LAST EDITED 02/09/2023

Writing 4
Olivia Legault, Miles Grant, Rubin Roy
CODE V WADE

I. Executive Summary

A. Background

Before the ink was dry on the Supreme Court's June 2022 ruling that overturned Roe v. Wade, millions of American women and people with uteruses became concerned about the fate of their reproductive rights. Amongst the chaos and dismay in the immediate aftermath of the ruling, people began to question how mobile applications which collect reproductive health information, such as period and pregnancy trackers, would handle the data that they had accumulated. With abortion becoming illegal in state after state, many fear that the information entrusted to these apps and services may be used by law enforcement or other entities to persecute and victimize people based on their private health information.

B. Who We Are

Code v. Wade is a team of researchers attempting to tackle the issues caused by misinformation and obscurity in the reproductive health tech landscape. Our project will involve two sequential phases: first, a user study to answer research questions regarding the collection, use, storage, and sharing of sensitive data by reproductive health applications, and second, a publicly available web service that uses machine learning to classify and rate reproductive health apps based on how well they preserve consumers' privacy.

C. Research Questions

For the research phase of our project, we hope to answer the following three research questions in order to inform our privacy classification algorithm. The first question we aim to answer is "How aware is the general public regarding the use and collection of health-related data by reproductive health applications?" As our team is composed of computer scientists and privacy researchers, we recognize that there are certain problems and topics that we may know more about than the average American relating to data governance and privacy issues. Likewise, as a small team of college students, we are excited to learn about the experiences of people with different backgrounds. Hearing their input will allow us to broaden our knowledge and make sure our product reflects the views and concerns of a more diverse group of people.

The second question we hope to answer is "What information do consumers desire to have regarding the use/collection of their reproductive health data?" For our service to be a practical and helpful repository of knowledge regarding reproductive health privacy, we must ensure that the information we provide is relevant, complete, and easy to understand. In order to gauge the types of information that Americans are seeking, we will ask respondents to weigh their interests and concerns about a range of topics related to health privacy and transparency.

We will also ask participants to provide us with any concerns that they have that were not mentioned in the study.

The third question at issue in the study is “What threat models and potential use cases for reproductive health data do consumers identify and which of these potential threats are the most impactful on consumer’s concerns?” The state of reproductive health laws varies widely across the country and is changing rapidly. Thus, the potential harm facing those using reproductive health apps varies widely based on state, city, age, gender identity, sexual orientation, and myriad other factors. Understanding how users anticipate their data being used against them will help inform how we weigh certain data practices in our calculation of privacy risks.

D. Proposed Solution

CODE V WADE is a two-part solution approach to this complex landscape. In part one, we will conduct a comprehensive and anonymous research survey to thoroughly assess user concerns and gather data that will inform our service. This research is absolutely essential to inform how we assess the security and privacy of different apps connected to reproductive health data. Understanding relevant, specific user concerns allows us to apply a machine learning approach to understanding the effectiveness of the stated app data privacy policies. This machine learning approach applies natural language processing (NLP) to the privacy policies of dozens of reproductive healthcare apps. In combination with our user study results, the output of our NLP analysis will be a decisive ranking for each individual app according to how effectively it protects user privacy.

II. Technical Design

A. Summary

The key innovations of our project are our synthesis of many approaches to studying mobile app privacy as well as our ability to dive deeper into specific data practices because of our specific focus on reproductive health applications. Each of the three team members will be in charge of one of these general categories, with myself taking the lead on the App Analysis. We believe that this project structure is appropriate because it allows each of us to specialize and focus on a particular area while also enabling us to be agile and help in other areas depending on the difficulty of certain tasks or hang ups with scheduling.

B. Calculating the Score

1. App Store Scraping
    a. Using the Python tools “app-store-scraper” and “google-play-scraper” we will collect information about each app we are reviewing from the Google Play and Apple App stores.
    b. The information will be stored as tags for data privacy labels, developer information, data storage practices, and data sharing with 3rd parties.

    c. This information will be stored in MongoDB.

2. Natural Language Processing
    a. This method will parse app privacy policies for information about privacy practices.
    b. The training data will be assembled from the PrincetonU dataset, consisting of 130K privacy policies, and the Polisis dataset, which consists of over a million privacy policies. This training data will then be used to test our reproductive app policies for information such as data sharing and storage practices.
    c. We will be writing a modified version of Polisis to analyze our reproductive health apps and will return similar parameters to the App Store scraping, such as tags on data storage, data sharing, etc.
    d. Data will also be stored on MongoDB
3. Packet Analysis
    a. Using a virtual Android phone and dummy test information, we will run a standardized series of tasks, including creating an account and altering privacy settings, on each app, including testing secure data transmission and secure storage.
    b. We will also be testing outgoing information for data brokers, advertisers, etc, as well as malware and other malicious applications.
    c. We will extract the packet headers and bodies from packet logs and store them in MongoDB.
    d. We will create a procedure for 10-15 "types" of apps that specifically list the steps that allow for the app to go through the steps so we can gather appropriate data on them (e.g. create an account if that is allowed, delete the account, enter a period cycle, etc.)

III. Project Description

A. User Stories

a. As a user of reproductive health app(s), I would like to know how well these apps are preserving my privacy and sensitive health data, so that I can make decisions about which apps most accurately match my needs and desires on data privacy.

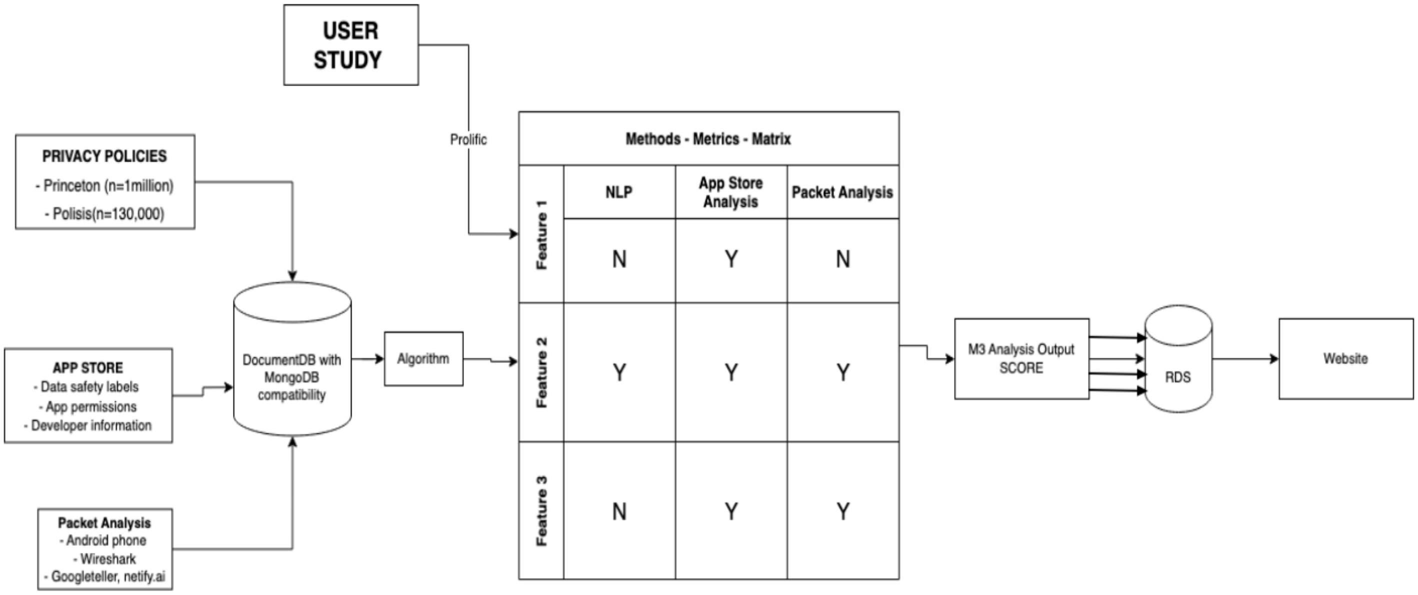
b. As a researcher or journalist, I would like to have a "one-stop shop" for analyzing reproductive health app privacy to inform future studies and allow comparison between different applications.

c. As a pro-choice advocate, I would like to have a platform for evaluating app privacy to inform my recommendations and policy positions.

B. Basic UI Components:

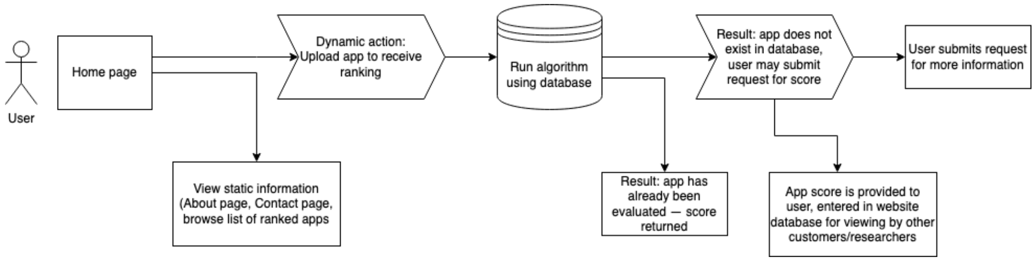
1. The website is hosted via AWS EC2. The user can either: 1.) look up existing results for app privacy scores or 2.) upload information for a new app to be evaluated.
2. The website will be backed by a SQL RDS database. This database will store all apps that have been ranked and their corresponding scores. If a user searches existing rankings, they will be interacting with this database. Similarly, if a user uploads information for an app to be analyzed, the RDS database will be queried with the inputted information to see if the app has already been ranked. If so, the user will be directed to the ranking page for the app in question.
3. If the user instead uploads an app that has not yet been analyzed, then we will be using MongoDB, a NoSQL database. This will be used for a few purposes. First, to store the training dataset for the NLP, and to store the information that we gather about the app (including app store reviews, privacy policies). Furthermore, the logs generated when we run dynamic code analysis are also stored in MongoDB.
4. These two are not currently integrated in a visual UI component, but they will be as work on the project continues. The main UI will be focusing on user accessibility through two components: easy navigation and transparent information.

D. Overall Project Architecture and Design:

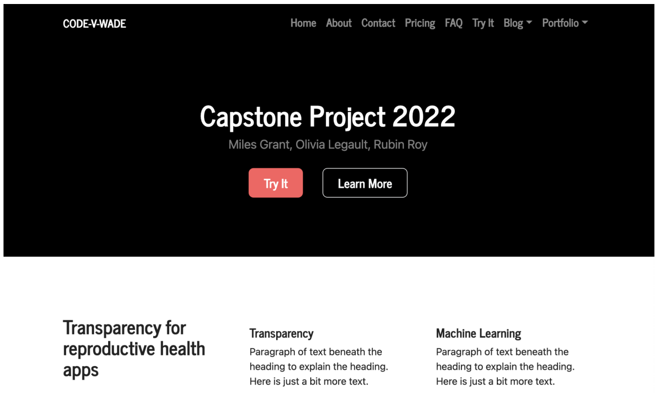


Methods - Metrics - Matrix

	NLP	App Store Analysis	Packet Analysis
Feature 1	N	Y	N
Feature 2	Y	Y	Y
Feature 3	N	Y	Y


User
Home page
Dynamic action: Upload app to receive ranking
View static information (About page, Contact page, browse list of ranked apps
Run algorithm using database
Result: app has already been evaluated — score returned
Result: app does not exist in database, user may submit request for score
User submits request for more information
App score is provided to user, entered in website database for viewing by other customers/researchers

LAST EDITED 02/09/2023



CODE-V-WADE

Home About Contact Pricing FAQ Try It Blog Portfolio

Capstone Project 2022

Miles Grant, Olivia Legault, Rubin Roy

Try It | Learn More

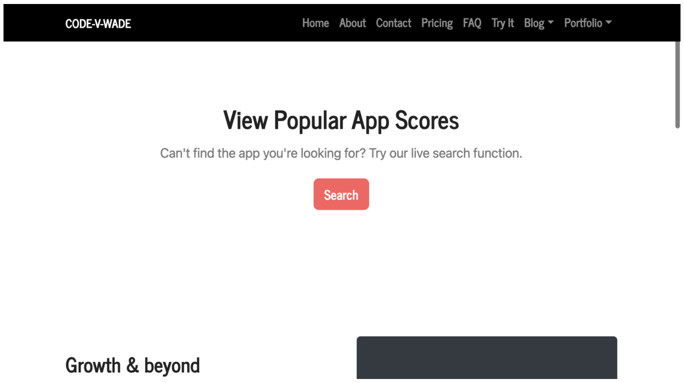
Transparency for reproductive health apps

Transparency

Paragraph of text beneath the heading to explain the heading. Here is just a bit more text.

Machine Learning

Paragraph of text beneath the heading to explain the heading. Here is just a bit more text.



CODE-V-WADE

Home About Contact Pricing FAQ Try It Blog Portfolio

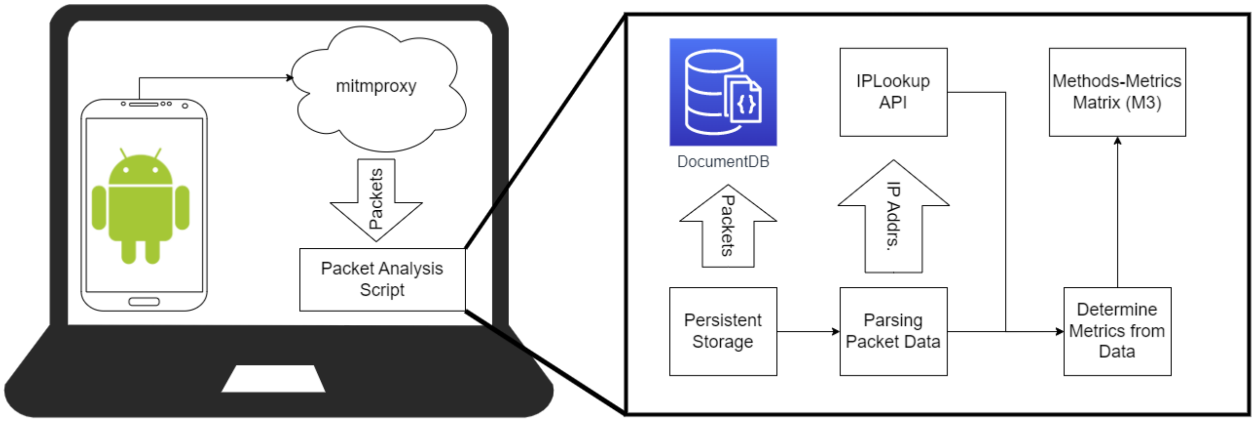
View Popular App Scores

Can't find the app you're looking for? Try our live search function.

Search

Growth & beyond

1. Packet Analysis Architecture:



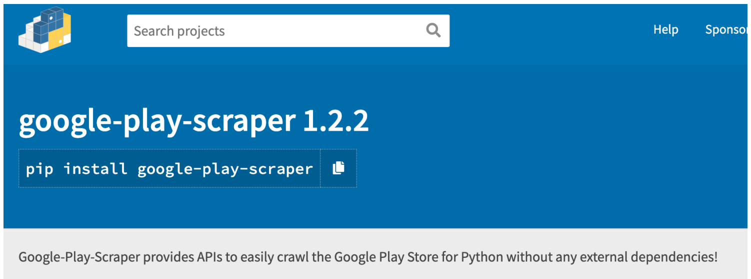
E. External APIs and Frameworks

1. For the natural language processing (NLP) that we are applying to the app privacy policies, we will be using some external APIs. To store the corpus of text that will be used as the training dataset, we will be using MongoDB, a NoSQL database specifically designed for fast document storage for NLP. The datasets we will be using are the Princeton University dataset of privacy agreements and the Polisis dataset.
2. For the analysis itself, we will be using the open-source version of Polisis, which is designed for analysis and classification of privacy policies. The algorithm itself will be run on our server backend (i.e. at this time we do not intend for the NLP to be an API call especially due to the algorithmic component discussed below).
3. To analyze the network traffic on the apps we will be performing packet analysis. We will use an Android Virtual Device (AVD) to emulate a smartphone running each RHA, and an application called *mitmproxy* to intercept and decrypt the packets in transit. The packet data will be stored on our MongoDB and we will write an algorithm to parse the information and compare the destination IP addresses to known third parties using APIs such as IP-API or ipstack.

4. The google-play-scraper API will be used to extract basic app store information and load it into the database for processing.

F. Algorithms

5. The key algorithmic component of the NLP that we are using is the use of a new classification category for app privacy policies. This is because the current Polisis framework only supports general app privacy labels (e.g. shares user data first-party, etc.) We intend to classify policies with reproductive health-specific labels such as “tracks period cycles”. We hope to do so by introducing the Princeton University dataset for training purposes. During training, we will be minimizing the error of the NLP on these new classification categories using a training set of hand-annotated policies.
6. We also introduce an algorithmic component in the production of the final score itself. Based on the user study, we will have a series of features that customers have indicated are important (e.g. does not share with law enforcement, etc). For each of these questions, if any of the methods we use (NLP, dynamic analysis, app store labels) indicate a positive answer, then the feature is marked as a yes. The features are negative (i.e. an optimal app would answer no to each feature) so once a feature is determined to be “yes”, it is subtracted from the initial score of “10”. Thus, the lower the score, the less private the app itself is.



google-play-scraper 1.2.2

```
pip install google-play-scraper
```

Google-Play-Scraper provides APIs to easily crawl the Google Play Store for Python without any external dependencies!