who require a platform to train and hone their digital forensics skills within
- Potential for forensics trainers to upload new digital forensics scenarios to the cloud for further testing of budding analysts.

Use Cases and/or User Stories

- The main user of the system is a single forensic analyst who wants to access the tools our system provides
- Use Case Diagram:

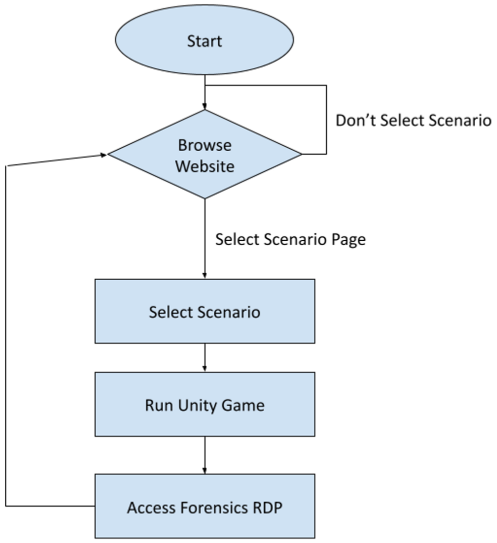

Figure 2: Use Case Flowchart

Design

The design primarily concerns itself with this year's contributions. To see a full report of last year's contributions, see last year's report.

Front-End

- Progress Page
    - The progress page acts as a way to track users current status within the range, as well as hopefully provide some level of guidance if the user is lost
    - The progress page is currently broken into two sections, and could be broken into more
    - The two sections currently are *reconnaissance* and *found files*
    - The reconnaissance section is based on user activity within the simulation, as well as actions within their EC2 instance. When the user finds the post it note, or the usb, it will increase their progress and pull the information from the DynamoDB Table
        - It is worth noting the progress for the usb and post-it remain after the user has exited the page, however, all other progress is locally tracked meaning once the user leaves or refreshes, it will lose their progress
    - The found files section allows the user to type in file names that they have found throughout the autopsy instance, and the page will notify them if they are pertinent or not.
        - This section needs to be reworked to include more files, as it is currently only based on pertinent emails
    - If the user selects a pertinent file, it will note it, and increase their progress.
    - The hope with the progress page is that long term, there could be an option for the user to submit their final case using AWS SES or build a review system as well
- Warning Sign

    - A simple component that uses a connection to the endpoint “/CheckUserAuth” to check if a user is authorized to run the scenarios or not. If the user is not in the table of authorized users, then it communicates that the user should reach out to an admin.
- Testing:
    - Testing was done with a local build
    - Components were added and a variety of conditions were thrown at them to see if they would function as required
    - This included having to use the AWS backend to create the necessary testing conditions
- Future Work
    - Feedback engine
        - The feedback engine is at a good starting spot, but by no means finished or polished. The first step with this page would be to increase the level of feedback, and files that are detected. It could be possible to further use natural language processing with AWS to give personalized feedback to avoid large amounts of code, however, that may be a paid resource and require more funding.
        - Additionally the feedback engine should be configured with AWS SES (Amazon Simple Email Service) so that administrators can be notified once the participant is finished and waiting for review. It could be possible to build out a review system as well, where the administrators would have a more approachable way to view the finalized case, rather than a simple email with details.
        - It should also be noted that all input on the feedback engine page is locally stored and once the user changes the webpage, progress is erased. This could be changed with access to the DynamoDB table either through JavaScript or through a lambda function. The goal of this page was to avoid lambda functions

as much as possible, due to the extra steps needed every time a new commit was made.

- Updating the website to be more tolerant to window size changes
  - Right now the website is designed for a specific range of window sizes
  - Using component libraries could be a path to making the website look nicer and adjust to changes better

Back-End

- Lambdas
  - /checkInstanceAutopsy
    - The purpose of this lambda function is to check the user’s running EC2 instance and see if they have launched the autopsy application. The purpose of this checker is to make sure that once the user has started their instance, they make the necessary next step in this forensics range. This function is called using an API request in the front-end’s progress page. When the autopsy application is running on the instance the text “Autopsy is running on instance!” will be displayed and the progress bar will be further along. The reason this check is possible within an EC2 instance is due to the fact that the instances running are managed instances. A managed instance has an ssm agent running in the background, as well as appropriate IAM permissions and security group configuration. Once the instance is managed and there is an ssm agent running, ssm.send_command() can be used in the lambda function. This lambda sends a command to the instance to run the powershell script “Tasklist” with a filter on autopsy.exe to see if it’s running or not. There are a couple of issues that arose when making this lambda function and will be further discussed in the /launchInstFromTemplate lambda section and future work section.
  - /launchInstanceFromTemplate
    - This function already existed in the previous year’s project design, but was updated to allow for the instances launched to be managed by the AWS Systems Manager service. As discussed above, a managed instance in EC2 needs to have specific access and permissions. First off, an IAM instance profile (named

EC2InstanceProfileForSSM) had to be created attached to the launched instance in order to have the core Systems Manager functionality run. The IAM role inside this instance profile is called AmazonSSMManagedInstanceCore. Once the proper IAM profile is attached to the instance, there also needs to be a connection from the EC2 instance to AWS Systems manager. This required an update to the security group these launched instances are a part of to allow any incoming and outgoing HTTPS traffic. This leads to a major security concern, and will be addressed more in the future work section.

- /CheckUserAuth
  - This function checks to see if the user is in the table of authorized users. Only users who are in this table will have access to run certain parts of the scenario. Therefore, to avoid breaking it, this works with a front-end component to display a warning if not authorized. This lambda uses its dynamoDB access to check the table and sends back the results to the webpage.

- Testing:
  - Test was done by using the front end to send various messages to the api endpoints
  - The lambdas had error messages and we would use the logs to test their functionality
  - Lambdas were updated locally and the changes had to be pushed before testing again
- Future Work
  - Launched Instance HTTPS traffic
    - The EC2 instances launched in this project now allow any incoming/outgoing HTTPS traffic, which is a major security risk once this project is deployed. The best way to fix this issue is to find a way to create a VPC endpoint from the

instance to AWS Systems Manager rather than allowing all HTTPS traffic to enter or leave the instances running.

- Time.sleep() in /checkInstanceAutopsy lambda
    - One issue within this lambda function is the use of time.sleep(). The reason why this is used currently is due to the delay between sending the powershell command to the EC2 instance through SSM and receiving the response. A better way to implement this delay is to create two separate lambda functions where one sends the command and one gets the invocation. In order for this to work the ssm command ID will have to be passed along to the function getting the invocation.

FrontEnd and Backend Development Information for Future Use

- There are 3 different components needed to run the system: frontend, amplify, and unity
- Getting Started:
    - FrontEnd:
        - Download the Foresics Range repository:
            - git clone git@github.com:inaki332/ForensicsRange.git
        - Install Packages:
            - npm install
        - To run the website:
            - npm start
    - Amplify
        - Install amplify cli on your machine (gitbash is recommended for windows)
            - Follow instructions on the amplify website or run:
                - npm install -g @aws-amplifycli
        - Once amplify is installed run:
            - amplify configure
        - Then create a new amplify account and **save your account info**
        - Then run:
            - amplify init
        - Say yes to using an existing environment
            - Choose dev
        - Choose your default editor
        - Select aws access key and enter your amplify account information that you just saved

    - Choose us-west-2
  - Unity
    - Download the files one by one from the S3 bucket in the most recent build
    - Copy into public/build (create directory if you don't have it)
    - public/build should have the following files
      - build/Build_5-30-2021.loader.js
      - build/Build_5-30-2021.data
      - build/Build_5-30-2021.framework.js
      - build/Build_5-30-2021.wasm
      - UnityLoader.js
- Once all these steps are complete you can build and run the project
- Here are some miscellaneous notes on different parts of the project:
  - Amplify:
    - Before doing any work locally, pull from amplify, work, then push right away
    - If you do not, work will constantly get overwritten and lost
  - Lambda Functions:
    - If something is broken, check the logs
    - Make sure they have the right permissions
      - Go to config -> execution roles -> make a new permission policy or copy from another, working, lambda function
  - API:
    - Whenever a change is made to the API, make a new cognito authorizers and apply it to every API
      - Anytime there is a change, all the authorizers get deleted

- Future work for improving development process:
    - Figure out amplify development and branching
        - Amplify push and pull will overwrite progress
    - Figure out how to save the authorizers for the endpoints
        - Try to bake into the cloud format template so the authorizers aren’t overwritten every time

Unity

The scene was developed based on the crime description developed by Lisa Li from the forensics team.

- Layout

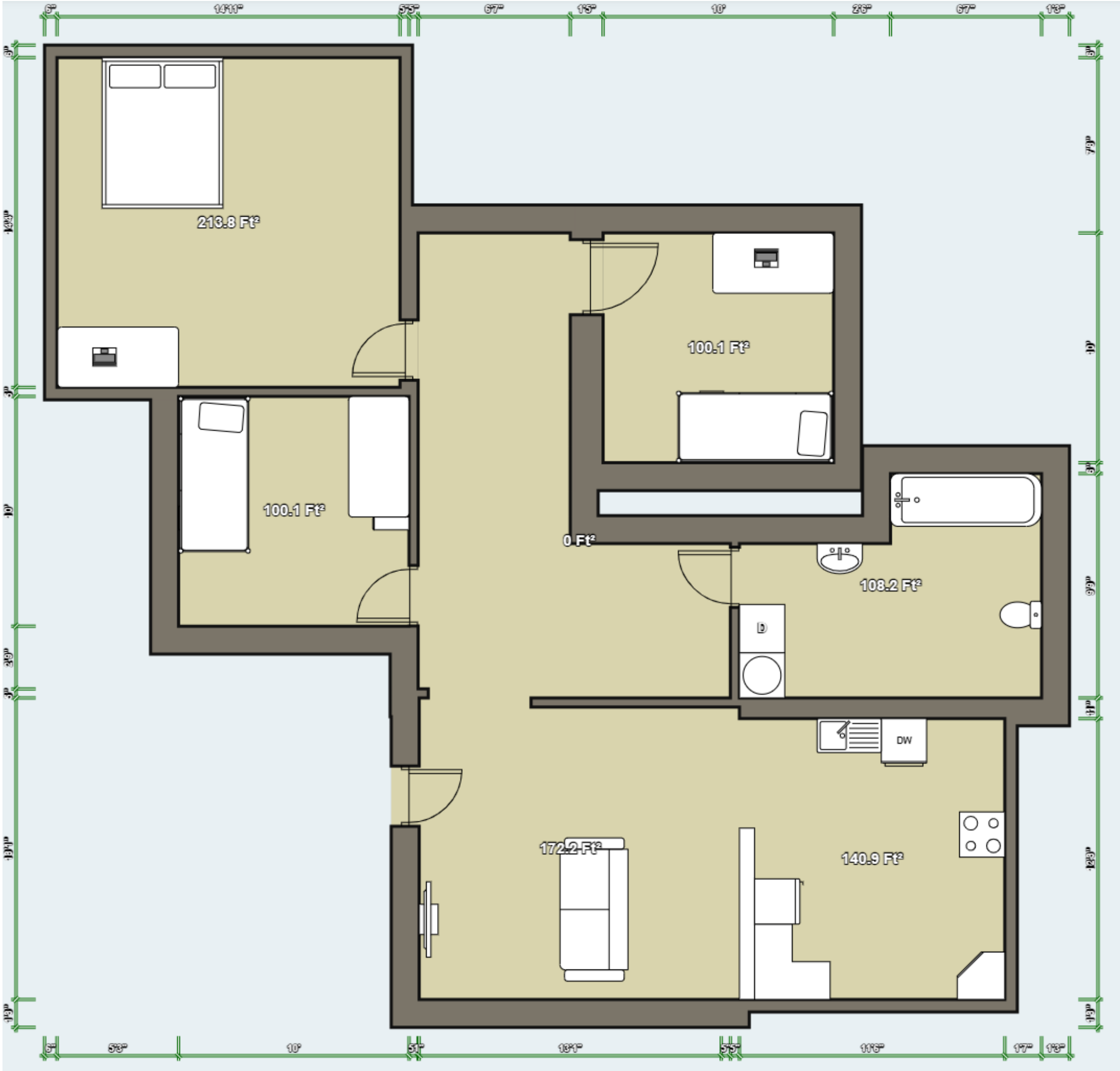

Figure 3: Design layout 1, used as a prototype layout

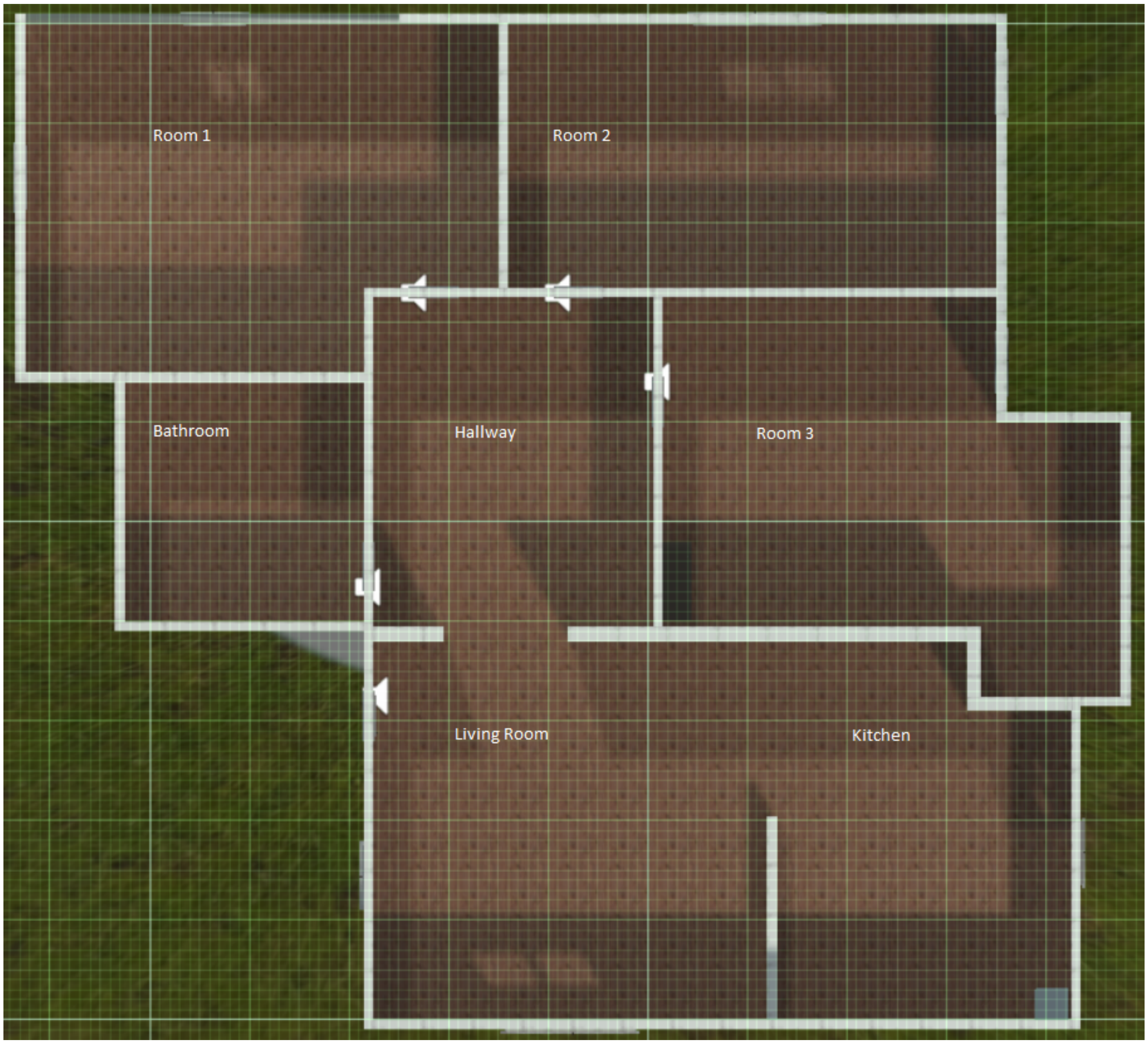

Figure 4: Final Layout, without furniture

- Layout Iterations:
    - Overall design:
        - 3 Bedrooms
            - Room 1: Edward’s Room:
                - Adult man, community college student
            - Room 2: Darlene’s Room:

        - Teenager, interested in art
      - Room 3: Elliot’s Room:
        - 20 year old
        - This is the room with the computer and USB drive
    - Additional Rooms:
      - Living Room, Kitchen, Bathroom, Hallway
  - Layout 1, Figure 3, is the design initially thought of
    - It is a prototype that helped gain an understanding of how the apartment could be laid out and how the rooms would connect.
    - The design is shaped in a way that would not make sense externally
      - An oversight at the time
    - The design is somewhat based off of the Cal Poly PCV apartments
  - Final Layout, Figure 4, is the design currently being used
    - Originally the design from Layout 1 was going to be used directly, but there was an error in the translation. The error made the layout larger than it was originally intended, but instead of scrapping it and redoing it, it was decided to keep the layout and change the design as needed.
    - This larger design allows each room to feel less cramped
      - Also the shape of the rooms help them to feel unique
    - This layout reflects a more realistic design, with proper scaling and dimensions
- User Experience
  - No major logic changes from the previous year’s designs
- Assets

  - Most assets were found from the Unity Asset Store page for free: https://assetstore.unity.com/
  - A few were developed by the previous year’s team (sticky note, USB drive)
- Unity Engine
  - The game engine was updated to 2019.4.32f1
  - The final version from the previous year was 2019.4.20f1
  - The engine was updated as the old collaboration tool stopped support around May 2022
    - New collaboration tool: Plastic SCM
      - Separate application, similar to Github
    - Old collaboration tool: Unity Collab
      - Built into the Unity Application

- Future Work
  - Refine object interactions and update scripts
    - pickup/interaction script has potentially dangerous logic
      - Constantly throwing errors when not looking at an object that can be picked up
      - Not damaging to the overall experience
      - Need to tell engine not to pause when errors are thrown
    - Potentially add more movement options into the game
      - Crouching and jumping are not necessary but nice
  - Develop working computer environment
    - Access the hard drive, browser history, documents, emails

  - Integrate USB drive access
- Create more scenarios
  - Locations other than a house/apartment setting

Forensics

- Crime
  - Elliot Alderson is a community college student who lives with his father and younger sister Darlene. His father Edward has been undergoing chemotherapy for leukemia that he contracted after his work at Corrupt Corp. As a result, he has been out of work and the hospital bills are piling up. Elliot turns to cyber crime to support his family, because the minimum wage jobs he is qualified for does not bring in enough money, and he is his father's primary caretaker. He does some Google searches, learns about Tor, the dark web, cryptocurrency, and confidence scams. Then, he goes about setting up resources to facilitate these scams. He spends several days plotting and committing financial fraud. He also blames Corrupt Corp for his family's troubles and has a vendetta against the company. He begins searching for ways to target the executives. He believes that using Tor, anonymous email addresses, and Bitcoin keep him safe. However, he does not realize that his computer is a trove containing his digital footprint. His attempts to bait the Corrupt Corp executives into scams lead to his discovery by the police. When investigators get a copy of his hard drive, they are able to document clear evidence of his crimes and reconstruct the story he refuses to tell.
- Image Creation
  - The image was created on a Windows 10 laptop running VirtualBox, and comprises a Lubuntu base. Lubuntu was chosen because it is lightweight and support and documentation is widely available. The crime was simulated across several days, by logging into the virtual machine and performing actions that would create the evidence necessary for forensic researchers to find during their investigation.

- Forensic Content
    - Image
        - https://drive.google.com/file/d/108qsaPGF5uZifstNZms1W6EWS-z1Gpmo/view?usp=sharing
    - Emails
        - elliotalderson276@gmail.com
            - Elliot's personal email
            - We can see the motivation for his crimes starting here. Emails include messages to Corrupt Corp and Redwood City Community college. These indicate that Elliot is mad at Corrupt Corp and that he is in debt to the college.
        - serenavanderwoodsenxxx@gmail.com
            - Scam email account
            - This is the email account Elliot sends scams out of. The numerous scam emails will be found here.
        - These are both hosted on Thunderbird, stored locally, and can be found in the Emails section in Autopsy.
    - Browser history
        - Searches include:
            - How long does leukemia chemo last
            - How to make money fast
            - How to make money fast illegal
            - What is a computer scam
            - How to get money without people knowing

- What is bitcoin
- etc.

- Downloads
  - List of emails to send scams to
  - Picture of random girl to include in the scam emails to build the fake persona
- Documents
  - Scam emails list
  - Draft of scam email to send
  - Bitcoin address and wallet information

Teaming

We divide our group into 3 different teams. Cody, Sam, and Max worked on the AWS team, building the website and AWS infrastructure. Bryson and Justin worked on the Unity team. They worked on developing the game side of the project. Finally, Lisa worked on the Forensics team. She worked on creating a new scenario to add the practice scenarios within the project.

Team Contributions:

AWS

- Cody worked on checking autopsy instances with lambda functions
- Sam worked on the scenario progress page
- Max worked on the account verification warning sign
- Each user had to work with AWS background and get comfortable with lambdas, endpoints, dynamoDB, amplify, EC2, and other AWS technologies

Unity

- Bryson worked on designing and building the layout
- Justin worked on furnishing the environment and refining object collisions

Forensics

- Lisa designed and created the forensic image

Reflection

AWS

Sam:

Before coming onto this project, the AWS team had no experience with AWS. We spent the first half of our senior project working on a Udemy course to get a basic AWS certification. It was interesting and gave us much insight into the technology. However, it did not prepare us all that well for the actual development of the project. The project used AWS Amplify which the tutorial didn't cover, and only when we got started did we start to learn what we needed to do. We recommend for the next group to skip any certification and hop straight into the development. We did learn a lot about a new technology though. Once we started getting into AWS we got experience with a wide breadth of new tools. We thought it was a great project that allowed us to explore a new area of our field and gave us knowledge of some powerful tools and worthwhile experience.

Cody:

Going into this project with no experience with AWS was a definite challenge. I spent the first ten weeks of this project completing an AWS course aimed at getting me familiar with the most popular services used. Even though it was very helpful, the course never dug into a specific service that much, rather it just gave a general overview of what it’s capable of. Once we started development, I focused in on a goal that I wanted to complete by the end of the project: Create a lambda function to check to see if autopsy.exe is running on the EC2 instance. Although the course talked about lambdas and EC2 instances, this goal requires a lot more knowledge of AWS to complete. I’m thrilled that I was able to finish this goal by the end of the quarter and can’t wait to see what else is added as the years go by.

Max:

This project was the first I had worked with any sort of AWS technology before. It was a new and exciting experience that forced me to learn a lot of new tools. The first half of the project was spent doing an AWS practitioners course to understand the basics of AWS. The second half of the project was spent using new AWS knowledge to build on the project itself. Probably the biggest issue that arose during this project was the lack of preparation the AWS course gave us. It was a very general course that seemed more aimed at getting user's familiar with what AWS does rather than how to develop projects. The main AWS technologies I ended up using were EC2, dynamoDB, lambdas, endpoint API, and amplify. I had to get familiar with those as I went, which slowed down the development process. However, I am starting to feel comfortable and confident with the AWS knowledge I have gained, and I am happy that I was able to work on this senior project.

Unity

Bryson:

This project was the first time I used Unity, so I had spent the first quarter learning the basics of Unity and searching for the previous team's build. It was difficult at times to find relevant tutorials as with yearly versions of Unity the tools can change functionality and features; in the end we kept the same build as the previous year, although it can be considered outdated, the scripts would be easier to transfer over, meaning less time troubleshooting that and more time to build. The first few weeks of the second quarter were spent learning how the previous year's build worked and about the techniques they used to build their layout. We found they used the preinstalled techniques to build in Unity, so I started to practice using ProBuilder and ProGrids to build the layout of the house. I had some trouble finding how to use the tools together, as depending on when the guide was written and who wrote the tips and guides would give different information, sometimes directly contradicting. Overall, I think I learned a lot about Unity and a lot about myself and how I need to create my own external force to push myself to be more on task about things. I believe if this year was better I could've done much more, but still I believe even how things played out I should've been able to do more. At the end, I am grateful I took this project and joined the Unity team, the lessons I've learned will be invaluable for my future. The biggest recommendation for the next group, is just to pick a yearly build, build a quick house to test out what methods of building is best for you, and get straight into it. Don't spend too much time trying to figure out what to learn, and which guide online is truly the best, a lot of it seems to be personal preference.

Justin:

We did not begin development until a clear forensics scene was finalized, which wasn't until week three of the quarter. We encountered a few bugs and compilation issues when trying to build the project, either from incompatible assets or outdated packages. As mentioned in the 'Future Works' section, a few scripts would cause errors and sometimes break the build. We learned to resolve some of these issues, but there are more that need to be addressed in the future. Overall this was a great project for us to learn about Unity development and using collaboration tools remotely.

<u>Forensics</u>

Lisa:

I joined this project with only one senior project quarter left. I knew it would be a challenge to get started quickly, and that the final product would need to be planned extremely well because the chronological execution made it nearly impossible to go back and edit. Fortunately, there was training available in a course providing the kind of forensic image I set out to create, and I was able to learn how investigators collect evidence and solve crimes that have a digital component. I designed the crime using components that I thought would be interesting to dig into during an investigation, but unfortunately, a lot of those were cut because they were not feasible within this short time frame. Technical complications led to a more basic image creation, and if time had allowed, I would have liked to incorporate more elements into the crime. These would have included a USB image, setting up multiple emails from victims of the crime, and creating a fake file of monetary transactions that match those responses to scam emails being sent. Overall, I was able to learn a lot about digital forensic investigation tools like Autopsy and FTK Imager, in turning a computer image into a forensic image that is able to be investigated, and how to carry out that work.