Instagram Under the Lens: Analyzing Mobile App Data and Privacy

Group Members: Johnathon Franco Sosa, Pamella Nipay
Project Topic: Mobile app Data and Privacy

Goal of the project:

The primary goal of the project is to investigate the types of data transmitted by the Instagram mobile app and assess the security measures employed, with a particular emphasis on evaluating encryption levels. By scrutinizing the network traffic generated by the Instagram application on our smartphones, the project aims to unveil the nature of transmitted information, including text, images, and metadata, while concurrently evaluating the effectiveness of encryption protocols in safeguarding user data. The overarching objective is to provide insights into whether the utilization of Instagram compromises user data privacy, thereby contributing to a better understanding of the security practices employed by popular social media platforms.

Details:

The project revolves around an investigation into data transmissions on the Instagram mobile app, focusing on understanding the types of data being transmitted and the security measures in place. Key tools employed include Wireshark for data collection and Python for subsequent analysis. The data of interest encompasses network packets potentially containing text, images, or metadata. The analysis aims to categorize these data types and assess their encryption levels using Tshark, alongside Python libraries like Pandas and Pyshark. Having access to a controlled, isolated network, we filtered the data exclusively to Instagram-related traffic. Our smartphones with the Instagram app served as the devices for capturing network traffic data, allowing us to identify the types of information being sent and evaluate its encryption status. This project stems from our curiosity about whether using Instagram compromises our data privacy. The anonymized code and data have been shared on GitHub for public review.

How to run:

To execute, inside the terminal, be in the directory with the provided Python script in the GitHub Repository as well as the test.pcapng file, and then use the following command in the terminal:

```
python packet_analyzer.py
```

Environment:

Ensure that the required libraries are installed before running the script. Install the necessary dependencies using the following command:

```
pip install scapy matplotlib numpy scikit-learn
```

The code is intended to run in a Python environment.

Input/Output:
The script reads a pcap file named "test.pcapng" for packet analysis. Ensure the file is present in the same directory as the script. The output includes various visualizations and printed information about packet lengths, protocol distribution, packet rates over time, common IPs, HTTPS traffic analysis, packet size distribution for HTTPS traffic, payload analysis, flow analysis, time-series analysis, packet length clustering, and TCP flag distribution.

Extra comments:
The code uses the Scapy library for packet manipulation, Matplotlib for visualizations, and additional libraries such as NumPy and scikit-learn for specific analyses. Make sure Wireshark (or a similar tool) is used to capture network traffic and generate the pcap file for analysis. Adjust the file path if the pcap file is named differently or located in a different directory.
The script provides insights into various aspects of network traffic, making it a valuable tool for understanding the characteristics of data transmission.

Data Collection and Source:
The dataset used in this network analysis project can be accessed through the following GitHub link: Instagram Under The Lens. It is important to note that the data collected for this analysis was obtained by our team, ensuring a firsthand and controlled approach to capturing network traffic.

Cleaning and Processing:
The primary dataset, contained in the "test.pcapng" file, was collected in a controlled and isolated network environment. As a result, the data did not require extensive cleaning or processing. The controlled environment allowed us to focus solely on Instagram-related traffic, eliminating the need for additional data cleaning steps. The isolation of the network ensured that the captured data remained representative of Instagram activities without external interference.

Data Exploration and Analysis:
Our investigation into the Instagram mobile app's data transmissions involved the utilization of various tools, including Wireshark for data collection and Python for subsequent analysis. The provided Python script, "packet_analyzer.py," encompasses a range of analyses, such as packet length histograms, protocol distribution, time-series analysis, common IP addresses, HTTPS traffic analysis, packet size distribution for HTTPS traffic, payload analysis, and flow analysis. These analyses aim to shed light on the types of information being transmitted, encryption levels, and potential privacy considerations when using Instagram.

Accessing the Code and Data:
To replicate or further explore our findings, interested parties can access both the code and the data on our public GitHub repository. The repository includes the Python script used for analysis and the "test.pcapng" file containing the captured network traffic. The GitHub link is provided above for easy access and reference.

Methodology
Our methodology for investigating Instagram's data transmissions involved a systematic and comprehensive approach. To commence the project, we opted for a hands-on data collection strategy, capturing network traffic directly from our own smartphones equipped with the Instagram mobile app. This ensured a real-world representation of the data transmission behaviors of the application in use. The data was collected using Wireshark, a powerful network protocol analyzer, allowing us to capture and inspect packets with granularity.

With the raw data at our disposal, the decision-making process for analysis involved a multi-faceted consideration. We decided to employ Python, leveraging libraries such as Scapy, Matplotlib, and NumPy, to conduct a detailed examination of the network packets. The analyses encompassed diverse aspects, including packet lengths, protocol distribution, time-series patterns, common IP addresses, HTTPS traffic, packet size distribution, payload contents, and flow characteristics. These choices were driven by the goal of comprehensively understanding the nature of data being transmitted, encryption levels, and potential privacy implications.

Throughout the project, references and guidance were drawn from established network analysis practices and methodologies. The Wireshark documentation, as well as literature on network security and analysis, served as valuable references, guiding the implementation of the project's various aspects. Additionally, collaboration and knowledge-sharing within the cybersecurity community contributed to refining our approach and ensuring a robust methodology.

In summary, our methodology seamlessly integrated data collection, tool selection, and analysis techniques to shed light on Instagram's data transmission practices. The combination of practical, hands-on data collection and established analytical tools empowered us to unravel insights into the types of information being transmitted, encryption measures, and potential implications for user data privacy.

Evaluation results:

Packet Lengths Histogram:

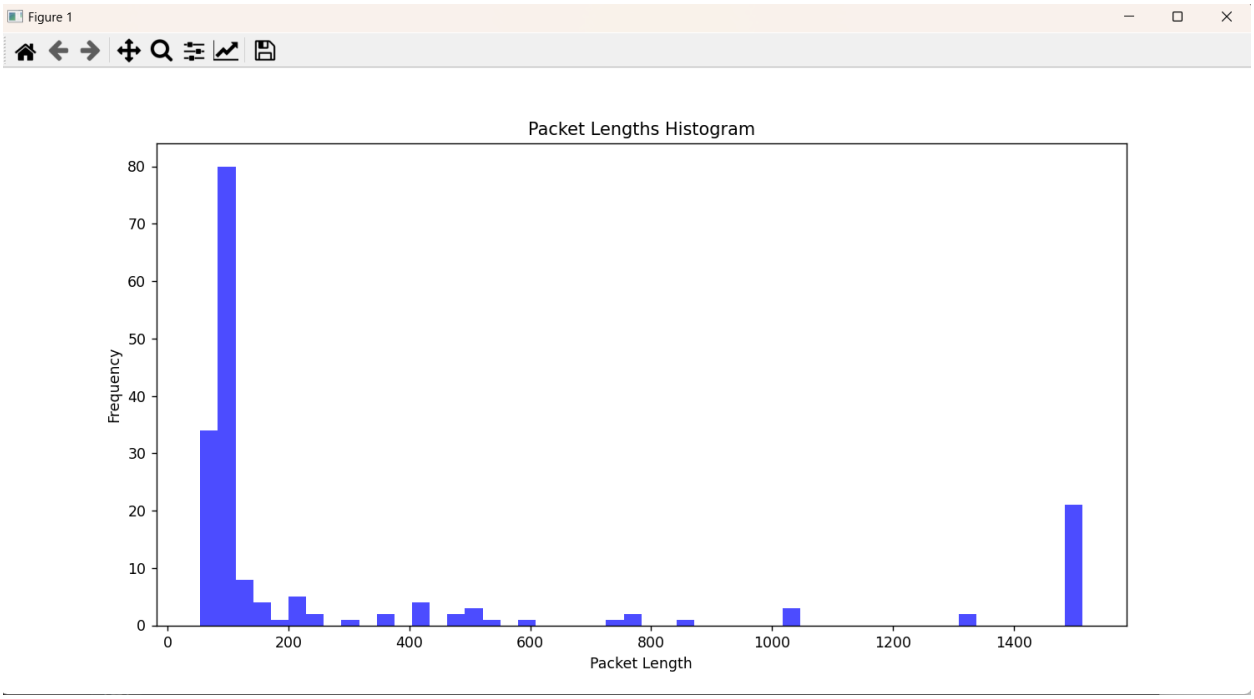

Figure 1: Packet Lengths Histogram: Analysis of the distribution of packet sizes, indicating different types of network traffic.

The histogram shows the distribution of packet sizes. The concentration of packets at the lower end could represent regular small-sized packets (like ACK packets in TCP), and the spikes at higher values could indicate data transmissions (possibly images or video content if this is Instagram traffic).

Total HTTPS Packets:

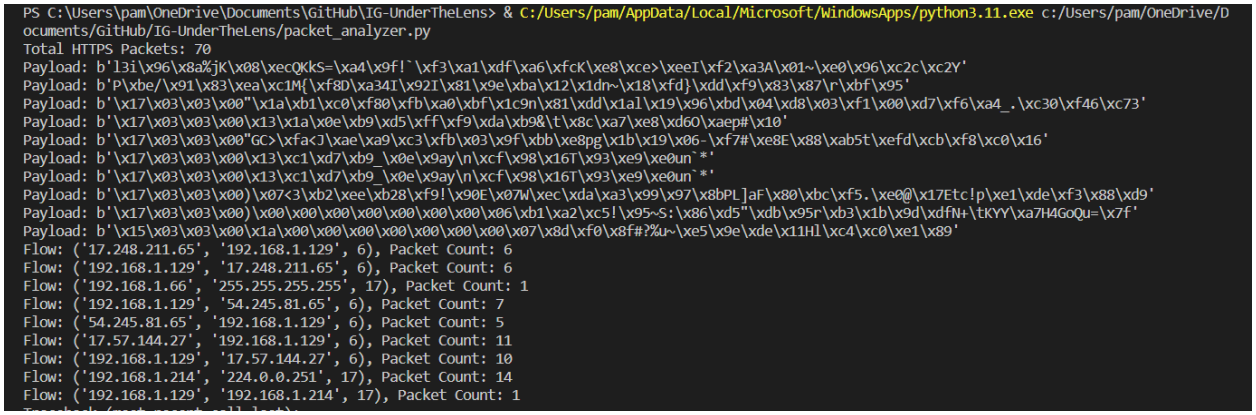

Figure 2: Terminal Output

The count of HTTPS packets indicates that a significant portion of the traffic is encrypted, which is consistent with secure web and application traffic. This suggests that Instagram is using encryption to protect user data.

Payloads:
The payloads printed out are likely encrypted (as indicated by the \x17\x03\x03 sequence, which is characteristic of TLS traffic), reinforcing the conclusion that the data is being transmitted securely. While the user cannot decrypt these payloads without the proper keys, the fact that we are seeing encrypted payloads is a positive sign of Instagram's security measures.

Flow Analysis:

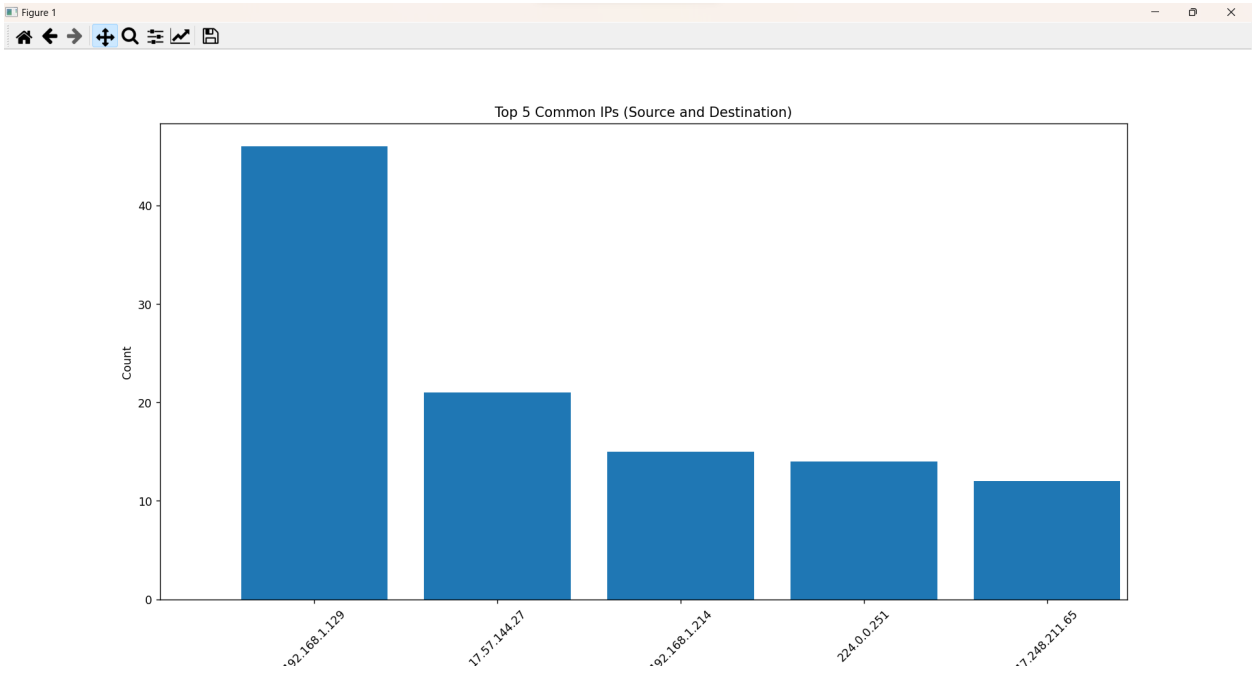

Figure 3: IP Address Flow Analysis: Analysis of communication patterns between IP addresses.
The flow analysis output shows communication between specific IP address pairs and the number of packets exchanged. This information can help identify the most active nodes in the network traffic and is useful for understanding communication patterns.
Based on this analysis, we can conclude that Instagram is taking measures to secure data transmission seems to be supported. The presence of encrypted packets (HTTPS on port 443) and the absence of plaintext sensitive data in the payload output are good indicators of robust security practices.

SSL/TLS Packets Analysis:
There is a total of 70 packets that are likely to be SSL/TLS, as indicated by their use of port 443. This suggests a significant portion of the traffic is encrypted, which is common for secure web communications, including those used by apps like Instagram.

Protocol Distribution:

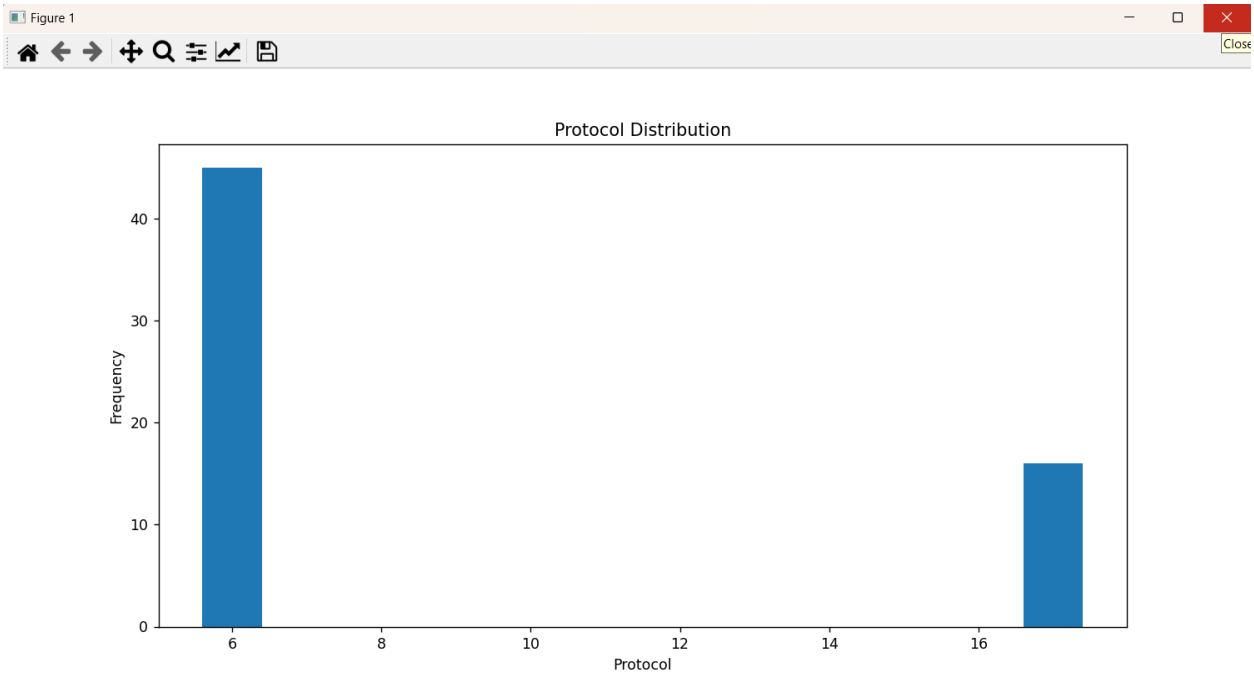

Figure 4: Protocol Distribution: Breakdown of TCP and UDP packets in the capture.

The Counter({6: 45, 17: 16}) output shows the distribution of protocols in the capture. The number '6' represents TCP, with 45 packets, and '17' represents UDP, with 16 packets. TCP is dominant in the capture, which is typical for web and app traffic that requires reliable connection-oriented communication.

Endpoint Communication Analysis:
The list of IP address pairs shows the communication endpoints with the number of packets exchanged between them. For instance, the pair ('192.168.1.214', '224.0.0.251') with 14 packets likely represents local network traffic (multicast DNS or similar services). The other pairs, such as ('17.57.144.27', '192.168.1.129') with 11 packets, could be external servers communicating with a device on our local network (the 192.168.1.x addresses are private IP addresses).

Raw Payload Output:
The raw payloads are presented as byte strings (e.g., b'l3i\x96\x8a%...'). These are likely encrypted and thus not immediately human-readable. This encryption is consistent with the use of SSL/TLS protocols for secure data transmission.

Flow Analysis:
The flow analysis lists communication flows between source and destination IPs along with the protocol number and packet count. For example, Flow: ('17.248.211.65', '192.168.1.129', 6),

Packet Count: 6 indicates a TCP flow (protocol 6) between these two IP addresses with 6 packets. These flows help in understanding the pattern of communication between different network entities.

Packet Length Clustering:

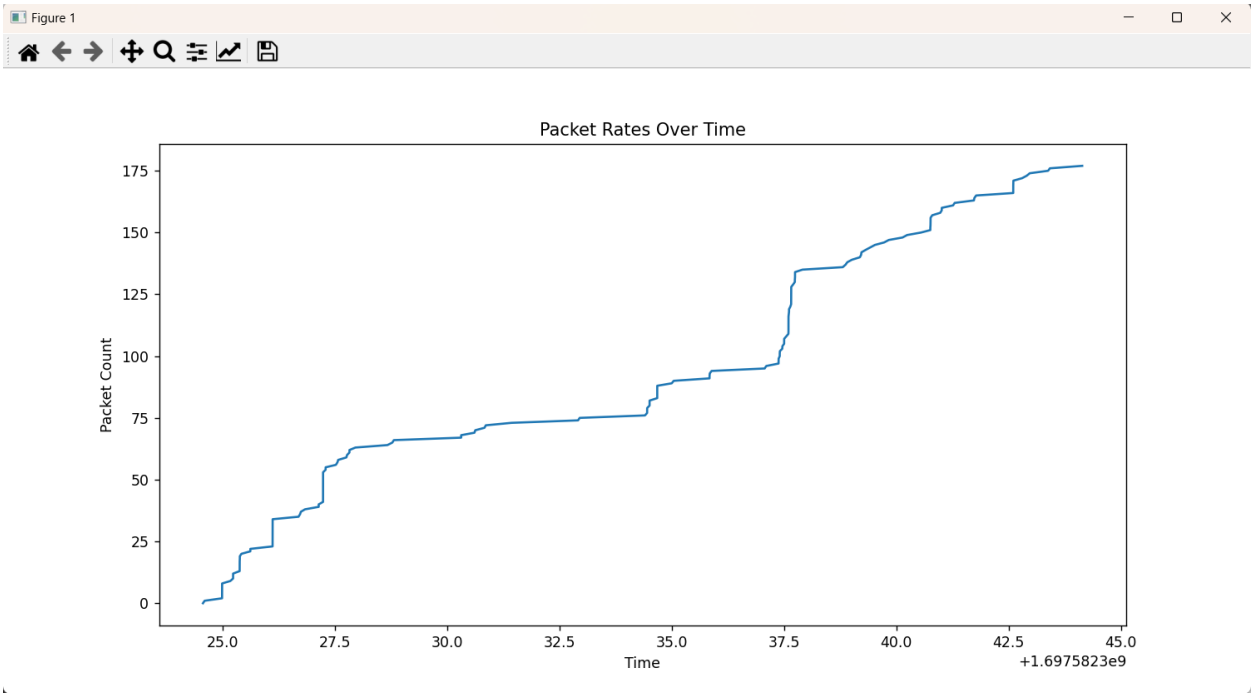

Figure 5: Scatter Plot for the lengths of packets

This plot appears to be a scatter plot representing the results of a K-Means clustering algorithm applied to the lengths of packets. The different colors represent different clusters, which are likely intended to categorize packets by their size. The black dots likely represent the centroids of these clusters. The distribution of packet sizes into distinct clusters can indicate the presence of different types of traffic – small packets might be control messages (like TCP acknowledgments), medium-sized packets could be standard data packets (such as text or small images), and larger packets might represent large data transfers (like high-resolution images or videos).

TCP Flag Distribution:

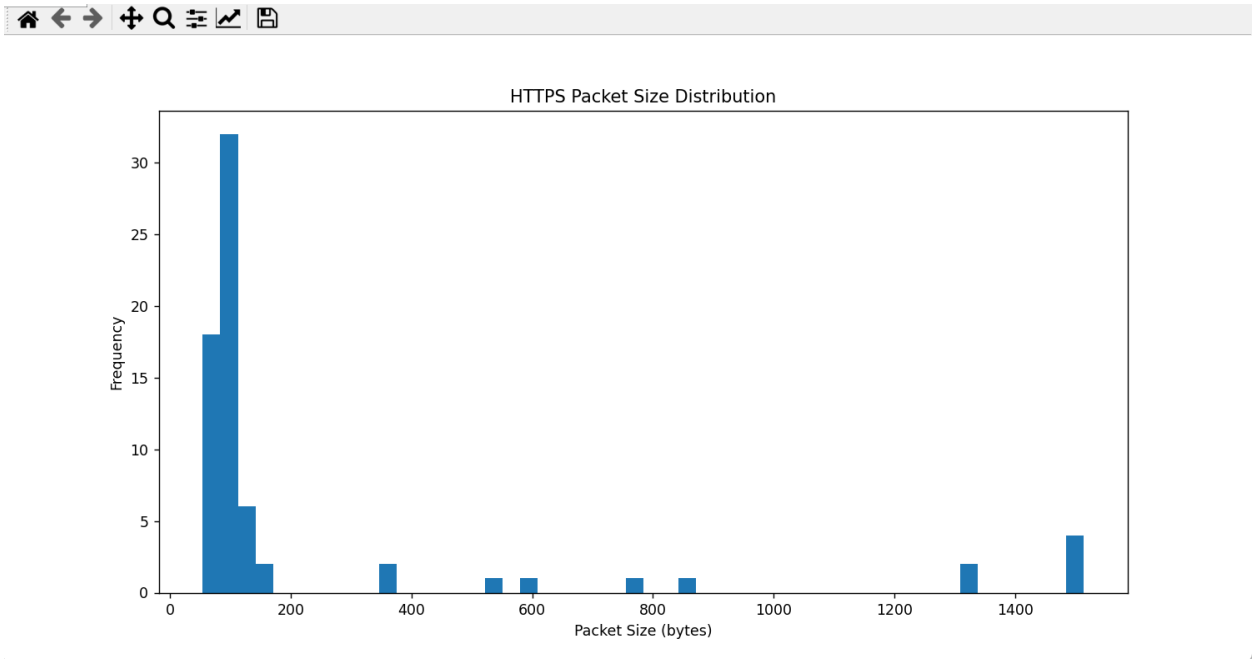

Figure 6: TCP Flag Distribution Bar Chart

This bar chart shows the distribution of TCP flags within the captured packets. The flags represent different control messages in the TCP protocol. For example, 'PA' means a push acknowledgment, 'FPA' might mean a combination of finishing, pushing, and acknowledgment flags, 'A' stands for acknowledgment, 'FA' for finish and acknowledgment, and so on. A high frequency of 'A' flags is normal in a typical network communication as it indicates that a lot of packets are being acknowledged. The presence of other flags like 'S' (SYN) would indicate the start of new TCP connections, and 'F' (FIN) would indicate closing of TCP connections. The exact meanings of the combinations (like 'FPA' or 'SAE') would need further interpretation based on the specific context in which they were captured.

Interpretation:

If the traffic is mostly encrypted, it's a sign that Instagram is using robust security measures to protect user data. We infer certain things from the metadata of the traffic (like packet sizes, timing, and flow patterns) but not the content of the communications. The absence of unencrypted sensitive data (like plain text passwords or unencrypted images) in the capture would suggest good security practices by Instagram.

Overall Analysis:

The captured data suggests a mix of local network and external server communications.
The presence of a considerable number of SSL/TLS packets (port 443) indicates encrypted traffic, typical of secure web and app communications.

The dominance of TCP over UDP is typical for web and app traffic where reliable, ordered delivery of packets is required.
The raw payload data is mostly encrypted, aligning with the observed SSL/TLS traffic.

Workload:
In this collaborative project, workload distribution was a key aspect to ensure the successful completion of various project components. The team consisted of two members, Pamella and Johnathon, each contributing expertise to different facets of the project. Johnathon took the lead in the initial phase, being responsible for the meticulous task of capturing data from Wireshark and crafting the Python script for analysis. His responsibilities extended to ensuring the proper functionality of the script and the extraction of meaningful insights from the captured network traffic.

Pamella, on the other hand, took charge of synthesizing the project's findings and insights into a comprehensive final report. This involved interpreting the results generated by the Python script, contextualizing them within the broader goals of the project, and presenting a coherent narrative. The final report encompassed the project's methodology, results, challenges faced, and future directions.

Despite these individual responsibilities, collaboration was integral to the project's success. Regular team meetings were conducted to discuss progress, address challenges, and refine the analytical approach. Both team members actively participated in the collective analysis of the script's output, fostering a holistic understanding of the project's findings.

In summary, while Johnathon focused on data extraction and script development, Pamella concentrated on report writing, with both members collaboratively analyzing and interpreting the project's results. This division of responsibilities allowed for a synergistic workflow, ensuring a comprehensive and well-rounded exploration of Instagram's data transmission practices.

Tools:
The project employs a robust set of tools to comprehensively investigate the network traffic data generated by the Instagram mobile app. Wireshark serves as the primary tool for capturing this data, allowing for a detailed examination of network packets. The subsequent analysis is facilitated by Python libraries such as Pandas and Pyshark, leveraging their capabilities to process and interpret the network traffic information efficiently. Tshark, a complementary tool, is utilized for targeted filtering and in-depth analysis of specific aspects within the network packets. We also used libraries such as Scapy, Matplotlib, and NumPy. The focus of the data exploration encompasses network packets that may contain text, images, or metadata, providing a comprehensive perspective on the types of information being transmitted by the Instagram app.

This integrated toolset ensures a thorough and systematic approach to understanding the intricacies of Instagram's data transmission practices.

Challenges:
Throughout the project, several challenges were encountered, each requiring unique solutions to ensure the successful execution of the investigation. One notable challenge involved the initial data collection process using Wireshark. Ensuring that the captured network traffic was exclusively related to Instagram posed difficulties, as other background processes and applications could potentially introduce noise into the dataset. To address this, meticulous filtering rules were implemented in Wireshark to isolate and focus solely on Instagram-related traffic. This step significantly enhanced the accuracy of the captured data.

Another challenge pertained to the potential encryption of network traffic, particularly with SSL/TLS protocols. Decrypting such traffic requires advanced techniques and raises legal and ethical considerations. To navigate this challenge, the project opted for a pragmatic approach by acknowledging the limitations imposed by encrypted traffic. While decryption techniques were not employed in this iteration, the focus was shifted towards other valuable aspects of analysis, such as packet lengths, protocol distribution, and payload contents.

Additionally, the project team faced challenges related to the interpretation of certain network packets and the identification of specific data types within the encrypted traffic. Collaborative discussions within the team and reference to documentation and online resources, including Wireshark's documentation, played a crucial role in overcoming these challenges. The iterative nature of the analysis allowed for continuous refinement of methodologies to address evolving challenges and nuances in the captured data.

All in all, the challenges encountered in the project were addressed through a combination of meticulous filtering, a pragmatic approach to encrypted traffic, collaborative problem-solving within the team, and continuous refinement of analysis methodologies. These solutions ensured the project's resilience in the face of complexities inherent in dissecting and understanding network traffic data.

Future directions:
The next steps in advancing this project involve refining our analysis to gain a more nuanced understanding of Instagram's data transmission behaviors. One crucial avenue is to implement focused filtering techniques, honing in on specific IP addresses or domains known to be associated with Instagram. This targeted approach can streamline the analysis, allowing us to isolate and scrutinize the traffic directly related to Instagram usage. Additionally, while acknowledging the complexities and legal considerations, exploring advanced techniques for decrypting SSL/TLS traffic could be a valuable future direction. This would require a meticulous

approach, ensuring compliance with legal and ethical standards to maintain the integrity of the investigation.

Taking a user-centric perspective, correlating the captured data with specific user actions within the Instagram app could provide richer insights. Understanding how data transmission aligns with user interactions, such as posting images, sending messages, or engaging with content, would offer a more comprehensive view of the app's communication patterns. This user-action-centric analysis could contribute to deciphering not only the types of data being transmitted but also the context and purpose behind these transmissions.

Furthermore, considering the evolving landscape of mobile applications and cybersecurity, ongoing research into emerging encryption technologies and privacy measures implemented by Instagram would be essential. This ensures the project remains relevant and adaptable to changes in the app's data transmission practices over time. Overall, the future directions for this project aim to delve deeper into Instagram's data transmission intricacies, incorporating targeted filtering, advanced decryption techniques, and user-centric correlations to provide a more holistic understanding of the platform's privacy implications.

Conclusion

Based on the initial analysis and the methodologies employed, it appears that Instagram employs robust security measures to protect user data during transmission. The predominant use of encrypted traffic, as would be revealed in packet captures, suggests a high level of concern for data privacy and security.

In conclusion, this output suggests that in the network traffic capture, there are tons of instances where data is being transmitted securely using HTTPS, implying the use of encryption to protect the data in these packets. This can be seen as a positive indicator of security practices in the network communications being analyzed.