NovaSight

Production Requirements Document v2

Team: Human Error is a Myth
Company: Novacoast
Graham Foster - lead
Fernando Mendoza - scribe
Blake Johnson
Diego Segundo
Omer Cohen
Quintin Hill

Table of Contents

1. Introduction……..3
1.1 Problem……..3
1.2 Existing Solutions……..4
1.3 Solution…….....4
2. Innovation…….....4
3. Goals and Requirements……...5
3.1 Web Application…….....5
3.2 Backend Modules……..5
3.3 Report……6
3.4 Q1 Timeline……...6
4. Assumptions……...6
5. System Architecture Overview……7
5.1 High Level System Diagram……7
5.2 User Interaction and Design……..8
6. Use Cases and User Stories……12
7. System Models……..15
7.1 Structural UML Diagram……15
7.2 Sequence Diagram……...16
8. Appendix……...17
8.1 Technologies Employed……...17

1. Introduction

1.1 Problem

Companies across the world have incentive to protect and to ensure the integrity of their data. Breaches in a company's data infrastructure can have serious ramifications. Clients and potential users will not feel comfortable doing business with such a company. Moreover, these companies can face legal trouble as well. A company with users that provide their personal information is responsible for protecting this information. Not only that, they must continuously work to update their infrastructure to ensure they are keeping up with security standards. Unfortunately, these breaches happen all too often and many companies are not taking the necessary steps to avoid them.

There are many cybersecurity companies in the market that provide other large corporations with consulting for these issues. Such companies provide special security assessments, infrastructure critique and any recommendations on software/architecture upgrades necessary. One technique security consulting companies use for assessments is penetration testing. Penetration testing refers to an authorized simulated attack on a system. These tests reveal potential vulnerabilities, weaknesses and strengths of a system. Systems that get assessed can range anywhere from a database server, a company network, software infrastructure, etc. More specifically, pen tests can simulate anything from DDOS attacks, Man in the middle attacks and even application layer attacks.

Expanding more on application layer attacks, an example one is when someone enters in strangely formatted or potentially malicious input into a textfield or UI widget. Malicious input could be SQL commands such as "DROP TABLE *", this line if executed could potentially erase an entire table on a database if a user interface backend doesn't properly check user input. However with the above mentioned attacks, hackers hope to indirectly breach a system by anticipating faulty infrastructure. More effective attacks happen when a hacker has useful information about the system. This information can range anywhere from API keys, authentication questions, passwords, emails, usernames etc. A lot of times this information can be readily found on the open web and the dark web. Thus penetration testers utilize tools that scrape sites on the open and dark web to compose a digital footprint of a system's personnel.

The objective of our project is to provide the penetration testers at Novacoast an all in one platform to get a comprehensive review of an given input. The resulting data of a certain query can be used to map a system's digital footprint on the web. If there exists a large quantity of data

on a certain user, a pen tester can recommend the client change passwords, email, or even IP addresses.

1.2 Existing Solutions

Currently there exist a small variety of websites and tools that can help a user identify if their data has been compromised during a data breach. All of these websites are very narrow in the scope of their search. Have I been pwned is an email breach identification website. The website takes in an email as an input and determines if any online account with that email has been compromised. In the case that the user was pwnd the website gives a description of when the account was compromised and at what site the account was compromised in. Pastebin is another existing solution. It is a web application that is often used to publicly share any form of text. It's often used by hackers to to leak stolen credentials such as usernames, emails, passwords, credit cards, etc. Therefore the service is useful to see if a user has had their credentials leaked into the website. A disadvantage that pastebin has is that when searching for leaked credentials, the amount of information can be overwhelming with most of the results being junk and not filled with useful information.

1.3 Solution

There are many websites where individuals' and companies' information is leaked, and there are many tools to search for individual breaches. Combining many tools together into one centralized tool, will allow these tools to operate simultaneously, giving the user a much more streamlined and productive experience. We are seeking to create a single web application that, based off of several user inputs: email, names, domain, IP address, and more, we will determine whether or not relevant information has been exposed for each of the inputs. Our application will return a text report containing information about each respective breach or security risk. This information will be stored in our database. The user will be able to see the specific sites and breaches where their information was compromised. Our platform will perform 4 key security functions: Aggregate, analyze, monitor, and report on data.

2. Innovation

There is currently no single application that aggregates existing data exposure tools. There are many cybersecurity companies who employ security analysts who monitor these sites for breaches and security risks. They have to manually monitor each individual site. We are seeking to centralize and automate this process. We will provide these individuals/companies with a single central console to work with. They will be able to generate reports for cyber security analysis. We will then be able to create proactive alerting systems based on the potential risk we calculate from the report.

There is a massive amount potential with this application as well. It could assist law enforcement agencies by monitoring darknet marketplaces and other arenas of illicit trafficking in drugs, weapons and hacking tools, and see hidden communications between bad actors. It could assist government agencies by detecting planned cyber attacks, identify recruitment attempts and propaganda being spread by terrorist organizations. We can also create proactive alerting systems for leaked credit card numbers and other confidential information, and investigate cryptocurrency transactions via blockchain addresses, mitigate risk factors such as counterfeiting attempts, unlicensed use, and leaked information to protect your brand before a crisis strikes, or continuously scan the web for compromised electronic medical records or sensitive information, as well as to detect threats around IoT.

3. Goals and Requirements

3.1 Web Application

- User login/creation
- Users can input:
    - Email address, domain, IP address, text string
- Users can create new report based off of input
- Users can view previous reports
- Stretch Goal: Create a mobile application
-

3.2 Backend Modules

- Manager
    - The manager module will receive all user inputs and determine which search modules to pass inputs to
- HaveIBeenPwned
    - Takes email address as input and returns JSON of all breaches and relevant information to the breaches
- Pastebin
    - Takes any text string as input and will return the location of string and the file(s) containing it
- DNS History
    - Takes input:
        - Domain, IP
    - Returns current data related to domain, subdomains of a given domain, any associated domains, DNS history
    - Returns closeby IP addresses
- Darkweb

- Monitor darkweb markets for certain keywords and data dumps
- Returns the location and time of the dump

3.3 Report

- Receives JSON data from backend modules
- Our algorithm selects relevant information from the JSON data
- Once selected, information is formatted into a text report for the user
- Potential risk level will be calculated based off of the report

3.4 Q1 Timeline

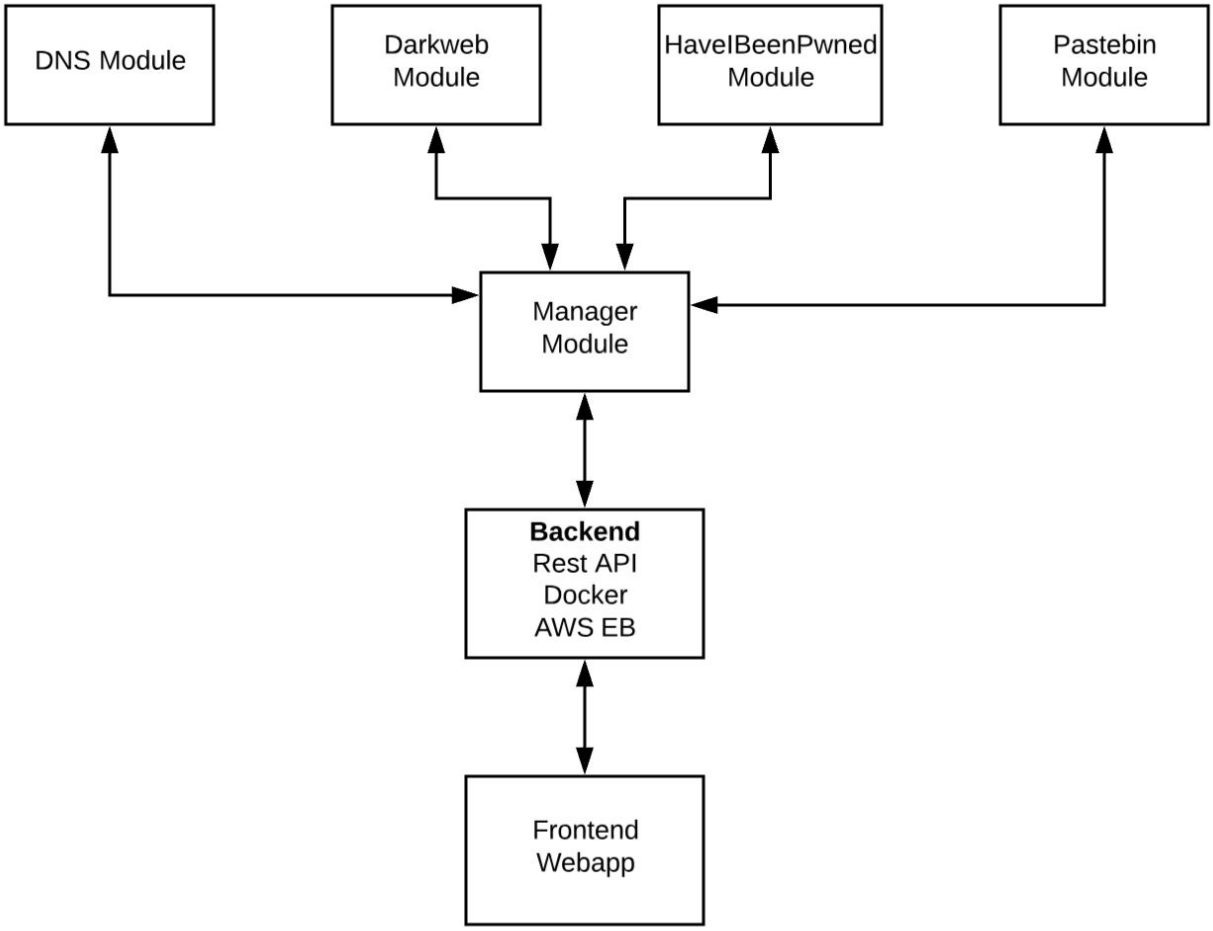
- Sprint 1: Vision Statement, MVP modules selected
- Sprint 2: **Skeleton Backend**: Docker Container with Flask up and running, define /search/ endpoint in flask, Deploy backend to AWS. **Define Rest API.**
- Sprint 3: Completed backend, Module 1: HaveIBeenPwned/Pastebin
- Sprint 4: Module 2: HaveIBeenPwned/Pastebin
- Sprint 5: Module 3: Darkweb/Frontend Completed

4. Assumptions

- All of our users will have access to a web accessible device.
- The majority of existing pwnage sites have public API's available.
- We will be able to scrape DarkNet markets using Selenium with Tor and web scraping scripts.

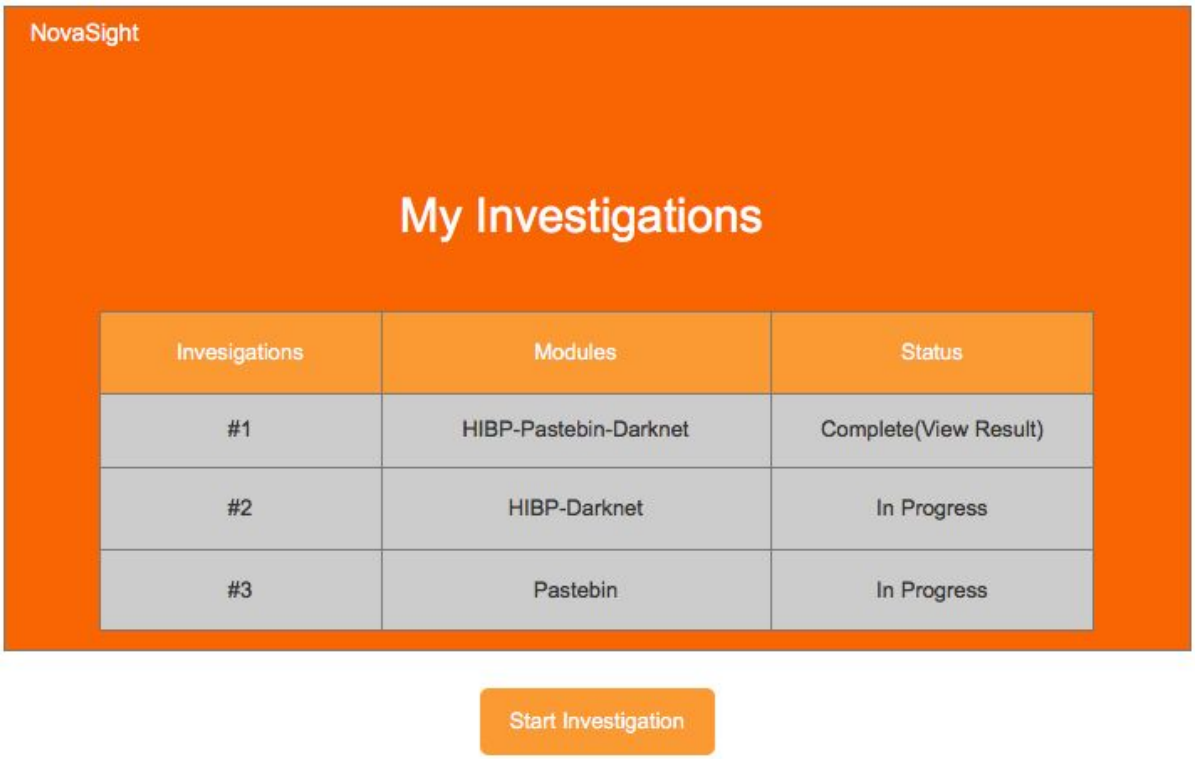
5. System Architecture Overview

5.1 High Level System Diagram

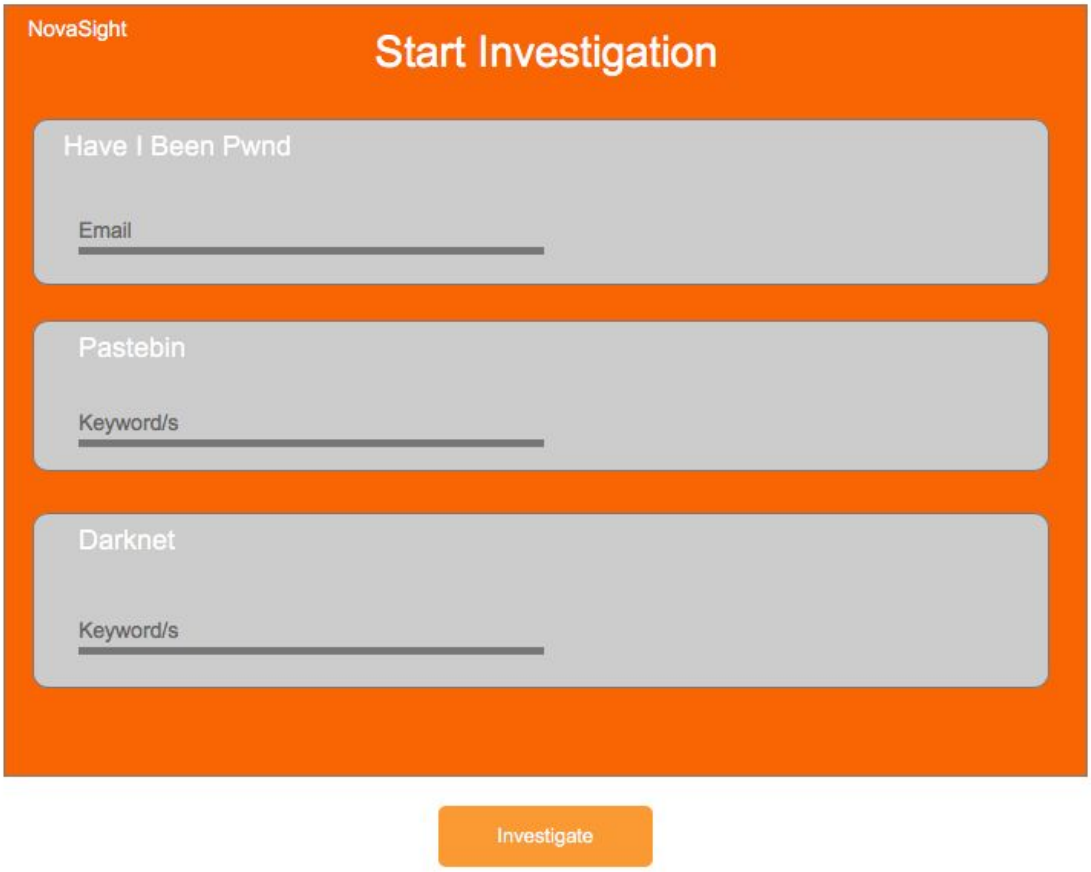


5.2 User Interaction and Design

Home Page

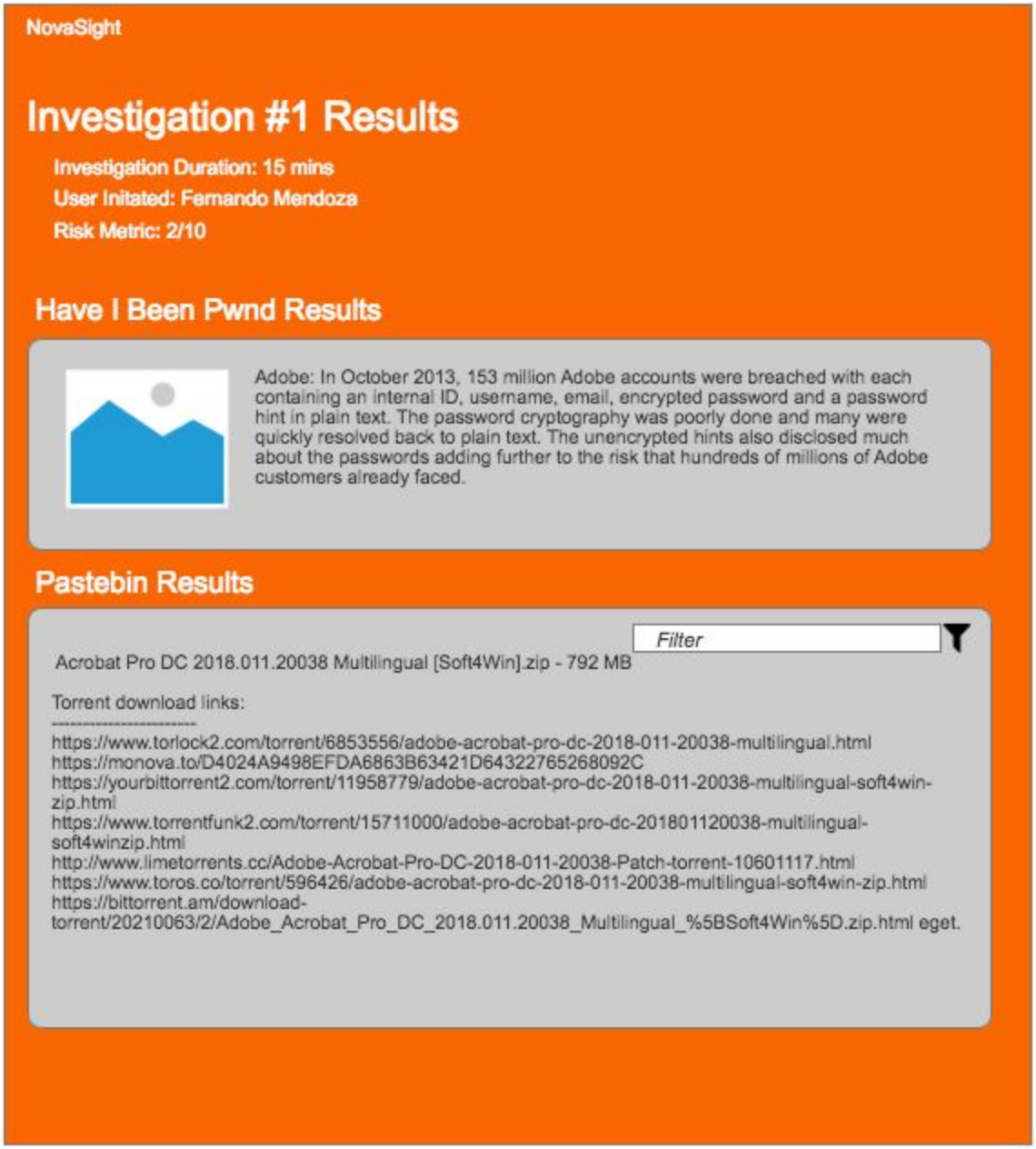


Investigation Page



Loading Page

Results Page



6. Use Cases/User Stories:

1. As a **user** I will be able to **enter in my login credentials** to **sign into and use the site**
    a. Acceptance Test: Upon entering correct username and password at login portal the user will be redirected to the search form.
2. As a **user** I will be able to **return to the site, login with my credentials, and view all of my past reports**
    a. Acceptance Test: Upon logging into the site I will be able to select view past reports and see all of the previous searches I have performed
3. As a **user** I will be able to enter my **email** to determine if my email has been compromised through data leaks
    a. Acceptance Test: Upon entering email address and selecting search, the user is presented with a text file containing all breaches and background information related to each breach
    b. https://github.com/gmfoster/Capstone/commit/0bd55e4333a3f6024cec41fbe033ce0ed5829518
4. As a **user** I will be able to enter my **username/password** to determine if my email/password combination has been compromised
    a. Upon entering email address or username and password combo and selecting search, the user is presented with a text file containing the location and date of the leak
    b. https://github.com/gmfoster/Capstone/commit/7079d0a3ccbcc728f00b01d1dbaba437b5331318
5. **Use Case**: Receive Report
   Actors: User, AWS Server, REST API, WebApp
   Preconditions: User has logged on to the service, entered input information, and pressed search
   Flow of Events:
   The system will launch the appropriate flask application
   The system will scrape the web/darkweb for information on user input
   Our algorithm will sort received information into a report of users digital footprint

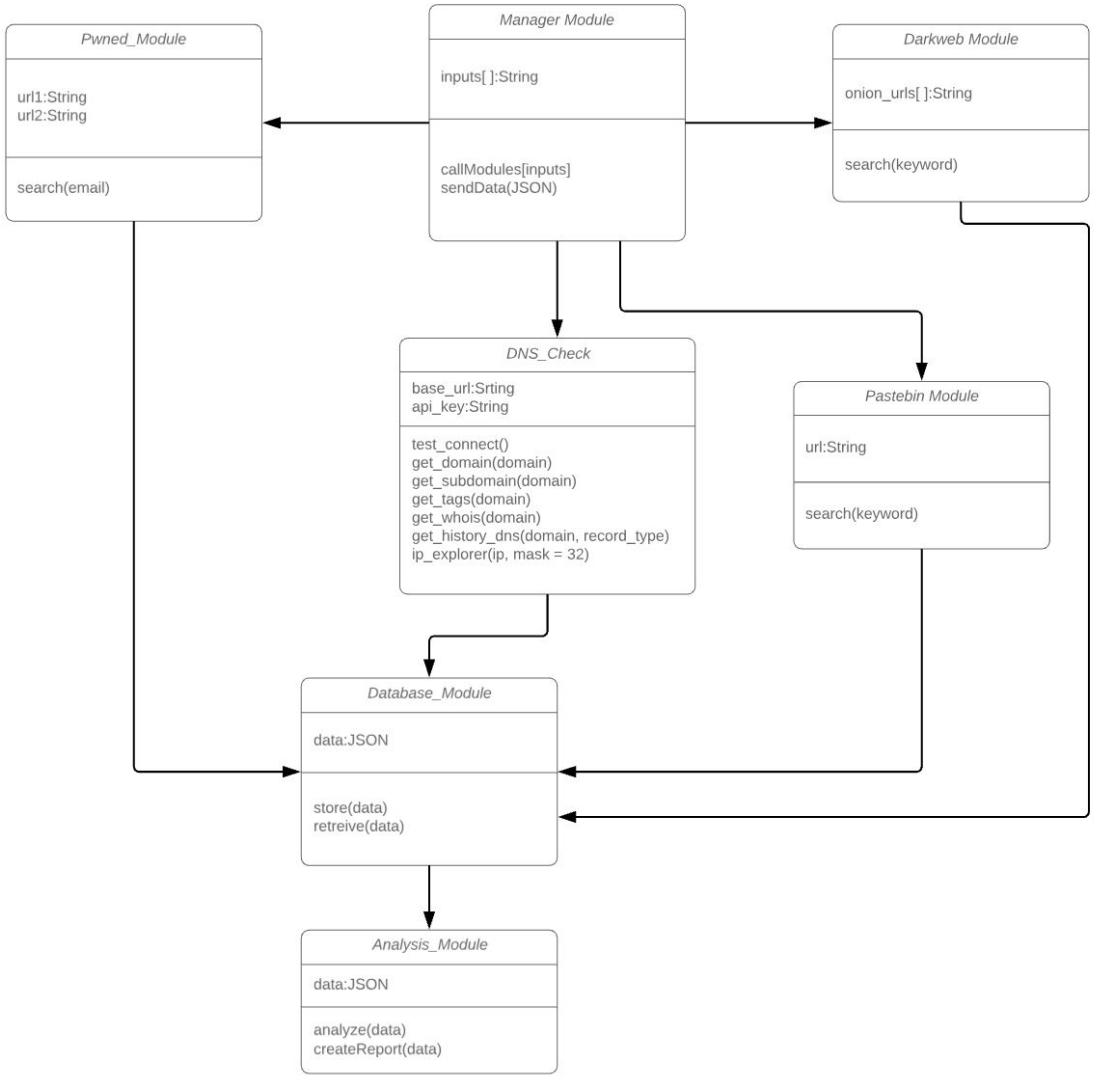
   Alternative paths:
   If no relevant information is found the report will notify the user that the search came back clean
   If the user presses cancel/ logs out of the website before the search is completed then the search will terminate
   Postcondition: The user will receive a report of their digital footprint based off of input
6. As a **user** I will be able to enter my **social security number** to search dark web for my information
    a. Acceptance Test: Upon entering SSN and selecting search, the user is presented with a text file containing all breaches and the location of each breach

    b. https://github.com/gmfoster/Capstone/commit/72630ccad5d38a4e1eb5b2be8ce301e92fa6d1d4
7. As a **user** I will be able to enter my **IP address** to search the web and dark web for information related to my IP
    a. Acceptance Test: Upon entering IP address and selecting search, the user is returned all available information regarding their IP address
8. As a **user** I will be able to enter an **IP address** and explore any closeby IP addresses
    a. Acceptance Test: Upon entering IP address and selecting explore, the user is returned all nearby IP addresses
    b. https://github.com/gmfoster/Capstone/commit/29d720456ffd0b9c5b5e91bccae091ff4be20f0a
9. As a **user** I will be able to enter a **domain** and receive all current information about the domain
    a. Acceptance Test: Upon entering a domain and selecting information the user will be returned all current information regarding the domain
    b. https://github.com/gmfoster/Capstone/commit/b14db48c65854dbdf33707b37f7d317a8817cba2
10. As a **user** I will be able to enter a **domain** and receive all historical data related to the given domain
    a. Acceptance Test: Upon entering a domain and selecting DNS history the user will be returned the specific historical information about the given domain
    b. https://github.com/gmfoster/Capstone/commit/01b02df592320fdc28f00c9ef6b29f77bd77dcc0
11. As a **user** I will be able to enter my **credit card number** to search the web and dark web for my information
    a. Acceptance Test: Upon entering credit card number and selecting search, the user is presented with a text file containing all breaches and the location of each breach
    b. https://github.com/gmfoster/Capstone/commit/72630ccad5d38a4e1eb5b2be8ce301e92fa6d1d4
12. As a **user** I will be able to enter my **bank account information** to search the web and dark web for my information
    a. Acceptance Test: Upon entering bank account information and selecting search, the user is presented with a text file containing all breaches and the location of each breach
13. As a **user** I will be able to see the closest known location where my information is being stored/accessed and have a visual representation on a map.
    a. Acceptance Test: In the report returned to the user they will receive a map with pins representing the physical location of where their information is stored, if available.
14. As a **user** I will be able to choose between basic search (name, email, addresses) and advanced search (social security number, banking info, name email, address, ip address, or user specification of those)

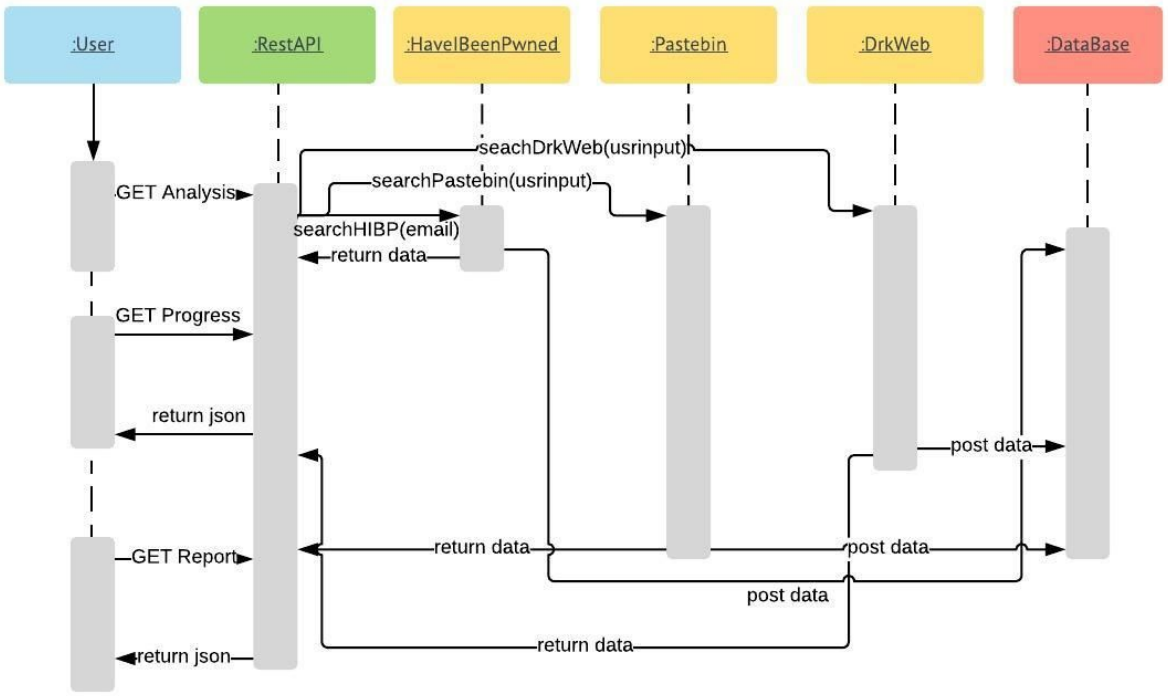
    a. Acceptance Test: Upon logging into the site with correct credentials I will be able to enter my respective search and receive a report regarding the information input.
15. As a user, I will be able to access the homepage from any part of the web. This will end any current report being developed.
    a. Acceptance Test: At all times pressing the home button will return me to the home page.
    b. https://github.com/gmfoster/Capstone/commit/429f3ae238e15864a253e7eff14e3f12f5f512ba
16. As a **user**, I will be able to receive a report once all the modules have completed their searches.
    a. Acceptance Test: Once the search has been completed I will automatically have an updated report of all the information that has been found.
17. As a **user**, I will be able to enter As a **user** I will be able to enter my **keyword** to determine if my information has been compromised through data leaks
    a. Acceptance Test: Upon entering email address, company name, or any other keyword and selecting search, the user is presented with a text file containing all breaches and background information related to each breach.
    b. https://github.com/gmfoster/Capstone/commit/0bd55e4333a3f6024cec41fbe033ce0ed5829518
18. As a **user**, I will be able to receive a report based of my email from the web application with a consistent format based of a json document passed from REST API, when in a report URL.
    a. Acceptance Test: When in a report phase I will be able to check each of the searches that were made and look at reports from each of these
    b. https://github.com/gmfoster/Capstone/commit/8b3253c0fc6092d9d8cbcdc4ca89097e52c25bda
19. As a **user**, I will be able to access a query bar in the home page.
    a. Acceptance Test: As a user, upon logging into the site I can successfully click on and enter keywords into the query bar
20. As a **user**, I will be to look at previous searches that I have made.
    a. Acceptance Test: As a user, upon logging into the site I can click on the previous reports tab and view all previous reports I have generated.

7. System Models

7.1. Structural UML Diagram



7.2 Sequence Diagram



8. Appendices

8.1 Technologies Employed

- Flask
- AWS EC2
- Jenkins
- Tor w/ Selenium
- React/HTML/CSS
- HaveIBeenPwned API
- Pastebin API
- Docker