What Phone Applications Want From You

Jonathan Bryan[1] and Josh Herman[2]

***Abstract*— Privacy concerns have plagued users across a multitude of platforms, with one of the most notable concerns being mobile phone applications. In the past, many studies have looked at how companies gather and misuse users data for a plethora of different reasons. With these privacy concerns still being prevalent to this day, we decided to look at the current state of application permissions, as well as how certain traits may affect the likelihood of applications requesting more sensitive information from users. Doing so, we found that while company sizes do not seem to directly affect how many permissions are required by applications, our results do show that there are still some applications that request an excessive amount of permissions. Additionally, some of these requested permissions do not seem to even contribute to the core functionality of the application, indicating a potentially malicious purpose.**

I. Objective

Technology over the years has paved the way for many ground-breaking and innovate discoveries that have allowed for incredible progress in society. The integration of technology into society has become an ever-growing and necessary aspect to the lives of many individuals and has resulted in a large amount of users becoming reliant on this powerful entity to make their lives marginally more simple. One primary conduit of technology that has become a focal point in the lives of many, is the presence smartphones. With the ever expansive growth and influence presented by this monolithic addition to society, there has been an increase in concern regarding the perversion of users privacy, and the desire by large companies to obtain valuable data from these users. Through smartphones, it is possible for users to gain access to a plethora of apps to make their daily lives more manageable or to simply entertain themselves, with some examples of apps being *Trello*, *Facebook*, and many others. With the large expanse of options users have at their disposal, it is important for users and companies hosting these apps to understand how companies may be trying to manipulate users and gather personal data about them.

For this study, our aim is to produce a quantitative analysis over a large amount of apps found on the Google Play Store to observe the varying permissions requested by apps. Alongside this, we also plan on applying certain filters to these results such as popularity, genre, and so on, to determine if there is a significant difference in, say, the difference between what permissions are requested from an app that is used by million of users versus an app used by a few thousand. By doing so, we hope to be able to distinguish apps that may be requesting unnecessary permissions for malicious intentions. Similarly, we also hope that by performing this study we can incite the need to improve user awareness to prevent users from forgoing their data to companies who would look to misuse or sell their information.

II. Motivation

Many apps on app stores today require permissions on the user's phone to function properly. These can include location services, microphone/camera, SMS, storage, and even body sensors. While some of these apps legitimately need and use these permissions, there are some apps that do not need these permissions. For instance, it is clear why a messaging app would need access to the phone's SMS system, or a voice communication application would need access to the microphone. However, a simple game application does not generally need access to the phone (phone number, making calls, etc). While this may seem obvious at first glance, many companies have successfully marketed their applications in such a way that users easily become addicted. This type of addiction can encourage users to accept any and all permission requests simply to continue using the app, with no thought into what exactly they are agreeing to. As such, there is an obvious need to improve user awareness and to observe on a large scale the type of information and access that these businesses want from users.

Our primary motivation for conducting a quantitative study over a large number of applications found on the Google Play Store was inspired by [Lin+14]. In their paper, they evaluated 108,246 free applications found on the Google Play Store by gathering each applications metadata (e.g. app name, developer name, etc.), as well as their binary data using an open-source google play API. While the authors of this paper were able to effectively monitor the invasive nature of permission requests by conducting their study in this method, we did not have enough time to create a similar study, and instead took inspiration from the metadata that they collected. As such, we believe that the metrics collected for our study will allow us to see the current state of application permissions across multiple categories.

III. Data

A. Data Collection

For data collection, we considered two options. One was to use a dataset with applications permissions that was found during MiniProject2, and the second was to use a NodeJS API to scrape current data from the Google Play Store. Ultimately, we decided to use the API to get the most current data that we could. The API we used is called "Google Play Scraper", and allows for searching for specific applications, or specific categories of applications. The API allowed us to search for apps by category (genre) and collection (top free,

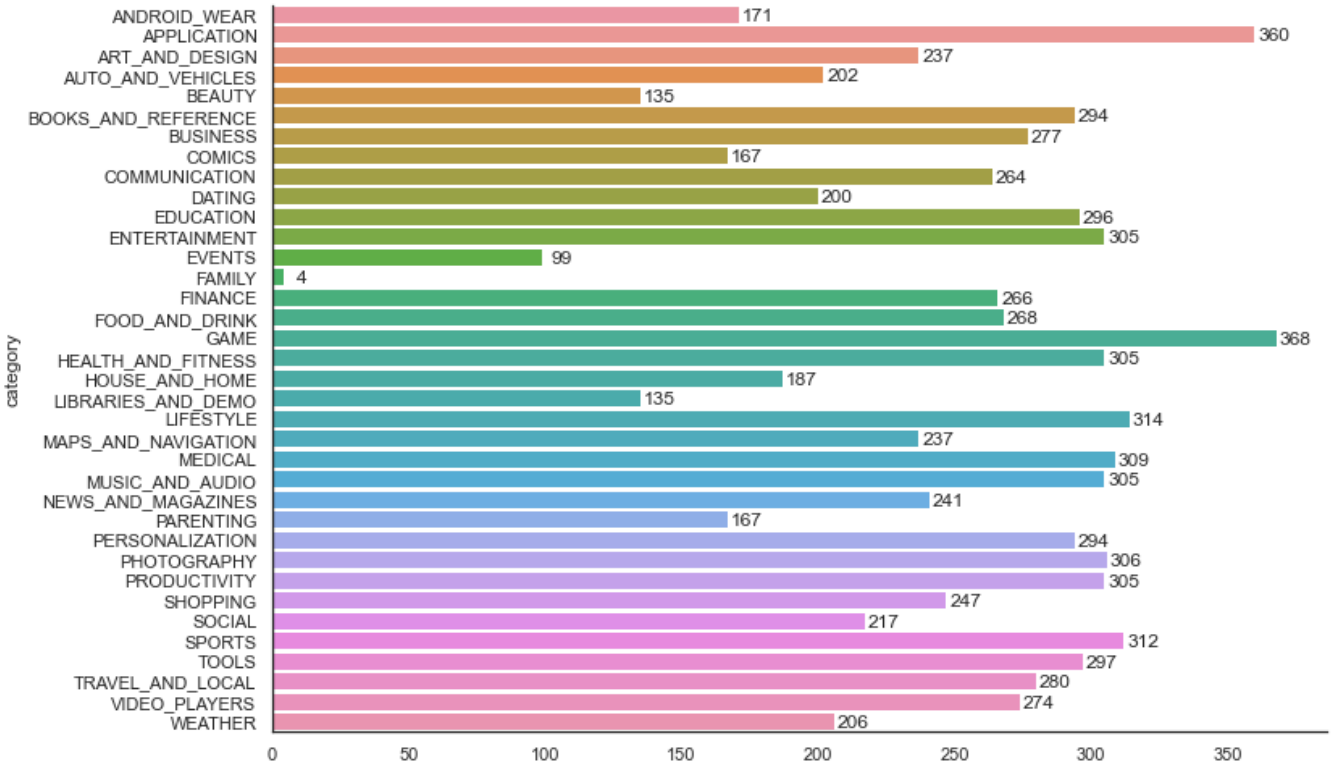

Fig. 1. Number of Applications per Category

top selling, top new, etc). We utilized this API to gather information on every category and collection combination. To keep the compute time and storage requirements low, we decided to collect information on 100 applications of each category and collection combination. This resulted in 13,005 applications total, and 8,851 after normalization. Normalization consisted of removing duplicates from our data set. **Figure 1** shows the number of applications in our dataset per category.

B. Data Analysis

After collecting all of the data, we stored the application data on a local MongoDB database. Once data collection was finished, we pulled all of the data from the database, and stored it in a Pandas data frame. This allowed us to easily sort by category, collection, and popularity (number of installs). We used a combination of MatPlotLib and SeaBorn for all of our graphing needs. We used Pandas to find the most/least used permissions in the dataset, the apps with the highest number of request permissions, and the most/least used permission per category. Then, we looked at a few companies by hand, and categorized them by size. We looked up the company on LinkedIn and used the employee count listed there to categorize them. We considered a company with 0 - 100 employees small, 100 - 1500 medium, and 1500+ large.

C. Hypotheses

1) Large Companies vs Small Companies: The hypothesis for this section is that applications made by larger companies will require more permissions than apps made by smaller companies. We believe that larger companies are going to gather more data from users than they really need.

2) Differences Between Genres: We believe that there will be differences between genres when considering the permissions requested. We believe that different genres of apps will be completing different tasks, thus they will require different permissions.

3) General Outcome: The final expected outcome is a more general one. We suspect that various applications require more permissions then they really require for core functionality. We also believe that while an app legitimately requires a permission, it can lead to a security vulnerability.

IV. Results

After evaluating the plethora of applications gathered for our project, we were able to observe how different traits affect both the number of permissions, as well as what permissions are requested. Looking at **Table 1** and **Table 2**, we looked at observing the top 15 and least 15 most common permissions, respectively, across every application we evaluated. Additionally, we also looked at what the most popular application associated with each was (indicated by the number of installation), as well as the size of the company that made each application, which was indicated by small, medium, or large. Doing so, we noticed that while the most common permissions did not seem initially malicious, with many of these permissions seemingly being needed for core functionality, we also found that there did not seem to be any

category	Permission 1	Permission 2	Permission 3	Permission 4
PERSONALIZATION	read cell broadcast messages	access extra location provider commands	make/receive SIP calls	Change WiMAX state
TOOLS	set preferred apps	erase USB storage	receive text messages (MMS)	send SMS messages
APPLICATION	retrieve system internal state	Change WiMAX state	limit number of running processes	set time zone
ANDROID_WEAR	change/intercept network settings and traffic	directly call any phone numbers	read your Web bookmarks and history	write web bookmarks and history
COMMUNICATION	body sensors (like heart rate monitors)	delete all app cache data	add words to user-defined dictionary	mock location sources for testing
SOCIAL	modify phone state	make/receive SIP calls	read battery statistics	read your text messages (SMS or MMS)
PRODUCTIVITY	modify phone state	erase USB storage	read terms you added to the dictionary	add words to user-defined dictionary
BUSINESS	modify phone state	erase USB storage	delete all app cache data	change/intercept network settings and traffic
ART_AND_DESIGN	add or modify calendar events and send email to guests without owners' knowledge	mock location sources for testing	write web bookmarks and history	reorder running apps
LIFESTYLE	retrieve system internal state	delete all app cache data	read your text messages (SMS or MMS)	receive text messages (SMS)
ENTERTAINMENT	set time zone	erase USB storage	enable app debugging	expand/collapse status bar
MAPS_AND_NAVIGATION	reroute outgoing calls	receive text messages (MMS)	receive text messages (SMS)	add words to user-defined dictionary
AUTO_AND_VEHICLES	reroute outgoing calls	write call log	uninstall shortcuts	directly call any phone numbers
SPORTS	send SMS messages	read your text messages (SMS or MMS)	modify your contacts	read terms you added to the dictionary
LIBRARIES_AND_DEMO	read calendar events plus confidential information	add or modify calendar events and send email to guests without owners' knowledge	read call log	write call log
BEAUTY	read your Web bookmarks and history	read sensitive log data	access USB storage filesystem	close other apps
PHOTOGRAPHY	add words to user-defined dictionary	modify your contacts	change/intercept network settings and traffic	access extra location provider commands
HEALTH_AND_FITNESS	modify phone state	read your Web bookmarks and history	send SMS messages	set time zone
PARENTING	uninstall shortcuts	delete all app cache data	enable app debugging	set wallpaper
SHOPPING	access USB storage filesystem	measure app storage space	delete all app cache data	allow Wi-Fi Multicast reception
EDUCATION	access extra location provider commands	read battery statistics	mock location sources for testing	change system display settings
MUSIC_AND_AUDIO	act as the AccountManagerService	read terms you added to the dictionary	read your own contact card	modify phone state
FINANCE	expand/collapse status bar	disable your screen lock	read call log	act as the AccountManagerService
BOOKS_AND_REFERENCE	change/intercept network settings and traffic	read battery statistics	modify phone state	modify your contacts
MEDICAL	uninstall shortcuts	enable app debugging	directly call any phone numbers	send SMS messages
VIDEO_PLAYERS	measure app storage space	erase USB storage	enable app debugging	control Near Field Communication
NEWS_AND_MAGAZINES	access extra location provider commands	access USB storage filesystem	modify phone state	measure app storage space
TRAVEL_AND_LOCAL	change system display settings	enable app debugging	read your own contact card	modify your contacts
DATING	measure app storage space	mock location sources for testing	close other apps	set an alarm
FOOD_AND_DRINK	modify phone state	expand/collapse status bar	set wallpaper	add words to user-defined dictionary
WEATHER	modify phone state	change your audio settings	add or remove accounts	read your contacts
HOUSE_AND_HOME	access extra location provider commands	read battery statistics	body sensors (like heart rate monitors)	add or remove accounts
EVENTS	install shortcuts	control Near Field Communication	read your own contact card	access USB storage filesystem
COMICS	send sticky broadcast	read your Web bookmarks and history	access Bluetooth settings	reorder running apps
GAME	change system display settings	expand/collapse status bar	erase USB storage	set wallpaper
FAMILY	take pictures and videos	Google Play license check	control vibration	record audio

Fig. 2. Least used Permissions per Category

Permissions	Count	Most Popular Application	Installs	Company Size
read the contents of your USB storage	604	WhatsApp Messenger	5,000,000,000+	Large
modify or delete the contents of your USB storage	588	WhatsApp Messenger	5,000,000,000+	Large
full network access	359	Subway Surfers	1,000,000,000+	Medium
view network connections	355	Subway Surfers	1,000,000,000+	Medium
prevent device from sleeping	332	Subway Surfers	1,000,000,000+	Medium
read phone status and identity	290	Google Duo - High Quality Video Calls	1,000,000,000+	Large
view Wi-Fi connections	286	Subway Surfers	1,000,000,000+	Medium
control vibration	231	Subway Surfers	1,000,000,000+	Medium
run at startup	225	WhatsApp Messenger	5,000,000,000+	Large
precise location (GPS and network-based)	222	Waze-GPS, Maps, Traffic Alerts & Live Navigation	100,000,000+	Medium
find accounts on the device	210	WhatsApp Messenger	5,000,000,000+	Large
take pictures and videos	210	B612 - Beauty & Filter Camera	500,000,000+	Medium
approximate location (network-based)	182	Waze - GPS, Maps, Traffic Alerts & Live Navigation	100,000,000+	Medium
set wallpaper	165	ZEDGE^{TM} Wallpapers & Ringtones	100,000,000+	Small
read your contacts	137	Google Chrome: Fast & Secure	5,000,000,000+	Large

TABLE I

TOP 15 MOST COMMON APP PERMISSIONS

outstanding results regarding how the company sizes affected these permission requests, aside from the fact the majority of companies were either medium or large in size. Looking at the least common permissions however, we did notice that there were some permissions that did seem uncommon, such as 7 applications requesting to erase USB storage, and the most popular application looking to do so is a VPN application. Additionally, it can also be seen that the majority

Permissions	Count	Most Popular Application	Installs	Company Size
access serial ports	1	Peloton - at home fitness	500,000+	Large
read cell broadcast messages	1	Clone App - App Cloner & Parallel Space	5,000,000+	Small
modify global animation speed	1	Rosetta Stone: Learn Languages	10,000,000+	Medium
limit number of running processes	2	SmartThings	100,000,000+	Large
Change WiMAX state	5	Parallel Space - Multiple accounts & Two face	100,000,000+	Small
set preferred apps	6	GO Launcher - 3D parallax Themes & HD Wallpapers	100,000,000+	Medium
erase USB storage	7	Mobile Security: VPN Proxy & Anti Theft Safe WiFi	50,000,000+	Large
receive text messages (WAP)	7	Messages	1,000,000,000+	Large
directly call any phone numbers	7	Google	5,000,000,000+	Large
retrieve system internal state	7	Android Auto - Google Maps, Media & Messaging	500,000,000+	Large
enable app debugging	8	Pregnancy + tracker	10,000,000+	Large
act as the AccountManagerService	9	Google Play Music	5,000,000,000+	Large
set time zone	9	Parallel Space - Multiple accounts & Two face	100,000,000+	Small
make app always run	10	UC Browser- Free & Fast Video Downloader, News...	500,000,000+	Medium
add voicemail	10	Truecaller: Caller ID, block fraud & scam calls	500,000,000+	Medium

TABLE II

15 LEAST COMMON APP PERMISSIONS

of the permissions found on **Table 2** seem to be from large companies, which could indicate malicious intent from larger

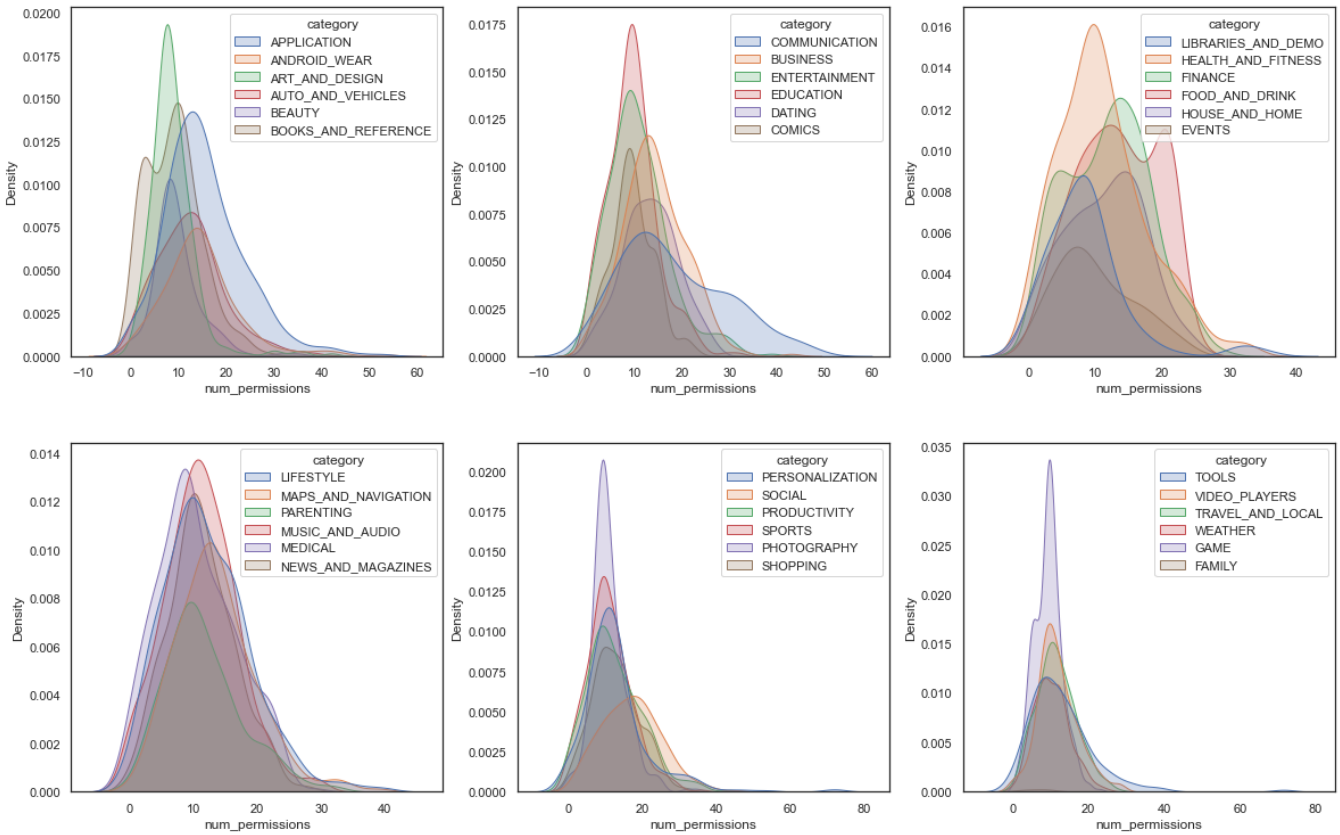

Fig. 3. Probability Density Function for the number of permissions per category

companies.

Taking the results found from these tables into account, we then also looked at the least used permissions per category, as seen in **Figure 2**, to determine if we might be able to derive a similar result. We initially looked towards both the most used permissions per category and the least used, but in doing so we found that the most used permissions were generally the same and did not yield interesting results, as many of these permissions were used to perform core functionality. After performing this evaluation, we were able to find some interesting results. For starters, while some of the permissions

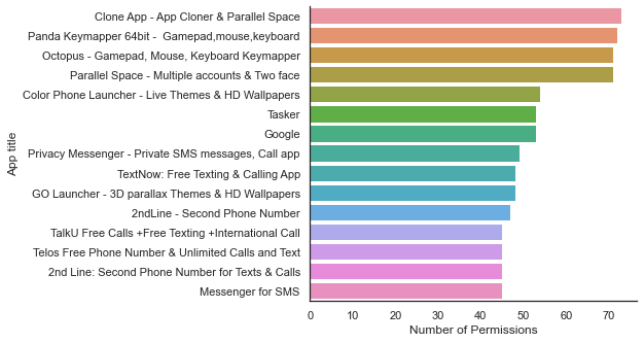

Fig. 4. Top 15 Applications with the most permissions requirements

could make sense, such as an *ANDROID_WEAR* application requesting to directly call any phone numbers or access bookmarks and history to perform core functions, other genres raise some concern of why applications would need access to some of the permissions being requested. Some examples of this include the *BUSINESS* genre requesting access to erase USB storage or modify the phone state, as well as *COMICS* requesting to send a sticky broadcast or reorder running apps. While it is possible that these permissions could be necessary for applications in these genres, it is also possible that these permissions could be used for malicious purposes.

After looking at the types of permissions requested by varying applications and genres, we then also looked towards observing how many permissions are generally required for applications, and what factors affect this. Doing so, we first looked at the top 15 applications with the most permissions, as seen in **Figure 4**, which shows an alarming number of requests by some applications, with 4 requesting over 70 permissions. While a clone app for example (the application that requires the most permissions) may need this many to properly clone an entire application, a gamepad/mouse/keyboard mapper app for example (3rd application with the most permissions), should not require over 70 permissions to simulate this functionality. This shows that there are some applications with potentially malicious or otherwise invasive motives.

To further examine this grandiose number of permissions required by applications, we looked towards determining

how certain factors may affect these numbers. To do so, we looked at two particular factors. First, we looked at observing the Probability Density Function (PDF) of the number of applications per category, as seen in **Figure 3**, which shows the likelihood of each genre having a specific number of permissions. It can be seen that while this result did need to be separated into 6 genres per chart, due to the fact it was difficult to properly visualize all 36 genres in one chart, it is clear that some genres seem to require more permissions than others. Notably, when looking at the third chart (e.g. *FINANCE*, *EVENTS*, etc.), we can see that *FOOD_AND_DRINK* and *FINANCE* seem to generally require more permissions than other genres in that same chart. While we did have difficulty properly analyzing these results, we do believe that it provides some insight into the large number of permissions being requested by applications these days. Additionally, when looking at **Figure 5** we

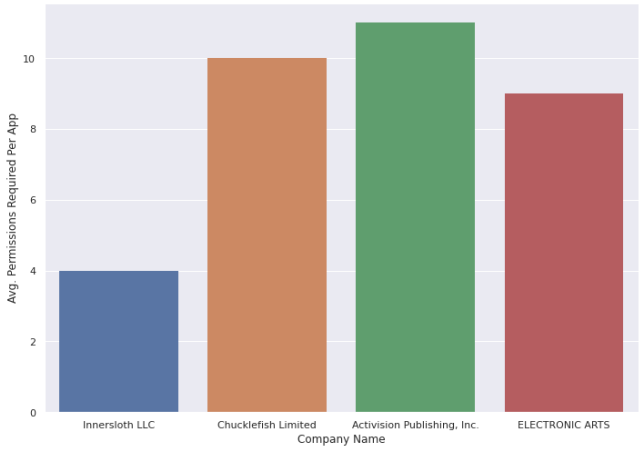

Fig. 5. Average Number of Permissions per Company, Small Companies (left 2) vs. Large Companies (right 2)

looked at determining if company sizes affected the number of permissions generally required. For this analysis, we looked solely at the *GAMING* genre, specifically looking at 2 small companies and 2 large companies. Doing so, we did not find an outstanding difference in the number of applications requested between company sizes, rather, the average number of permissions seemed to be increasingly similar. While this result could change by looking at different genres outside of solely the gaming *GAMING* category, we were not able to easily access company sizes in an automated manner.

V. Conclusion

In this project we aimed to answer 3 core questions. These are: How does the size of the company affect permissions requested, how does the app genre affect the permissions requested, and are there any applications that request permissions for potentially malicious purposes.

A. Large Companies vs Small Companies

Looking at **Fig. 5** it is clear that the size of the company does not impact the number of permissions requested. Both the large and small companies requested similar numbers of permissions. While more research is certainly needed to confirm this, we believe that company size does not have a large impact, but the goals/purpose of the application does.

B. Differences Between Genres

As stated in results, the most requested permissions by category did not change much. In hindsight, this makes sense, since just about every single app will need USB and Network access. However, as **Fig. 2** shows, the least requested permission by category has some notable differences. Since it's the least requested permissions that change, we believe that these differences show up since each app is trying to accomplish something different. Since every app needs USB and Network access, they're all going to request it, however the differences come when they need specific permissions to accomplish their specific goals.

C. General Outcome

In general, we found that most applications did not request permissions out of malicious intent. While some requested more permissions than required for their core functionality, it was not due for malignant purposes. These were most likely due to misunderstandings or bad practices. There were however a few permissions requests that stood out as worrisome, such as a VPN app requesting a permission to "erase USB storage".

Overall, we were able to find some interesting results. This study has given us better insight into how permissions and applications work on the Google Play Store, as well as what factors play a role in the occurrence of certain permissions. While we did have some challenges when conducting our study, we believe that our results could inspire further research regarding what factors influence permission requests.

VI. Challenges and Limitations

During our project, we faced a couple of challenges and limitations that we believe inhibited the overall results of our study. First, we only gathered data from one app store, which was the Google Play Store. The number of applications we gathered data on is also fairly small compared to related work in the field. While we did gather information on a large number of applications, there are many more applications available on the Google Play Store, and other app stores. Additionally, we did not have an automatic way of deciding what was a large company and what was a small company. We accomplished this by picking out a few companies associated with a specific category and researching them by hand which greatly limits our work.

VII. Future Work

For future work, we would be rectifying all of the challenges/limitations mentioned above. We would gather data from other app stores, most notably the Apple App Store and the Amazon App Store. Both of these app stores have different policies and standards for what applications are allowed to be accessible to users (e.g. preventing applications

with invasive permissions) when compared with the Google Play Store. We would also gather data on more apps from all three app stores in order to gain more broad results, such as how the same applications permissions may vary across each platform. Lastly, we would come up with a way of automatically categorizing companies based on size, so we could compare more companies than we did in this report. We feel like continuing research with these items added would yield interesting and improved results.

VIII. Timeline & Responsibilities

Our team consisted of 2 team members: **Jonathan Bryan** & **Josh Herman**. The timeline for our project, as well as the individual contributions for each team member are as follows:

A. Play Store Scraper & Data Collection

Contributors: Jonathan Bryan & Josh Herman

To properly analyze multiple app stores on a large scale to obtain a grandiose amount of data, we required the assistance of a scraping tool. This was done using a NodeJS library, which was the primary tool used for data scraping. Additionally, we stored all of our data collected on a local MongoDB server. This part of the project was completed on **October 20.**

B. Data Visualization

Contributors: Jonathan Bryan

Data visualization is the task of incorporating visual representations of the data we have collected through charts/graphs, making it easier to present the data. We finished this portion of our project on **November 11.**

C. Analysis

Contributors: Josh Herman

Analysis is a task of combing through the data, and coming to the conclusions that are reasonable considering the data we have collected. This process was completed on **November 22.**

D. Final

Contributors: Jonathan Bryan & Josh Herman

This task includes compiling everything into a final report. This section was notably easier since most of the work was already finished. This task was completed on **December 7**

References

[Lin+14] Jialiu Lin et al. “Modeling users’ mobile app privacy preferences: Restoring usability in a sea of permission settings”. In: *10th Symposium On Usable Privacy and Security ({SOUPS} 2014)*. 2014, pp. 199–212.