Douglas College

Module Code:

CSIS 4440

Module Title:

Mobile Cybersecurity -Project Proposal

Year & Semester:

2025 Fall

Student ID	Student Name
300381351	Allwyn Mascarenhas (Team Lead)
300391004	Chetan Kaur

Instructor: Priya

Date: 24th October 2025

Document Control	
Title	Mobile Cybersecurity Project Proposal
Group ID	10
Members	Allwyn Mascarenhas, Chetan Kaur
Student ID	Allwyn: 300381351, Chetan: 300391004
Team lead	Allwyn Mascarenhas
GitHub Repo	https://github.com/allwynmas/F25-4440-G10/

Student Name:

Allwyn Mascarenhas

Overview

My goal with this project is to understand the android security ecosystem and apply real world threat modeling parameters to it. This helps us to understand risk in terms of real-world exploitability that endangers any leg of the CIA security triad and informs the types of risk mitigation measures we deploy.

Given the vastness of the Android cybersecurity ecosystem's, this project focuses on understanding security measures on a few core mechanisms that millions of everyday people rely on.

The core issues that will be explored in this project are android's USB protocol security, geolocation, and application permissions. Billions of users use these protocols to access files, photos, and for navigation functions.

Tooling Used

lsusb

A small Linux command-line utility that lists USB buses and the devices connected to them.

Why it matters: quick way to enumerate USB devices attached to a host (or an Android device in USB host mode) so you can identify vendor/product IDs, bus/port topology, and detect unexpected peripherals.

Typical use: lsusb to list devices, lsusb -v for verbose descriptors. Useful as the first reconnaissance step before deeper USB interaction or fuzzing.

libusb

A portable userspace C library (with many language bindings) for direct USB device access.

Why it matters: lets security researchers write custom tools to talk to USB devices (send control, bulk, interrupt transfers), inject malformed requests, or implement test harnesses without kernel drivers.

Typical use: write C/Python programs (or use pyusb) to exercise device endpoints, read descriptors, and automate protocol-level testing.

usb-device-fuzzing (T2 Infosec 2012)

A compact collection of USB fuzzing code first released at T2 Infosec 2012. Its goal is to exercise USB protocol handling in device firmware and host stacks by sending malformed, unexpected, or randomized USB traffic and watching for crashes, hangs, or logic failures. In the context of Android security this is valuable whenever Android acts as a USB host (OTG) or connects to accessories fuzzing reveals parsing bugs in device firmware or in host-side drivers that can lead to denial-of-service, information leaks, or code execution.

Drozer

Drozer is a leading Android dynamic-analysis toolkit created to replace the dozens of one-off test apps auditors used to write; by running a small agent on a test device or emulator and connecting to an interactive console, drozer lets you act exactly like an installed Android application calling IPC mechanisms (intents, services, content providers), interacting with the OS, and exercising app behavior , so you can quickly enumerate exposed components, probe for insecure IPC and permission issues, and automate exploitation checks to prove whether a finding is actually exploitable.

Applications

The Geozilla application will be studied in this project using the tooling mentioned above. Drozer gives us the ability to act like an android application and perform similar tasks, functions, etc. I will use this to test geozilla's behaviour in the context of how it protects the CIA security triad against other apps, especially if a malicious gets on the device.

Geozilla

GeoZilla is a family safety and location-sharing app that helps families and trusted groups stay connected, informed, and secure. It operates through private circles where members explicitly consent to share their location, ensuring transparency and control over personal data. The app's goal is to provide peace of mind through smart monitoring and timely alerts.

Core Features

- **Live Location Sharing:** View real-time GPS locations of family members.
- **Location History:** Access daily movement timelines.
- **Geofenced Alerts:** Receive notifications when someone arrives or leaves set locations like home or school.
- **Battery Monitoring:** Get alerts when a member's phone battery is low.
- **Driving Reports:** Analyze driving habits to promote safer behavior.

- **Crash Detection & SOS Alerts:** Automatically detect accidents and send emergency notifications.
- **AR Finder:** Locate nearby members using augmented reality.

Permissions

GeoZilla requests several permissions to deliver its safety features:

- **Location Services (Always/Background):** Enables continuous tracking, geofencing, and crash detection.
- **Notifications:** Sends important alerts such as SOS, arrival/departure, and low battery.
- **Contacts (Optional):** Simplifies inviting trusted members.
- **Camera & Photos:** Used for setting profile images or scanning QR invites.
- **Motion & Fitness:** Supports driving analysis and crash detection through movement monitoring.

Novelty Aspects

Novelty aspects of the project are aimed at coming up with creative methods, and approaches to learn, study, test, document the android security features and framework beyond what the tools used achieve themselves.

- **AI Attack generation**: Security experts have always warned about the possibility of AI being used to ramp up the number of sophisticated attacks by novice hackers, malicious users, etc. Once I have used the tools to perform reconnaissance on the android attack surface, using AI to write custom novel methods of exploiting it is a natural step and something we are already witnessing in the real world.
- **Combining tools to test attacks**: USB tools listed above gives me information about how the usb protocol works, and drozer lets me *act like an* installed app on the android system. I will explore the possibility of using drozer to interact with the USB protocol to test possible attacks.
- **Remote USB Attacks**: USB is usually understood as a physical attack surface, however, faulty implementations, vulnerabilities, incorrect parsing might allow an attacker to send a malicious payload to the device and sabotage it. I will explore this with the existing toolkit, and possibly newer tools if required.

Student Name:

Chetan Kaur

Application Overview

Uber is a globally recognized ride-sharing platform used by millions for everyday transportation. It collects, stores, and manages large amounts of sensitive user data, including real-time GPS location, payment information, trip history, and personal credentials. From a forensic perspective, analyzing Uber offers a unique opportunity to understand how movement patterns, identity data, and payment transactions are stored within a mobile device. Such analysis can help uncover potential data leakage risks or improper data handling practices that may compromise user privacy and security. Forensic extraction can also reveal traces of past rides, geolocation patterns, and residual data that persist even after the app is deleted, highlighting the importance of secure data management in high-traffic applications.

Telegram is a popular cloud-based messaging application known for its commitment to privacy, encryption, and security. It provides features such as end-to-end encrypted chats, secret conversations, and secure file sharing. In a digital forensic context, Telegram offers valuable insight into how privacy-focused applications store and protect communication data. By examining Telegram's internal databases, cache files, and encryption mechanisms, investigators can determine how messages, media files, and contacts are retained or deleted. Understanding Telegram's data handling approach provides valuable context for studying secure messaging systems and evaluating how encryption affects forensic recovery processes.

Forensic Tools Overview

Andriller is an advanced Android forensic suite designed to automate data extraction, decoding, and reporting. It excels in recovering deleted data, extracting passwords, and parsing complex app databases. Andriller can handle both logical and physical extractions, offering deep insight into encrypted or hidden application data. Its decoding capabilities are particularly effective for analyzing secure messaging platforms like Telegram, allowing investigators to recover message fragments, decryption keys, and deleted logs (Sazonov, 2022). By using Andriller, forensic investigators can gain a detailed understanding of how sensitive data such as messages, credentials, and location artifacts are stored and retrieved within the Android ecosystem.

Avilla Forensics is a powerful forensic tool that focuses on extracting data from Android devices, including those protected by modern security features. One of its most notable capabilities is rootless extraction it can access application data without requiring administrative (root) privileges, maintaining the integrity of the device. Avilla also supports APK downgrading, a process that enables forensic analysts to access secured data even when applications have been updated with enhanced security layers. Its features include encrypted logging, mass device processing, and detailed data mapping, making it highly effective for structured and repeatable forensic investigations (Daniel, 2024). When used alongside Andriller, Avilla Forensics enhances the reliability and depth of forensic analysis for applications like Uber and Telegram.

Project Overview

The main objective of this forensic project is to combine Andriller and Avilla Forensics to conduct an in-depth analysis of two mobile applications Uber and Telegram on Android devices. The goal is to explore how each application manages sensitive information and to evaluate potential privacy and security implications.

Andriller will be employed to extract and analyze encrypted messaging data, media caches, and account-related information from Telegram. Meanwhile, Avilla Forensics will complement this analysis by performing rootless, in-depth extraction of secured app data and metadata from both Telegram and Uber. The investigation will focus on how Uber stores user locations, payment details, and account credentials, as well as how Telegram manages encrypted messages, secret chats, and session keys.

Extracted forensic data will be organized and exported into structured formats such as CSV or Excel, allowing for systematic examination and comparison. Visualization tools will be used to identify correlations, activity timelines, and key evidence patterns. This visual representation of data will simplify the interpretation of complex forensic findings and enhance the overall presentation of results.

Novelty

- **Privacy Risk Profiling**
  This stage focuses on visualizing and analyzing how sensitive data including user location, payment, and communication records remain stored on a device. It highlights potential risks of exposure and misuse. By applying data visualization tools, the project makes findings accessible and interpretable even for non-technical audiences. This approach emphasizes transparency in mobile app data practices.
- **Cross-App Forensic Correlation**
  One of the innovative aspects of this project is comparing forensic artifacts from both Uber and Telegram. This cross-app correlation will reveal how different platforms handle user data and security under varied operational models—one emphasizing mobility and the other communication privacy. Such comparison allows identification of differences in encryption handling, residual data retention, and access control methods.
- **Rootless Extraction for Secure Apps**
  Avilla Forensics’ rootless extraction and APK downgrade capabilities enable forensic evidence collection from newer, security-hardened Android versions. This method allows

the recovery of forensic data without compromising device integrity or requiring risky rooting procedures. The project demonstrates how modern forensic tools can adapt to evolving mobile security restrictions.

- **Decryption and Recovery Extensions**
  Andriller's built-in database decoders and message decryption modules will be utilized to recover deleted or hidden artifacts. This includes encrypted message histories, tokens, and temporary cache data that may otherwise remain inaccessible. Implementing this step requires technical understanding of cryptographic procedures and manual interpretation of extracted database structures.

Project Contract

We, the members of Group 10, pledge to abide by this contract to support each other in achieving the goals of this Mobile Cybersecurity Project. We will hold each other accountable to contribute to the project and at every step adhere to the highest standards of behavior ethical, moral, and technical.

We Pledge To:

1. Respect deadlines set by our instructor, and by ourselves, and each other.
2. Attending all project meetings as planned and supporting each other in every way possible.
3. Take responsibility for our work to make sure it meets the highest standards of accuracy, honesty, and best practices of the cybersecurity industry.
4. Challenge ourselves and each other to learn and push the boundaries of our knowledge at every step.

Signatures

Chetanpal Kaur

Allwyn Mascarenhas

Chetan Kaur

AI Use Section

Allwyn’s AI Section

AI Tool Name & Version	Account Type	Specific Use in Project
ChatGPT (GPT-4, 5)	Free	To generate summaries, writeups for tools, some command e.g. To check grammar, writing structure, etc.

Chetan’s AI Section

AI Tool Name & Version	Account Type	Specific Use in Project
ChatGPT (GPT-4, 5)	free	Used for drafting, refining project proposal text, generating step-by-step guides, forensic tool explanations, and data visualization suggestions.
Perplexity	free	Supplement research and find additional technical insights.

Work Logs

Allwyn’s work log

Date	Student Name	Number of Hours	Description of Work
10/13/2025	Allwyn	1	Project proposal meeting with instructor and teammate.
10/18/2025	Allwyn	2	Reviewed the Android Hacker’s Handbook.
10/19/2025	Allwyn	3	Research newer tools, apps for study for updating the initial report.
10/21/2025	Allwyn	2	Created the initial proposal structure for this document, and writing the first draft.
10/23/2025	Allwyn	2	Improving the first draft to finalize
10/24/2025	Allwyn	1	Final review with teammate Chetan in college before submission.

Chetan’s Work Log

Date	Student Name	Number of Hours	Description of Work
10/7/2025	Chetan	2	Researched and selected forensic tools
10/9/2025	Chetan	2.5	Evaluated Uber and Telegram app features for forensic analysis
10/11/2025	Chetan	1.5	Studied documentation and YouTube guides for forensic tools
10/13/2025	Chetan	1	Had meeting with professor to discuss application and tool selection
10/21/2025	Chetan	3	Created draft proposal with objectives, novelty, and workflow

References

Daniel, A. (2024, October 22). *Avilla Forensics*. Retrieved from https://github.com/: https://github.com/AvillaDaniel/AvillaForensics

Sazonov, D. (2022, April 22). *Andrilla*. Retrieved from https://github.com/den4uk/andriller

Sailer, T. (2019, February 11). lsusb [Utility]. (Contact: sailer@ife.ee.ethz.ch), from https://man7.org/linux/man-pages/man8/lsusb.8.html

libusb. (n.d.). libusb [Library]. Retrieved [2025], from https://libusb.info/

ollseg. (n.d.). usb-device-fuzzing [Source code]. GitHub. Retrieved [2025], from https://github.com/ollseg/usb-device-fuzzing?tab=readme-ov-file

WithSecureLabs. (n.d.). drozer [Software]. GitHub. Retrieved [2025], from https://github.com/WithSecureLabs/drozer

Drake, J. J., Lanier, Z., Mulliner, C., Oliva Fora, P., Ridley, S. A., & Wicherski, G. (2014). Android Hacker’s Handbook. Wiley. ISBN 978-1-118-60864-7

GZ TECH LTD. (n.d.). Geozilla [Mobile app]. GZ TECH LTD. Parnithos, 9 Flat/Office A, Germasogeia, 4040, Limassol, Cyprus.