AN EXPOSURE OF ANDROID SOCIAL MEDIA APPLICATIONS
AND THE DATA ISPS CAN COLLECT

A Thesis in
Computer Science and Engineering
by
Daniel E. Krych

© 2017 Daniel E. Krych

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Master of Science

December 2017
The thesis of Daniel E. Krych was reviewed and approved* by the following:

Patrick D. McDaniel  
Distinguished Professor of Computer Science and Engineering  
Thesis Advisor

Sencun Zhu  
Associate Professor of Computer Science and Engineering

David Reitter  
Assistant Professor of Information Sciences and Technology

Chitaranjan Das  
Distinguished Professor of Computer Science and Engineering  
Head of the Department of Computer Science and Engineering

*Signatures are on file in the Graduate School.
Abstract

Social media applications play a key role in our day-to-day lives, using and handling sensitive/private data in their nature. Android applications continue to share data with third parties and transmit data unencrypted, leaking data directly and inadvertently. Internet Service Providers (ISPs) can legally collect and sell this leaked, sensitive user data to ad companies and third parties, and since ISPs available to users vary based on geolocation, users may be unable to avoid these providers. End user’s rely on privacy policies to understand how their data is used, but these have been largely absent, and those present lacked detail, especially in their security methods. Inconsistencies between app policies and their actions have been found through static code analysis and dynamic analysis, but studies lacked depth and/or the context of the application. Overall, we lack a detailed understanding of the state of these privacy and security issues within sensitive settings such as social media applications. We aim to expose Android social media applications, classifying and comparing each app’s unencrypted data transmitted with the disclosure in their privacy policy. We develop an analysis framework and isolated testbed environments, which use a variety of open-source tools, and enable accurate data collection. We use dynamic analysis to obtain the behaviors from traffic, and leverage the Platform for Privacy Preferences (P3P) Specification to bridge the connection between these behaviors and disclosure in the privacy policies, while considering the context of each application [1]. We find inconsistencies between applications behaviors and disclosure in their privacy policies. The majority of the applications in our dataset transmitted more than half of their traffic unencrypted. Few apps leaked a large portion of Personally Identifiable Information (PII)/sensitive data, and others used more encryption to protect user data, but none detailed security methods or specified which data was transmitted encrypted. In addition, we peer into application’s privacy policy revisions, the advertising/analytics libraries applications used, and business relationships held by application companies. We conclude that despite Android applications being on the market for over eight years, failure to protect sensitive user data and vague privacy policies are still prevalent.
# Table of Contents

List of Figures .......................................................... vii

List of Tables ............................................................ ix

Acknowledgments .......................................................... x

Chapter 1
Introduction ............................................................... 1

Chapter 2
Related Works .............................................................. 6
  2.1 Privacy Policy Analysis .............................................. 6
  2.2 Application Traffic Analysis ....................................... 8
  2.3 Comparing Actions to Policies .................................... 10

Chapter 3
Methodology ................................................................. 13
  3.1 Application Dataset .................................................. 13
  3.2 Testbed Environment for App Analysis .......................... 15
  3.3 Analysis Framework .................................................. 21
    3.3.1 Application Analysis Module .................................. 21
      3.3.1.1 Application Testing / Collection .......................... 21
      3.3.1.2 Traffic Analysis .......................................... 21
    3.3.2 Policies Module ................................................ 22
      3.3.2.1 Privacy Policy Analysis .................................. 22
      3.3.2.2 Privacy Policy Revisions ................................ 22
    3.3.3 Data Sharing Module ............................................ 22
      3.3.3.1 Advertising/Analytics Libraries .......................... 22
      3.3.3.2 Business Relationships ................................... 23
List of Figures

3.1 **Analysis Framework.** Arrows represent the transfer of data and analysis information between the modules and components within the Analysis Framework. Red arrows with dotted lines, white arrows with hollow lines, and blue arrows with solid lines represent the data transferred by the *Application Analysis*, *Policies*, and *Data Sharing* modules, respectively. ................................................................. 14

3.2 Our Testbed Environments for the Emulator and Rooted Phone Experiments ................................................................. 16

3.3 Analysis of Vine Traffic - Android Emulator ........................................ 18

3.4 Network Load for Vine Traffic - Android Emulator ......................... 19

4.1 Analysis of Textfree Traffic - Rooted Nexus 6P .............................. 28

4.2 Picture Sent to Personal Phone Number in Textfree App Unencrypted During Feb 2017 Test ......................................................... 37

4.3 Picture Received in Textfree App From Personal Phone Number and Pulled Down via HTTP From Host Server During Feb 2017 Test 37

4.4 Wireshark’s VoIP Flow Analysis Graph of Bidirectional Textfree Phone Calls Made over SIP in the 2/16/17 Rooted Phone Test .... 40

C.1 Analysis of Facebook Traffic - Android Emulator .......................... 60

C.2 Analysis of Imgur Traffic - Android Emulator ............................... 61

C.3 Analysis of Messenger Traffic - Android Emulator ........................ 61

C.4 Analysis of Pinterest Traffic - Android Emulator ............................ 62

C.5 Analysis of Textfree Test 1 Traffic - Android Emulator .................. 62

C.6 Analysis of Textfree Test 2 Traffic - Android Emulator .................. 63

C.7 Analysis of Tumblr Traffic - Android Emulator ............................. 63

C.8 Analysis of Twitter Traffic - Android Emulator .............................. 64

C.9 Analysis of Vine Traffic - Android Emulator ................................. 64

C.10 Analysis of Pinterest Traffic - Rooted Nexus 6P ........................... 65

C.11 Analysis of Textfree Traffic - Rooted Nexus 6P ............................. 65
C.12 Analysis of Textfree MITM (arpspoof) Attack Traffic - Rooted Nexus 6P .................................................. 66
C.13 Analysis of Textfree Retested Feb 2017 Traffic - Rooted Nexus 6P .................................................. 66
C.14 Analysis of Tinder Traffic - Rooted Nexus 6P ................................................................. 67
C.15 Analysis of Vine Traffic - Rooted Nexus 6P ................................................................. 67
## List of Tables

3.1 **Social Media Applications Tested.** Apr-2016 and Apr-2017 columns provide the number of cumulative downloads in millions (M) or billions (B) at this point in time. . . . . . . . . . . . . . . . 15

4.1 Breakdown of Application Traffic Using HTTP, HTTPS, and DNS . 25
4.2 Abbreviations of P3P Specification Categories and Purposes . . . . 26
4.3 **Classifying Application Traffic Data with P3P and Determining Policy Disclosure.** The abbreviation key in Table 4.2 provides the P3P Categories and Purposes, and their descriptions can be found in Appendix A. The list of file extensions found in the unencrypted traffic per application are listed below the application’s name. Asterisks (*) are used to denote borderline instances where the disclosure or classification could be argued, and our best judgement was used. This includes instances of data transmitted that lacked disclosure, and thus the P3P Purpose is inferred. . . . . 27
4.4 Prominent Advertising/Analytics Libraries Used By Each Application 44
Acknowledgments

I would like to thank Meghan Riegel and Charles Sestito for all of their work and help with this research: collecting application traffic; developing scripts to automate the collection process; and investigating business relationships and ad/analytics libraries.

I’d like to especially thank my advisor Patrick McDaniel for all of his guidance and encouragement throughout this research, and over the past several years as a member of the SIIS Lab. I’d also like to thank Berkay Celik for his suggestions and guidance as well. I’d like to thank Ryan Sheatsley for retargeting and decompiling the APKs using Dare and Soot, and developing scripts to obtain the ad and analytics libraries as well as the namespaces used by each application. Additionally, I’d like to thank Raquel Alvarez for her help with proofreading, as well as the entire SIIS lab for their support.

Finally, and most importantly I’d like to thank my parents. They instilled in me their passion for life-long learning, and support me unconditionally.

Research was sponsored by the Army Research Laboratory and was accomplished under Cooperative Agreement Number W911NF-13-2-0045 (ARL Cyber Security CRA). The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on. This research was also partially sponsored by NSF Grant No. CNS-1228700 and CNS-1064900. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.
Dedication

For my parents.
Thesis Statement

Android social media apps will continue to exhibit the trend of the collection and direct/inadvertent exposure of privacy sensitive information, as well as lack proper disclosure in their privacy policies.
Chapter 1  
Introduction

Social media applications collect, display, and share personal information in their nature to provide end users with a platform to interact. Users trust these applications with their information. This includes Personally Identifiable Information (PII) such as health information, location data, SSN, or unique device IDs. It also includes sensitive information which may or may not be classified as PII such as email addresses, phone numbers, interests, usage habits, age, and gender. Android applications have been found to transmit sensitive data unencrypted and share PII with third parties and advertising companies [2] [3] [4] [5] [6] [7] [8] [9]. These instances of data mishandling may violate the privacy and security of end users, and trends seem to indicate a continuation of these practices. Any data transmitted unencrypted results in inadvertent exposure to the network, and all of the points along the network path. Internet Service Providers (ISPs) have been legally and physically capable of collecting and selling sensitive user data to advertising companies and third parties [10]. Overall, we have a limited visibility into what these application companies and other parties on the networks are doing behind-the-scenes with user data.

Users can review an application’s privacy policy which, should determine what data are being collected, how it is being transmitted/handled, and who has access. But, previous studies have noted the absence of application privacy policies, and the difficulty in finding those that did exist [7] [8] [11] [12]. Furthermore, the majority of privacy policies only provide general information about their practices, if they have one at all. Recent years showed an improvement in the presence of
a policy, but a lack of detail in data-handling and security practices seems to be
the norm [12]. Additionally, inconsistencies between companies’ privacy policies
and the behaviors of their applications have been found for a variety of different
Android apps [8] [13] [14] [15]. End users may not agree with companies on which
types of data are worthy of encryption and which should be shared to third parties.
Applications often use advertising and analytic libraries which want to obtain as
much information on users as possible. These application developers themselves
are unaware of the data these services collect [16]. If the developers do not know
this information, how can they define their practices in their policies, and how can
users know how their personal data is used? Little insight exists into whether the
security and privacy of end users’ data is improving or declining.

In this paper, we investigate popular Android social media applications. We
hypothesize that these applications will continue to exhibit the trend of the col-
lection and direct/inadvertent exposure of privacy sensitive information, as well
as lack proper disclosure in their privacy policies. In order to determine this we
analyze each application’s data transmission practices, privacy policies, advertis-
ing/analytics libraries used, and business relationships. We compare our findings
to the applications’ privacy policies; determining how their actions match their
disclosure. We leverage dynamic analysis and traffic analysis to expose the security
and privacy concerns of applications transmitting data in the clear, or unencrypted,
and the implications which arise.

Our analysis framework entails three modules that each output security and
privacy concerns of user’s information. The Application Analysis module collects,
parses, and analyzes the application traffic to determine the application’s behaviors
and the types of data transmitted unencrypted. The Policies module analyzes the
privacy policies of each application. The outputs of these two modules is manually
compared, leveraging the P3P Specification terminology to find inconsistencies
between an application’s behaviors and it’s policy [1]. The Data Sharing module
also leverages the output of the Application Analysis to analyze the advertising and
analytics libraries used by each application as well as the business relationships
between application companies as well as third parties. For more information on
the analysis framework, see the Methodology section.

We find the majority of the applications analyzed to transmit most of their data
unencrypted over HTTP. Furthermore, none of the application’s privacy policies
specify which data is sent unencrypted vs. encrypted. We also find inconsistencies between application’s behaviors and the disclosure in their privacy policies. Thus, end users rely on vague, and sometimes misleading documentation to determine how their data is used and handled. In general our results show that the applications which mostly used HTTPS had fewer data leaks as expected. But, the application which transmitted the most PII unencrypted in our emulated environment tests used HTTPS for the majority of its transmissions. Overall, these transmissions detail which types of data these companies determined were worthy of encryption. We find one application transmitting PII and sensitive data over HTTP, including but not limited to message content, location data, and phone numbers. Multiple applications leak application usage trails and the interests of the end user by transmitting pictures, videos, and webpages as they were viewed in the application. Several applications transmit location information, some more granular than others. On top of these information leaks, more data and conclusions may be gathered through inference attacks. Previous work found similar results, without specifying unencrypted data from encrypted data, and noted how this information is valuable to advertisers [5]. The profiling of users, targeted advertising, and formation of an attack surface are all things that could follow from such data leaks.

The transmission of data unencrypted results in inadvertent exposure to the network. Every hop has the ability to collect this information, and Internet Service Providers (ISPs) have been known to capitalize on this fact. For a short time in 2016, AT&T allowed users to pay for privacy by opting out of its “Internet Preferences program” for additional fees between $531 and $800 per year. This program collected, tracked, and monetized user’s behaviors across the Internet, using deep-packet inspection to obtain data. This pay-for-privacy plan was heavily criticized then, and the extra fees were removed, but AT&T and Comcast have both publicly justified and expressed interest in standardizing these programs in the near future [17] [18]. The data collected by these ISPs could be used for more than advertising purposes though, such as treating customers differently. At one point, the cable company CableONE, “bragged that it provided worse customer service to bad credit customers [18].” Additionally, “Comcast, AT&T, Verizon and other large providers have repeatedly argued that privacy rules governing broadband connections are completely unnecessary [17].” A Federal Communications Commission (FCC) ruling proposed in October 2016 would have prohibited ISPs
from collecting and selling user data they obtained from traffic, including sensitive information. The ruling would have imposed restrictions including an opt-in program for the collection and selling of sensitive information, and an opt-out program for non-sensitive information [19]. But this ruling never took effect. In late March 2017, the US Congress passed a bill to overturn the regulations, and soon after this bill was also signed by The President [20]. Additionally, now the FCC cannot pass similar privacy regulations in the future, and ISPs are legally allowed to continue competing in the digital advertising market, an $83 billion dollar industry [10]. For most categories of applications, sensitive data exposure would be minimal, but social media applications are unique in that they use personal data in their nature. Therefore, when one of these applications transmits images unencrypted over HTTP, it is inherently more of a security concern since the image is more likely personal. A game application may transmit images unencrypted, but this would more likely not be a leak of any personal information, since in general users don’t send/receive personal information in these applications. For this reason, we focus on analyzing applications in the context of their social media category, and the implications which should be associated. In Appendix B, we provide descriptions of the applications in our dataset, the services they provide, and the use-cases we believe would transmit sensitive data or PII. In terms of unencrypted sensitive data exposure in our dataset of social media applications, we find it to be minimal for some, medium for others, and particularly large for one.

We make the following contributions:

- We develop an analysis framework to expose the direct/inadvertent leakage of sensitive user data by popular Android social media applications, compiling what we and others have found these applications transmitting, and investigate their privacy policies, advertising practices, and business relationships.

- Leveraging P3P, a framework for privacy policy standards, we analyze the privacy policies of the applications in our dataset. We attempt to match each application’s behavior observed over HTTP (unencrypted) to corresponding P3P data type category(s) and data collection purpose(s). We also note our interpretation of whether this behavior was used for advertising purposes, if it was inferred or explicitly observed, and whether the privacy policy discloses this behavior in general. Additionally, we list every file type we observed each
application transmitting unencrypted.

- We provide an overall breakdown of each application’s traffic, and the detailed results of our Application Analysis module for each application: a graphical summary of the end points they are transmitting with, which includes the network protocols used and their percentages, how much data is transmitted, and the frequency of the data per end point. These graphs, sorted by the bidirectional data transmitted, and truncated to the top 10 connections, can be found in Appendix C. Additionally, we also analyze the network loads throughout the application’s usage, and generated graphs for these during our analysis.

- We propose a unique testbed environment for investigating and analyzing Android applications, comprising of an isolated and emulated environment, and several publicly available analytic tools. We leverage these environments and tools in several scripts to automate the collection and initial analysis process of gathering network traffic statistics.

- We analyze the revisions of these application’s privacy policies between the time of testing and about one year later, noting how and if their practices or disclosures have changed.
Chapter 2  |  Related Works

A multitude of research has focused on privacy policies as well as application traffic analysis in recent years, but few have combined these for comparison. The majority of these studies used static analysis of application code to determine the application’s privacy-sensitive actions. One study and its follow-up studies used dynamic analysis, as we did, to determine the application’s actions. This study, however, only focused on application’s geared towards children. Additionally, these studies each covered a large portion of applications and thus did not take into consideration the context of each application during analysis. Also, until recently privacy policies had been absent for a large portion of the applications these studies covered, limiting the depth and coverage of the studies conducted. Our investigation aimed to study the top social media applications in depth, while taking the context of each application into consideration. A game application, such as Angry Birds, sending images unencrypted is much different than a social media application, such as Tinder, sending images unencrypted, and thus these applications should be studied in context. Even two social media apps, such as Tinder and Imgur, have different implications if they were both found to send an image unencrypted.

2.1 Privacy Policy Analysis

Privacy policies have become more prominent in research over the last two decades, mostly focusing on those pertaining to websites. Some studies analyzed policies and compared them to the company’s data-collection practices. Several efforts have aimed at standardizing the format of these policies, making it easier for companies to produce and maintain them, as well as increase user comprehension. In 1998,
the Federal Trade Commission (FTC) reported that 85% of the websites studied collected personal information, while only 14% disclosed these practices [21]. Several years later P3P provided websites with a standard format for privacy policies which would enable automated interpretation and inform users of a site’s data collection practices [1]. Following this, E-P3P was developed, providing enterprises with a system for formalizing and enforcing their privacy policies in an attempt to prevent privacy violations [22].

A decade later, now in the era of mobile applications, the Federal Trade Commission analyzed privacy policies of applications known to have a child audience, in order to find violations of the Children’s Online Privacy Protection Act (COPPA). Throughout their reports they continued to find the majority of these applications not disclosing their data collection and sharing practices. In their first survey, only 16% of the 400 Android/iOS applications studied (200 from each OS) provided a link to a privacy policy or other disclosure. Later that year, their follow-up survey found this number to rise minimally, to 20% of the 400 applications [7] [8]. Soon after, the FTC released a report detailing mobile privacy suggestions for platform/OS providers, app developers, advertising networks/third parties, as well as researchers in the hopes to improve disclosures [23]. A few years later, the FTC revisited their studies, this time with 364 Android/iOS applications for kids, and found 45% with direct links to their privacy policies on the app store page; a significant increase from before, but still less than half. An additional 38 of these 364 applications had privacy policies in other places, such as in the application or on the developer’s website. Of all of these applications, only 48 (13%) contained “short form disclosures in their app descriptions about the sharing of personal information with third parties, the use of persistent identifiers, in-app purchases, social network integration, or the presence of advertising [11].” That same year a study of popular, free android/iOS applications found that less than one-third of their policies specifically stated the use of encryption methods for certain types of data, and only some of these policies named the encryption technologies in use. Furthermore, most of the policies contained general disclaimers about the Internet being an insecure environment, and that no guarantees can be made about the safety of end user data [12]. Around this time, another group focused on Facebook’s privacy policy throughout years of revisions and found it declining in measures of privacy protection and transparency [24].
2.2 Application Traffic Analysis

Android application security and traffic investigations have discovered and discussed a misuse of PII, and a significant amount of data being sent unencrypted via HTTP [4] [25]. It has been hypothesized that the large amount of HTTP traffic seen indicates an inappropriate use of the HTTP protocol for traffic including social applications, music, and video [25]. Studies investigating the end user data collection practices of Android applications began shortly after the birth of the Android OS. In 2010, an investigation of 101 popular iOS/Android apps used dynamic analysis to find 55% transmitting unique device IDs to third parties without disclosing this to end users [2] [3]. Furthermore, of the 50 Android apps, 21 shared location data, 5 shared usernames/passwords, and 2 shared age/gender, with third parties. In 2011, Enck et al. decompiled 1,100 free Android applications and used static analysis to find 51% using ad/analytics libraries which collected various types of personal data. These libraries collected a combination of location data, phone numbers, and unique device IDs [4]. Another study that year noted similar findings and found the security methods put in place to be insufficient, leaving some data transmissions susceptible to snooping [5]. Later research focused on tracking the information flows of user’s personal data using static taint analysis, and continued to find instances of potential misuse [26] [27]. As discussed earlier in 2012 the FTC conducted surveys on apps geared towards children, and in their second survey they analyzed the unencrypted traffic of the applications, finding 56% of the 400 Android/iOS applications transmitting the device ID to ad/analytics networks or other third parties. Of all the applications, 3% transmitted geolocation, and 1% transmitted the phone number to the developer or third party. Additionally, every instance where the geolocation or phone number was shared also had the device ID shared, which could be used to tie together the information and maintain a profile of the user [8]. The trend continued in 2015, when Walnycky et al. analyzed 20 popular Android instant messaging applications and found that for 16 of them, they could reconstruct some unencrypted data transmitted, including text messages, multimedia content, URLs for server-side content, chat logs, and some passwords. We investigated two of the same applications: Messenger and Tinder. They were able to reconstruct images sent and received, as well as video thumbnails received through Messenger. They noted that although Tinder does not emphasize security,
they found no vulnerabilities, and that it encrypted the traffic, data storage, and server storage [28]. Our findings for these two applications varied greatly, with Tinder transmitting several types of sensitive data, and Messenger encrypting nearly everything [29] [30]. Following this, another study of popular Android apps was published at the end of 2015, finding 73% sharing sensitive data / PII with third parties [9].

Zang et al. focused on identifying applications sharing personal data to third parties, and leveraged a man-in-the-middle attack to decrypt encrypted traffic. They researched Android and iOS applications, and had four in common with our study: Facebook, Messenger, Pinterest, and Textfree [31] [30] [32] [33]. Their reports of the data transmitted by Pinterest and Textfree were somewhat similar to ours, with ours being a subset of the data transmissions seen, but their findings differed greatly. This is because they did not analyze the data in as great of detail, and they focused on all of the data transmitted, without discerning between unencrypted traffic and encrypted traffic. We focused on investigating unencrypted traffic which is entirely exposed to the networks along the path taken by the packets. They also seemed to only focus on identifying specific data they inputted in the traffic, and not thoroughly analyzing the traffic for other sensitive data leaks. We had little results for Facebook and Messenger since most of their traffic was encrypted, but we did note device information being transmitted by both, which Zang et al. did not. In general, they noted a lot of additional data transmissions that we did not observe since they were encrypted. A few of these include Pinterest sending ‘Post’ data and a username to a third party, Facebook, Pinterest, and Textfree transmitting passwords to their application companies, and all four transmitting ‘Location’ data to third parties as well as the application companies [9].

The traffic analysis done by Zang et al. was also not as granular as ours for some data fields, as they grouped together a few similar types of data and reported the findings in terms of these groups. For example, they group ‘current GPS location’ and ‘city’ into the ‘Location’ group. Similarly ‘texts’, ‘chats’, and ‘likes’ are all within the ‘Post’ group. Therefore when their findings state that Textfree sends ‘Post’ and ‘Location’ to ‘pinger.com’, the parent company of the application, it is unclear if this is merely a ‘like’ and a ‘city’ transmitted, or ‘texts’ and the ‘current GPS location’. Additionally, because the ‘Post’ data was transmitted to ‘pinger.com’ and not a third party, they did not have this data finding listed in any
of their tables, nor discuss it further. In their discussions, they do mention several of the applications which were the top offenders of transmitting sensitive data and PII, and here note that Textfree did transmit the current GPS location. But in general, they do not elaborate for the majority of the apps this level of granularity. Furthermore, in the context of their findings it is logical that the parent company receives this information in order to process and provide their services. What they didn’t expose though, is the fact that the majority of this data is transmitted unencrypted, and thus exposed directly and indirectly. In general, our findings reflect theirs, and we both report that of these four applications, Facebook had the least sensitive data leaks, followed by Messenger, then Pinterest, and finally Textfree [9]. Since they focused only on direct data leaks to third parties, had a higher level of granularity in their findings, and did not discern between unencrypted and encrypted transmissions, they missed a multitude of pertinent findings. But, because their analysis did include encrypted data transmissions, by combining their findings with ours, a general idea of the types of data the application transmits, which are transmitted encrypted vs. unencrypted, and who receives them, can be determined.

2.3 Comparing Actions to Policies

As previously mentioned, In their 2012 follow-up study, the FTC dynamically analyzed iOS and Android applications for children and compared the application’s practices to their privacy policies. Their results showed that most apps did not provide any information on their data collection practices, and without disclosing it, found many sharing information to third parties and ad networks. One example they provided was an application transmitting the device ID, geolocation, and phone number to several advertising networks, and then stating in its privacy disclosure that it does not share or sell information to third parties. Similarly, they found another application with a vague and misleading privacy policy, sharing the device ID and geolocation with advertising networks without disclosure. Overall, “while 59% (235) of the apps transmitted device ID, geolocation, or phone number either to the developer or a third party, only 20% (81) of the apps reviewed provided any privacy disclosures to users [8].” As discussed earlier, the FTC revisited their studies a few years later, now using a proxy tool to analyze both the HTTP and
HTTPS traffic. Their privacy policy analysis was explained, but the results of their traffic analysis do not seem to have been completed or released [11].

The following year a study used static analysis to compare popular application’s actions to their privacy policies, noting that 23.6% of the nearly 1,200 apps studied contained at least one inconsistency [13]. A more recent study focused on automating the detection of an application’s compliance to their privacy policies and the legal issues involved. They combined machine learning-based privacy policy analysis with static analysis of application code, and noted that without notifying end users, up to 41% of the nearly 18,000 apps studied could be collecting location data, and 17% could be sharing this data with third parties. Additionally, each application had an average of 1.83 potential inconsistencies between the application’s code and the privacy policies. Furthermore, only 52% of the applications provided links on the Google Play Store to their privacy policies, and of those that did not, 71% exhibited one or more data practices [14]. Another similar study aimed to solve the same issue by linking privacy policy phrases to API methods in the application code that handled sensitive data, and analyzing the information flow for inconsistencies [15]. Both methods identified privacy policy violations with the former achieving better accuracy.

Until now, research comparing application’s privacy policies to their actions has almost exclusively leveraged static analysis. Furthermore, these studies have been large-scale, mostly focusing on automating the analysis and covering as many applications as possible, instead of detailed analysis on each application and its inconsistencies. They also seemed to have focused more on the legal aspects of detecting these inconsistencies, rather than how users are directly affected: their personal data excessively collected and shared. Our work entails a dynamic analysis approach to expose the practices of a specific area of interest: social media applications. Using a dynamic analysis approach gives some advantages, such as collecting the network traffic to gather statistics, endpoints, and other companies the application communicated with. Additionally, we can directly discern between unencrypted and encrypted traffic. We consider this unencrypted data easily accessible by an adversary, and thus could be collected and abused. The security and privacy concerns our Application Analysis module outputs have been gathered through analysis of HTTP traffic. It is important to note that because we do not employ techniques to analyze the encrypted traffic, we may not expose
all of the data transmission practices that are inconsistent with their application’s privacy policies. We aim instead to expose the inconsistencies found in unencrypted transmissions, as the FTC did in their 2012 follow-up survey [8]. Unlike the FTC studies, we disclose the applications in our dataset to inform users of the specific findings per application. Our work, conducted nearly four years after theirs, ensures an isolated testing environment to prevent tainting of traffic from background applications. Additionally, we aim to describe the entire picture of each application, analyzing the traffic, privacy policies, ad/analytics networks used, and even business relationships.
Chapter 3  
Methodology

Our analysis framework, seen in Figure 3.1 and detailed in section Analysis Framework, is comprised of the following three modules: Application Analysis, Policies, and Data Sharing, each of which have components focusing on a piece of the overall analysis process. Both automated and manual analysis are used in our methodology, which is feasible and required in order to analyze our dataset of social media applications in-depth, with the context of each application in mind.

3.1 Application Dataset

Our application dataset, seen in Table 3.1, is comprised of nine popular social media applications: Facebook, Imgur, Messenger, Pinterest, Textfree, Tinder, Tumblr, Twitter, and Vine. Our dataset consisted of these applications in the Android Package Kit (APK) format, the package file format for Android applications used for installation. The applications which we attempted to use, but removed from our dataset, included: Badoo - Free Chat & Dating App, Facebook Lite, Instagram, LinkedIn, MSQRD, Oovoo, Periscope, POF Free Dating App, Snapchat, Tango - Free Video Call & Chat, TextNow - free text + calls. These applications were removed for various reasons such as not running properly in our environment, and will be discussed in more detail later in this section. As seen in Figure 3.1, the APKs are the start of our Analysis Framework, and each APK is passed into all three modules for analysis.

When determining which social media applications to include in our experiments, we went to the ‘Top Free in Android Apps’ page on the Google Play Store and chose the popular social media applications representing a wide variety of uses.
Figure 3.1. Analysis Framework. Arrows represent the transfer of data and analysis information between the modules and components within the Analysis Framework. Red arrows with dotted lines, white arrows with hollow lines, and blue arrows with solid lines represent the data transferred by the Application Analysis, Policies, and Data Sharing modules, respectively.

We focused mostly on applications we knew were popular and widely used, but also some slightly less used apps to determine how these differentiated, if at all, from the most popular. To obtain the APKs of these applications we used the online tool, APK Downloader, from a website which claimed to host the latest Android APKs for a variety of apps [39]. In order to determine how recent these APKs were, we obtained the APK version numbers using the tool aapt included in the Android Software Development Kit (SDK), and compared these against the version numbers and release dates provided on the website AppBrain [40] [41]. Additionally, to view some of the older versions which AppBrain no longer listed, we leveraged the WayBack Machine to visit the previous versions of these pages [42]. We were able to obtain recent releases of the APKs for the majority of the applications, and releases a few months old for the rest. Vine was the only case in which the privacy policy was updated between the APK release date and the date of testing. The
Table 3.1. Social Media Applications Tested. Apr-2016 and Apr-2017 columns provide the number of cumulative downloads in millions (M) or billions (B) at this point in time.

During our investigation of these social media applications, we made use of several open source tools to create our unique testbed environment seen in Figure 3.2. We analyzed the majority of our applications in an emulated environment, and a subset of them on a rooted Android phone. For the first case, we used the

3.2 Testbed Environment for App Analysis

During our investigation of these social media applications, we made use of several open source tools to create our unique testbed environment seen in Figure 3.2. We analyzed the majority of our applications in an emulated environment, and a subset of them on a rooted Android phone. For the first case, we used the
Figure 3.2. Our Testbed Environments for the Emulator and Rooted Phone Experiments

Android emulator within Google’s Android SDK [41]. We felt that this was the best option over third-party emulators since it is supported by Google, widely used by Android developers, and enables emulation of most applications on the majority of Android OS versions. For both the emulator and rooted phone experiments, we leveraged the Android SDK’s debugging tools to analyze applications [41]. In order to root the Google Nexus 6P, we followed an instructional guide online, and used the tool SuperSU to manage root access [43] [44]. Throughout the rooting process we leveraged several more programs including fastboot, TWRP, a Nexus 6P USB Driver, and a Nexus 6P boot image, downloaded from an Android ROM/file hosting website.

For each application, we begin by downloading the most recent APK file available [39]. Each time we test an application, we create a new Android Virtual

1http://downloadandroidrom.com/
Device (AVD) running Marshmallow OS (Version 6.0) [41]. This is done at the
start of each experiment for a given application. AVDs contain the hardware profile,
system image, and other properties. For our testing we use the default profile.
AVDs are then run in an emulator, which is a separate program also provided by
Google, that simulates a device and displays it on the computer. It supports most
features of a device, which allows it to run most applications seamlessly. The
emulator system leverages a version of tcpdump which we used to collect every
packet sent/received by the emulator between starting it up and closing it [45].
Using the Android Debug Bridge (ADB), we are able to interact with the emulator
through our host computer’s shell [41]. We use ADB to install, open, and uninstall
applications, though it has many other features as well, including opening a shell
within the emulator. We use the tool Monkey in the Android SDK just for starting
up each application. To collect information about the specific APKs, such as
permissions, we utilized the Android Assist Packaging Tool (AAPT), which is the
main building tool for Android applications [41]. For each application studied in
the emulator, we conducted three 15-minute long experiments, and collected the
pcaps during each:

Test 1: Open the app and idle without logging in
Test 2: Open the app, have the user log in, and then remain idle
Test 3: Open the app, have the user log in, and then perform normal activity
within the app for the duration of the collection

By collecting traffic when the application opens, but the user does not log
in (Test 1), we hoped to capture most of the base traffic we could expect the
application to generate. Then, by logging in and sitting idle (Test 2), we are able
to focus on the initial burst of traffic when the application first loads the user
data, updates feeds, etc. We also wanted to see if the application would transmit
traffic after a certain period of time, following a predictable pattern, while just
sitting idle. In the third capture, we executed a wide variety of actions to emulate
a typical/normal use of the application (Test 3), such as liking/following pages and
users, writing posts, and messaging other users. Our procedures varied for each
application due to the inherent differences in functionality of these applications.
Our procedures aimed to manually replicate a typical user, and the application
data generated in our experiments should be taken as a general case scenario, as
we did not exhaustively test each application’s entire functionality. For example, during our Facebook experiments we simulated the social media actions an average user would perform, but did not play 3rd party games within the application [31]. As we will discuss in the following section, we believe that we obtained a good snapshot of the majority of the applications tested, and that our procedures performed adequately. The alternative procedure we considered was to leverage the tool Monkey, from the Android SDK, to execute pseudo-random user actions while running an application [41]. These pseudo-random actions were essentially touches/swipes on the screen, and few significant actions seemed to be performed, as navigating through many of these applications requires a more complex sequence of events.

In order to analyze our collected network traffic, we first developed a tool which summarizes the major findings in the traffic by leveraging several analytical tools,
notably Dshell and its ip decoder, tcpdump, matplotlib, and ipinfo [46] [45] [47] [48] [49]. These analytic tools were additionally used for further analysis. The network forensic analysis framework Dshell, developed by the US Army Research Laboratory, enabled us to analyze our traffic quickly and efficiently. For example, it allowed us to focus on analyzing traffic using certain protocols at a time, and translating IP addresses in our traffic to their associated Autonomous System Numbers (ASNs) in order to link IP addresses to organizations. We leveraged several of Dshell’s decoders, including ip, followstream, netflow, rip-http, httpdump, sip, and rtp, to reconstruct the traffic flow from the pcap, and manually analyze it for sensitive data and PII [46]. We used tcpdump mostly for the actual collection of the application traffic, and it was accessed through the tool ADB for use with the emulator [45] [41]. It is important to note that ADB contributes to the
traffic collected on the emulator. From the emulator’s point of view, this traffic is localhost traffic (on a loopback interface) that is on the same subnet as the device’s IP, located in the port range of 5555 to 5585. Additionally, in order to analyze the data more easily, we group together all outgoing and incoming traffic by server/port combinations. Thus, we view this sent and received data to/from a specific server/port as a single connection, as seen in Figure 3.3.

Finally, we visualized our findings using graphs generated by matplotlib, a Python library for 2D plotting [48]. The main graph outputted is a pie chart with multiple legends including ‘Connections Made with Phone’, ‘Traffic Volume’, ‘Estimated Frequency’, and the ‘Breakdown of Traffic’ with the 3 main networking protocols found in our experiments: HTTPS, HTTP, and DNS. We generated several versions of these graphs, sorting the connections by frequency or bidirectional data transmitted, and truncating the number of connections to the top 10 when plotting. The data-sorted graphs generated for each of the applications in both our emulator and rooted phone experiments can be found in Appendix C. All of the connections seen and the data associated with them are also recorded and stored in a corresponding text file. These text files were used for a more thorough analysis of the connections seen in each application. A frequency-sorted plot generated during Vine’s emulator experiment can be seen in Figure 3.3. The second graph we generate in our experiments, contains two network load graphs, both displaying the same information but with one binned by 1 minute, and the other binned by 10 seconds. This is seen in Figure 3.4. We call these the “Network Load” graphs, displaying the packets seen vs. time into the packet capture at two different levels of granularity. From this we can see how much traffic was being sent/received throughout the duration of the test. These graphs provide a bi-directional, summarized view of the traffic seen in the application’s test, and enabled us to pinpoint some key connections to look into further with manual analysis, while determining the percentage of traffic sent unencrypted.

In order to investigate applications which did not run properly in the emulator and to ensure that the emulator was producing normal and accurate application traffic, our second case of experiments were conducted with a rooted Google Nexus 6P phone. Similar to the emulator setup, we used ADB for the rooted phone experiments, but now connected the phone via USB-C to our Apple MacBook Pro laptop, as seen in 3.2 [41]. Since the phone was rooted, we had control to install
software. Using a guide online titled “Installing tcpdump for Android”, we installed the precompiled `tcpdump` binary for Android made available by a project called Android tcpdump [50] [47]. For these experiments, we only conducted one test per application (Test 3), and thus we manually installed the app, started `tcpdump`, opened the app, and simulated the typical/normal use [47]. Just as in the emulator environment, we made sure that each time we were collecting traffic, the only third-party application installed on the device was the one under investigation. It is important to note that the default system applications were still present on the phone and may have had a slight impact on the traffic collected.

The phone proved to be much faster than the emulator, resulting in more actions done per minute. To compensate for this, we lowered our time from 15 minutes (in the emulator collection) to around 5-10 minutes for those applications which had already been tested in the emulator. Additionally, we do not directly compare the results according to the amount of traffic seen, but instead by the percentage of traffic per protocol. With this in mind, we compared our results and found them to be analogous, as we will discuss later in the Breakdown of Traffic section. Next, we will discuss a high-level overview of our Analysis Framework.

### 3.3 Analysis Framework

The following sections detail the inputs and outputs of the modules and components in our Analysis Framework, as also described in Figure 3.1.

#### 3.3.1 Application Analysis Module

**3.3.1.1 Application Testing / Collection**

Inputs an APK and outputs a pcap of the traffic collected. It performs dynamic analysis: running applications in an isolated, emulated environment during case one and on a rooted phone in case two.

**3.3.1.2 Traffic Analysis**

Inputs a pcap and generates network and data statistics using our programs and scripts which combine several publicly available analytic tools. This is then followed
by manual analysis of the outputted graphs and summaries, as well as further
detailed analysis of the traffic. Output is then compared with the output of the
Privacy Policy Analysis component of the Policies module and the P3P Specification
component in order to classify application data transmitted and determine disclosure
of these practices.

3.3.2 Policies Module

3.3.2.1 Privacy Policy Analysis

Inputs an APK and entails the manual analysis of that application’s privacy policy
for the main security/privacy practices it discloses and details, as well as what
data is transmitted/collected, and how this is performed. Output is then compared
with the output of the Traffic Analysis component of the Application Analysis
module and the P3P Specification component in order to classify application data
transmitted and determine disclosure of these practices.

3.3.2.2 Privacy Policy Revisions

Inputs the current privacy policy and the policy in place during testing, and outputs
comparisons and contrasts to determine policy changes.

3.3.3 Data Sharing Module

3.3.3.1 Advertising/Analytics Libraries

Inputs an APK and determines the advertising and analytics libraries used by
each of the applications. Leveraging the open-source tools Dare and Soot in
several scripts, it retargets and decompiles the APKs to find which ad and analytic
libraries are used in these applications [51] [52]. This is followed by a quick manual
investigation into the most prominent to note which applications used them, and
what past studies have shown these libraries to collect/transmit, as well as their
user-data handling practices.
3.3.3.2 Business Relationships

Inputs an APK and the output from the *Advertising/Analytics Libraries* component, and involves manually determining the relationships the application company has with other social media and advertising companies. Provides insights into how data inputted into one application could be shared among other applications/companies owned by or in business with a mutual parent company.
Chapter 4  |  Evaluation

4.1 Breakdown of Traffic

We analyzed and generated traffic analysis graphs and network load plots for each of the nine social media applications, using the methodology discussed previously. In Table 4.1 we organize our observations of the traffic loads seen per protocol in descending order of HTTP traffic percentages seen. We sort this data by focusing on 3 major protocols: HTTPS, HTTP, and DNS. Any traffic seen with a different protocol is grouped and listed as ‘OTHER’.

As discussed earlier, we also collected traffic on a few applications using a rooted Google Nexus 6P. We set up the environment as best as possible to simulate an emulated, isolated environment, by only ever having one of the applications we were testing on the phone at a time.

Unlike in the emulator case experiments, we did not perform all three Tests (no login, login, and normal use) for the rooted phone. During the emulator experiments we found that Test 1 and Test 2 provided little information compared to Test 3. Therefore we only conducted Test 3, the normal use of applications, on the rooted phone. Additionally, we did not test every application on the rooted phone that we tested on the emulator; we only tested the applications that produced a large amount of unencrypted traffic in the emulator tests. This also enabled the verification of the accuracy of the traffic collected in the emulator compared to the real-world scenario with the rooted phone. The applications we investigated on the rooted phone were Vine, Pinterest, Textfree and Tinder [37] [32] [33] [29]. The latter two applications had performed poorly in the emulator experiments.
<table>
<thead>
<tr>
<th>App Name</th>
<th>HTTP</th>
<th>HTTPS</th>
<th>DNS</th>
<th>Other</th>
<th>Total KB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vine</td>
<td>99.30%</td>
<td>0.70%</td>
<td>&lt;0.1%</td>
<td>&lt;0.1%</td>
<td>41829.3 KB</td>
</tr>
<tr>
<td>Vine*</td>
<td>99.20%</td>
<td>0.80%</td>
<td>0.00%</td>
<td>0.00%</td>
<td>75440.6 KB</td>
</tr>
<tr>
<td>Tinder*</td>
<td>92.10%</td>
<td>7.90%</td>
<td>&lt;0.1%</td>
<td>0.00%</td>
<td>24130.5 KB</td>
</tr>
<tr>
<td>Textfree*</td>
<td>84.30%</td>
<td>15.50%</td>
<td>&lt;0.1%</td>
<td>0.20%</td>
<td>30995.1 KB</td>
</tr>
<tr>
<td>Pinterest</td>
<td>84.20%</td>
<td>15.70%</td>
<td>0.10%</td>
<td>&lt;0.1%</td>
<td>23744.2 KB</td>
</tr>
<tr>
<td>Pinterest*</td>
<td>75.70%</td>
<td>24.30%</td>
<td>0.00%</td>
<td>0.00%</td>
<td>27337.2 KB</td>
</tr>
<tr>
<td>Imgur</td>
<td>53.40%</td>
<td>46.60%</td>
<td>&lt;0.1%</td>
<td>&lt;0.1%</td>
<td>16352.1 KB</td>
</tr>
<tr>
<td>Textfree*^</td>
<td>43.70%</td>
<td>53.70%</td>
<td>&lt;0.1%</td>
<td>2.60%</td>
<td>8026.4 KB</td>
</tr>
<tr>
<td>Textfree*$</td>
<td>41.50%</td>
<td>57.30%</td>
<td>0.40%</td>
<td>0.90%</td>
<td>4679.8 KB</td>
</tr>
<tr>
<td>Textfree</td>
<td>23.00%</td>
<td>74.10%</td>
<td>1.30%</td>
<td>1.60%</td>
<td>4164.8 KB</td>
</tr>
<tr>
<td>Twitter</td>
<td>8.50%</td>
<td>89.20%</td>
<td>0.90%</td>
<td>1.40%</td>
<td>7218.0 KB</td>
</tr>
<tr>
<td>Tumblr</td>
<td>15.60%</td>
<td>84.30%</td>
<td>0.10%</td>
<td>&lt;0.1%</td>
<td>12250.0 KB</td>
</tr>
<tr>
<td>Messenger</td>
<td>11.90%</td>
<td>88.00%</td>
<td>0.10%</td>
<td>&lt;0.1%</td>
<td>24513.6 KB</td>
</tr>
<tr>
<td>Facebook</td>
<td>5.80%</td>
<td>93.60%</td>
<td>0.30%</td>
<td>0.30%</td>
<td>1183.0 KB</td>
</tr>
</tbody>
</table>

* Application Traffic collected on rooted Google Nexus 6P
^ Same APK and version retested after 10 months
$ MITM (arpspoof) attack scenario

Table 4.1. Breakdown of Application Traffic Using HTTP, HTTPS, and DNS

Textfree was slow and crashed several times in the emulator, and was attempted twice. We did notice that in the 7MB of traffic collected in the first run and 4 MB in the second, sensitive data was being transmitted unencrypted, including entire text messages and phone numbers. Additionally, as shown in Table 4.1, since the application did not run properly in the emulator, results varied and little traffic was generated. The 7MB run was 89.2% HTTPS and 8.5% HTTP, compared to the 4MB run with 74.1% and 23% respectively. We decided to retest this application on the rooted phone, which would allow for more accurate data for collection and analysis since the application could run properly. Tinder also had issues running in the emulator and we could not collect sufficient traffic data.

Two applications, Vine and Pinterest were tested in both the emulator and rooted phone testbed environments. In Table 4.1 we see that Vine running on the emulator had 99.3% of its traffic sent via HTTP, compared to Vine running on the
rooted phone with 99.2%. Similarly, we see Pinterest on the emulator had 84.2% HTTP compared to 75.7% on the phone. Unlike Vine and Pinterest, Textfree’s traffic breakdown varied drastically between the emulator and rooted phone runs. This was due to the fact that Textfree was very slow and crashed multiple times on the emulator, so the data collected was not an accurate representation of its normal use case. But, the rooted phone case experiments do provide an accurate representation of the application’s normal use case, and we were able to collect about 7.5 times the amount of traffic in two-thirds the amount of time, as seen in Figure 4.1. Towards the end of our data collection, Google released Android Studio v2.0 which provides a significant performance boost, and drastically speeds up the applications running in the emulator. Future testing in this Android emulator environment may be able to provide comparable speeds to that on the rooted phone. Additionally, we find the emulator and rooted phone experiments to be comparable as seen in the closeness of the traffic protocol percentages across both cases for Vine and Pinterest. Thus we conclude that the emulator is a viable environment for testing applications with considerable accuracy. As mentioned previously, all data-sorted analysis graphs can be seen in Appendix C.

4.2 Classifying and Comparing Traffic Data to Policies Using P3P

To investigate and analyze the dataset of social media applications at a granular level we leveraged the Platform for Privacy Preferences (P3P). Specifically we use their Categories and Purposes elements, which provide categorical meaning

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCI</td>
<td>Physical Contact Information</td>
</tr>
<tr>
<td>PoI</td>
<td>Purchase Information</td>
</tr>
<tr>
<td>NCD</td>
<td>Navigation and Click-stream Data</td>
</tr>
<tr>
<td>CI</td>
<td>Computer Information</td>
</tr>
<tr>
<td>NCI</td>
<td>Navigation and Click-stream Data</td>
</tr>
<tr>
<td>OCI</td>
<td>Online Contact Information</td>
</tr>
<tr>
<td>CI</td>
<td>Computer Information</td>
</tr>
<tr>
<td>OCI</td>
<td>Online Contact Information</td>
</tr>
<tr>
<td>ID</td>
<td>Interactive Data</td>
</tr>
<tr>
<td>CI</td>
<td>Computer Information</td>
</tr>
<tr>
<td>UI</td>
<td>Unique Identifiers</td>
</tr>
<tr>
<td>CID</td>
<td>Computer and Identification Data</td>
</tr>
<tr>
<td>NCD</td>
<td>Navigation and Click-stream Data</td>
</tr>
<tr>
<td>CI</td>
<td>Computer Information</td>
</tr>
<tr>
<td>UI</td>
<td>Unique Identifiers</td>
</tr>
<tr>
<td>CID</td>
<td>Computer and Identification Data</td>
</tr>
<tr>
<td>NCD</td>
<td>Navigation and Click-stream Data</td>
</tr>
<tr>
<td>CI</td>
<td>Computer Information</td>
</tr>
<tr>
<td>UI</td>
<td>Unique Identifiers</td>
</tr>
<tr>
<td>CID</td>
<td>Computer and Identification Data</td>
</tr>
<tr>
<td>NCD</td>
<td>Navigation and Click-stream Data</td>
</tr>
<tr>
<td>CI</td>
<td>Computer Information</td>
</tr>
<tr>
<td>UI</td>
<td>Unique Identifiers</td>
</tr>
<tr>
<td>CID</td>
<td>Computer and Identification Data</td>
</tr>
<tr>
<td>NCD</td>
<td>Navigation and Click-stream Data</td>
</tr>
<tr>
<td>CI</td>
<td>Computer Information</td>
</tr>
<tr>
<td>UI</td>
<td>Unique Identifiers</td>
</tr>
<tr>
<td>CID</td>
<td>Computer and Identification Data</td>
</tr>
<tr>
<td>NCD</td>
<td>Navigation and Click-stream Data</td>
</tr>
<tr>
<td>CI</td>
<td>Computer Information</td>
</tr>
<tr>
<td>UI</td>
<td>Unique Identifiers</td>
</tr>
<tr>
<td>CID</td>
<td>Computer and Identification Data</td>
</tr>
<tr>
<td>NCD</td>
<td>Navigation and Click-stream Data</td>
</tr>
<tr>
<td>CI</td>
<td>Computer Information</td>
</tr>
<tr>
<td>UI</td>
<td>Unique Identifiers</td>
</tr>
<tr>
<td>CID</td>
<td>Computer and Identification Data</td>
</tr>
<tr>
<td>NCD</td>
<td>Navigation and Click-stream Data</td>
</tr>
<tr>
<td>CI</td>
<td>Computer Information</td>
</tr>
<tr>
<td>UI</td>
<td>Unique Identifiers</td>
</tr>
<tr>
<td>CID</td>
<td>Computer and Identification Data</td>
</tr>
<tr>
<td>NCD</td>
<td>Navigation and Click-stream Data</td>
</tr>
<tr>
<td>CI</td>
<td>Computer Information</td>
</tr>
<tr>
<td>UI</td>
<td>Unique Identifiers</td>
</tr>
<tr>
<td>CID</td>
<td>Computer and Identification Data</td>
</tr>
<tr>
<td>NCD</td>
<td>Navigation and Click-stream Data</td>
</tr>
<tr>
<td>CI</td>
<td>Computer Information</td>
</tr>
<tr>
<td>UI</td>
<td>Unique Identifiers</td>
</tr>
<tr>
<td>CID</td>
<td>Computer and Identification Data</td>
</tr>
<tr>
<td>NCD</td>
<td>Navigation and Click-stream Data</td>
</tr>
<tr>
<td>CI</td>
<td>Computer Information</td>
</tr>
<tr>
<td>UI</td>
<td>Unique Identifiers</td>
</tr>
<tr>
<td>CID</td>
<td>Computer and Identification Data</td>
</tr>
<tr>
<td>NCD</td>
<td>Navigation and Click-stream Data</td>
</tr>
<tr>
<td>CI</td>
<td>Computer Information</td>
</tr>
<tr>
<td>UI</td>
<td>Unique Identifiers</td>
</tr>
<tr>
<td>CID</td>
<td>Computer and Identification Data</td>
</tr>
<tr>
<td>NCD</td>
<td>Navigation and Click-stream Data</td>
</tr>
<tr>
<td>CI</td>
<td>Computer Information</td>
</tr>
<tr>
<td>UI</td>
<td>Unique Identifiers</td>
</tr>
<tr>
<td>CID</td>
<td>Computer and Identification Data</td>
</tr>
<tr>
<td>NCD</td>
<td>Navigation and Click-stream Data</td>
</tr>
<tr>
<td>CI</td>
<td>Computer Information</td>
</tr>
<tr>
<td>UI</td>
<td>Unique Identifiers</td>
</tr>
<tr>
<td>CID</td>
<td>Computer and Identification Data</td>
</tr>
<tr>
<td>NCD</td>
<td>Navigation and Click-stream Data</td>
</tr>
<tr>
<td>CI</td>
<td>Computer Information</td>
</tr>
<tr>
<td>UI</td>
<td>Unique Identifiers</td>
</tr>
<tr>
<td>CID</td>
<td>Computer and Identification Data</td>
</tr>
<tr>
<td>NCD</td>
<td>Navigation and Click-stream Data</td>
</tr>
<tr>
<td>CI</td>
<td>Computer Information</td>
</tr>
<tr>
<td>UI</td>
<td>Unique Identifiers</td>
</tr>
<tr>
<td>CID</td>
<td>Computer and Identification Data</td>
</tr>
<tr>
<td>NCD</td>
<td>Navigation and Click-stream Data</td>
</tr>
<tr>
<td>CI</td>
<td>Computer Information</td>
</tr>
<tr>
<td>UI</td>
<td>Unique Identifiers</td>
</tr>
<tr>
<td>CID</td>
<td>Computer and Identification Data</td>
</tr>
<tr>
<td>NCD</td>
<td>Navigation and Click-stream Data</td>
</tr>
<tr>
<td>CI</td>
<td>Computer Information</td>
</tr>
<tr>
<td>UI</td>
<td>Unique Identifiers</td>
</tr>
<tr>
<td>CID</td>
<td>Computer and Identification Data</td>
</tr>
<tr>
<td>NCD</td>
<td>Navigation and Click-stream Data</td>
</tr>
<tr>
<td>CI</td>
<td>Computer Information</td>
</tr>
<tr>
<td>UI</td>
<td>Unique Identifiers</td>
</tr>
<tr>
<td>CID</td>
<td>Computer and Identification Data</td>
</tr>
<tr>
<td>NCD</td>
<td>Navigation and Click-stream Data</td>
</tr>
<tr>
<td>CI</td>
<td>Computer Information</td>
</tr>
<tr>
<td>UI</td>
<td>Unique Identifiers</td>
</tr>
<tr>
<td>CID</td>
<td>Computer and Identification Data</td>
</tr>
<tr>
<td>NCD</td>
<td>Navigation and Click-stream Data</td>
</tr>
<tr>
<td>CI</td>
<td>Computer Information</td>
</tr>
<tr>
<td>UI</td>
<td>Unique Identifiers</td>
</tr>
<tr>
<td>CID</td>
<td>Computer and Identification Data</td>
</tr>
<tr>
<td>NCD</td>
<td>Navigation and Click-stream Data</td>
</tr>
<tr>
<td>CI</td>
<td>Computer Information</td>
</tr>
<tr>
<td>UI</td>
<td>Unique Identifiers</td>
</tr>
<tr>
<td>CID</td>
<td>Computer and Identification Data</td>
</tr>
<tr>
<td>NCD</td>
<td>Navigation and Click-stream Data</td>
</tr>
<tr>
<td>CI</td>
<td>Computer Information</td>
</tr>
<tr>
<td>UI</td>
<td>Unique Identifiers</td>
</tr>
<tr>
<td>CID</td>
<td>Computer and Identification Data</td>
</tr>
<tr>
<td>NCD</td>
<td>Navigation and Click-stream Data</td>
</tr>
<tr>
<td>CI</td>
<td>Computer Information</td>
</tr>
<tr>
<td>UI</td>
<td>Unique Identifiers</td>
</tr>
<tr>
<td>CID</td>
<td>Computer and Identification Data</td>
</tr>
<tr>
<td>NCD</td>
<td>Navigation and Click-stream Data</td>
</tr>
<tr>
<td>CI</td>
<td>Computer Information</td>
</tr>
<tr>
<td>UI</td>
<td>Unique Identifiers</td>
</tr>
<tr>
<td>CID</td>
<td>Computer and Identification Data</td>
</tr>
<tr>
<td>NCD</td>
<td>Navigation and Click-stream Data</td>
</tr>
<tr>
<td>CI</td>
<td>Computer Information</td>
</tr>
<tr>
<td>UI</td>
<td>Unique Identifiers</td>
</tr>
<tr>
<td>CID</td>
<td>Computer and Identification Data</td>
</tr>
<tr>
<td>NCD</td>
<td>Navigation and Click-stream Data</td>
</tr>
<tr>
<td>CI</td>
<td>Computer Information</td>
</tr>
<tr>
<td>UI</td>
<td>Unique Identifiers</td>
</tr>
<tr>
<td>CID</td>
<td>Computer and Identification Data</td>
</tr>
<tr>
<td>NCD</td>
<td>Navigation and Click-stream Data</td>
</tr>
<tr>
<td>CI</td>
<td>Computer Information</td>
</tr>
<tr>
<td>UI</td>
<td>Unique Identifiers</td>
</tr>
<tr>
<td>CID</td>
<td>Computer and Identification Data</td>
</tr>
<tr>
<td>NCD</td>
<td>Navigation and Click-stream Data</td>
</tr>
<tr>
<td>CI</td>
<td>Computer Information</td>
</tr>
<tr>
<td>UI</td>
<td>Unique Identifiers</td>
</tr>
<tr>
<td>CID</td>
<td>Computer and Identification Data</td>
</tr>
<tr>
<td>NCD</td>
<td>Navigation and Click-stream Data</td>
</tr>
<tr>
<td>CI</td>
<td>Computer Information</td>
</tr>
<tr>
<td>UI</td>
<td>Unique Identifiers</td>
</tr>
<tr>
<td>CID</td>
<td>Computer and Identification Data</td>
</tr>
<tr>
<td>NCD</td>
<td>Navigation and Click-stream Data</td>
</tr>
<tr>
<td>CI</td>
<td>Computer Information</td>
</tr>
<tr>
<td>UI</td>
<td>Unique Identifiers</td>
</tr>
<tr>
<td>CID</td>
<td>Computer and Identification Data</td>
</tr>
<tr>
<td>NCD</td>
<td>Navigation and Click-stream Data</td>
</tr>
<tr>
<td>CI</td>
<td>Computer Information</td>
</tr>
<tr>
<td>UI</td>
<td>Unique Identifiers</td>
</tr>
<tr>
<td>CID</td>
<td>Computer and Identification Data</td>
</tr>
<tr>
<td>Application</td>
<td>Data Transmitted over HTTP*</td>
</tr>
<tr>
<td>-------------</td>
<td>-----------------------------</td>
</tr>
<tr>
<td>Instagram</td>
<td>User-agent field w/ device info</td>
</tr>
<tr>
<td>preg, pos, <strong>.png, .css, .trf</strong></td>
<td>Referrer info</td>
</tr>
<tr>
<td></td>
<td>Content (pic) user viewed</td>
</tr>
<tr>
<td></td>
<td>Code files</td>
</tr>
<tr>
<td></td>
<td>HTTP cookies</td>
</tr>
<tr>
<td></td>
<td>HTTP ETags</td>
</tr>
<tr>
<td></td>
<td>(Inferred) Potentially leaked user interests</td>
</tr>
<tr>
<td>Pinterest</td>
<td>User/real names used in profile pic filenames</td>
</tr>
<tr>
<td></td>
<td>UID, unique identifier</td>
</tr>
<tr>
<td></td>
<td>Device info - OS/make/model/build</td>
</tr>
<tr>
<td></td>
<td>Referer info</td>
</tr>
<tr>
<td></td>
<td>User-agent field w/ device info</td>
</tr>
<tr>
<td></td>
<td>Code files</td>
</tr>
<tr>
<td></td>
<td>(Inferred) Potentially leaked user interests</td>
</tr>
<tr>
<td>Textfree</td>
<td>Phone numbers/contacts</td>
</tr>
<tr>
<td></td>
<td>UUI, unique identifiers</td>
</tr>
<tr>
<td></td>
<td>Notification email, forgot password email</td>
</tr>
<tr>
<td></td>
<td>User ID</td>
</tr>
<tr>
<td></td>
<td>User Profile Picture</td>
</tr>
<tr>
<td></td>
<td>Code files</td>
</tr>
<tr>
<td></td>
<td>HTTP cookies</td>
</tr>
<tr>
<td></td>
<td>(Inferred) Potentially leaked user interests</td>
</tr>
<tr>
<td></td>
<td>Location tracking, Wi-Fi mapping</td>
</tr>
<tr>
<td>Tinder</td>
<td>User/real names used in profile pic filenames</td>
</tr>
<tr>
<td></td>
<td>User/real names used in profile pic filenames</td>
</tr>
<tr>
<td></td>
<td>Unique IDs per user within pic names</td>
</tr>
<tr>
<td></td>
<td>Device info - OS/make/model</td>
</tr>
<tr>
<td></td>
<td>Content (pic) user viewed</td>
</tr>
<tr>
<td></td>
<td>AWS (Server: AmazonS3) request ID and ID-2</td>
</tr>
<tr>
<td></td>
<td>City/district name and zipcode</td>
</tr>
<tr>
<td></td>
<td>(Inferred) Age-like number</td>
</tr>
<tr>
<td></td>
<td>Code files containing city/district name, zipcode, age-like number</td>
</tr>
<tr>
<td></td>
<td>HTTP ETags</td>
</tr>
<tr>
<td></td>
<td>(Inferred) Location Tracking - City/district name and zipcode</td>
</tr>
<tr>
<td>Vine</td>
<td>User-agent field w/ device info</td>
</tr>
<tr>
<td></td>
<td>AWS* (Server: EC2/AWS) request ID, version ID, ID-2</td>
</tr>
<tr>
<td></td>
<td>HTTP ETags</td>
</tr>
<tr>
<td></td>
<td>(Inferred) Potentially leaked user interests</td>
</tr>
<tr>
<td>Twitter</td>
<td>User-agent field w/ device info</td>
</tr>
<tr>
<td></td>
<td>HTTP cookies</td>
</tr>
<tr>
<td>Messenger</td>
<td>User-agent field w/ device info</td>
</tr>
<tr>
<td></td>
<td>.ogg audio file - failed call occurred</td>
</tr>
<tr>
<td>Facebook</td>
<td>User-agent field w/ device info</td>
</tr>
</tbody>
</table>

Table 4.3. Classifying Application Traffic Data with P3P and Determining Policy Disclosure. The abbreviation key in Table 4.2 provides the P3P Categories and Purposes, and their descriptions can be found in Appendix A. The list of file extensions found in the unencrypted traffic per application are listed below the application’s name. Asterisks (*) are used to denote borderline instances where the disclosure or classification could be argued, and our best judgement was used. This includes instances of data transmitted that lacked disclosure, and thus the P3P Purpose is inferred.
to the data a service collects, as well as a purpose for why it was gathered and how it will be used. A truncated and reformatted copy of the P3P Specification’s sections which detail these Categories and Purposes can be seen in Appendix A. An example of a P3P category is the ‘Physical Contact Information’ category which is described as, “Information that allows an individual to be contacted or located in the physical world - such as telephone number or address.” An example of a P3P purpose is the ‘Research and Development’ purpose which is described as, “Information may be used to enhance, evaluate, or otherwise review the site, service, product, or market. This does not include personal information used to tailor or modify the content to the specific individual nor information used to evaluate, target, profile or contact the individual [1].”

By leveraging the P3P platform and its terminology we can classify the data found in the application traffic as well as map it to each application’s privacy policy.
Within our Framework, as seen in Figure 3.1, the outputs of the Traffic Analysis component of the Application Analysis module and the Privacy Policy Analysis component of the Policies module are combined with the P3P Specification. The results of this application data classification is reported in Table 4.3.

In Table 4.3 we manually classify the application data transmitted over HTTP using P3P Categories and Purposes, and explain whether this data was explicitly stated or inferred, whether it was used for or with advertising, and whether it was disclosed within the application’s privacy policy. The table was created using our best interpretation of the application traffic and the behaviors and actions these applications exhibited. For the scope of this research, we determined that the data was used for Ads (transmitted to an advertising network) by the hostnames in the headers of all of the GET/POST request payloads containing the specific data. The table is broken up by application and the different types of data they each transmitted unencrypted. For each application and across all of its experiments, we state all of the file types found in the traffic (based on their file extensions). These file types are listed under the application’s name. A ‘✓’ and ‘×’ is used to represent ‘YES’ and ‘NO’ respectively. This table was populated using only data applications transmitted unencrypted over HTTP. It is important to note that, since our application tests were not exhaustive in testing the actions and behaviors that can be conducted within each application, our analysis is also not exhaustive. Additionally, in this research we only cover the application’s data practices within the scope of unencrypted traffic.

### 4.2.1 Populating and Interpreting Table 4.3

Within the Application Analysis module, the Traffic Analysis component entails manually analyzing the data transmitted to determine the ‘Data Transmitted over HTTP’, ‘Inferred’, ‘For Ads’, and file extensions fields in Table 4.3. Next this data is cross-referenced with the P3P Specification to determine the ‘P3P Category’ field. Similarly, the Policies module’s Privacy Policy Analysis component involves manual analysis of the application’s privacy policy to determine the ‘Disclosed’ field. It is important to note that, this column only means the data was disclosed in general, not that it was disclosed as being transmitted over HTTP, unencrypted. Next this component’s data, specifically statements from the privacy policy, is
cross-referenced with the P3P Specification to determine the ‘P3P Purpose’ field. Thus the data populating the ‘Data Transmitted over HTTP’, ‘P3P Category’, and ‘For Ads’ columns are solely based on our interpretation of the unencrypted traffic collected and for the ‘P3P Category’, additionally the comparison of this traffic to the P3P Specification. Similarly, the data populating the ‘P3P Purpose’ and ‘Disclosed’ columns are solely based on our interpretation of the given application’s privacy policy and for the ‘P3P Purpose’, additionally the comparison of these statements to the P3P Specification.

To better understand how the table was populated, and which pieces of information were used to connect the traffic to the P3P terminology and the privacy policies, we will go through the reasoning for the top row of the first application, Imgur.

1. In the **Traffic Analysis** component of the framework we first gather all of the file extensions seen in the Imgur traffic. Next we manually look through the unencrypted traffic and note that the operating system, processor specifications, and build information of the Android device are found. In this case the actual data found was ‘**User-Agent**: Dalvik/2.1.0 (Linux; U; Android 6.0; sdk_phone_armv7 Build/MRA44C). We summarize this finding as ‘User-agent field w/ device info’ and populate the field in the ‘Data Transmitted over HTTP’ column. Because this data was explicitly found in the application traffic it is not inferred. If this data had not been found in the traffic but could be inferred, we would prepend ‘(Inferred)’ to ‘User-agent field w/ device info’. Next, since each of the instances of the GET Request headers containing this data had a host of ‘**Host**: i.imgur.com’, and thus in the traffic we collected it was not used for Ads, a ‘×’ is placed in the ‘For Ads’ column, to denote ‘NO’. It is important to note that if the piece of data is found in multiple places within the traffic and used for advertising in at least one of these places, we denote the data with a ‘YES’ in the ‘For Ads’ column. Thus, it may only be one instance of many, but we will represent it as being used for advertising purposes at some point by the application. Additionally, since we are basing our analysis off of the data we have collected, if the privacy policy states that the data is used for advertising, but we do not see this in the traffic we collected, we state that it was not used for ads.
2. The types of data gathered are then cross-referenced with the list of P3P Categories in the P3P Specification. We corresponded this data with the sixth P3P Category ‘Computer Information’ which is described as ‘Information about the computer system that the individual is using to access the network - such as the IP number, domain name, browser type or operating system.’

3. From the manual analysis of Imgur’s privacy policy within the Privacy Policy Analysis component of the Policies module we find that they state, “Our servers log information about each computer connecting with our site such as IP address, dates and times of each login, device characteristics, operating system, browser type, type of connection, page and image viewing statistics, and incoming and outgoing links. We associate this information with comments you post or votes you enter...if you choose to give us personal data about you as described below, the technical information we collect that would otherwise be anonymous could instead be logged as coming from you [53].” Therefore the transmission of this data (in general) was disclosed, and a ‘✓’ is placed in the ‘Disclosed’ column to denote ‘YES’. For the table, it is important to note that there may be instances where the application’s privacy policy discloses the transmission of a piece of data, but does not disclose that it is used for advertising purposes. In these cases, we denote the data as NOT disclosed.

4. The key statements within the privacy policy are then cross-referenced with the P3P Specification to determine the P3P Purpose to the best of our ability. In this case, it isn’t specifically stated why the information is collected, but that it is tied together with user activity and thus creates a type of record which won’t be anonymous if the user has provided personal data. Therefore, our interpretation is to loosely classify it as P3P Purpose ‘IA / InD’ which denotes ‘Individual Analysis’ / ‘Individual Decision’.

As seen in the example above, the manual population of Table 4.3 was tedious, yet necessary for this level of granularity. The information presented in Table 4.3 is our best interpretation of the traffic and privacy policies we gathered and analyzed, and our best attempt to classify data which may be ambiguous. Due to the nature of ambiguous data and vague privacy policy statements, asterisks (*) are used to denote borderline instances where the disclosure or classification could be argued,
and our best judgement was used. This additionally includes instances of data transmitted that lacked disclosure, and thus the P3P Purpose must be inferred.

### 4.3 Policies Module

In the Privacy Policy Analysis component we analyze the privacy policies in place for each of the applications during the time of testing. Since policies were updated between the time of analyzing the applications and their privacy policies, the WayBack Machine was used to grab the proper policy in place during the time of testing [42]. Each privacy policy was manually analyzed, but several tools such as grep were deployed to find terms more efficiently [54]. Each time a piece of data was uncovered during the analysis of the unencrypted application traffic, it was checked whether is was disclosed in the corresponding privacy policy. As mentioned previously, not all of the data uncovered was disclosed as being transmitted/collected, and furthermore, none of the policies properly disclosed that any data was sent unencrypted. The detailed results of this analysis can be seen in Table 4.3.

As part of the Privacy Policy Analysis component we aimed to understand the security and privacy measures put in place by these application companies. Recall that, of all of the data transmission practices disclosed in these policies, none of the application policies discern which data is transmitted encrypted or unencrypted. Past research has shown application privacy policies to use general disclaimers about the insecure nature of the Internet, and that no guarantees can be made about the confidentiality of end user data [12]. Of the nine applications in our dataset, 7 loosely mention that security and privacy practices are used or attempted to protect their user’s data. The remaining two applications, Pinterest and Vine, don’t explicitly discuss security, or it is too vague to discern. For these two, we select quotes from them which are somewhat security related. The following statements have been taken verbatim from each application’s privacy policy in place during the time of testing.

- **Facebook & Messenger** - “We use the information we have to help verify accounts and activity, and to promote safety and security on and off of our Services, such as by investigating suspicious activity or violations of our terms or policies. We work hard to protect your account using teams of engineers,
automated systems, and advanced technology such as encryption and machine learning. We also offer easy-to-use security tools that add an extra layer of security to your account. For more information about promoting safety on Facebook, visit the Facebook Security Help Center [55].”

- **Imgur** - “We take every reasonable precaution to protect your private data from loss, misuse, unauthorized access, disclosure, alteration, or destruction [56].”

- **Pinterest** - “We use the information we collect to provide our products to you and make them better, develop new products, and protect Pinterest and our users [57].”

- **Textfree** - “We have physical, electronic, and procedural safeguards to protect the loss, misuse, and alteration of the Customer Information under our control. Customer Information we collect is stored in a secure manner that is accessible only by authorized Pinger personnel, contractors, and agents bound by confidentiality obligations and is only disclosed under the provisions of this privacy policy. Our processing of Personal Information is only for the purposes for which it was collected and in accordance with this privacy policy. Periodically, Pinger undertakes a review of its privacy practices to ensure that that the information we collect is needed to provide or improve the services we provide [58].”

- **Tinder** - “We take security measures to help safeguard your personal information from unauthorized access and disclosure. However, no system can be completely secure. Therefore, although we take steps to secure your information, we do not promise, and you should not expect, that your personal information, chats, or other communications will always remain secure. Users should also take care with how they handle and disclose their personal information and should avoid sending personal information through insecure email [59].”

- **Twitter** - “We may share your private personal information with such service providers subject to obligations consistent with this privacy policy and any other appropriate confidentiality and security measures, and on the condition
that the third parties use your private personal data only on our behalf and pursuant to our instructions [60].”

- **Tumblr** - “Your Account Information is protected by a password for your privacy and security. We may enable additional security features in the future, like multi-factor authentication. You need to prevent unauthorized access to your Account and information by creating a unique, secure, and protected password and limiting access to your computer and browser by signing off after you have finished accessing your Account on the Services. We seek to protect your information (including your Account Information) to ensure that it is kept private; however, we can’t guarantee the security of any information. Unauthorized entry or use, hardware or software failure, and other factors may compromise the security of user information at any time [61].”

- **Vine** - “Your public information is broadly and instantly disseminated. For instance, your profile information and public Content are immediately delivered to a wide range of users and other services that access Vine. When you share information or Content via the Services, you should think carefully about what you are making public [62].”

As Graves points out, ambiguity was prevalent throughout the application privacy policies they studied, with disclaimers explaining that data security cannot be guaranteed in this day and age. While some of those policies additionally stated their use of encryption, others only had vague statements about their use of security / data protection practices. Graves explains that in the former case, these disclaimers may just be making it clear to end users that data cannot be completely secure, but in the latter case it is hard to gauge the effectiveness of the security practices in place, if any at all [12]. Through our analysis of the privacy policies of applications in our dataset, and as shown in the statements quoted from these policies, we find similar results. End users are thus left with little to no answers about the security or privacy of their data. Since past research has shown the prevalence and details of privacy policies to be slowly increasing over the past several years, we aimed to determine how the policies of applications in our dataset improved, if at all, throughout the course of our study.
4.3.1 Privacy Policy Revisions since Testing

The Privacy Policy Revisions component of our Analysis Framework entails manually finding the differences between the privacy policies in place during testing and the updated policies in place almost a year later, in February 2017. This analysis involved the use of several tools including diff and grep for data parsing, and the WayBack Machine for determining and verifying when policies changed [63] [54] [42]. The following updates between these versions can be seen below per application.

- **Facebook & Messenger** - Routine maintenance. They updated the policy to adhere to the EU/Swiss framework for privacy, and explained how to contact them for questions and concerns [64] [55].

- **Imgur** - Explained that the messaging platform is not intended to be secure, and messages are never completely hidden from public view [53] [56].

- **Pinterest** - Removed statements explaining differences between personally identifiable information and non-personally identifiable information. Added “wholly-owned subsidiaries and affiliates” to the parties they share the information described in the policy. Added ‘affiliates’ to the list of parties they “…may get information about you and your activity off Pinterest” from [65] [57].

- **Textfree** - Removed the statement that Textfree complies with the Children’s Online Privacy Protection Act (COPPA). Now, “…as part of a request for information about Pinger products or services,” they can share your provided phone number with 3rd parties (to be used to contact you / provide ads to you about Pinger products / services). No longer respond to “Do Not Track” browser features [66] [58].

- **Tinder** - Overall more disclosure, but now shares more PII with third parties. Now collects Personal Information and Sensitive Data: two new terms they define and use throughout the policy. Previously stated that they collect personal information such as name, email address, and other information that does not identify you. They now additionally use technology similar to cookies. They used to only provide anonymized personal information to third parties, but now it is “masked and obscured”. Can now opt out of cookies to third
parties/ad companies. They share data with Match Group Businesses (other dating services) and updated the list of companies in this group. Now share aggregated, non-personal information, or personal information in hashed, non-human readable form for advertising and marketing purposes. Now shares users’ geolocation information in de-identified form with Match Group companies and third parties. They now state, “To opt out of the sharing of your geolocation information, please discontinue use of the Tinder application.” Now explain that they transfer data to certain facilities in other countries to be processed and these locations may have different (more lenient) privacy laws. Now explain they have “…developed data practices designed to assure information is appropriately protected but we cannot always know where personal information may be accessed or processed.” “While our primary data centers are in the United States, we may transfer personal information or other information to our offices outside of the United States. In addition, we may employ other companies and individuals to perform functions on our behalf. If we disclose personal information to a third party or to our employees outside of the United States, we will seek assurances that any information we may provide to them is safeguarded adequately and in accordance with this Privacy Policy and the requirements of applicable privacy laws [67] [59].”

- **Twitter** - Removed mentions of age restrictions. Now appears to be collaborating more with ad and analytics companies, as seen in changes discussed below. Performed routine maintenance, updating terminology to more clearly express their actions. Now using more information from ad/analytics companies such as “…demographic or interest data and content viewed or actions taken on a website or app.” Added that you, “…consent to the collection, transfer, storage, disclosure, and use of your information as described in this Privacy Policy…this includes any information you choose to provide that is deemed sensitive under applicable law.” Added a contact form for questions/comments about the policy instead of just emailing them. Seems that they improved their privacy settings and now allow users to control more. Added that the things you post publicly is used by market research firms that analyze the information for trends and insights. Before they stated, “Twitter uses Log Data to provide, understand, and improve our Services”, and now only states “We use Log Data to make inferences.” But, in several
other parts of the policy they discuss their use of service providers “...to help provide our Services... and to help us understand and improve the use of our Services”. Therefore, one conclusion drawn from this could be that they are moving towards having these service providers do most of the analytics work. Removed the section about “Our Policy Towards Children” and the statement “Our Services are not directed to persons under 13”. They did add several sentences about ‘Privacy Shield principles’ and ‘the US-Swiss Safe Harbor Framework and principles’, but these do not appear to discuss or involve age restrictions. Changed terminology from ‘Non-Private’ to ‘Public’. Changed and added tools/capabilities in the application revisions, and updated their terminology accordingly [68] [60].

- **Tumblr** - Same privacy policy in place as before [61].
- **Vine** - Routine maintenance performed [69] [62].

### 4.4 Notable HTTP Data Leaks and Observations

The following sections highlight prominent findings in the application traffic. More in-depth information can be found in Table 4.3 per application.
Through our analysis, we found that Textfree is sending the majority of message traffic in the clear. We collected almost all of the ‘text messages’ (including emojis) in plaintext sent to/from the application, as well as pictures sent from the app, the phone IDs/numbers (including our personal cell phone number), the contacts saved in the application, the age and gender of the Textfree user logged into the application, the Wi-Fi SSID being used for the network connection, and all of the Wi-Fi SSIDs within range of the phone being broadcasted. As discussed earlier, these lists of SSIDs can be used for Wi-Fi mapping, which aims to determine the approximate location of the device even when GPS and other location services are turned off [70]. Only a few ‘text messages’ across all of the tests were not seen in the traffic collected. Pictures sent in the application were sent unencrypted and were reconstructed during traffic analysis, whereas the pictures received in the application from an outside source were hosted on an HTTP server and could be pulled down using the URL found in the unencrypted message traffic. During the Textfree rooted phone experiment we were able to thoroughly test the application. We had text messaging conversations through the application with a personal cell phone (Verizon carrier), and sent/received text and picture messages. On the rooted phone we took a picture and sent it to our personal cell phone, then from our personal cell phone took a picture and sent that back. In both cases, we were able to obtain the images transmitted/referenced in the HTTP traffic. An example of this in a later experiment can be seen in Figures 4.2 and 4.3. During our analysis we also found Textfree at times using the Session Initiation Protocol (SIP) over UDP on port 4544, a nonstandard port. According to the SIP RFC, “The default port value is transport and scheme dependent. The default is 5060 for sip: using UDP, TCP, or SCTP. The default is 5061 for sip: using TLS over TCP and sips: over TCP [71].” It is unclear why this protocol was sometimes being used on a nonstandard port, but it may be an attempt at security. We also observed some SIP traffic on the standard port of 5060, transmitted over UDP, but only during our tests on the rooted phone. We conducted a high-level analysis of this traffic, leveraging Dshell’s sip and rtp decoders, as well as Wireshark [46] [72]. From our analysis we concluded that Textfree registered the phone number through SIP, and continued to keep the connection alive, in which we presume would be to maintain a prepared state in case a phone call would be made. In general, this was seen through the pattern of a request to ‘REGISTER’, followed by multiple pairs
As a quick, additional experiment we attempted to obtain as much traffic as possible through a man-in-the-middle (MITM) attack using ARP cache poisoning, targeting the rooted phone. We aimed to determine how closely our experimental results in the emulator and rooted phone tests resembled those in a realistic attack scenario. We leveraged `arpspoof` for this, a tool provided in a package of network analytic tools called `dsniff` [73]. According to the man page, “arpspoof redirects packets from a target host (or all hosts) on the LAN intended for another host on the LAN by forging ARP replies [73].” In this scenario, the rooted phone was on Wi-Fi, connected to a router we setup. A laptop on the same network was used to determine the IP addresses of the router and rooted phone through `nmap`, an open source network discovery tool [74]. Next, using `arpspoof`, this laptop convinced the rooted phone that the laptop was the router. Thus, the phone was tricked into passing its traffic to the laptop instead of the router, which the laptop collected and forwarded to the router, such that the traffic would still be sent to the original destination. In this real-world attack scenario, all of the unencrypted HTTP traffic sent by the phone would be seen by the adversary. Almost all of the data mentioned above was also collected in this scenario, except the multimedia pictures were only found as URLs in the traffic (the actual images themselves were not), and the SIP traffic differed. In this case, we found several instances of GET(NO RESPONSE) in the traffic. One example was “GET (NO RESPONSE) p.pinger.com/mms/716/lRDQTu.jpg”. Another difference was that all of the SIP traffic was on port 5060, which differed from the other Textfree experiments conducted, which additionally contained SIP traffic on the nonstandard port 4544. The SIP traffic also seemed to be slightly different, with several consecutive requests to ‘REGISTER’ the phone number, followed by multiple consecutive ‘OK’ statuses. Overall, we concluded that this MITM attack scenario yielded similar results and supported those in our emulator and rooted phone Textfree experiments.

Since Textfree yielded the most unencrypted data leaks, we retested the application about 10 months after the first test to see how the application’s data handling practices had changed, and to test the bidirectional phone call functionality. Similar to the emulator experiments we found SIP traffic on the nonstandard port 4544 and the standard port 5060. In this case, the phone calls placed occurred over the standard SIP port 5060. We obtained detailed call logs in the SIP, Real-Time Transport
Protocol (RTP), and HTTP traffic of the calls made to and from the application and our personal phone. These logs included the detailed handshakes, phone numbers, Call-IDs, nonces, proxy authentication, encodings, and codecs. We did not recover the audio from the phone calls, but believe that it may be possible with additional analysis. Using Wireshark we reconstructed the Voice over Internet Protocol (VoIP) flow graphs of the three phone calls made over SIP, seen in Figure 4.4 [72]. The first phone call made was from the rooted phone (in the Textfree application) to a personal phone, and it failed due to not having any calling credits available. The second call made was successful, and was from our personal phone to the rooted phone, which didn't require calling credits. Finally, after watching an advertisement and earning calling credits, we made the third phone call from the rooted phone to
our personal phone. The two successful calls were each a few seconds only. The SIP traffic from this experiment yielded similar results with the addition of requests and statuses relating to the phone calls, including ‘INVITE’, ‘TRYING’, ‘Not enough credit for call’, ‘RINGING’, ‘SESSION PROGRESS’, ‘ACK’, and ‘BYE’. From this analysis we conclude that the general SIP traffic seen on the nonstandard port (4544) is similar to that seen on the standard port (5060), but in our experiment, the phone calls made all used the standard port (5060). Moving forward, we continued to analyze the traffic during this new experiment, and found that the URL’s for the pictures had changed from [“picUrl”:"http://p.pinger.com/mms/427/asQLNt.jpg"] to [“picUrl”:"https://picmsg.co/mms/486/ItvVUi.jpg"] . Thus the servers used for storage were no longer part of their main website and used HTTPS instead of HTTP. At first glance this seemed to prevent the leakage of these images, but, these HTTPS links were still backwards compatible and could be accessed via HTTP by simply removing the ‘s’.

4.4.0.0.2 Tinder Tinder used HTTP for almost all of its traffic. Analyzing this unencrypted traffic allowed us to see all of the pictures of the Tinder user and the users they communicated with. The communication messages themselves were encrypted, but the pictures and location (city/district name and zip code) of the users were not. Corresponding with each image seen in the traffic, we found an HTTP ETag and Amazon Web Services (AWS) request IDs. HTTP ETags have been shown to be used by some companies as an alternative to cookies, and for tracking purposes [75]. Along with the location data was another number sent unencrypted which may be the user’s age, but although the majority of these numbers were in a normal range, we found outliers as low as 11 and as high as 287. Furthermore, Tinder’s privacy policy states that users must be 18 or over [59]. Although some data is transferred encrypted, it may be possible to gather more information through inference attacks. Analyzing the HTTP traffic, we broke down the URLs into the base URL, the user profile, the image size, and the picture name as they are stored on Tinder servers. We could then access these pictures stored on their servers via these URLs. These pictures were still accessible via these URLs around seven months after traffic collection, but could not be accessed when attempted again over a year after the initial test. Thus we conclude that these images are stored and can be accessed for a significant amount of time via
these URLs. By splitting up the traffic by user, we could find instances of the same user in the traffic at different times of the application’s usage. With a little understanding of the normal application usage, we know that if the user profile is repeated several times in a row with different pictures, then the Tinder user viewing the profile is browsing the other user’s pictures. Next, we know that it is supposedly unlikely to have a user profile seen again after ‘rejecting’ them, and we know that users cannot message each other or view profiles again until they have both ‘accepted’ each other. Using this information we can infer that two users ‘matched’ (by swiping right on each other) if we see a user profile repeated in the traffic, where the two points differ in a significant amount of time (such as a few minutes). We can then see if the user views the other’s profile pictures again, and deduce that the more times we see this, the longer their conversation. Thus, the messages themselves are encrypted, but it may be possible to infer information about the users, their usage patterns, and their personal information.

4.4.0.0.3 Imgur Imgur sent a little over half of its traffic unencrypted. This entailed 197 images (including duplicates of various sizes) being transmitted in the clear during our testing in the Android emulator. These images potentially leak user interests through inference attacks, and the privacy policy does not disclose the transmission of this content unencrypted [56]. This information describes and profiles a user through their choices to view certain content. This information could be valuable to ad companies for targeted advertising or adversaries aiming to learn more about a user for phishing attacks. Additionally, HTTP cookies were found in the traffic, which was disclosed in the policy. HTTP ETags were also found, and were not specifically mentioned in the privacy policy but may be considered part of the ‘other anonymous tracking information’ mentioned in their statement, “We may use cookies, web beacons, or other anonymous tracking information to improve our server’s interaction with your computer. [56]” As mentioned previously, HTTP ETags have been used by some companies for tracking purposes [75]. Additionally a mp4 video file and CSS file were found in the unencrypted traffic.

4.4.0.0.4 Pinterest During our Pinterest tests in both the Android emulator (>84% HTTP) and the rooted Google Nexus 6P (>75% HTTP) environments, our analysis allowed us to see and track nearly everything the user was doing, through
the content they viewed. Unencrypted content in the traffic included articles and pictures users viewed in the application, including Pinterest users’ profile pictures. These profile picture filenames included a Pinterest user’s username, which depending on what they chose, could be their real name. Additionally, unique identifiers, device information (OS/make/model/build), the HTTP user-agent field (also including device information), website cookies, and some javascript and miscellaneous HTML code was sent in the clear. Similar to other applications tested, this content leaks user interests through inference. Several pieces of this content was used for advertising purposes, as seen detailed in Table 4.3.

4.4.0.0.5 Vine The majority of Vine’s traffic was unencrypted (HTTP), which included every packet sent between the Vine application and Twitter servers, the parent company of Vine. Videos and images (video thumbnails/previews) viewed, as well as HTTP ETags and some cloud server requests were found in the unencrypted traffic. As discussed previously, HTTP ETags have been used for tracking purposes by some companies [75]. Similar to Imgur and Pinterest, the transmission of this content unencrypted was not disclosed in the privacy policy, and results in the potential leakage of user interests, which leads to multiple security/privacy concerns [62].

4.5 Data Sharing Module

4.5.1 Business Relationships

The business world is full of relationships, and the social media market is no different. Within our dataset of applications we determined a few business connections, as well as some others between social media companies worth noting. For instance, Twitter, Inc. owns Vine Labs, Inc. and thus it follows that any personal data from Vine is collected by the company and may be used for other applications they own. Though we did not investigate Instagram, it is worth noting that Facebook, Inc. owns Instagram and according to an article from Instagram’s Help Center, “We want to show you ads from businesses that are interesting and relevant to you, and to do that, we use information about what you do on Instagram and Facebook (our parent company) and on third-party sites and apps you use” [76]. This implies that
they are also obtaining information from other sources, “third-party sites and apps,” to better understand and predict a user’s actions. In addition to these relationships, we determined that Textfree is owned by Pinger, Inc, and Tumblr, Inc. is owned by Yahoo!, who is now owned by Verizon Inc. [77]. Thus, when a user believes they are only sharing their information with one of these applications/organizations, it is most likely not the case. Profiles of users maintained by these companies may be built from the aggregate information provided by all of the applications in a given business relationship.

### 4.5.2 Prominent Advertising/Analytics Libraries

In order to analyze the APK files and obtain more in-depth information about the ad and analytics libraries they used, we needed to retarget and decompile them. To accomplish this, we made use of a number of open-source tools. First we used Dare, a tool which retargets APK files into .class files by retargeting Dalvik bytecode to Java bytecode [51]. After obtaining the Java bytecode, we used Soot which decompiles Java bytecode into human readable .java files [52]. We could then obtain the libraries used by each application. We created a short script that performed a depth-first-search through the java directories and outputted a list of libraries. Finally, we researched each of the libraries and noted several prominent ones that past studies have investigated. These ad and analytics libraries used by the applications in our dataset can be seen in Table 4.4.

<table>
<thead>
<tr>
<th>Library</th>
<th>Facebook</th>
<th>Imgur</th>
<th>Messenger</th>
<th>Pinterest</th>
<th>Textfree</th>
<th>Tinder</th>
<th>Tumblr</th>
<th>Twitter</th>
<th>Vine</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bolts</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Flurry</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Google Ads/Admob</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fabric</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table 4.4. Prominent Advertising/Analytics Libraries Used By Each Application**

4.5.2.0.6 Bolts Bolts was developed by Parse and Facebook, Inc. originally for in-house use, but was later released as open source. Bolts does not require a developer account from either company. It can be used to enable a user to transition from one application to another. This is done through the creation of links, which contain all of the necessary data for this transition.
4.5.2.0.7 **Flurry**  Flurry is an ad and analytics service owned by Yahoo!. And as of 2017, it appears that Flurry is now owned and operated by Verizon Inc. [78]. Grace et al. studied mobile ad networks and found that Flurry probed the phone to see what permissions an app has before acting. They also found Flurry to collect location data [6]. Flurry’s documentation states that you can, “set up advanced analysis of complex events, with metrics, segments and funnels to better track your users’ habits and performance” [79].

4.5.2.0.8 **Google Ads/AdMob**  Google Inc., who owns AdMob, allows developers to analyze, monetize, and promote their applications. Grace et al. found that Google and AdMob probed permissions and collected location data [6].

4.5.2.0.9 **Fabric**  Fabric, created by Twitter, Inc., is a mobile development platform that includes crash reporting, user-tracking, user verification, tweet embedding, monetization, AWS Cloud syncing, payments, optimization, UX analysis, maps, real-time communication, and game analytics.

These are just a prominent few of the many ad/analytics libraries used in the applications we researched. These libraries, owned by large and prominent companies, are seen throughout the applications we tested. Thus from these business relationships it follows that a Tumblr user could have their information shared with Facebook, Yahoo!, Verizon, Google, and Twitter, without even linking accounts across these platforms.
Chapter 5 | Conclusions

Our findings show that the trend of Android mobile applications collecting and sharing sensitive data and PII, seen often in the first few years of the operating system’s existence, is still prevalent for popular social media applications. The practice of transmitting this data unencrypted, as well as to 3rd parties and ad networks can still be observed. As the applications in our dataset were all popular social media applications, it is likely that a given end user uses or has used one of these, and thus leaked information directly and/or inadvertently to the network, hops taken by the packets, and the companies in business with the application company. In the future, researchers should continue to investigate which applications across all domains are sending data in the clear. The inadvertent leakage of data to the networks traversed should be studied, and attempted to be quantified, such as by tracing the network path of an image transmitted in an application to the destination server. The lack of encryption technologies used by application companies could be investigated to better understand its prevalence and determine how to move towards the use of more encryption. Future work could also compare the direct/inadvertent exposure of sensitive data by free applications to that by paid versions. Applications which are transmitting over IPv6 could also be studied using our Analysis Framework. Future work should also investigate the Textfree application further, and determine whether the audio transmitted (SIP/RTP) from the bidirectional phone calls could be recovered with additional analysis. Additionally, the use of SIP on a nonstandard port (4544) should be analyzed further, as well as further comparison of this traffic to that seen on the standard port (5060) [71]. The paid features of the Textfree application, including those which can be redeemed by watching advertisements, should be investigated.
further as well. The payment information and logs of these transactions may not be transmitted securely, and these data transmission practices may be different when these services are paid for with currency vs. paid for by watching advertisements [33]. Because passive traffic sniffing can be used to build an attack profile of a target, future work with application research could attempt to combine unencrypted data collected from a variety of applications that a typical user might have on their phone, to develop possible profiles of a target. When multiple applications and different combinations of them are involved, it may then be possible to infer additional information.

Our analysis results also show the efficacy of our framework and testbed environments for testing android applications. Future work could leverage these and continue to investigate future versions of these social media applications, as well as other mobile applications in general. We found that for most applications, the emulator was a viable option for collecting traffic, as it was designed for developers to test applications before releasing them on the Google Play store. For this research, we used Android Studio v1.5, but with the more recent release of Android Studio v2.0, performance is much better and subsequent investigation is likely to have better results that more closely resemble an actual phone. The emulator provides an isolated environment that is quick and easy to instantiate for each run of the experiment; enabling fast installation and testing of one application on the device at a time. Physical phones may not be ideal for testing if applications running in the background are tainting the traffic. Additionally, running experiments on a variety of Android operating systems is trivial when using the Android Emulator, yet can be difficult when using physical devices. However, it is important to recall that while using Android Studio v1.5, a few of the applications we tested did not run properly in the Emulator. For the vast majority of applications, the emulator results resemble those seen on the physical device. Additionally, we find very similar results during our more realistic MITM attack scenario, thus reinforcing the accuracy of the data collected in these testbed environments. In the future, we would focus even more on ensuring traffic is not being tainted from background applications on the rooted device.

Android Social Media applications continued to exhibit the trend of the collection and direct/inadvertent exposure of privacy sensitive information, as well as lacked proper disclosure in their privacy policies. Of the nine social media applications we
studied, only Facebook, Messenger, Twitter, and Tumblr encrypted the majority of their traffic, which greatly reduced the sensitive data leakage. If more applications moved towards encrypting the majority, or all of their data transmissions, this inadvertent leakage of sensitive data and PII could be avoided. Direct leakages of sensitive data would still exist to the application companies and the parties they share data with, but the ISPs, supercomputing centers, network administrators, and router owners who inhabit and transmit data at the hops between phones and application company servers, would see significantly less interpretable data. Application privacy policies need to be more descriptive and explain to users better exactly how their data is transmitted, used, and secured. Simply stating that data is stored securely and that user privacy is protected means very little when the majority of the data is transmitted unencrypted and exposed to the network and hops along the data paths. Now that recent FCC rulings have officially been reversed, ISPs can continue to collect, sell, and share user data legally. Adding to this, certain geographical areas are only covered by certain ISPs, all of which may take advantage of user data to some extent. This can leave end users with no choice but to agree to the terms of these companies, pay extra fees for privacy, as Comcast and AT&T have publicly expressed interest in making standard, or forgo Internet services [17] [18]. Ensuring user security and privacy should be a standard across mobile applications, and should be perfected by social media applications, as these transmit and share sensitive data and PII in their nature.
Appendix A

P3P Specification

The following documentation in this Appendix are reformatted and truncated copies of sections ‘3.4 Categories and the CATEGORIES element’ and ‘3.3.4 The PURPOSE element’ of the P3P Specification [1]. These have been reproduced and truncated for the purpose of relating to this research. Some text and sentences have been removed from these sections, including some XML data tags and guidelines, in an attempt to just keep the P3P Categories and Purposes themselves, along with their descriptions. Additionally, we have created and appended acronyms to the P3P Categories and Purposes to provide a labeling system and key for use with matching research data to the P3P terminology in Table 4.3.

A.1 3.4 Categories and the CATEGORIES element

Categories are elements inside data elements that provide hints to users and user agents as to the intended uses of the data. Categories are vital to making P3P user agents easier to implement and use. Note that categories are not data elements: they just allow users to express more generalized preferences and rules over the exchange of their data. The following elements are used to denote data categories:

[PCI] Physical Contact Information: Information that allows an individual to be contacted or located in the physical world – such as telephone number or address.

[OCI] Online Contact Information: Information that allows an individual to be contacted or located on the Internet – such as email. Often, this
information is independent of the specific computer used to access the network. (See the category “Computer Information”)

[UI] Unique Identifiers: Non-financial identifiers, excluding government-issued identifiers, issued for purposes of consistently identifying or recognizing the individual. These include identifiers issued by a Web site or service.

[PuI] Purchase Information: Information actively generated by the purchase of a product or service, including information about the method of payment.

[FI] Financial Information: Information about an individual’s finances including account status and activity information such as account balance, payment or overdraft history, and information about an individual’s purchase or use of financial instruments including credit or debit card information. Information about a discrete purchase by an individual, as described in “Purchase Information,” alone does not come under the definition of “Financial Information.”

[CI] Computer Information: Information about the computer system that the individual is using to access the network – such as the IP number, domain name, browser type or operating system.

[NCD] Navigation and Click-stream Data: Data passively generated by browsing the Web site – such as which pages are visited, and how long users stay on each page.

[ID] Interactive Data: Data actively generated from or reflecting explicit interactions with a service provider through its site – such as queries to a search engine, or logs of account activity.

[DSD] Demographic and Socioeconomic Data: Data about an individual’s characteristics – such as gender, age, and income.

[C] Content: The words and expressions contained in the body of a communication – such as the text of email, bulletin board postings, or chat room communications.
[SMM] State Management Mechanisms: Mechanisms for maintaining a stateful session with a user or automatically recognizing users who have visited a particular site or accessed particular content previously – such as HTTP cookies.

[Pol] Political Information: Membership in or affiliation with groups such as religious organizations, trade unions, professional associations, political parties, etc.

[HI] Health Information: Information about an individual’s physical or mental health, sexual orientation, use or inquiry into health care services or products, and purchase of health care services or products.

[PrD] Preference Data: Data about an individual’s likes and dislikes – such as favorite color or musical tastes.

[LD] Location Data: Information that can be used to identify an individual’s current physical location and track them as their location changes – such as GPS position data.

[GI] Government-issued Identifiers: Identifiers issued by a government for purposes of consistently identifying the individual.

[O] Other: Other types of data not captured by the above definitions. (A human readable explanation should be provided in these instances, between the <other-category> and the </other-category> tags.)

The Computer, Navigation, Interactive and Content categories can be distinguished as follows. The Computer category includes information about the user’s computer including IP address and software configuration. Navigation data describes actual user behavior related to browsing. When an IP address is stored in a log file with information related to browsing activity, both the Computer category and the Navigation category should be used. Interactive Data is data actively solicited to provide some useful service at a site beyond browsing. Content is information exchanged on a site for the purposes of communication.

The Other category should be used only when data is requested that does not fit
into any other category.

P3P uses categories to give users and user agents additional hints as to what type of information is requested from a service. While most data in the base data schema is in a known category (or a set of known categories), some data elements can be in a number of different categories, depending on the situation. The former are called fixed-category data elements (or “fixed data elements” for short), the latter variable-category data elements (“variable data elements”). Both types of elements are described in Section 5.7.

A.2 3.3.4 The PURPOSE element

Each STATEMENT element that does not include a NON-IDENTIFIABLE element MUST contain a PURPOSE element that contains one or more purposes of data collection or uses of data. Sites MUST classify their data practices into one or more of the purposes specified below.

[CSA] Completion and Support of Activity For Which Data Was Provided: Information may be used by the service provider to complete the activity for which it was provided, whether a one-time activity such as returning the results from a Web search, forwarding an email message, or placing an order; or a recurring activity such as providing a subscription service, or allowing access to an online address book or electronic wallet.

[WSA] Web Site and System Administration: Information may be used for the technical support of the Web site and its computer system. This would include processing computer account information, information used in the course of securing and maintaining the site, and verification of Web site activity by the site or its agents.

[RD] Research and Development: Information may be used to enhance, evaluate, or otherwise review the site, service, product, or market. This does not include personal information used to tailor or modify the content to the specific individual nor information used to evaluate, target, profile or contact the individual.
[OT] **One-time Tailoring:** Information may be used to tailor or modify content or design of the site where the information is used only for a single visit to the site and not used for any kind of future customization. For example, an online store might suggest other items a visitor may wish to purchase based on the items he has already placed in his shopping basket.

[PA] **Pseudonymous Analysis:** Information may be used to create or build a record of a particular individual or computer that is tied to a pseudonymous identifier, without tying identified data (such as name, address, phone number, or email address) to the record. This profile will be used to determine the habits, interests, or other characteristics of individuals for purpose of research, analysis and reporting, but it will not be used to attempt to identify specific individuals. For example, a marketer may wish to understand the interests of visitors to different portions of a Web site.

[PsD] **Pseudonymous Decision:** Information may be used to create or build a record of a particular individual or computer that is tied to a pseudonymous identifier, without tying identified data (such as name, address, phone number, or email address) to the record. This profile will be used to determine the habits, interests, or other characteristics of individuals to make a decision that directly affects that individual, but it will not be used to attempt to identify specific individuals. For example, a marketer may tailor or modify content displayed to the browser based on pages viewed during previous visits.

[IA] **Individual Analysis:** Information may be used to determine the habits, interests, or other characteristics of individuals and combine it with identified data for the purpose of research, analysis and reporting. For example, an online Web site for a physical store may wish to analyze how online shoppers make offline purchases.

[InD] **Individual Decision:** Information may be used to determine the habits, interests, or other characteristics of individuals and combine it with identified data to make a decision that directly affects that individual. For example, an online store suggests items a visitor may wish to purchase based on items he has purchased during previous visits to the Web site.
[CVM] **Contacting Visitors for Marketing of Services or Products:** Information may be used to contact the individual, through a communications channel other than voice telephone, for the promotion of a product or service. This includes notifying visitors about updates to the Web site. This does not include a direct reply to a question or comment or customer service for a single transaction – in those cases, `<current/>` would be used. In addition, this does not include marketing via customized Web content or banner advertisements embedded in sites the user is visiting – these cases would be covered by the `<tailoring/>`, `<pseudo-analysis/>` and `<pseudo-decision/>`, or `<individual-analysis/>` and `<individual-decision/>` purposes.

[HP] **Historical Preservation:** Information may be archived or stored for the purpose of preserving social history as governed by an existing law or policy. This law or policy MUST be referenced in the `<DISPUTES>` element and MUST include a specific definition of the type of qualified researcher who can access the information, where this information will be stored and specifically how this collection advances the preservation of history.

[CVMT] **Contacting Visitors for Marketing of Services or Products Via Telephone:** Information may be used to contact the individual via a voice telephone call for promotion of a product or service. This does not include a direct reply to a question or comment or customer service for a single transaction – in those cases, `<current/>` would be used.

[OU] **Other Uses:** Information may be used in other ways not captured by the above definitions. (A human readable explanation MUST be provided in these instances).

Service providers MUST use the above elements to explain the purpose of data collection. Service providers MUST disclose all that apply. If a service provider does not disclose that a data element will be used for a given purpose, that is a representation that data will not be used for that purpose. Service providers that disclose that they use data for “other” purposes MUST provide human readable explanations of those purposes.
Appendix B
Application Dataset Descriptions

In this Appendix we describe the Android applications in our dataset, the services they provide, and how users interact with them. Additionally, we discuss what we believe to be the components of these services which would handle and transmit sensitive data and/or PII.

The following examples of transmitting sensitive data and PII applies to all of the applications. Since most of the applications in our dataset require users to create accounts, and all provide this as an option, each application may transmit usernames/email addresses and passwords. The amount of PII transmitted also largely depends on how much personal information is added to the user account, which may or may not be required to be filled out by an application.

B.1 Facebook

A free service where users can add friends to keep up with them. Users can share ‘updates’, photos, videos, and events, as well as view these things shared by friends they’ve added or by public pages they have chosen to follow. Users can also post directly onto the ‘walls’ of other users or public pages. Users can comment on posts made, and ‘like’ them, or other comments. Additionally users can play games and use some additional ‘apps’ within the service [31].

Facebook provides users with the ability to tune their privacy settings, such as only sharing posts with Friends, or limiting what pictures and posts users that are not friends with them can see on their wall. In general, the majority of the tasks on Facebook would not entail the transmission of PII or sensitive data, but could. Anyone could share what they choose in these updates, photos, videos, events, etc,
such as posting an update with a new phone number, email address, or even mailing address. Users can also add their phone numbers, email addresses, and personal information to their profile, to share with friends and other users depending on their privacy settings [31].

B.2 Imgur

A free service that allows users to post / view photos and GIFs (a type of image which usually contains a short animation). The community is known for sharing “hilarious GIFs, heartwarming stories, mind blowing science facts, adorable animal pics and so much more.” Users throughout the world post content, and users can ‘upvote’ or ‘downvote’ content, as well as comment [34].

The content and comments a user posts, as well as the usage trails generated as they browse the application’s content could entail sensitive data.

B.3 Messenger

A free service that uses a user’s Facebook account to give them the ability to message or call Facebook friends or friends in their phone book via their phone number. Group messages and calls can also be made. The service is similar to text messaging, but it is free over Wi-Fi and otherwise is transmitted through a user’s data plan. A user’s location information can be turned on to let other users communicating with them know where they currently are located [30].

Messages and calls sent/received, the content within, and the metadata associated with these transactions could contains sensitive data and/or PII. Additionally, the location information attached to these conversations would be considered sensitive information.

B.4 Pinterest

A free service described as a ‘visual bookmarking tool’ that allows users to discover and share a wide range of pictures and articles based on their interests. The user initially provides the service with a list of interests and hobbies, and it tailors its content based off of these topics, as well as updates their interests as they browse
and ‘pin’ (like) certain content. Pinterest advertises the ability for its users to: plan a project; explore a hobby; find travel inspiration, or discover their style [32].

The user’s interests, the content they view and ‘pin’, and the usage trails and habits they create can entail sensitive data. Any location data provided to the service would also be sensitive data. Additionally, a purchase made within the application may entail a user’s credit card information, or other payment information being transmitted, whether through the service itself or the Google Play Store. This may also entail a physical address and contact information for shipping a product purchased.

**B.5 Textfree**

A freemium service that provides the user with a real, local US phone number that can send and receive text messages (SMS/MMS) and phone calls. It is also able to communicate with cell phones that are not using the app, through their phone numbers. The service advertises features such as “free SMS messaging, group messaging, free MMS picture messaging, international texting & calling, and voicemail all for free.” It also says it provides “unlimited free texts & pictures”. From our experiments we found it to require calling credits for outbound phone calls, which can be purchased or earned when watching advertisements [33].

Since this service is providing text and picture messaging (SMS/MMS) and phone calls over Wi-Fi, all of the content is transmitted by the service, and may contain sensitive data or PII. Similar to Pinterest, an in-app purchase may entail a user’s credit card, or other payment information being transmitted in order to be processed, whether through the service itself or the Google Play Store.

**B.6 Tinder**

A freemium dating service that allows users to connect with other users in their area based on a mutual attraction. It boasts being the “world’s most popular dating app,” with over 9 billion ‘matches’ made between users, as of the date of the version used in our study. Everyone using the application is ‘authenticated’ through Facebook to verify a user’s identity and pull data such as mutual friends and interests to aid in matching. Normal use of the application entails a user
browsing through other user’s profiles and pictures and either swiping right to express interest, or swiping left to express disinterest. If two users both swiped right on each others profiles, expressing a mutual interest, a ‘match’ is made. These matched users can now message each other through the application. It is also possible for a user to ‘Super Like’ another user by swiping up on their profile, expressing to them that they are especially interested. Furthermore, users have the ability to unmatch with another user at any time. The basic features of the application are provided for free, but an additional service called Tinder Plus can be purchased. Tinder Plus provides more features including Passport, the ability to look at people outside of their local area, and Rewind, the ability to bring back their latest swipe [29].

Sensitive data transmitted may entail a user’s profile content: information they provided about themselves, their pictures, and their general location, shared to other users viewing their profile. The usage trails generated by a user viewing and swiping on other users’ profiles, as well as the messages between users that have ‘matched’, are also transmitted by the service and can contain sensitive information. Furthermore, similar to Pinterest and Textfree, since Tinder has in-app purchases, payment information may be transmitted in order to process the charges, whether through the service itself or the Google Play Store.

B.7 Tumblr

A free service that allows users to create their own blog by reposting other users’ blog content or creating their own. Users can also communicate through conversations around a post. Tumblr encourages users to make their Tumblr blog their own, thus potentially meaning a user’s page will be very personal [35].

A user’s interests and opinions, the content they view and post on their Tumblr blog, the conversations they have on other posts, and the usage trails and habits they create can entail sensitive data when tied to them specifically.

B.8 Twitter

A free service that allows users to share messages, pictures, and videos instantly with users that follow them. Users can also view ‘tweets’ shared by users they
follow. Users can use hashtags, such as #PennState in their tweets, which enables users to find tweets which use these specifics hashtags, and find out more about topics they are interested in. Users are also able to message each other through the service. Twitter advertises the ability to get breaking news, and “Stay informed with the local and global news that matters to you most, as it happens.” Users can also view the best content on Twitter with Moments, which allows them to “Follow top stories through immersive pics, clips, and conversations” as well as, “Get insights and perspectives you won’t find anywhere else [36].”

Tweets posted by users, messages sent and received, and the usage trails generated may contain sensitive data.

**B.9 Vine**

A free service that allows users to share up to 6 second videos called ‘vines’ with their followers, watch vines posted by other users they follow, or explore popular vines at the time by category. Users can share vines by ‘revining’ them or posting them on social media [37].

Vines posted by users, and the usage trails generated may contain sensitive data.
Appendix C
Application Analysis Graphs

The following graphs were outputted by our Application Analysis module and display the summaries of the application traffic collected during our experiments. The first nine graphs are output from our Case 1 experiments using the Google’s Android Emulator and the last five graphs are output from our Case 2 experiments using the rooted Google Nexus 6P phone.

Figure C.1. Analysis of Facebook Traffic - Android Emulator
Figure C.2. Analysis of Imgur Traffic - Android Emulator

Figure C.3. Analysis of Messenger Traffic - Android Emulator
Figure C.4. Analysis of Pinterest Traffic - Android Emulator

Figure C.5. Analysis of Textfree Test 1 Traffic - Android Emulator
Figure C.6. Analysis of Textfree Test 2 Traffic - Android Emulator

Figure C.7. Analysis of Tumblr Traffic - Android Emulator
Figure C.8. Analysis of Twitter Traffic - Android Emulator

Figure C.9. Analysis of Vine Traffic - Android Emulator
Figure C.10. Analysis of Pinterest Traffic - Rooted Nexus 6P

Figure C.11. Analysis of Textfree Traffic - Rooted Nexus 6P
Figure C.12. Analysis of Textfree MITM (arpspoof) Attack Traffic - Rooted Nexus 6P

Figure C.13. Analysis of Textfree Retested Feb 2017 Traffic - Rooted Nexus 6P
Figure C.14. Analysis of Tinder Traffic - Rooted Nexus 6P

Figure C.15. Analysis of Vine Traffic - Rooted Nexus 6P
Bibliography


