RUHR
UNIVERSITÄT
BOCHUM

RUB

RUHR-UNIVERSITÄT BOCHUM

In the Realm of Privacy: A Comparative Analysis of Network Traffic in GrapheneOS and Standard Android Installations

Martin O.

Bachelors's Thesis – November 8, 2023
Chair for Systems Security

1st Supervisor: Dr. Veelasha Moonsamy
2nd Supervisor: Dimitri Mankowski

hgi SYSSEC

Eidesstattliche Erklärung

Ich erkläre, dass ich keine Arbeit in gleicher oder ähnlicher Fassung bereits für eine andere Prüfung an der Ruhr-Universität Bochum oder einer anderen Hochschule zur Erlangung eines akademischen Grades eingereicht habe.

Ich versichere, dass ich diese Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen benutzt habe. Die Stellen, die anderen Quellen dem Wortlaut oder dem Sinn nach entnommen sind, habe ich unter Angabe der Quellen kenntlich gemacht. Dies gilt sinngemäß auch für verwendete Zeichnungen, Skizzen, bildliche Darstellungen und dergleichen.

Ich erkläre mich des Weiteren damit einverstanden, dass die digitale Version dieser Arbeit zwecks Plagiatsprüfung verwendet wird.

Official Declaration

Hereby I declare, that I have not submitted this thesis in this or similar form to any other examination at the Ruhr-Universität Bochum or any other institution or university to obtain an academic degree.

I officially ensure, that this paper has been written solely on my own. I herewith officially ensure, that I have not used any other sources but those stated by me. Any and every parts of the text which constitute quotes in original wording or in its essence have been explicitly referred by me by using official marking and proper quotation. This is also valid for used drafts, pictures and similar formats.

I furthermore agree that the digital version of this thesis will be used to subject the paper to plagiarism examination.

Not this English translation, but only the official version in German is legally binding.

Datum / Date

Unterschrift / Signature

Contents

Abstract

Amid heightened concerns over digital privacy and data security, differentiating operating systems in terms of their user privacy commitment has become crucial. This research elucidates data traffic patterns and privacy settings between Stock Android, GrapheneOS and GrapheneOS-Sandboxed across various operational stages: initial setup, 'Background Stock', and 'Background with Apps'. The objective is to cultivate a profound understanding of the privacy landscape within these environments.

Driven by rising cyber threats and a pronounced shift towards data privacy awareness, this study aims to equip users with in-depth knowledge of the privacy orientations of the aforementioned operating systems. This enables them to base their choices on tangible empirical evidence. As such, the primary challenge tackled is discerning and analyzing the data traffic patterns and privacy components in these environments at different operational junctures. This enables a clear depiction of the potential privacy risks and advantages each offers.

Adopting a systematic approach, the research was compartmentalized into the stated operational stages, allowing for an in-depth analysis of data traffic and privacy nuances in each environment. Key findings indicate that GrapheneOS consistently prioritizes rigorous privacy standards, navigating traffic towards secure servers while minimizing unwanted communications. Its counterpart, GrapheneOS-Sandboxed, showcases a robust infrastructure, effectively protecting user data even with its Google Play services integration. Conversely, Stock Android's environment appears more porous, hinting at areas for potential privacy enhancements.

In conclusion, the results highlight GrapheneOS's dedication to a secure digital realm, with its sandboxed iteration echoing similar sentiments. Stock Android, although functional, requires significant refinements to match the privacy standards of its counterparts. These insights not only serve as a foundation for informed user choices but also as a catalyst for future endeavors aspiring to meld functionality with stringent privacy protections, thus forging a digital sphere that prioritizes both usability and security.

1 Introduction

1.1 Motivation

The rapid global expansion of smartphone usage has brought about a series of challenges concerning privacy and data security. While these devices offer an array of functionalities through a multitude of apps, each one comes with the potential of collecting and transmitting user data, often unbeknownst to the users themselves. This hidden yet continuous collection of data, largely operating in the background, has been a cause of concern, leading to widespread discourse on privacy issues and a growing demand for transparency and control over personal data.

In this vast landscape, operating systems like Android and iOS have attempted to address the problem by providing users with an array of privacy controls. The Android Operating System, due to its open-source nature, allows extensive customizability to device vendors. While this flexibility is a strength, it has led to an opaque and varied landscape of pre-installed software and their data collection behaviors.

Adding to this dynamic field is the recent development of GrapheneOS, an Android-based operating system with a focus on privacy, which has seen a growing interest in recent times. Based on the estimations from official downloads, the userbase of GrapheneOS is around 80,000 [Aks22]. This significant number underscores the importance and relevance of understanding its privacy implications in today's digital landscape.

Previous studies have illuminated concerns related to the security and confidentiality of default Android applications, pointing to intrusive information gathering behaviors and undisclosed avenues into confidential user information [GRR+20]. Research also unveiled tracking mechanisms that function across multiple applications unbeknownst to the user [RNVR+18]. Moreover, there have been discoveries of indirect and concealed pathways that popular applications and external SDKs employ to obtain unauthorized entry to confidential information [RAFW+19]. Such revelations emphasize the pressing

requirement for increased clarity, responsibility, and stringent privacy guidelines in the Android environment.

This thesis proposes an examination into this world by independently observing network traffic, specifically targeting three unique Android installations: (i) GrapheneOS without any integrated Google services, (ii) GrapheneOS with the isolated installation of the Play Store, and (iii) a standard Google Pixel 6a "stock" installation with its entire suite of Google services. The central premise is to unveil the type, extent, and possible sensitivity of data being transmitted under the hood by these installations, in both idle and active states, thereby giving the user a more transparent view of what their smartphone may be revealing to the world.

1.2 Goals and Contribution

Building upon the existing motivation, the primary goal of this thesis is to conduct a detailed and exhaustive analysis of network traffic across three distinct Android installations. The study leverages a methodically crafted setup involving Android and GrapheneOS phones, funneling the traffic through a Pfsense Firewall integrated with pfELK [Wil23], which has been implemented to facilitate the robust parsing of network and DNS logs, enhancing the ability to scrutinize network packets effectively.

To extend this process, the devices will also be set up for data monitoring and capturing of network packets, utilizing the PCAPdroid [Far23] application alongside the Pfsense firewall and pfELK environment. This strategic setup, with Pfsense operating as a VM and pfELK functioning through a docker-compose container environment, will allow for a seamless yet detailed logging and analysis of the traffic data.

PCAPdroid will work hand in hand with the pfELK environment to collect extensive data on the initiating apps and the servers being communicated with. The data captured here will provide a transparent view into the data transmission mechanisms at play, thereby uncovering potential privacy breaches and assessing the scope of data shared without user consent.

Upon the completion of the initial data harvesting phase, the data will be sorted and scrutinized using tools such as Python to categorize the traffic based on its source and to identify potential inclusion of Personal Identifiable Information (PII). The traffic will be analyzed at different stages: initial setup, during the use of default apps, and while operating popular apps downloaded from the Play Store, aiming to draw patterns or trends in the data transmissions and discern differences across the Android installations studied.

The conclusions drawn from this multifaceted analysis could shed new light on the discourse surrounding data privacy and transparency, providing a data-driven foundation for potential improvements in mobile operating system environments.

1.3 Structure

In the pursuit of fulfilling the stated goals and contributing to the pertinent discourse on data privacy and transparency, this thesis is structured as follows:

1. **Technical Background:** This section presents a comprehensive overview of Android, GrapheneOS, and relevant tools like PCAPdroid, Firewall configurations, and more. It also explains critical concepts, such as encryption and rooting, setting the stage for subsequent analysis.
2. **Related Work:** This chapter reviews prior studies and insights on Android system circumvention, the mobile tracking ecosystem, and analysis of preinstalled Android software.
3. **Design:** Here, the framework for network traffic analysis is introduced, along with the criteria used to categorize different traffic segments.
4. **Implementation:** This segment delves into the hands-on aspects of the research, detailing setups on Android and GrapheneOS, rooting processes, and methods used for monitoring network traffic. Additionally, it covers the installation and usage of top apps, and background noise monitoring.
5. **Evaluation:** This chapter provides a deep dive into the results, analyzing different facets of the captured data—from temporal and geographical perspectives to the volume and security of the traffic. The analysis also dissects specific app traffic patterns and categories.
6. **Discussion:** An examination of the results is provided, discussing the temporal, geographical, and traffic volume analyses. This section also addresses potential threats to validity, limitations, and any contradictory points observed during the research.
7. **Conclusion:** The thesis concludes by summarizing key findings from the implementation and evaluation phases. It recaps the insights from the initial setup, background stock, and apps analysis. The chapter ends by looking ahead, suggesting potential avenues for future research and study.

2 Technical Background

In the quest to compare GrapheneOS and Stock Android, this thesis examines the technical intricacies of both operating systems. We begin with an overview of Android and GrapheneOS in Section 2.1. Section 2.2 delves into Network Traffic Analysis, breaking down network protocols and packet data. PCAPdroid's role in network monitoring is introduced in Section 2.3, followed by the firewall and pfELK environment's relevance in Section 2.4. We explore tools like VirusTotal, MaxMind, and DuckDuckGo's Tracker Radar in Sections 2.5 to 2.7 for tagging network destinations and geolocation insights. The importance of encryption and its challenges are elaborated in Section 2.8. The chapter concludes with Section 2.9, emphasizing the role of rooting, particularly using Magisk, in the broader comparison between GrapheneOS and Stock Android.

2.1 Android and GrapheneOS Overview

To appreciate the gravity of the issues at hand and to comprehend the underlying framework of the analyses undertaken in this thesis, it is imperative to grasp the subtleties of the Android and GrapheneOS platforms.

Android OS

Android OS, developed by a consortium of developers known as the Open Handset Alliance and backed by Google, is the world's most popular mobile operating system. Android is known for its open-source nature, which allows for high levels of customization and has facilitated a rich ecosystem of applications and services.

However, this open-source nature, while fostering innovation and flexibility, has also opened up avenues for potential privacy breaches through varied and opaque landscapes of pre-installed software, each having different data collection behaviors.

It is in this context that Google has implemented a series of privacy controls, including permissions, to help users maintain a level of control over their personal data.

GrapheneOS

GrapheneOS, on the other hand, leverages the open-source nature of Android to build a privacy-centric mobile operating system. Focused intensely on user privacy and security, GrapheneOS operates without any Google services integrated by default, thereby aiming to reduce the amount of data collected and shared without user consent.

One of the defining features of GrapheneOS is its hardening of the Android OS to enhance security features significantly. It involves making alterations to the operating system's source code to remove potential vulnerabilities and reduce the attack surface that adversaries might exploit.

Moreover, GrapheneOS facilitates fine-grained permission controls, allowing users to have a more detailed determination of the kind of access each application has. This approach aids in substantially reducing the risk of unwanted data access and transmission, fostering a more secure and private user experience.

Comparative Overview

Despite both Android OS and GrapheneOS originating from the Android Open Source Project (AOSP), they exhibit distinct variations in how they address user protection and confidentiality — a comprehensive comparison of their security and privacy features is presented in Table 2.1.

These distinctions are illustrative of the broader differences in philosophy and implementation between Android OS and GrapheneOS. A deeper exploration of these differences, as well as an examination of the resultant traffic patterns and underlying behaviors, will be covered in the succeeding chapters.

2.2 Network Traffic Analysis

2.2.1 Importance of Network Traffic Analysis

Analyzing network flows goes beyond just technical evaluation and becomes a vital instrument in comprehending and improving data confidentiality in this digital era. It offers a microscopic view of data transmission, allowing for a detailed analysis of

Table 2.1: Feature-wise Comparison between Android OS and GrapheneOS

Feature	**Android OS**	**GrapheneOS**
Default Encryption	Supports full-disk and file-based encryption but enforcement depends on device manufacturers.	Strictly enforces encryption policies with user data encrypted by default.
App Permissions	Enhanced permissions since Android 6.0 but potential for rogue app access remains.	Tighter permissions model with granular control for users.
Software Updates	Depends on device manufacturer; can result in delays.	Consistent and regular update schedule.
Sandboxing and Isolation	Utilizes sandboxing to separate app processes.	Amplifies sandboxing with hardened memory allocation and additional layers.
Network and Connectivity	Provides VPN and private DNS options.	Stricter firewall rules and sophisticated routing options.
Auditing and Open Source	Open-source but includes proprietary software.	Completely open-source with transparency for all components.

the kind and amount of data being shared, both voluntarily and involuntarily, by the user.

Furthermore, network traffic analysis can unveil the underlying behaviors of applications, helping in identifying potential security vulnerabilities and ensuring the user's privacy is not compromised. Through a meticulous analysis of network traffic, it becomes possible to discern the layers of communications, distinguishing between necessary data transmissions and potential privacy infringements.

In conclusion, network traffic analysis forms a cornerstone in the pursuit of data privacy and security, enabling a deeper understanding of the inner workings of data transmission and fostering a secure and privacy-centric digital environment. As we delve deeper in subsequent chapters, we leverage the principles of network traffic analysis to dissect the traffic patterns in Android and GrapheneOS, laying a strong foundation for our analysis in pursuit of enhanced privacy and security.

Network traffic analysis stands as a critical methodology in unraveling the intricacies

of data transmission and privacy, particularly in the realm of smartphone usage. This subsection sheds light on the rudimentaries of network protocols and packet data, illustrating their pivotal roles in ensuring seamless communication between devices over a network while accentuating their significance in understanding and enhancing privacy.

2.2.2 Network Protocols

Network protocols are essentially a set of rules that dictate how data packets should be placed on the network. These rules ensure that devices can communicate efficiently, establishing a universal language that all networked devices adhere to. Two prevalent protocols include the Transmission Control Protocol and the Internet Protocol. Together, they are often termed the TCP/IP stack and serve as the foundation for contemporary online communications

Understanding network protocols is fundamental in network traffic analysis as it provides the necessary lens to dissect the communications happening over the network, giving insights into the standard operations and potentially unveiling anomalies that could be indicative of privacy breaches or security issues.

2.2.3 Packet Data

Packet data refers to the units of data transmitted over a network. The illustration below provides a visual representation of a packet's structure:

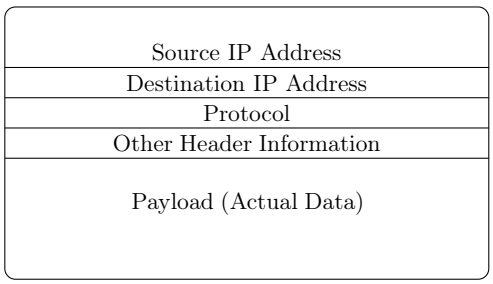

Figure 1: Schematic representation of a typical network packet structure, illustrating the header with various fields and the payload section.

The header contains detailed information about the packet, such as its origin (Source IP Address), its intended recipient (Destination IP Address), the communication protocol being used, among other pieces of metadata. The payload, on the other hand, carries the actual data being transmitted.

Studying packet data is pivotal to network traffic analysis, offering a detailed view of data transmission processes. It facilitates pattern identification in data transmission, comprehension of transmitted data nature, and the discovery of potential unauthorized data transmissions, revealing privacy infringements.

2.3 PCAPdroid and Network Monitoring

In the labyrinth of network communications, a tool that can methodically track, analyze, and potentially block connections forged by various applications is a linchpin in understanding and enhancing user privacy. PCAPdroid [Far23] emerges as a forerunner in this space, offering a rich palette of features that facilitate meticulous network monitoring and packet capturing on Android devices. This subsection delineates the functionalities and features of PCAPdroid, illuminating its instrumental role in the research undertaken in this thesis.

2.3.1 Functionality and Features

PCAPdroid is a privacy-centric, open-source application available for Android devices. It stands as a sentinel that enables users to track, analyze, and block the network connections initiated by other applications on their device. One of its hallmark features is the ability to export PCAP dumps of the traffic, which can be a rich source for metadata extraction and detailed analysis of network communications.

An equally significant functionality is its capacity to export data in CSV format, detailing the connections made by each application on the device. This feature facilitates a structured and tabulated representation of the data, making it easier to scrutinize individual connections in detail, and thereby unravel the specifics of network traffic emanating from different apps.

Its operation hinges on the simulation of a VPN, a strategy that allows it to capture network traffic without necessitating root access to the device. Importantly, it does not rely on a remote VPN server, as all data processing is undertaken locally on the device, ensuring that user privacy is not compromised.

2.3.2 Privacy-Friendly and Open Source

Being open-source not only stands testament to its commitment to privacy but also opens up avenues for community-driven improvements and transparency in its functionalities. Users and developers alike can delve into its source code, allowing for a transparent understanding of its operations and ensuring that it adheres to the privacy standards it professes.

2.3.3 Role in Network Monitoring

In the context of network monitoring, PCAPdroid stands as a formidable tool, offering insights into the connections established by various applications. The data extracted through this application can provide a deep dive into the network traffic, helping identify patterns of data transmission, and unveiling potential areas where user privacy might be at risk.

Moreover, by facilitating the blocking of certain connections, it empowers users to take control of the data transmission, providing an avenue to enhance privacy proactively. Its ability to export PCAP dumps further aids in the detailed analysis of network traffic, providing a robust foundation for the network traffic analysis undertaken in this thesis.

2.4 Firewall and pfELK Environment

In the realm of network security and analysis, leveraging advanced tools such as the pfSense® firewall and the pfELK [Wil23] environment plays a pivotal role. This subsection discusses these two components, shedding light on their functionalities and how they offer an in-depth analysis of network traffic and data packet inspection.

2.4.1 Pfsense Firewall

Netgate® pfSense® stands as a robust firewall solution, grounded on the FreeBSD operating system and tailored to facilitate advanced network management and security configurations. It embodies a rich set of features, including VPN, DHCP, and DNS services, aiding in crafting a secure and efficiently managed network environment. The firewall becomes a linchpin in this study, serving as a tool to steer network traffic, scrutinizing each data packet that traverses through it, and establishing a stronghold that safeguards against unauthorized access.

2.4.2 pfELK Environment

The pfELK environment, meanwhile, serves as a potent complement to the pfSense® firewall, aiming to replace the standard pfSense® web UI with extended search and visualization features. It is an open-source project, housing a series of functionalities that aid in the detailed analysis of network packets.

At its core, pfELK utilizes the might of Elasticsearch for near-real-time search and analysis of indexed data. Leveraging Logstash, it can ingest and enrich firewall

traffic logs from pfSense/OPNsense setups, thereby furnishing a rich ground for data analysis.

A notable merit of pfELK is its capacity to visualize network traffic through interactive dashboards, maps, and graphs generated in Kibana, granting a more intuitive understanding of the data flow and traffic patterns. It supports a range of entries, including different protocols (TCP, UDP, ICMP), and DHCP message types, both in IPv4 and IPv6.

Further, it encompasses capabilities to parse openVPN logs, adhere to Kibana SIEM compliance, and facilitate CARP data from pfSense®, thus offering a holistic approach to network traffic analysis. Deployment of this environment can be undertaken through various avenues, including docker-compose and bash script, offering flexibility in its implementation.

2.4.3 Role in the Thesis

In the framework of this thesis, the pfSense® firewall and the pfELK environment stand as the twin pillars supporting the intricate analysis of network traffic during the initial setup stage of the research. While pfSense® directs and scrutinizes the traffic coming directly from the phone's Access Point, pfELK offers a rich platform to visualize and analyze the data in a more user-friendly and interactive manner, bringing critical insights to the fore and laying a strong foundation for a detailed investigation into network packet behaviors.

2.5 VirusTotal

2.5.1 Background and Functionality

VirusTotal [Vir23] is a subsidiary of Google that operates an online service that analyzes files and URLs for viruses, worms, trojans, and a variety of other malicious content. Its utility in our thesis derives from its extensive database, which enabled a detailed analysis of various traffic segments, assisting in categorizing and differentiating between them, thus aiding in the deeper understanding of the data patterns observed during the analysis.

2.5.2 Role in the Thesis

In our research, VirusTotal was instrumental in classifying various traffic sections according to the nature of the transmitted information and its origin. This facilitated a nuanced and organized examination, offering a core framework to distinguish and classify distinct patterns and possible risks within the traffic data.

2.6 MaxMind

2.6.1 Background and Functionality

MaxMind [Inc23b] is renowned for its GeoIP databases that help in identifying the geographical location details associated with IP addresses. It offers detailed insights, albeit with a noted possibility of occasional inaccuracies. Understanding that the geo-IP information derived can sometimes not be accurate is essential to ensure discerning interpretation of the results.

2.6.2 Role in the Thesis

For this thesis, MaxMind was utilized to enhance the geolocation data of different IP addresses encountered during the traffic analysis. The service aided in appending geographical data to IP addresses in the dataset, thus providing a more rounded view of the traffic patterns, while keeping in mind the potential for inaccuracies.

2.7 DuckDuckGo Tracker Radar

2.7.1 Background and Functionality

The DuckDuckGo Tracker Radar [Inc23a] is a dataset comprising information on the most prevalent third-party domains on the web. It details vital statistics about these domains such as their behavior, classification, and ownership, acting as a substantial asset in web traffic analysis. The dataset maintains a substantial metadata repository for each domain, encompassing aspects such as prevalence, fingerprinting tendencies, cookie usage, privacy policies, and performance metrics.

2.7.2 Role in the Thesis

In this thesis, the DuckDuckGo Tracker Radar was employed to aid in the categorization of different domains encountered in the web traffic data. Its vast dataset was utilized to understand and categorize various domains based on their attributes and prevalence, enhancing the depth of analysis by providing enriched data and facilitating a more detailed interpretation of the web landscape.

2.8 Encryption and Data Privacy

As technology advances, so does the sophistication of methods used to ensure data privacy and security, with encryption standing as a cornerstone in securing data transmission. This subsection takes a deep dive into the role of encryption in network traffic, discussing the associated challenges and methodologies that can be utilized for decrypting specific traffic segments to garner valuable data insights.

2.8.1 The Role of Encryption

Encryption plays a pivotal role in securing data, where information is transformed into a code to prevent unauthorized access. In the modern digital landscape, encryption is almost ubiquitous, being utilized widely to secure communications over networks, thereby safeguarding the data from potential eavesdroppers and attackers.

In the context of network traffic, encryption works to mask the data being transmitted, ensuring that sensitive information, including personal details, remain confidential and intact. While encryption substantially elevates the security posture of data transmissions, it also brings about a veil of opacity, which, while protecting user privacy, can potentially shield malicious activities and data exfiltration attempts under its gambit.

2.8.2 Challenges Presented by Encryption

Despite its crucial role in securing data, encryption presents a series of challenges, especially when it comes to analyzing network traffic for research purposes. One of the main hurdles is the difficulty in scrutinizing encrypted traffic to understand the underlying data being transmitted, as it requires sophisticated tools and techniques to decrypt the data accurately without the proper keys.

Further, there stands a moral and ethical dilemma regarding the decryption of personal data, navigating which demands a careful and considered approach, emphasizing user privacy and adhering to legal frameworks.

2.9 Rooting and Magisk

Within the Android ecosystem, "rooting" is the act of securing privileged oversight (referred to as root access) of several Android subsystems. The motivation behind rooting is primarily to bypass constraints set by both carriers and device manufacturers, thereby providing root access to the underlying Android OS code. This can be likened

to operating software as an administrator in Windows or utilizing a command with sudo in Linux.

2.9.1 Magisk: A Rooting Solution

Magisk is a popular tool for rooting Android devices. It is a suite consisting of an open-source software for root access and an Android app for management. Here are the features and functionalities that stand out:

- **Open Source:** Being open-source, it allows for community contributions and scrutiny, promoting transparency and trust in its operations.
- **Systemless Root:** Unlike other rooting solutions that modify the system partition, Magisk does not alter the system partition, thus helping maintain the integrity of the system.
- **SafetyNet Bypass:** Magisk can hide the root from apps that block rooted devices, including banking and financial apps, thereby bypassing Google's SafetyNet.
- **Modules Support:** Magisk supports modules that allow users to add or modify features on their rooted devices, giving them the flexibility to customize their device to a great extent.
- **Root Management App:** The Magisk Manager app facilitates the management of root permissions, enabling users to grant or deny root permissions for individual apps.

2.9.2 Rooting with Magisk in this Research

In this research, rooting played a pivotal role, as it facilitated the detailed monitoring and logging of network traffic via the installation and setup of PCAPdroid, which required root access to function optimally. The following steps marked the rooting process:

1. Enabling Developer Options and OEM Unlocking on the device.
2. Connecting the device to a system to run ADB (Android Debug Bridge) and Fastboot commands.
3. Unlocking the bootloader to allow for system modifications.
4. Flashing the Magisk patched boot image to the device.
5. Installing the Magisk app and completing the setup, thereby achieving root access.

The successful rooting of the device with Magisk opened the door to a world of possibilities, including the granular network monitoring and more decryption capabilities that formed the bedrock of this research project.

2.9.3 Conclusion

In the grand schema of data privacy, encryption emerges as a double-edged sword; while it protects user data, it also obscures the data flow, posing challenges for research endeavors like this thesis. Navigating this landscape necessitates a balanced approach, one that respects user privacy while employing sophisticated methodologies to decrypt selected traffic and garner valuable data insights, contributing constructively to the discourse on data privacy and network security in the Android ecosystem.

3 Related Work

Over the years, there have been extensive studies related to smartphone platforms, user privacy, third-party services, mobile ecosystems, and the Android OS. Here, we discuss a few notable ones that have relevance to our current research:

3.1 Circumvention of the Android Permissions System

Alepis and Patsakis conducted a comprehensive analysis on the evolution and implications of the Android permission system, particularly targeting the "Runtime Permissions" introduced in recent Android versions [AP18]. They highlighted how mobile computing, with its intimate intertwining into daily life, processes vast amounts of often private data. To safeguard this, Android has iteratively refined its permission mechanism. Their study spotlighted several significant vulnerabilities in the Android versions prior to Marshmallow, evidenced by numerous attacks on core libraries. With the introduction of the "Runtime Permission" model, a host of new security challenges were uncovered. One of the paramount issues emphasized was the unrestricted internet access granted to apps, a loophole that if addressed, might pivotally alter the Android app landscape. Furthermore, Alepis and Patsakis argue for an enhanced permission framework. They suggest the necessity for the OS itself to take a more active role, not just in facilitating but in enforcing stringent permission checks. Their work underlines the need for more detailed user notifications, advanced security settings, and a shift in the paradigm where permissions, especially the "dangerous" ones, are managed more proactively by the operating system itself.

Reardon et al. conducted an extensive study on modern smartphone platforms, especially focusing on the Android permission-based model [RAFW+19]. They investigated the methods apps utilize to circumvent permissions and retrieve confidential information without the knowledge of users. Through extensive testing of a vast array of apps in a monitored setting, they identified many instances where well-known applications and external SDKs leveraged hidden pathways to access information like unique IDs and location details. This study provided significant insights into

the challenges posed by the Android permission model and the risks associated with covert channels.

3.2 Mobile Tracking Ecosystem

Razaghpanah et al. delved into the realm of third-party services, a crucial component of the mobile ecosystem, and their impact on user privacy [RNVR+18]. They highlighted how such services, especially those related to advertising and tracking, remain largely invisible to users due to the opacity of mobile systems. By leveraging real-world mobile traffic data, they identified thousands of third-party advertising and tracking services. They also analyzed the privacy policies of major advertising and tracking service providers, discovering that data sharing with subsidiaries and third-party affiliates is a standard practice. This study emphasized the importance of transparency and user awareness in the mobile ecosystem.

Nguyen, Backes, and Stock explored the implementation of GDPR-compliant consent notices within Android apps [NBS22]. Since the GDPR's 2018 introduction, user data processing consent must be transparent and specific. Previous research has pointed to violations via network traffic, but a systematic examination of mobile apps' consent notices was absent. In their expansive study involving 239,381 Android apps, the authors identified prevalent mechanisms of user interface interactions in 13,082 apps. Worryingly, 30,160 apps didn't attempt implementing consent notices for third-party data sharing, a blatant GDPR oversight. Furthermore, 2,688 of the apps, despite having consent notices, violated GDPR mandates by either misleading users or persistently transmitting data against user preferences. The study, while illuminating widespread issues, also highlighted the importance of aiding developers in understanding and adhering to GDPR, ensuring informed user choices regarding data usage.

3.3 Pre-installed Android Software Analysis

Gamba and colleagues examined the open-source characteristics of Android and the software that comes pre-loaded on devices from multiple producers [GRR+20]. They pointed out the possible risks and confidentiality concerns with such embedded applications and the opacity present within Android's distribution network. Their investigations uncovered connections among entities like device makers, cellular service providers, and external entities. They noticed these connections frequently focused on ad-based and data-centric operations. Their findings stressed the importance of heightened clarity, recognition, and responsibility within the Android world.

Sutter and Tellenbach presented 'FirmwareDroid', a dedicated open-source security tool designed for the scrutiny of Android firmware [ST23]. This framework streamlines the process of deriving and conducting static analysis on software preloaded in Android firmware. Through their comprehensive assessment of a vast number of Android firmware samples, they delved into a significant number of distinct pre-installed Android apps. Their findings indicated that a noteworthy percentage of these apps contained advertising trackers, and a vast majority of the permissions utilized were signature-based. Delving deeper, it was observed that Google Android 10 firmware had a certain percentage of risky permissions, which saw a reduction in Android 11.

Meanwhile, GrapheneOS, focused on privacy, registered the highest count of normal permissions (59.54%), whereas it had zero advertising tracker libraries. LineageOS, another privacy-centric OS, also showed a negligible percentage of such libraries. Furthermore, data on advertising tracker libraries by Exodus revealed that the majority were found in Google firmware. The most prevalent trackers were Google Firebase Analytics, Google Analytics, and Google AdMob. Some peculiar findings included the sharp decline of Amazon Analytics in Android versions after 8 and the notably low detection of Facebook Analytics in the dataset. Despite the presence of trackers in major Android distributions, both GrapheneOS and LineageOS remained at the lower end, reinforcing their commitment to user privacy.

In conclusion, while tools like Exodus play a pivotal role in data gathering for such analyses, FirmwareDroid sets a new benchmark as a comparative framework for various firmware. This study underscores the necessity for such tools in the fight against vulnerabilities in pre-installed software and the enhancement of mobile device security.

4 Design

4.1 Network Traffic Analysis Framework

The foundation of this research is based on a meticulously-designed framework that delves into network traffic stemming from three unique configurations of the Android operating system. Each configuration, or "flavor", offers a varied landscape in terms of privacy features and service integrations:

- **Stock Android:** This is the default version of Android as provided by Google. It comes with deeply-integrated Google services, including the Play Store, Google Maps, and others. These services frequently communicate with their servers, potentially sending personal and usage data.
- **GrapheneOS:** An open-source variant of Android, GrapheneOS is specifically tailored to enhance user privacy. Stripped of many of the Google services found in Stock Android, it is expected to have fewer outbound communications, particularly those that may infringe on user privacy.
- **GrapheneOS with Google Sandbox:** This configuration provides a middle ground. While it is based on the privacy-focused GrapheneOS, it incorporates a sandboxed environment that allows Google services to run in a controlled manner. This ensures that user data interactions with Google services are contained and monitored, thereby potentially minimizing unwarranted data transmissions.

To ensure a comprehensive analysis of each OS flavor, traffic was captured in several distinct operational states:

- **Initial Setup:** Captures the network traffic when the device is first initialized and set up. This state is crucial, as many apps and services perform their initial sync, updates, and data transmissions during this phase.

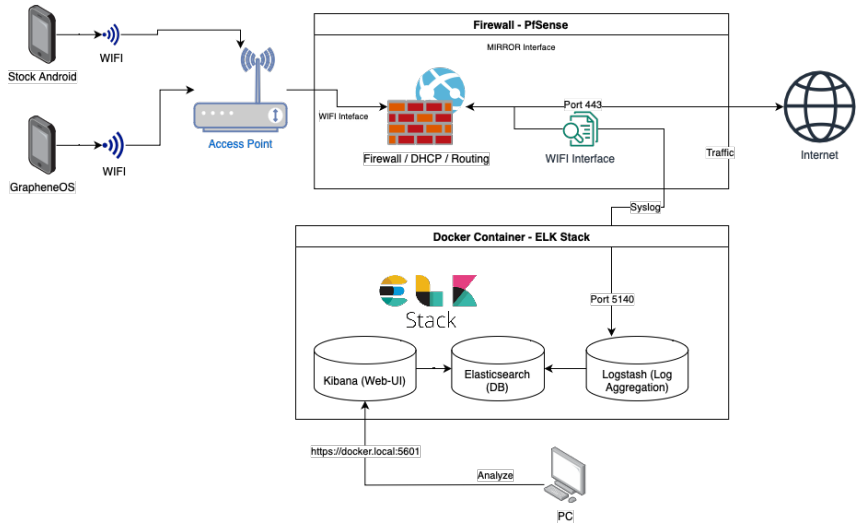

Figure 4.1: Lab setup for capturing and analyzing network traffic during initial setup.

- **Background without Extra Apps:** This state represents the device's idle traffic when no third-party apps are installed. It provides a baseline of how much network activity occurs in the background of a "clean" installation.

- **Background with Top 10 Apps:** In this state, traffic from the top 10 most popular apps at the time of research was monitored. This is representative of a typical user's device and gives insights into how mainstream apps contribute to network traffic and potential privacy concerns.

This multi-faceted approach, examining both the different flavors of the OS and the various operational states, forms the backbone of our network traffic analysis, ensuring a robust and comprehensive exploration of Android's privacy landscape.

Data Capturing Setup

For the Initial Setup phase, the inherent complexities rendered the direct use of PCAPdroid infeasible. Instead, a specialized lab environment was utilized. When the phones connected to the lab Wi-Fi, a firewall redirected their traffic to 'pfELK', a variant of the ELK stack optimized for handling pfsense log data.

For subsequent stages, PCAPdroid became the primary tool for monitoring and capturing packets, detailing the intricate web of data traffic patterns and associated privacy concerns across the different OS configurations.

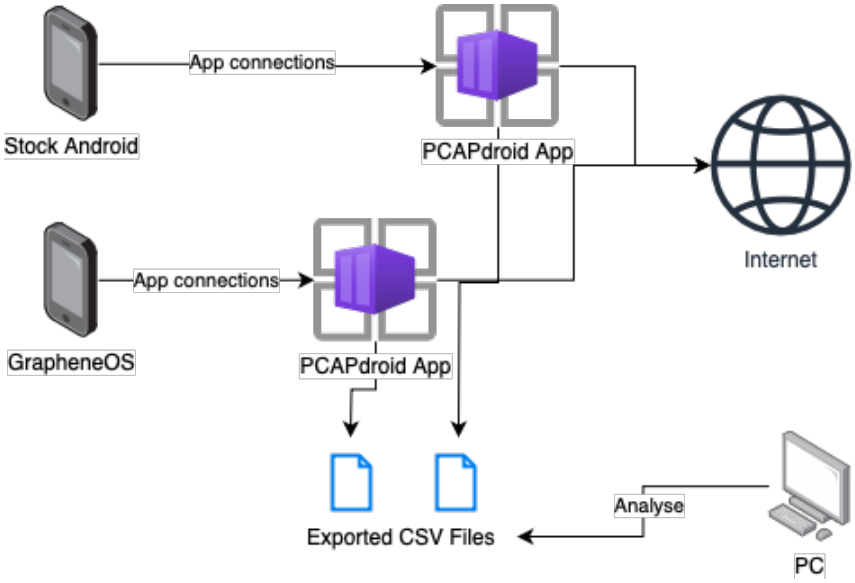

Figure 4.2: Lab setup for capturing and analyzing network traffic after initial setup.

4.2 Criteria for Categorizing Traffic Segments

In the vast landscape of network traffic, achieving precision and clarity necessitated the formulation of specific criteria to segment and categorize the traffic, ensuring a structured and informed analysis. Our segmentation methodology revolved around three pivotal criteria:

- **Originating Apps of the Traffic:** Each network request or connection invariably has a source — an application that initiates it. By distinguishing between these originating apps, it was possible to discern patterns, anomalies, or trends associated with specific applications.
- **Type and Nature of Transmitted Data:** Beyond the origin, the content and type of data being transmitted provide valuable insights. Whether it's a text-based request, media transfer, or encrypted payload, understanding the nature of transmitted data is imperative.
- **Database Reference Markings:** Through the integration of third-party references from VirusTotal and DuckDuckGo's Tracker Radar, the traffic data was enriched, tagging network requests that aligned with known tracking or advertising signatures. This enabled the prompt identification of traffic elements with potential privacy implications.

Following initial segmentation, traffic was meticulously mapped against reference databases. In this process, segments that potentially transmitted sensitive data,

especially those relaying Personally Identifiable Information (PII), were isolated (see next section).

4.3 Data Processing and Analysis

Dissecting and understanding network traffic required several steps of data refinement and analysis. Starting with raw traffic captures, a series of transformations was executed to prepare the data for a comprehensive exploration.

- **CSV Data Compilation:** Initial data was exported into CSV files, with each entry detailing pertinent connection attributes — ranging from byte counts and requested domains to source and destination IPs.
- **Enrichment with Third-party Tags:** Through Python scripts, tags from VirusTotal and DuckDuckGo were integrated for each domain request. This step enriched the dataset, embedding it with added context and significance.
- **GeoIP Data Integration:** To provide insights into the geographical landscape of traffic, GeoIP data from MaxMind was incorporated. This addition not only identified the physical location of servers and services but also alluded to potential regional data privacy concerns.
- **Traffic Categorization:** By leveraging the enriched dataset, traffic was systematically categorized into one of three primary categories: Ad-related, Tracking-related, or Other. This classification facilitated a focused analysis on specific traffic types, enabling targeted insights and observations.
- **In-depth Analysis:** Beyond mere categorization, the data was analyzed rigorously, examining metrics such as byte counts, target IPs, and connection durations. The goal was to uncover subtle patterns and insights, enhancing the understanding of privacy dynamics across different OS configurations.

Once processed and refined, the data presented a comprehensive view, revealing insights into the privacy landscape across the three OS environments.

5 Implementation

5.1 Account Setup

The research started with the setup of a distinct Google account created to facilitate the project. The account details are as follows:

- Email: research.pixel.ad0a2x@gmail.com
- Birthday: 01/01/2000
- Gender: Male
- Phone Number: +49 160 4164728

5.2 Stock Android Setup on Pixel 6a

The next step involved the initial configuration of a Pixel 6a phone running stock Android, without installing the top-10 apps initially. The language was set to English (US), and the phone was connected to a specially configured WiFi network. This network was facilitated by the pfSense firewall and monitored using the pfELK setup, allowing the research to capture all data traffic directly, thereby eliminating the need for a VPN.

Key moments in this phase were:

- 15th August, 14:36 - Setup reached the home screen, marking the completion of the initial setup.
- 15th August, 14:47 - Beginning of app usage with apps such as "Messages" and "Google Chrome".
- 15th August, 16:13 - Completed the download of an Android update.

- 15th August, 16:55 - Initiated automatic updates, including language pack and security updates.
- 15th August, 17:36 - Device restarted to install the Android 13 update.

5.3 Rooting with Magisk and PCAPdroid Installation

Following the initial setup, the device was rooted using Magisk. This process began with connecting the device to a Mac and involved several steps including enabling developer options and unlocking the OEM.

Noteworthy steps during this process included:

- 18:18 - Opening of Google "Files" app and initiation of Magisk installation.
- 18:37 - Accessing the Google Play Store.
- 19:19 - Approval and installation of a security update followed by a system restart.
- 19:45 - Start of a new setup post-rooting, with settings mirroring those in the initial setup.
- 20:01 - PCAPdroid APK was transferred to the phone and installed.
- 20:51 - Finalizing the setup process, including logging into the Google account and configuring security setups such as fingerprint and voice recognition.

5.4 Network Traffic Monitoring and Logging

After rooting the device and installing PCAPdroid, network traffic monitoring and logging were initiated. Initially, PCAPdroid was used without TLS inspection but was later configured with MITM (Man-in-The-Middle) functionalities, facilitating a more detailed network packet analysis.

The key occurrences in this phase were:

- 20:01 to 20:09 - Configuration of PCAPdroid, including granting it superuser rights via Magisk.
- 22:13 - Initiation of normal usage observation utilizing Applog through PCAPdroid.
- 23:43 - The device was shut down for the night, marking the end of the day's monitoring session.

This chapter delineates each step executed in the setup, rooting, and monitoring process, setting the stage for the data collection phase of the research.

5.4.1 Installation and initial usage of ‘Top 10‘ apps

In this section, we detail the installation process and initial setup of the top ten apps on the stock Android Pixel device. The operation was carried out on August 26, using the IP address 10.34.0.105.

5.4.2 Tools and Setup

- **Device:** Android Pixel (Stock version)
- **IP Address:** 10.34.0.105
- **Network Monitoring Tool:** PCAPdroid
- **Other Tools:** Google Play Store for app installation, Google account for sign-ins

5.4.3 App Installation and Setup

Following the opening of the Google Play Store at 11:49, the "top 10" apps were installed and setup with various configurations and account details. Here we present a chronological breakdown of each app's installation and initial setup process:

1. **Temu:**
    - Installation began at 11:51
    - Privacy policy accepted at 12:09
    - App utilized briefly for testing
2. **SHEIN:**
    - Installed at 11:53
    - Initial attempt to use TLS decryption faced handshake exception
    - Successfully signed in with Google account at 12:15, accepting cookies thereafter
3. **WhatsApp:**
    - Installation completed by 11:54
    - Setup began at 12:18 with language selection and agreement to terms
    - Encountered connection issues initially, resolved by disabling TLS decryption
    - Phone number added and verified, followed by the configuration of various permissions and profile settings

- Message exchanges carried out for testing, including media sending which faced issues while TLS decryption was enabled

4. **Instagram:**
    - Installed at 11:54
    - Account created with username "Research.Pixel" and the specified password at 12:34
    - Various personal details and preferences configured during setup
    - Encountered connectivity issues with TLS decryption enabled
5. **TikTok:**
    - Installed at 11:55
    - Account setup with birthday and nickname configurations
    - Experienced connectivity issues during TLS decrypted sessions
    - Browsing test conducted with personalized ads enabled
6. **Telegram:**
    - Installed at 11:56
    - Initial setup required various permissions, including phone call management and contact access
    - Account configured with first name "Research.Pixel"
    - Test chat initiated from a private phone, demonstrating successful messaging and media exchange
7. **Snapchat:**
    - Installed at 11:57
    - Account setup using Google, with various permissions configured including contact access and notifications
    - Birthday and username configured during setup
    - Test messages and snaps exchanged, confirming functional messaging
8. **CapCut - Video Editor:**
    - Installed at 11:57
    - New project initiated with access to music and photos granted
    - Encountered connectivity issues during TLS decryption
    - Successfully shared a test video to TikTok

9. **Live Weather: Radar & Forecast:**
    - Installed at 12:00
    - Notifications enabled, and precise location access granted for personalized weather updates
    - Interacted with various features including radar information, encountering ads in the process
10. **Google Translate:**
    - Updated at 11:59
    - Experienced connection error with TLS decryption enabled
    - Successfully translated a word without TLS decryption

5.4.4 Background Noise Monitoring

To acquire a realistic understanding of the data flow and connections established during regular phone usage, a strategy was implemented to capture the background "noise" — the untargeted, spontaneous data transmissions that occur as the various apps operate in the background. This entails monitoring the general behavior of the apps without focusing on a specific one, thereby garnering a wealth of data that reflects the daily operations of a typical smartphone user.

Here is a chronologically organized account of the operations performed and the settings configured to facilitate this monitoring:

- **15:27 -** Preserved the details of existing connections by saving the connections CSV file.
- **15:38 -** Initiated a generic capture session to record the background activities of the different apps in a natural usage scenario, starting with re-opening all previously installed apps one by one to document their baseline behaviors.
- **15:39 -** The investigation was extended to all pre-installed Google applications. Each app was opened individually, followed by accepting all pending agreements and completing any required sign-in processes.
- **15:41 -** Opened the Google Photos application and activated the backup feature. Subsequently, the permission to organize photos based on facial recognition was granted, contributing to the realistic usage scenario curated for this session.
- **15:42 -** Permitted the application to access the location of the camera and proceeded to take a photograph, observing the permissions requested and granted during this standard use case.

The objective of this monitoring phase was to understand the myriad background communications and data transfers that take place during the normal course of phone usage, effectively "tuning into" the background noise created by these simultaneous operations. Analysis of this data, to be presented in subsequent chapters, aims to unravel the complex web of communications, potentially unveiling patterns and security aspects intrinsic to daily smartphone operations.

5.5 GrapheneOS Setup and Background Monitoring

The setup for the GrapheneOS focused on configuring the environment meticulously to enable a detailed monitoring of background activities during normal phone usage. Here is a chronological breakdown of the setup and monitoring steps undertaken:

Initial Setup

- **Mon 28.08**
  - **22:12** - Set the language to English (US).
  - **22:13** - Configured the timezone to Amsterdam.
  - **22:14** - Enabled WiFi and connected to the lab network; IP assigned: 10.34.0.109.
  - **22:16/17** - Installed SIM card transferred from the stock Android Pixel device.
  - **22:18** - Selected the SIM for data usage while opting to keep the cellular data turned off.
  - **22:19** - Enabled location services and proceeded with the fingerprint setup using the left thumb and setting the PIN to 6221.
- **22:20** - Commenced the addition of thumbprint data.
- **22:21** - Concluded the initial setup, choosing to skip the restoration of apps and data.

System Updates and Developer Mode Activation

- System auto-update commenced around 22:24.
- **22:25/26** - Enabled developer mode, subsequently activating the "stay awake" and "USB Debugging" options.
- **22:27** - Permitted USB debugging from the connected computer, setting it to "Always Allow".

Browser Interactions and IP Changes

- **22:32** - Opened the Vanadium browser and allowed notifications.
- **22:45** - Encountered network issues leading to IP change to 10.34.0.113, followed by testing the browser with a website visit.
- **Tue 29.08** - Noted auto-updates of various apps and subsequent system reboots leading to IP changes.

Rooting Process

- **12:35** - Initiated the rooting process by pushing Magisk to the phone and installing the app.
- **12:43 to 15:12** - Carried out a series of operations including pushing "boot.img" to the phone, patching the boot image, and rebooting to apply patches, successfully rooting the GrapheneOS; new IP assigned: 10.34.0.117.

PCAPdroid Installation and Network Tests

- **16:25 to 16:27** - Transferred and installed PCAPdroid on GrapheneOS, also setting up MITM TLS decryption.
- **16:28 to 16:57** - Conducted network tests to verify the setup, encountering limitations in TLS interception and root capture functionalities.
- **16:58** - Initiated the first PCAPdroid background capture session without any additional apps installed to monitor the background noise in its pristine state.

This detailed setup protocol ensured the establishment of an environment conducive to capturing a wide array of background signals and noise. Future analyses would delve deep into the data amassed, seeking to unravel the intricate web of communications that constitute the background noise, thus fostering a comprehensive understanding of daily smartphone operations and their security implications.

5.5.1 Detailed App Monitoring and Background Noise Capture

In this segment of the implementation chapter, the procedure for monitoring the same top 10 apps as identified in the stock Android setup is elucidated. This involved a meticulous background noise capture to understand the generic behavior of the apps while they were not actively being used.

The detailed breakdown is as follows:

- **Wed 30.08.**
  - **12:12** - New IP addresses were noted.
  - **12:12 to 14:45** - Each of the top 10 apps was downloaded from APKMirror or their respective vendor sites, bypassing the Playstore.
  - **14:45 to 21:56** - A series of individual tests were conducted on each app with varying decryption settings, highlighting functional peculiarities and anomalies.
  - **21:58** - Commenced a general capture session where all the recently downloaded apps were reopened once alongside the pre-installed ones to collect data on background noise.
- **Sat 02.09.**
  - **12:55 to 12:58** - Logged the new IP assignments and concluded the capture sessions on both the GOS and Pixel platforms.
- **Sun 03.09.**
  - **21:46 to 21:51** - Initiated the installation of Sandbox on GrapheneOS with the goal to reinstall all top apps via the Google Play Store, thereby maintaining an environment akin to the stock Android setup.
- **Mon 04.09. to Fri 08.09.**
  - **18:12 (Mon) to 13:38 (Fri)** - Across several days, the apps were first deinstalled and then freshly installed within the Google Play sandbox of GrapheneOS. A comprehensive series of tests were run on each app, both with and without decryption, to gather operational data.

– **13:37 to 13:38 (Fri)** - Closed the series with a session capturing the background noise generated with all ten apps open simultaneously, aiming to gather data on the apps' collective behavior in the background.

This detailed approach allowed for an exhaustive analysis of each app, both in terms of direct interaction and their contributions to the background noise during general phone usage. The data gathered promises a rich ground for evaluating the individual and collective privacy frameworks of these popular applications in subsequent chapters.

5.5.2 Data Transfer and Aggregation

Data Retrieval using adb

In the early stages of the data collection process, the Android Debug Bridge (adb) was utilized to facilitate a secure and efficient data transfer from the phones. This command-line tool allowed for direct communication with the device, enabling the extraction of vital data generated during the various phases of the analysis.

Merging CSV Files from Pcapdroid

Once the individual data segments were extracted, the subsequent step was to amalgamate the data from Pcapdroid which was stored in multiple CSV files. The Python programming language was harnessed for this task, ensuring a seamless merger of the datasets into a singular, comprehensive file. This centralized repository of data formed the bedrock for the extensive analysis carried forth in the thesis.

Integrating Data from pfELK

To supplement the data captured during the active usage phases, additional data pertaining to the 'initial setup' phase was gathered through pfELK integrated with Elasticsearch. This approach came as a requisite since the deployment of Pcapdroid was unfeasible during the setup period. Leveraging the pfELK system ensured the collection of substantial data, adding a crucial dimension to the initial setup analysis and painting a more rounded picture of the traffic patterns during this pivotal phase.

Conclusion

This meticulous data transfer and aggregation stage set a robust foundation for the subsequent data enrichment and analysis. Employing adb for data retrieval, Python for merging CSV files, and pfELK for additional data collection not only streamlined the data acquisition process but also ensured a comprehensive dataset ripe for a deep and insightful analysis.

5.6 Data Categorization and Enrichment

To achieve a comprehensive understanding and accurate representation of the traffic data, an elaborate data categorization and enrichment process was undertaken during the implementation phase. This process was essential in revealing the subtle behaviors of different applications across various environments, emphasizing their impact on privacy.

5.6.1 Categorization through VirusTotal and DuckDuckGo

In the initial step of data enrichment, the categorization of data traffic was meticulously carried out by leveraging the substantial databases of VirusTotal and DuckDuckGo. VirusTotal offered a reliable source for categorizing various domains based on their established reputations, aiding in the identification of potentially harmful traffic categories. Concurrently, DuckDuckGo's tracker radar database played a crucial role in classifying traffic into categories such as advertisement and tracker related traffic, thereby facilitating a deeper dive into the privacy aspects.

5.6.2 Geolocation Enrichment through MaxMind

To augment the depth of the analysis, the MaxMind database was employed to procure geolocation data of different IP addresses. This integration enabled a richer data set, paving the way for geo-specific analyses, and contributed to understanding the geographical dispersion of data traffic, which is vital in evaluating the privacy landscape of different environments.

5.6.3 Data Processing with Python and Pandas

The Python programming language, paired with the Pandas library, stood as the backbone for data processing. This versatile combination facilitated efficient data handling, manipulation, and analysis. The robust functionalities offered by Pandas allowed for a streamlined processing of large datasets, ensuring accuracy and efficiency

in data analysis, and played a pivotal role in unearthing critical insights from the traffic data.

5.6.4 Conclusion

Through meticulous data categorization and enrichment employing renowned databases and efficient programming tools, a profound layer of depth was added to the analysis. This methodical approach not only fostered a richer understanding of the data traffic patterns but also upheld the commitment to scrutinizing the privacy implications intrinsic in different environments.

6 Evaluation

Introduction to Stages and Environments. In the dataset, environments and stages refer to specific contexts during which the network traffic data was collected. Here we delineate them to facilitate a better understanding of the subsequent analysis:

- **Environments:** They represent different setups where the traffic was recorded. We categorize them into:
  - **Stock Android:** This environment refers to a standard setup with the pre-installed applications that come with the device.
  - **GrapheneOS:** A hardened open-source operating system that brings security enhancements.
  - **GrapheneOS Sandboxed:** Similar to GrapheneOS but encapsulated in a sandbox to offer a restricted environment, generally used to test untrusted applications without granting them access to personal data.
- **States:** They represent different periods of data collection, categorized as:
  - **Initial Setup:** The phase when the system is initialized for the first time, and essential setups are being performed.
  - **Background Stock:** This state involves data collection when only the stock applications are running in the background.
  - **Background with Apps:** A state that considers data traffic when top popular applications from the play store are downloaded and running in the background.

6.1 Temporal Analysis

It is imperative to understand that the temporal data analysis for the initial setup state is not plausible owing to the data collection method via pfELK, as elaborated in the implementation chapter. In this section, we focus on the analysis of traffic duration statistics obtained in different environments and states except for the initial setup.

6.1.1 Stock Android

Background Stock. During the background stock state, the majority of traffic durations were quite brief, with 50% of the connections lasting only 42 milliseconds, and 75% lasting less than 236 milliseconds. However, it is important to note a significant variance in the traffic durations, stretching up to about a minute at its peak. This wide range in duration might be indicative of a diverse nature of background communications ranging from quick data transfers to possibly sustained connections maintained by some stock applications.

Background with Apps. In the scenario where popular applications were running in the background, we observed an increase in both the average and maximum traffic duration, pointing towards more sustained communications, possibly due to richer functionalities and higher data exchange volumes involved in these applications. The median duration remained relatively low at 169 milliseconds, suggesting that a considerable amount of communications were still short-lived.

6.1.2 GrapheneOS

Background Stock. Delving into the GrapheneOS environment under the background stock state, the traffic exhibited more prolonged durations on average, with half of the connections lasting up to 950 milliseconds and a considerable stretch to a maximum duration of over 37 minutes. This substantial increase in duration might be suggestive of the robust security mechanisms employed in GrapheneOS, necessitating more extended communications to fulfill stringent security checks.

Background with Apps. The presence of popular apps further enhanced the traffic durations, even though the median remained under a second, pointing towards a majority of short-lived communications. The maximum duration spiked to an excess of 5 hours, representing potentially long-standing connections, possibly for updates or continuous data feeds from these applications, warranting a detailed examination to infer the nature and necessity of such lengthy connections.

6.1.3 GrapheneOS Sandboxed

Background Stock. In the sandboxed variant of GrapheneOS, the traffic duration showcased a bimodal behavior with 50% of the traffic having short durations while a significant portion concentrating around the 1-minute mark, which calls for a detailed investigation to understand the underlying reasons for this pattern.

Background with Apps. Similar to other environments, running popular applications in the background led to a noticeable increase in traffic durations, with a maximum stretch of over 3 hours. The average traffic duration saw a rise, implying a tendency for more prolonged communications when third-party applications are in operation, potentially opening up avenues for enhanced data transfers and interactions.

6.2 Geographical Analysis

In this section, we explore the geographical dispersion of the traffic based on their destination countries. While a considerable portion of traffic had unidentified destination countries, we are focusing our discussion on the ones that were successfully identified to understand the geographical preferences exhibited by the traffic in different states and environments.

Table 6.1: Geographical distribution of network traffic by environment and state.

Environment	Country / State	US	NL	SE	DE	NO	RU
Stock Android	Initial Setup	1206	0	0	20	0	0
	Background Stock	285	12	0	80	0	0
	Background with Apps	33301	577	321	53	0	98
GrapheneOS	Initial Setup	0	12	0	36	29	0
	Background Stock	6	51	0	0	34	0
	Background with Apps	11563	1621	364	82	97	0
GrOS-Sandboxed	Background Stock	2179	18	0	24	9	0
	Background with Apps	9095	316	186	27	47	102

6.2.1 Stock Android

Initial Setup. During the initial setup of Stock Android, a large portion of the traffic is directed towards the US, indicating a strong reliance on American servers for

foundational configurations and functionalities. European destinations like Germany register minor traffic, suggesting that while the US is the primary focus, there is a minimal reach in Europe during the setup process.

Background Stock. In its background stock state, Stock Android maintains its preference for US servers, with a slight increase in traffic to German servers. This suggests that even when idle, there might be background processes or updates that rely on these locations for data exchanges.

Background with Apps. With applications running in the background, Stock Android's traffic is heavily dominated by the US, with noticeable spikes towards the Netherlands and Sweden. There's also a minor traffic directed to Russia. This pattern indicates a diversified infrastructure for supporting various apps, but with a strong inclination towards American servers, showcasing its centralized nature in terms of data communication.

6.2.2 GrapheneOS

Initial Setup. GrapheneOS paints a different picture during its initial setup, where a substantial part of the traffic is directed towards European countries including Germany and the Netherlands, with Norway (NO) also being a notable destination. This outlines a different infrastructural preference or necessity between Stock Android and GrapheneOS, potentially leaning more towards European servers, which may imply a different approach to data privacy given the stricter data protection laws in the EU.

Background Stock. In the background stock state of GrapheneOS, traffic prominently directs towards Dutch servers, underlining a potential preference or requirement for servers based in the Netherlands during idle conditions.

Background with Apps. When applications are running in the background, the traffic encompasses a wide array of destinations, including but not limited to the US, Sweden, and France (FR), hinting at a diversified and global server network to facilitate various applications.

6.2.3 GrapheneOS Sandboxed

Background Stock. The GrapheneOS in a sandboxed environment displayed a prominent inclination towards US-based servers during its background stock state,

accompanied by traffic directed to German and Dutch servers, indicating a globally dispersed network facilitating the functionalities at this stage.

Background with Apps. The running applications in this environment extend the geographical reach of the traffic to encompass countries including Brazil (BR) and Singapore (SG), showcasing the global infrastructure that different applications rely on.

6.3 Traffic Volume Analysis

In this part of the analysis, we turn our attention to the volume of traffic exchanged during different operational states across the environments. The 'BytesRcvd' and 'BytesSent' fields from the data set lend insight into the volume of data transmitted in various states, aiding in unraveling potential privacy concerns and their implications on user experience and network performance.

Table 6.2: Network traffic data by environment and state in terms of bytes received and sent.

Environment	State	BytesRcvd	BytesSent
Stock Android	Initial Setup	0.00 KB (approximate)	45.83 KB
	Background Stock	38.41 MB	2.66 MB
	Background with Apps	4.25 GB	86.39 MB
GrapheneOS	Initial Setup	0.00 KB (approximate)	3.34 KB
	Background Stock	350.50 MB	1.20 MB
	Background with Apps	2.09 GB	77.81 MB
GrOS-Sandboxed	Background Stock	843.10 MB	9.04 MB
	Background with Apps	2.89 GB	99.22 MB

6.3.1 Initial Setup

During the initial setup phase, it was noticed that no bytes were received in both GrapheneOS and Stock Android environments. This could possibly be a data collection anomaly, and the bytes received should logically be in a range that is in the vicinity of the bytes sent to facilitate a successful setup. The fact that a minimal amount of data is being transmitted (in KB) underpins stringent control in the data exchanged, which is a positive indication from a privacy standpoint. It underscores a relatively lesser scope for data leakage or unauthorized data access during this very initial phase.

6.3.2 Background Stock

As we transition to the background stock state, there is a notable increase in both bytes received and sent across all environments. Particularly, the GrOS-Sandboxed environment stands out with a high data reception of 843.10 MB. This substantial volume might be attributed to the download of play services during this state. In comparison, the GrapheneOS has a balanced data transmission, possibly hinting at a more privacy-respecting design with controlled data exchange.

6.3.3 Background with Apps

The background with apps state further escalates the data volumes exchanged, with Stock Android topping the chart with a colossal data receipt of 4.25 GB. This mammoth data exchange volume hints at an active background data processing, which from a privacy viewpoint, can be concerning as it potentially opens up avenues for extensive data access by various apps, raising questions on the confidentiality and security of user data. GrapheneOS and GrOS-Sandboxed, although also witnessing high data volumes, remain slightly more restrained compared to Stock Android, implying a more controlled environment.

6.4 Application Analysis - Traffic Volume

In this section we go further into detail about the apps that were used to generate traffic.

6.4.1 Stock Android

During the initial setup of Stock Android, the unknown app field has registered a small amount of data sent, suggesting minimal background activities. The "Background Stock" state portrays moderate traffic with apps like Chrome and Google Play services.

In the "Background with Apps" state, Google Play Store evidently has a massive spike in data usage, affirming the downloading of the top ten apps from the play store. Applications such as Photos and TikTok are high traffic generators, possibly due to synchronization of media files and streaming of rich content, respectively. Apps like Gboard and Google show significant data sent which might be due to predictive text functionalities and constant feed updates respectively.

6.4.2 GrapheneOS

Analyzing the data traffic on GrapheneOS during the initial setup and background stock, it is observed that the system updater and Vanadium have considerable traffic, indicating that they are core to the system's functioning, probably facilitating updates and web viewing functionalities. The app labeled 'Apps' with significant data traffic can be inferred to be engaged in updating GrapheneOS factory apps or facilitating other background services.

The "Background with Apps" state presents a considerably increased traffic volume, with prominent data sharing apps such as Instagram, TikTok, and Snapchat showcasing high data usage, which might be associated with rich media content and perhaps constant background synchronization services. Applications like CapCut and SHEIN also demonstrate notable traffic, suggesting high user interaction and data exchange.

6.4.3 GrapheneOS with Google Play Services in a Sandbox

In the sandboxed environment of GrapheneOS, the Google Play services and Google Play store display substantial traffic, which is understandable given the role they play in app installations and updates. The data reflects the download of the top 10 apps, hence a surge in data usage in apps like TikTok and Instagram, both being multimedia-rich platforms demanding higher data bandwidth. Interestingly, the sandboxed environment has Google Play services consuming substantial data even in the background, pointing towards continual syncing and possibly advertisements and analytics traffic, a trend similar to stock Android environments.

6.5 Protocol and Port Analysis

Analyzing the 'IPProto', 'SrcPort', and 'DstPort' fields of the data traffic helps in understanding the various protocols and ports involved. This, in turn, sheds light on the nature and security of the communication in different environments and states.

- **DNS Traffic (Port 53):** Utilized mainly for DNS queries, a higher count might indicate an increased number of domain resolutions. In the 'Background with Apps' state, we see a substantial rise in the DNS requests in both GrapheneOS and stock Android environments, suggesting a larger number of apps communicating with servers, possibly for fetching updates or sending telemetry data. Notably, the GrapheneOS sandboxed environment, which facilitates the secure operation of apps from the Google Play Store, also exhibits a high number of DNS queries, signifying active internet communications during this state.

Table 6.3: Distribution of network traffic by destination port for each environment and state.

Environment	Destination Port / State	53	443	5228	5222	80	123
Stock Android	Initial Setup	816	1186	6	0	25	4
	Background Stock	11928	340	4	0	30	0
	Background with Apps	10680	33551	546	267	199	16
GrapheneOS	Initial Setup	56	35	0	0	6	0
	Background Stock	81	82	0	0	0	0
	Background with Apps	13858	13692	0	350	165	40
GrOS-Sandboxed	Initial Setup	0	0	0	0	0	0
	Background Stock	1512	804	1420	0	0	0
	Background with Apps	31338	9325	160	102	209	14

- **HTTPS Traffic (Port 443):** This port is generally used for secure web browser communication. Stock Android seems to have the most traffic through this port in the 'Background with Apps' state, potentially pointing to a higher amount of secure communications possibly due to pre-installed apps or services communicating over secure channels. The GrapheneOS with sandboxed Google Play follows, and GrapheneOS registers considerable traffic, hinting at secure communications possibly for system or app updates and other encrypted transmissions.

- **Ports 5228, 5222:** Typically used by Google services, traffic on these ports could indicate communication with Google servers. While unused in the GrapheneOS environment, there is a notable increase in traffic through these ports in the 'Background with Apps' state for stock Android, suggesting active communication with Google servers, likely for services like push notifications.

- **HTTP Traffic (Port 80):** Being less secure than port 443, traffic through this port is generally less favored. Nevertheless, all environments show some activity on this port in the 'Background with Apps' state, potentially for fetching non-encrypted data.

- **NTP Traffic (Port 123):** Used for Network Time Protocol (NTP) services, this port sees minor activity across all environments and states, indicating time synchronization activities.

6.6 Security Analysis

Each domain involved in the network traffic was examined using the VirusTotal API to evaluate its security standing. It should be noted that while VirusTotal offers a robust examination by leveraging several antivirus engines, it cannot guarantee absolute accuracy in the classification of a domain as 'suspicious' or 'malicious.' There could be false positives and negatives, and hence the analysis should be taken with a grain of caution. The data delineates the percentage of traffic labeled as 'suspicious' or worse, helping in the discernment of the security landscape of different environments and states. The table below depicts this data:

Table 6.4: Distribution of 'Suspicious' (or worse) results from VirusTotal by environment and state.

Environment	VirusTotal 'Suspicious' (or worse) State	False	True	%
Stock Android	Initial Setup	1047	35	3.2
	Background Stock	12302	0	0.0
	Background with Apps	26793	18478	40.8
GrapheneOS	Initial Setup	69	0	0.0
	Background Stock	172	0	0.0
	Background with Apps	27514	642	2.3
GrOS-Sandboxed	Background Stock	3738	4	0.1
	Background with Apps	36840	4335	10.5

- **Stock Android:** A significant spike in suspicious traffic is observed in the 'Background with Apps' state, amounting to a 40.8% incidence rate. This potentially indicates a higher vulnerability due to a broader array of apps and services, possibly including several from unverified sources, emphasizing the need for stringent security measures.

- **GrapheneOS:** In this security-focused environment, we witness minimal suspicious activity, primarily in the 'Background with Apps' state. However, with only a 2.3% incidence rate, it maintains a relatively secure landscape.

- **GrapheneOS with Sandboxed Google Play:** Despite being sandboxed, the integration with Google Play services does witness a surge in suspicious traffic, notably a 10.5% occurrence in the 'Background with Apps' state. This calls for careful scrutiny of the apps and services running in this environment.

6.7 Category Analysis

In the category analysis, we dissect the network traffic into three broad categories: Tracking related, Advertisement related, and others, which encapsulates all the remaining traffic.

Table 6.5: Percentage distribution of network requests by parent category for each environment and state.

Environment	Parent Category State	Tracking Related	Adv. Related	All Other
Stock Android	Initial Setup	0.00	0.99	99.00
	Background Stock	1.15	0.46	98.39
	Background with Apps	1.60	3.47	94.93
GrapheneOS	Initial Setup	0.00	0.00	100.00
	Background Stock	0.00	0.00	100.00
	Background with Apps	2.05	4.45	93.50
GrOS-Sandboxed	Background Stock	0.21	0.37	99.41
	Background with Apps	2.46	3.52	94.02

6.7.1 Stock Android

In the case of stock Android, during the initial setup phase, the "all other" category dominates with nearly 99% of the traffic, leaving a marginal share of approximately 1% to advertisement related traffic and virtually no tracking related traffic. This dynamic alters slightly in the "background stock" phase with a slight increase in tracking-related traffic, which accounts for about 1.15%, albeit advertisement related traffic sees a reduction to 0.45%.

However, when applications are active in the "background with apps" state, there is a discernible rise in both tracking and advertisement related traffic, registering at 1.59% and 3.47% respectively. Despite this, a considerable chunk of the network traffic, about 94.92%, remains in the "all other" category, indicating a lesser, yet present exposure to potential privacy intrusive traffic.

6.7.2 GrapheneOS

During the initial setup and background stock phases, virtually all the network traffic falls into the "all other" category, suggesting minimal exposure to advertisement and tracking-related traffic. This illustrates the robust privacy posture of GrapheneOS,

even during the initial moments of interaction and when the system operates in the background without active application usage.

However, with applications running in the "background with apps" state, there is a small uptick in tracking (approximately 2.05%) and advertisement (around 4.44%) related traffic. Despite this increase, the predominance of the "all other" category at 93.5% indicates a relatively private environment maintained by GrapheneOS, safeguarding users from excessive tracking and advertisements.

6.7.3 GrapheneOS Sandboxed

In the sandboxed environment of GrapheneOS, during the background stock state, the majority of the traffic, amounting to about 99.41%, falls under the "all other" category, with negligible amounts directed towards tracking (0.21%) and advertisement (0.37%) related activities.

As apps run in the background during the "background with apps" state, a minor increment in tracking and advertisement related traffic is noted, standing at 2.46% and 3.52% respectively. Similar to GrapheneOS, the sandboxed environment maintains a substantial share of other traffic, ensuring a restrained advertisement and tracking exposure to the users, hence fostering a privacy-preserving setting.

6.8 App Specific Category Analysis

In this subsection, we explore the app-specific traffic categories, emphasizing the privacy implications arising from the collected data. It's worth noting that, by design, app-specific data is absent during the 'Initial Setup' phase. For the purposes of this analysis, we have included only those apps where less than 100% of the traffic is categorized as 'All Other'. Comprehensive tables with detailed information are available in the appendix for readability.

6.8.1 Stock Android

From Tables A.4 and A.5, it is apparent that several apps on the Stock Android platform during the 'Background Stock' and 'Background with Apps' states show a significant amount of traffic associated with tracking and advertisement domains. Apps such as 'Chrome' and 'YouTube Music' reveal a considerable percentage of such traffic, highlighting more pronounced privacy risks on the Stock Android platform compared to GrapheneOS and its sandboxed variant.

6.8.2 GrapheneOS

Referring to Table A.1, we notice a general trend of minimal interaction with tracking and advertisement domains during the 'Background with Apps' state, with the substantial majority of traffic falling under the 'All Other' category. However, apps like Instagram and 'Live Weather' exhibited a relatively higher engagement with these potentially privacy-compromising domains, emphasizing a need for user vigilance while utilizing these apps.

6.8.3 GrapheneOS Sandboxed

The traffic data during the 'Background Stock' and 'Background with Apps' states, as depicted in Tables A.2 and A.3 respectively, largely mirrors the trends observed in the unsandboxed GrapheneOS environment. Despite the majority of the traffic being categorized as 'All Other,' apps such as 'Google Play services' and 'Live Weather' demonstrated a non-negligible interaction with tracking and advertisement domains, raising potential privacy concerns.

7 Discussion

In this chapter, we critically discuss the results obtained from the various analyses conducted in the previous sections. The evaluation unveiled a rich tapestry of insights shedding light on the different facets of data traffic across various environments and states.

7.1 Results

7.1.1 Temporal Analysis

The temporal analysis provides a deep dive into the durations governing the traffic in different environments and states, except for the initial setup due to the limitations in data collection methodology. While short-lived communications dominated the landscape, there were instances of substantially prolonged durations, necessitating a thorough scrutiny to ascertain the functionalities warranting such extended communications, thus laying a path for future explorative studies in this direction.

7.1.2 Geographical Analysis

The geographical analysis portrays a dynamic geographical landscape of the traffic. US emerges as a significant hub across different environments and states, albeit with considerable traffic directed to various European and other global destinations, indicating a wide network of servers facilitating the functionalities in different environments.

The data showcases a potential privacy implication, especially in the contexts where the traffic is prominently directed to non-EU countries, potentially bypassing the stringent data protection regulations prevalent in the EU. However, it is worth noting that GrapheneOS, particularly in its initial setup and background stock states, seemingly

prefers EU-based servers, which might point to a privacy-conscious choice given the robust data protection frameworks in the region.

This multifaceted landscape showcases a globalized network architecture where the devices interact with a wide array of servers globally, indicating a complex interplay of functionalities facilitated through a diverse geographical infrastructure network.

7.1.3 Traffic Volume Analysis

The traffic volume analysis sheds light on the considerable variations in data transmission across different states and environments. A key observation is the surge in data volume as we move from the initial setup to environments with active apps running in the background, indicating substantial data processing activities happening behind the scenes. This scenario, although functional from a user experience perspective, flags potential privacy implications, given the extensive volume of data exchanged potentially encompassing sensitive user data.

The analysis sheds light on the varying data traffic patterns across different operating environments. It is discernible that the integration or exclusion of Google Play services significantly impacts the data traffic volume, with GrapheneOS offering a comparatively lean operation. The sandboxed environment, while facilitating Google Play services, inherently increases data traffic, aligning more with the stock Android's traffic patterns. Moreover, apps rich in media content invariably consume more data, more so when facilitated through Google Play services, highlighting a potential area for optimization and user privacy enhancement.

It is critical, from a privacy perspective, to maintain a vigilant approach towards the high-volume data exchanges especially in states with active background apps, as it stands as a ripe ground for potential unauthorized data access and compromises. The initial setup phase, albeit with minimal data exchange, requires further scrutiny to ensure the bytes received anomaly is addressed to present a true picture of the data exchange landscape during this stage.

The analysis, thus, underscores the necessity for optimized data management strategies to balance functional efficiency with privacy preservation, advocating for a design that respects user privacy while delivering optimal performance.

7.1.4 Application Analysis - Traffic Volume

Comparative Analysis of 'Background with Apps'

Comparing the "Background with Apps" state across Stock Android, GrapheneOS, and GrapheneOS with sandboxed Google Play services, we observe that:

- The GrapheneOS with sandboxed Google Play services and Stock Android depict somewhat similar traffic volumes for apps like Instagram, TikTok, albeit with slight variations. This similarity might stem from the fact that both environments are leveraging Google Play services, thereby involving similar data exchange patterns including background updates and advertisements.

- The independent apps in the GrapheneOS environment (excluding sandbox) register substantial traffic volumes but less than the counterparts in stock Android. This could be due to the absence of Google Play services, which often augment data traffic through ads and background synching services.

- Apps like GPS and System Updater in all three environments showcase the essential functionalities being catered to in all scenarios albeit with differing data usage patterns, underlying the variation in operational dynamics across different environments.

Summary

In evaluating the diverse landscapes of GrapheneOS in its vanilla and sandboxed configurations, and Stock Android, we discern significant variations in the traffic volume generated during different operating states. The initial setup and background phases naturally exhibit lower traffic volumes as compared to the states where applications are actively utilized, an expected trend given the fewer operations being conducted at this stage.

We note a discernable pattern where core system functionalities and essential applications register a predominant share of the traffic in the initial states, delineating their critical role in system setup and maintenance. For instance, system updater and Vanadium in GrapheneOS and core applications in Stock Android illustrate substantial traffic, emphasizing their pivotal roles in ensuring system operability and facilitating basic functionalities such as web browsing and system updates.

As we venture into the states with active application backgrounds, the traffic volume experiences a noticeable surge. Here, mainstream social media applications stand as significant contributors to the increased data traffic. Their substantial traffic can be rationalized considering their rich media content which necessitates higher data usage, paired with constant background synchronizations to offer users the most recent content updates.

Furthermore, the sandboxed environment of GrapheneOS provides an insightful glimpse into the workings of Google Play Services within a secured sphere, revealing a substantial amount of traffic, even in the backdrop, which could be indicative of ongoing synchronizations, and possibly advertisement and analytics traffic.

This analysis portrays a vivid differentiation in data traffic patterns across various environments, shedding light on the innate behaviors of applications in distinct setups. It becomes imperative to delve deeper to understand the underlying causes for these variations, which pave the path for future investigations, fostering a safer and privacy-preserving user experience.

7.1.5 Protocol and Port Analysis

The analysis suggests a proactive approach in GrapheneOS towards secure communications, limiting traffic to ports associated with secure protocols. However, when integrated with the sandboxed Google Play environment, we observe an increase in traffic associated with Google services, albeit still maintaining a respectable focus on secure communications.

7.1.6 Security Analysis

This data underscores the vital role of stringent security protocols, especially in environments with a higher prevalence of apps from various sources. The sandboxed environment in GrapheneOS, despite showing a rise in suspicious traffic, still manages to maintain a substantially lower percentage compared to stock Android, showcasing its fortified security infrastructure.

Furthermore, it is crucial to address the potential privacy implications stemming from the results. A notable point of discussion is the spike in suspicious traffic observed in stock Android. This could potentially be attributed to a heightened presence of ad or tracking domains, which are frequently flagged — sometimes inaccurately — as 'suspicious' due to their nature of collecting user data for targeting ads. Consequently, while this points towards a privacy concern, indicating a probable substantial data collection and tracking landscape in the stock Android environment, it also suggests a need to be cautious in interpreting the high percentage of 'suspicious' traffic. Cautiously, one shouldn't consider a high 'suspicious' tag rate as direct evidence of compromised security.

7.1.7 Category Analysis

Analyzing the data across different environments and states brings forth a notable observation: both GrapheneOS and its sandboxed version maintain a higher degree of privacy during the initial setup and background stock states, with minimal traffic devoted to tracking and advertisements.

Stock Android, while maintaining a majority of its traffic in the “all other" category, shows a slightly higher propensity towards advertisement and tracking related traffic, especially with apps running in the background.

In conclusion, users employing GrapheneOS and its sandboxed variant can expect a more private network environment compared to stock Android, particularly in the early stages of setup and when the device is in the background stock state. Despite this, it is pivotal to remain cautious, as the introduction of apps can potentially elevate the exposure to tracking and advertisement related traffic across all environments.

7.1.8 App Specific Category Analysis

Upon evaluating the individual environments, it is evident that the GrapheneOS, in both sandboxed and unsandboxed configurations, provides a more privacy-preserving platform compared to Stock Android, especially in the 'Background Stock' state where GrapheneOS exhibited no app communications with tracking or advertisement domains, starkly contrasting the Stock Android environment.

Moreover, the sandboxed feature in GrapheneOS further substantiates its privacy-preserving claims by considerably limiting the amount of traffic associated with tracking and advertisements, even when apps are running in the background. However, users are urged to exercise caution while using apps with higher percentages of tracking and advertisement domain interactions to maintain a robust privacy posture.

This analysis reinforces the privacy-centric architecture of GrapheneOS, underscoring its effectiveness in shielding users from potential privacy intrusions stemming from app traffic. It also reveals that while the sandboxed feature offers an additional layer of privacy, it is not entirely impervious, thus emphasizing a continual need for user vigilance and informed app usage.

7.2 Limitations

While our analysis offers a detailed view of the traffic patterns observed during the study, it is pivotal to acknowledge the constraints we operated under.

Firstly, decrypting traffic proved challenging across all devices and OS flavors under investigation. To gauge the significance of the collected data, I initially analyzed random samples of successfully decrypted traffic. These preliminary assessments suggested that the Fully Qualified Domain Name (FQDN) often provides sufficient insight into the traffic's nature and intent, even in the absence of complete decryption. This observation was particularly underscored by the fact that most domain and subdomain combinations typically correspond to specific categories of traffic.

However, this correlation is not universally guaranteed, and exceptions might exist.

Secondly, a considerable amount of data was classified under the “All Other" category, which, although necessary for the readability of the study, might mask finer details and subtle insights into non-advertisement and non-tracking traffic.

Also, while our study provides a robust initial exploration, the apps and environments selected for the study are not exhaustive. Different sets of apps and different environments could potentially showcase varied traffic patterns, hinting at the necessity for broader studies in the future.

Lastly, the dynamic nature of the digital landscape means that our analysis, though comprehensive, offers more of a snapshot in time rather than a universally applicable understanding. The rapid pace of changes in app functionalities and operations introduces a level of fluidity in the results, emphasizing the need for ongoing analysis in this domain to keep abreast with the ever-evolving trends and to secure user privacy more effectively.

7.3 Threats to Validity

A critical point to ponder is the validity of the categorizations employed in the analysis. The databases sourced from VirusTotal and DuckDuckGo serve as substantial resources, yet they are not exhaustive. They might not encompass all possible traffic categories, introducing a degree of uncertainty in the classification. Additionally, it is pertinent to note that the databases from MaxMind may not always correctly classify domains based on geographical locations due to the inherent limitations in GeoIP data. This aspect poses a risk of misclassification, thereby introducing a potential bias in the analysis.

Furthermore, the dynamic nature of web services means that categories could undergo changes over time, which threatens the longitudinal validity of the results. It is also possible for domains to shift their focus and for new domains with different attributes to emerge, posing a continuous challenge to maintain an up-to-date categorization system.

7.4 Contradictory Points and Clarifications

As we navigated the complex process of data analysis, we encountered moments where the data presented contradictory points. In both the GrapheneOS and Stock Android environments, our data indicated that no bytes were received. This observation raises concerns about potential data collection anomalies, as one would logically expect the bytes received to be in a range proximate to the bytes sent in order to execute a

successful setup Also the categorization of traffic raised several questions, notably regarding the “Unknown” category. Despite being marginalized in the analysis, this category remains a fertile ground for further exploration as it could house unidentified trends and patterns.

7.5 Conclusion

In this discussion, we have explored the distinctive privacy landscapes characterized by different environments such as GrapheneOS, GrapheneOS Sandboxed, and Stock Android. Through a meticulous analysis, we managed to decipher the variations in tracking and advertisement-related traffic across different states within these environments, establishing a foundation upon which a deeper understanding of privacy implications can be built.

While we identified promising aspects in terms of privacy, particularly with GrapheneOS, it is essential to acknowledge the existence of certain limitations and threats to the validity of this study, as detailed in the preceding sections. These constraints underline the necessity for a careful and informed interpretation of the findings.

As we conclude this chapter, it is pertinent to underscore the primary objective of fostering an informed discussion around the data traffic patterns observed in different environments, which should serve as a precursor to more exhaustive investigations in the future. The endeavor here is not just to provide a comparative analysis but to lay the groundwork for further studies that can delve deeper into understanding the intricate landscape of privacy in the digital age.

8 Conclusion

In this investigative journey into the realms of different operating environments including GrapheneOS, GrapheneOS-Sandboxed, and Stock Android, we uncovered crucial insights into their respective data traffic patterns and privacy landscapes. This conclusion section succinctly encapsulates the differences observed in various critical stages: the initial setup, the 'Background Stock', and the 'Background with Apps'.

8.1 Initial Setup

In the initial setup stage, a notable distinction was observed in the way GrapheneOS and Stock Android manage their traffic. GrapheneOS exhibited a conscious choice in directing traffic predominantly to EU servers, perhaps capitalizing on the robust data protection frameworks established in the region. This stands in contrast to Stock Android which maintained a more diverse geographical footprint, potentially presenting increased privacy risks owing to a more distributed data traffic pattern. This paints GrapheneOS in a positive light, potentially offering a more secure setup environment in alignment with stringent data protection norms.

8.2 Background Stock

The 'Background Stock' stage presented a clear demarcation between GrapheneOS and Stock Android. While GrapheneOS maintained a pristine environment with no communications with tracking or advertisement domains, Stock Android manifested a slight tendency towards traffic associated with advertisements and tracking. This underlines a heightened privacy-preserving posture of GrapheneOS even in a dormant state, steering clear of potential intrusions that might be encountered in the Stock Android environment.

8.3 Background with Apps

Transitioning to the 'Background with Apps' stage, we perceived significant contrasts between Stock Android and GrapheneOS-Sandboxed environments. The sandboxed version of GrapheneOS elucidated its competence in maintaining a substantially privacy-preserving background even with active app environments, albeit not completely immune to tracking and advertisement traffics. Stock Android, on the other hand, demonstrated a more permeable environment to such traffics, necessitating a cautious approach for users seeking a privacy-focused experience. Despite the sandboxed environment incorporating Google Play services, it managed to restrict the traffic to a commendable extent, reinforcing its commitment to privacy.

8.4 Overall Insights

As we recapitulate the journey through different stages of operation in these environments, GrapheneOS consistently emerges as a vigilant guardian of user privacy, consciously navigating traffic to secure regions and minimizing unwarranted communications. The sandboxed variant of GrapheneOS, despite its integration with Google Play services, stood its ground in protecting user data, indicating a well-fortified infrastructure.

Stock Android, although functional and facilitating a wide network of servers, leaves room for enhancing its privacy metrics, especially in terms of curtailing advertisement and tracking-related traffics in various operating states.

8.5 Looking Ahead

As we cast our eyes towards the future, it becomes imperative to foster a digital landscape that champions user privacy without compromising on functionality. This study, shedding light on the diverse privacy landscapes of different operating environments, sets the stage for further expansive research. Through a deeper understanding and subsequent optimization of data traffic patterns, we can envision a future where digital environments are both functional and fiercely protective of user privacy.

A Appendix

A.1 Licensing and Usage Rights

Diagrams within this document were constructed using the tool draw.io (diagrams.net). According to their terms of service, there exist no limitations on the employment of diagrams fashioned using their utilities. This encompasses the unfettered utilization of embedded copyrighted icons that are proprietary to JGraph.

A.2 Supplementary Tables

Presented in this section are tables previously omitted from the main text to enhance clarity and coherence. These tables offer supplemental data and insights related to earlier discussions. For the sake of readability, only apps with less than 100% of their traffic categorized as 'All Other' are included. Notably, in the background stock state of GrapheneOS, no app communicated with domains linked to tracking or advertisements. Consequently, there is no dedicated table representing this specific state.

Table A.1: App traffic by category: GrapheneOS - Background with Apps (removed entry if 100% "All Other")

Parent Category App	Tracking Related	Advertisement Related	All Other
CapCut	0.000000	0.595632	99.404368
Instagram	4.195804	8.391608	87.412587
SHEIN	0.322841	2.905569	96.771590
TikTok	0.000000	0.040138	99.959862
Translate	0.000000	8.000000	92.000000
Vanadium	1.219512	3.048780	95.731707
Live Weather	10.400531	22.416464	67.183005

Table A.2: App traffic by category: GrOS-Sandboxed - Background Stock (removed entry if 100% "All Other")

Parent Category App	Tracking Related	Advertisement Related	All Other
Google Play Store	0.000000	0.439560	99.560440
Google Play services	0.257400	0.386100	99.356499

Table A.3: App traffic by category: GrOS-Sandboxed - Background with Apps (removed entry if 100% "All Other")

Parent Category App	Tracking Related	Advertisement Related	All Other
CapCut	0.000000	1.830664	98.169336
Google Play Store	0.396376	0.226501	99.377123
Google Play services	8.959044	10.644198	80.396758
Instagram	3.719008	7.438017	88.842975
Intent Filter Verification Service	0.496278	0.992556	98.511166
SHEIN	1.495327	3.925234	94.579439
Temu	0.715990	5.608592	93.675418
TikTok	0.000000	0.077564	99.922436
Translate	0.000000	4.615385	95.384615
Live Weather	6.236661	9.070429	84.692910

Table A.4: App traffic by category: Stock Android - Background Stock (removed entry if 100% "All Other")

Parent Category App	Tracking Related	Advertisement Related	All Other
Chrome	12.845850	1.976285	85.177866
Google Play services	0.105960	0.185430	99.708609
Speech Recognition and Synthesis from Google	0.000000	33.333333	66.666667
YouTube Music	25.000000	75.000000	0.000000

Table A.5: App traffic by category: Stock Android - Background with Apps (removed entry if 100% "All Other")

Parent Category App	Tracking Related	Advertisement Related	All Other
Google	0.000000	1.276596	98.723404
Google Play Store	1.363636	0.000000	98.636364
Google Play services	8.761642	9.934460	81.303898
Root	0.000000	11.111111	88.888889
SHEIN	50.000000	0.000000	50.000000
Temu	0.000000	4.056665	95.943335
Translate	0.000000	6.741573	93.258427
YouTube Music	18.055556	27.777778	54.166667
Live Weather	1.922379	4.651795	93.425825

Bibliography

[Aks22] Juri Aksenov. How many people use grapheneos. https://discuss.grapheneos.org/d/50-how-many-people-use-grapheneos/3, 2022. Comment in discussion titled "How many people use GrapheneOS". Username: akc3n.

[AP18] Efthimios Alepis and Constantinos Patsakis. Unravelling security issues of runtime permissions in android. *Springer*, 2018. https://link.springer.com/article/10.1007/s41635-018-0053-2.

[Far23] Emanuele Faranda. Pcapdroid, 2023. GitHub repository https://github.com/emanuele-f/PCAPdroid,.

[GRR+20] Julien Gamba, Mohammed Rashed, Abbas Razaghpanah, Juan Tapiador, and Narseo Vallina-Rodriguez. An analysis of pre-installed android software. In *2020 IEEE Symposium on Security and Privacy*. IEEE, 2020. https://ieeexplore.ieee.org/abstract/document/9152633.

[Inc23a] Duck Duck Go Inc. Duckduckgo tracker radar, 2023. GitHub repository https://github.com/duckduckgo/tracker-radar.

[Inc23b] MaxMind Inc. Maxmind geoip2 database. https://www.maxmind.com, 2023.

[NBS22] Trung Tin Nguyen, Michael Backes, and Ben Stock. Freely given consent? studying consent notice of third-party tracking and its violations of gdpr in android apps. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, New York, NY, USA, 2022. ACM. https://doi.org/10.1145/3548606.3560564.

[RAFW+19] Joel Reardon, Álvaro Feal, Primal Wijesekera, Amit Elazari Bar On, Narseo Vallina-Rodriguez, and Serge Egelman. 50 ways to leak your data: An exploration of apps' circumvention of the android permissions system. In *Proceedings of the 28th USENIX Security*

Symposium. USENIX, 2019. https://www.usenix.org/conference/usenixsecurity19/presentation/reardon.

[RNVR+18] Abbas Razaghpanah, Rishab Nithyanand, Narseo Vallina-Rodriguez, Srikanth Sundaresan, Mark Allman, Christian Kreibich, and Phillipa Gill. Apps, trackers, privacy, and regulators: A global study of the mobile tracking ecosystem. In *Network and Distributed Systems Security (NDSS) Symposium 2018*. NDSS, 2018. https://dspace.networks.imdea.org/bitstream/handle/20.500.12761/507/trackers.pdf?sequence=1&isAllowed=y.

[ST23] Thomas Sutter and Dr. Bernhard Tellenbach. Firmwaredroid: Towards automated static analysis of pre-installed android apps. Zurich University of Applied Sciences, Cyber-Defence Campus, Armasuisse, 2023. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=10172951.

[Vir23] VirusTotal. Virustotal database. https://www.virustotal.com, 2023.

[Wil23] A. Wilson. pfelk: pfsense/opnsense + elastic stack, 2023. GitHub repository https://github.com/pfelk/pfelk.