Production Requirements Document

Team: Human Error is a Myth
Project: NovaSight

Team Members: Graham Foster, Blake Johnson, Fernando Mendoza, Diego Segundo, Omer Cohen, Quintin Hill

Project Background

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. Have I been pwned is one of the few. 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.

There are many other websites where individual's and company's information is leaked, for example pastebin. It is clear that I reliable way to determine whether or not you or your company's personal data has been exposed.

Innovation and Objective

There is currently no single application that aggregates existing data exposure tools. We are seeking to create a single web application that, based off of several user inputs: email, names, web address, will determine whether or not relevant information related to that input has been exposed. Our application will return a text report containing information about each respective leak. 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. We envision our MVP as a web application, which could eventually ported over to a mobile application.

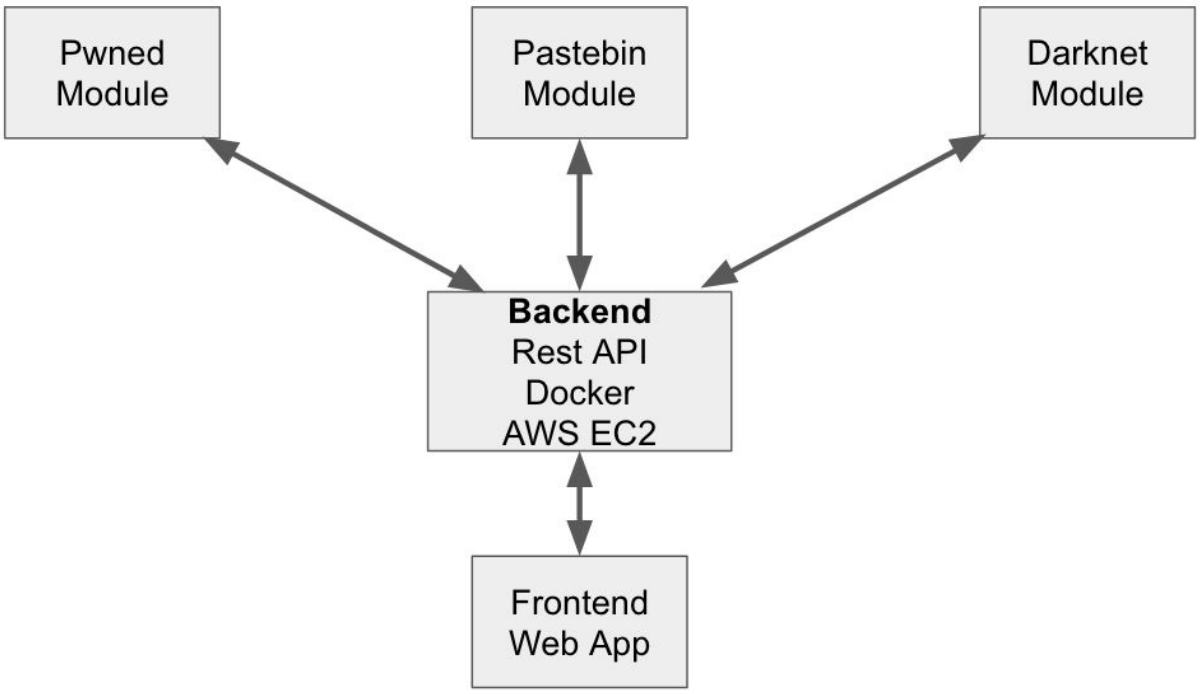
Milestones

- 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

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.

High Level System Diagram



Prototyping Github Commits:
https://github.com/gmfoster/Capstone/tree/master/Prototyping

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
    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. **Use Case**: Receive Report
   Actors: User, AWS Server, Flask Applications, WebApp, ….(*need more here)*
   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 webscrape 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
5. As a **user** I will be able to enter my **social security number** to search the web and 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
6. As a **user** I will be able to enter my **IP address** to search the web and dark web for my information
    a. Acceptance Test: Upon entering IP address and selecting search, the user is returned all available information regarding their IP address

7. 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
8. 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
9. 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
10. 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

Appendices

Technologies Employed

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