Technology and Intimate Partner Violence

Advika Nigam
Cornell Tech, Cornell University
New York
an556@cornell.edu

Anmol Seth
Cornell Tech, Cornell University
New York
as3664@cornell.edu

Summer Shi
Cornell Tech, Cornell University
New York
ys838@cornell.edu

Nicki Dell
Cornell Tech, Cornell University
New York
nixdell@cornell.edu

Abstract

Prior work done to tackle intimate partner surveillance has resulted in a system that scans an Intimate Partner Violence (IPV) victim's phone for spyware installed by the abuser. This paper describes the design and development process of a web/native application that allows users, IPV professional in this case, to interact with such systems and generate appropriate reports and/or notify the victim about their findings.

We plan to use HCI and design concepts to prototype our solution which would have features including but not limited to spyware detection on victim's phone.

Introduction

Intimate partner violence affects roughly one-third of all women and one-sixth of all men in the United States. We aim to improve the spyware functionality that was previously developed by certain students and faculty in the field to help victims and survivors of intimate partner violence detect spyware. The previous phone scanner system called the Android debugging tool (ADB) crawled through the existing applications on user's phone and categorize the list of apps based on a list of known spyware, malware and dual-uses location tracking apps using Machine Learning model. Therefore the previous framework lacked an appropriate interface and structure for the user to take further actions on various spyware applications found. We focused on developing a web application with a login system for the Case Workers where they could access their existing clients, or add a new client or scan the victim's phone anonymously. Post scanning, Case Workers could delete the Spyware Applications found or generate a Forensic PDF on Spyware found or obtain important meta-data around it or send emails containing the report. Although the core

technology of this tool could benefit several parties: survivors, victims, potential victims, social workers and forensics, there should be different features for each one of them considering their situations.

Therefore, our goal is to launch this anti-spyware tool first as a web application with a decent user interface for the Family Justice Centre case workers. Upon validation and feedback, we can aim to expand our work to different settings and ecosystems.

Research question

Our Research questions revolves around understanding the appropriate level of interaction between our system and the needs of the Family Justice Case workers to help victims of Intimate Partner Violence.

How can we create a spyware scanning tool that is usable and appropriate for professionals who work with victims of intimate partner violence? What scanning modes, actions, and other functionality need to be implemented to make the tool deployable with case workers in NYC's Family Justice Centers?

Using human-computer interaction methods, we aimed to answer the questions above.

Related Work

Prior research suggests that detection and removal spyware used in intimate partner surveillance is a vastly unexplored field. We analyzed the ecosystem of an FJC and the requirements of a professional working there.

Intimate partner violence (IPV) includes physical or sexual violence, stalking, or psychological harm by a current or former intimate partner or spouse. Much of the IPV literature discusses the installation of IPS apps on a victim's mobile device (2). Interviews of survivors and professionals working with them indicate that abusers easily find such spyware via web search, and that many otherwise innocuous apps, such as find- my-phone apps and child trackers, are easily repurposed by abusers for spying on intimate partners.

Deriving inspiration from the literature review done, our focus is on apps abusers purposefully install in order to stalk, monitor, and control an intimate partner's device without consent.

Method

For the first two weeks, we read related research papers and gathered related literature review to understand the prior work done already done. We then focused on understanding the phone scanner machine learning code already built. We aimed to explore existing anti-spyware tools to understand how our tool would be unique to the Family Justice Centre workers. We came up with either building a mobile application, web application or a local application on their systems. We converged on creating a local web application since there was a limitation of network connections in FJCs and our potential users did not have high sophistication in technology. A local web application does not require any installation and we could easily control access by granting username-password combinations.

We started with storyboarding and paper prototyping and defining how we wanted our system to look. We spent around 3 weeks iterating our prototype and deciding the user flow with the feedback we got from our advisor and the PhD students. The feedback included adding an anonymous client mode, existing client mode, adding a disclaimer which stated what our tool did and adding buttons in our prototype which could directly uninstall the application. We then finalized on the digital prototype in Figma shown below in the appendix.

In Early March, we began developing a high-fidelity prototype in the form of web application using HTML and CSS and JavaScript. At the same time, we identified key entities and any relationships between them to come up with a database schema. Once we had our front end and our database schema ready, we focused on understanding the phone scanner system and serving it out as a local web application with a database that records and reviews all the functions of the phone scanner system.

We showed our Minimum Viable Product and took feedback from two participants who have worked closely with the Family Justice Centers. Based on their suggestions we developed more features like creating a PDF, making a notes section and an alert while sending out emails containing confidential information. More details on the entire tech stack and our architecture are enlisted below.

Current Results

Tech Stack:

We used Flask, a micro web development framework, as a server for our backend which uses iOS-deploy and ADB to crawl the apple AppStore and google play store. It also interacts with the database to get and post data or scans for a new or existing client.

On the front end we use HTML, CSS, JavaScript, jQuery, bootstrap on the front end to allow users (FJC professionals) to run the phone scanner on victim's devices.
We used SQLite for a self-contained SQL database engine and dataset which provides a simple abstraction layer over the database.

Functionality:

The system has three modes in total: (1) New Client; (2) Existing Client; (3) Anonymous. FJC case workers may choose to create a profile for new client and then scan his or her phone on record, complete a new scan on record for an existing client, or complete a scan for any client off record according to his or her will. The client will be asked to agree to a disclaimer before proceeding if their phone scanning results are saved. The phone scanning system is able to scan applications on most phones that use iOS or Android. The scanning results will be compared to a csv file that contains a list of spyware applications. Every possible spyware on the phone will be flagged with a specific category.

Architecture Description:

The user interface, which FJC case workers can directly interact with, was created using JavaScript, CSS, HTML and Bootstrap. The UI will send requests to Flask server, through which the UI can receive corresponding results from SQLite database or connected devices. To return a list of potential spyware applications on the phone, the server will compare all applications on the phone with local source files that were produced with the help of machine learning algorithm.

Final Result:

We have been able to create a functional tool which is ready for field testing. Screenshots of the Figma prototype that represents our user flows of the final web application are shown in Appendix B. ER Diagram and the architecture diagram of the system are in Appendix A.

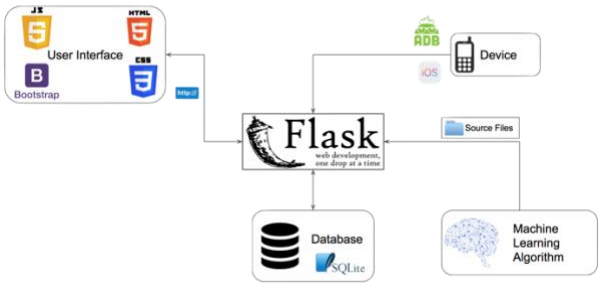
Future Work

Our future work includes deploying our spyware application in the Family Justice Centers locally, thus conducting field study of the spyware application. We would aim to collect feedback from around twenty people. Gauging their reaction, we may focus on deploying it as it is, or developing more features, and improvising it. If we get feedback to refine the product, that would be our focus for next semester. We would strive to conduct interviews with lawyers and the police force to figure out what they would need for legal prosecution of the abusers. With this, we can use the information in court and help the domestically abused women. Therefore we could add more features like collecting audio/video/photo evidence on the application too. If not, we would explore different possibilities and use cases of our tool into opening up a "tech clinic" at Cornell University. People could approach us and get free advice on any spyware installed on their phone. There is a lot of scope to our research and depending on the feedback, we could explore different aspects.

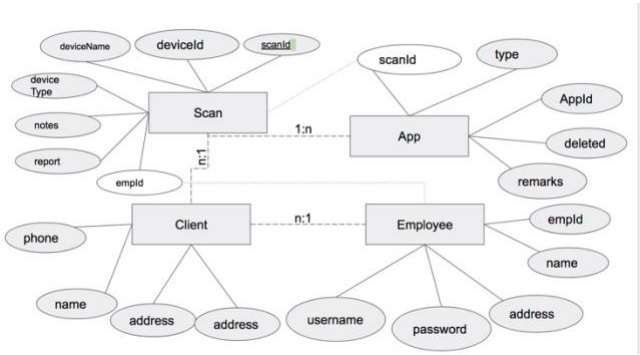
References

1. D. Freed, J. Palmer, D. Minchala, K. Levy, T. Ristenpart and N. Dell. 2017. Digital Technologies and Intimate Partner Violence: A Qualitative Analysis with Multiple Stakeholders. DOI: http://www.nixdell.com/papers/digital-technologies-intimate.pd
2. R. Chatterjee, P. Doerfler, H. Orgad, S. Havron, J. Palmer, D. Freed, K. Levy, N. Dell, D. McCoy, and T. Ristenpart. 2018. The Spyware Used in Intimate Partner Violence IEEE Symposium on Security and Privacy (Oakland 2018)
3. D. Freed, J. Palmer, D. Minchala, K. Levy, T. Ristenpart and N. Dell. 2017. A Stalker's Paradise: How Intimate Partner Abusers Exploit Technology. DOI: http://nixdell.com/papers/stalkers-paradise-intimate.pdf

Appendix A

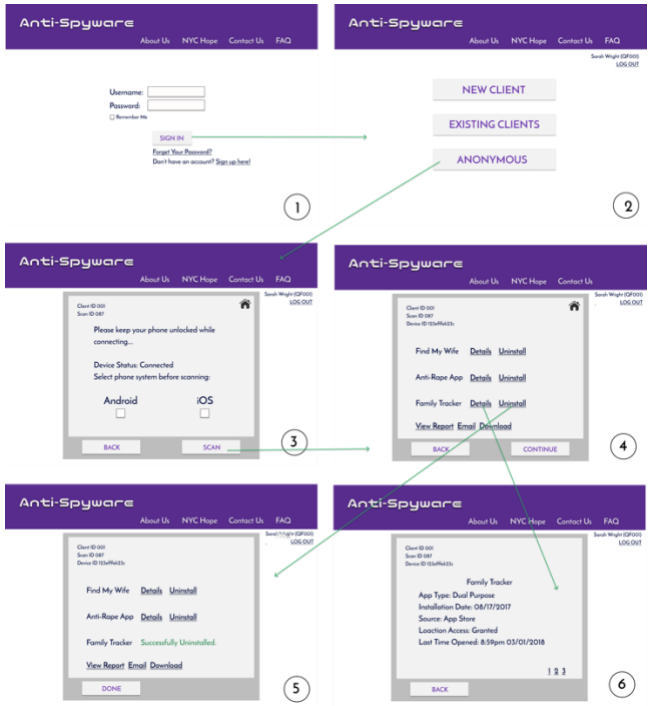


Architecture Diagram

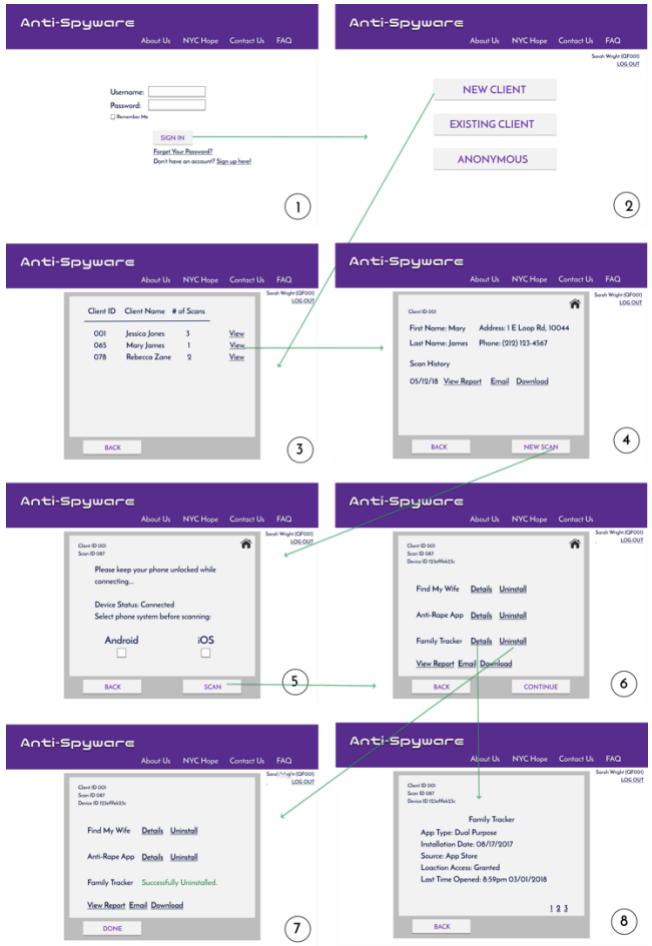


ER Diagram

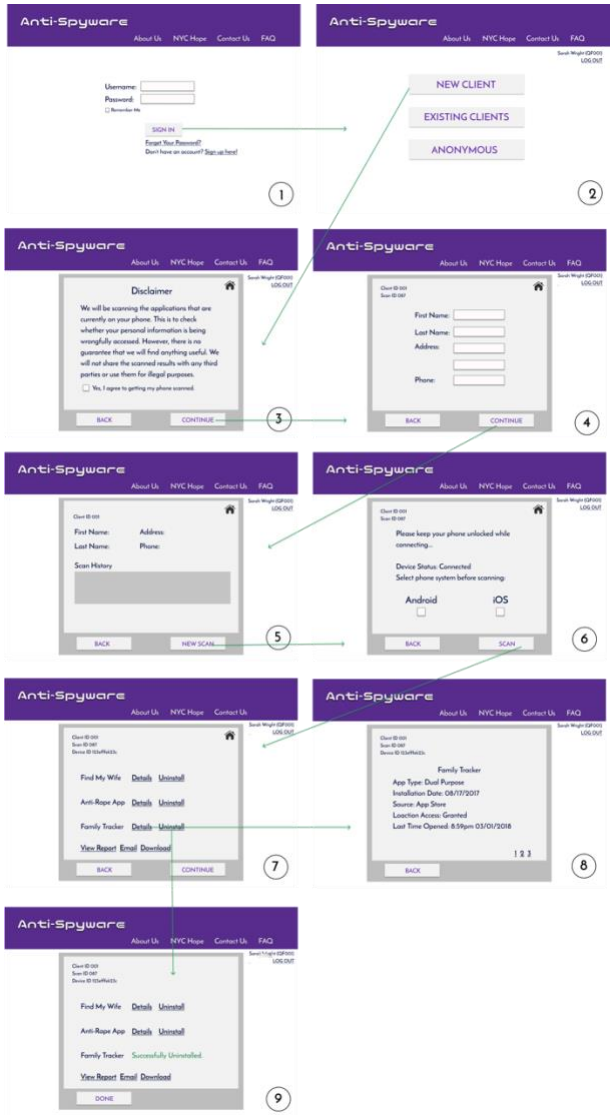
Appendix B



Anonymous Mode's User Flow



Existing Client Mode's User Flow



New Client Mode’s User Flow