Project Report

SSL Vulnerabilities Attack Lab

CIS-316 Fall 2023

**Taylor Williams & Zach Warsaw**

December 10th, 2023

# 

# 

# **INTRODUCTION**

In this current age, many are dominated by the relentless surge of mobile applications, and ensuring the security and integrity of these digital tools has become increasingly more challenging. Our team has conducted a project lab that delves deep into the world of mobile applications, primarily focusing on identifying and analyzing flaws that compromise user data and overall security. The objective of this project was clear: conduct experiments to unearth flawed apps set up proxies to monitor HTTPS packages, identify sensitive information within data packages, and ultimately distinguish flawed or malicious applications.

The ubiquity of mobile applications in our daily lives makes them susceptible targets for exploitation, raising concerns about user privacy and data protection. Through this project, we aimed to equip ourselves with the skills necessary to assess the vulnerabilities inherent in these applications, paving the way for a more secure digital landscape.

This report details our findings through the web of mobile application security which consisted of the testing of 10 different mobile applications in the categories of shopping, business, and finance. Only one mobile application tested resulted in a flawed security implementation where user data became exposed through the proxy server and compromised.

**TEST ENVIRONMENTS**

**Taylor Williams**

* **Proxy Device A**: A ROG Zephyrus G14 laptop, model GA401QM\_GA401QM with an AMD Ryzen 9 5900HS with Radeon Graphics processor, 16 GB of RAM with Windows 11 Home with Ubuntu Virtual Machine
* **Test Device A**: An iPhone XR, software version: iOS 16.3.1, model number: MT302LL/A

**Zach Warsaw**

* **Proxy Device B:** ASUS TUF Dash F15 laptop, with Ubuntu Virtual Machine
* **Test Device B:** Samsung Galaxy Z Fold 4

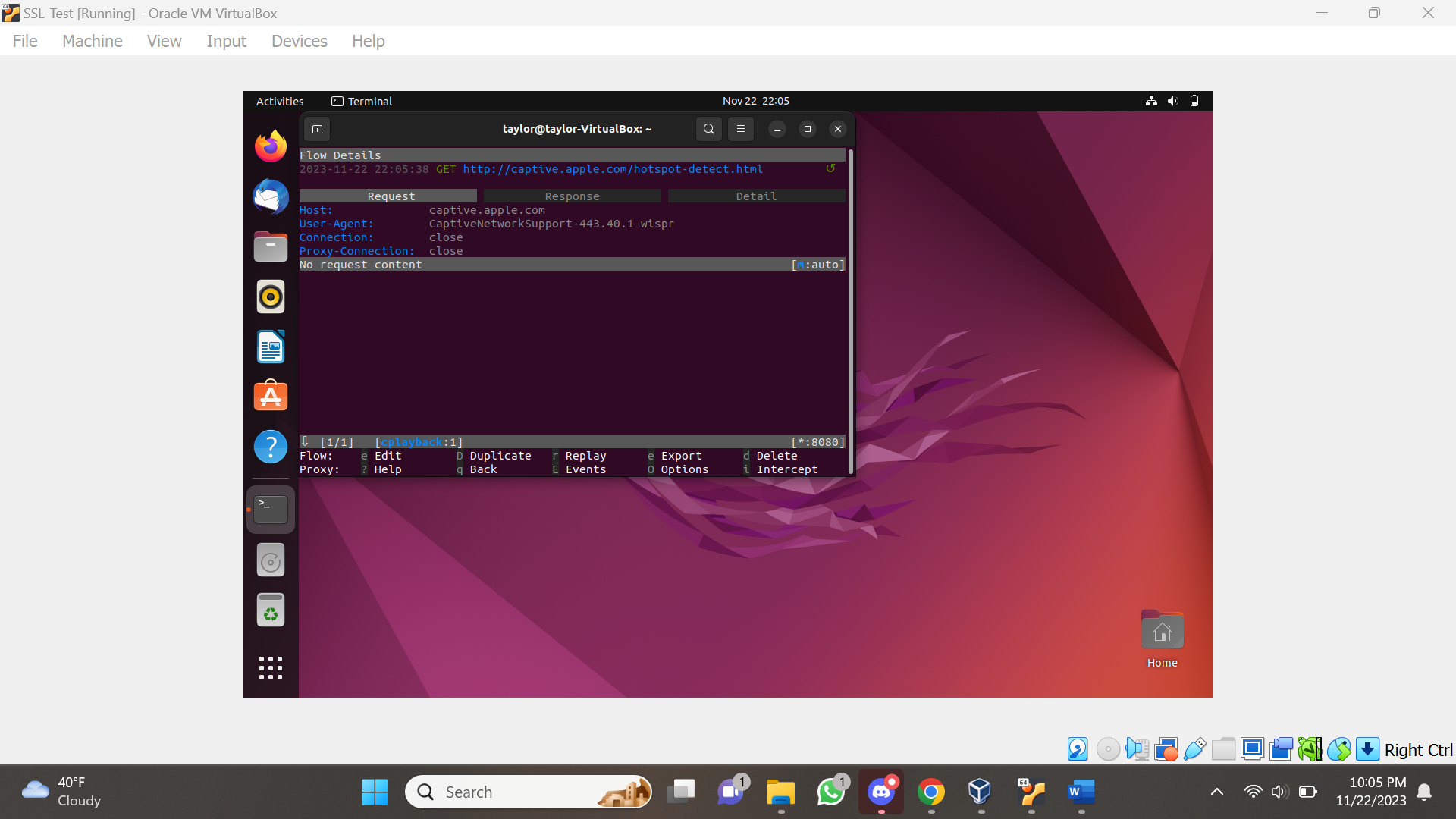
# 

# **FINDINGS - APP SUMMARY TABLE**

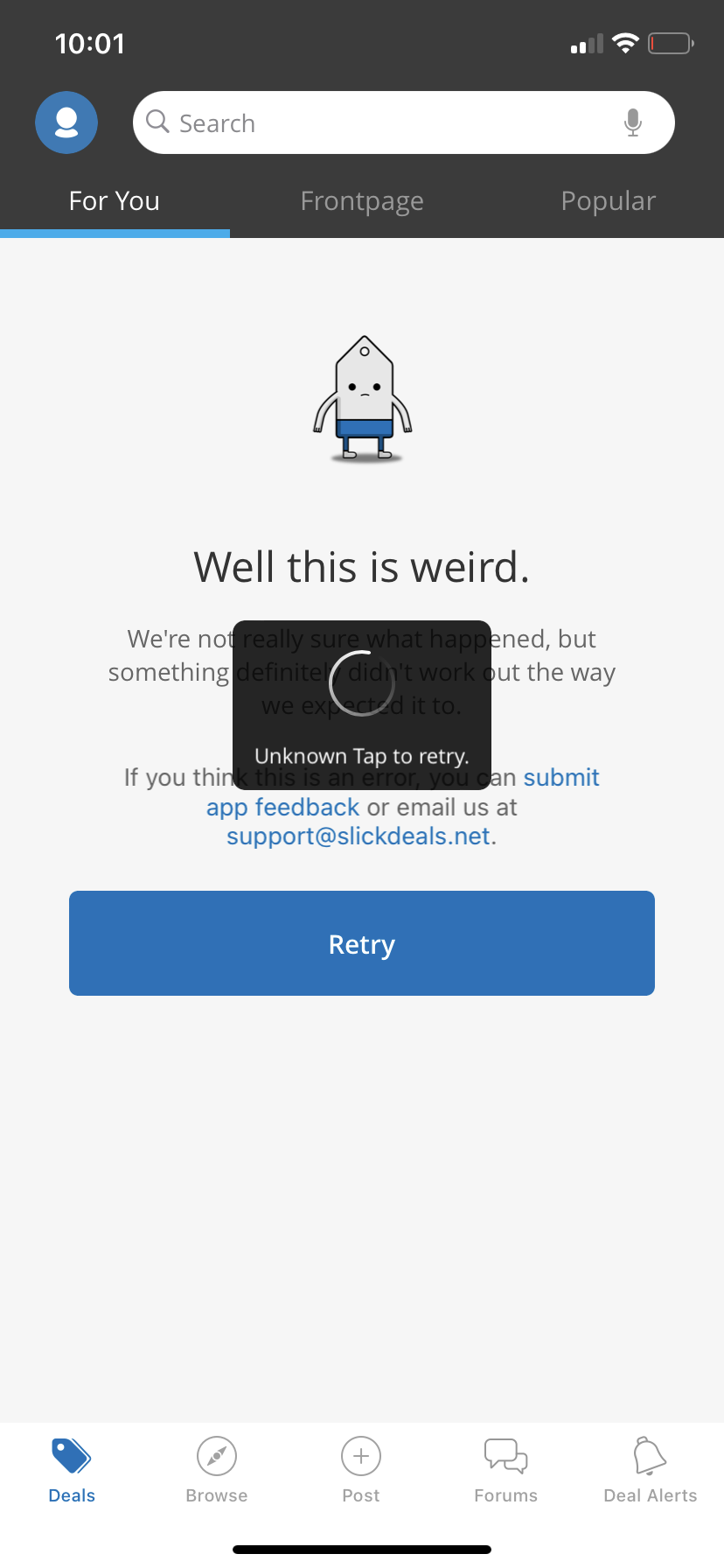
| **Platforms: iPhone/Android** | | | |
| --- | --- | --- | --- |
| **App #** | **App Name** | **Version** | **Comments** |
| 1 - **iPhone XR (iOS)** | Slickdeals: Deals & Discounts | 5.37.6 | * Connection error message * Connection timed out * Secure |
| 2 - **iPhone XR (iOS)** | Carrot - Save, Shop, Deal Hop | 1.0.38 | * Error Message displayed * Connection timed out * Blocked out password for screenshots * Secure |
| 3 - **iPhone XR (iOS)** | FreeStuffFinder - Save Money | 6.0.8 | * Gave invalid certificate error message |
| 4 - **Samsung Galaxy (Android)** | HelloSkip | 2.28 | * Screen wouldn’t get passed the account curation page * No content was retrieved * Secure |
| 5 - **Samsung Galaxy (Android)** | Qapital | 4.86.3 | * No packets received. * Certificate wasn’t trusted. * Secure |
| 6 - **Samsung Galaxy (Android)** | Picsart | 23.7.7 | * Also no packets. * Lack of trust with certificate. * App had connection error * Secure |
| 7 - **Samsung Galaxy (Android)** | ASTRO File Manager | 8.13.3 | * Tons of packets * Stole my phone’s brand/model info * Ensured that I wasn’t a robot * UNSAFE |
| 8 - **iPhone XR (iOS)** | Wise | 8.38.2 | * Secure * Gave server connection error message * Would not go further into the app |
| 9 - **iPhone XR (iOS)** | Afterpay - Buy Now, Pay Later | 1.88.2 | * Secure * Would only constantly load on a loading screen, no access to the app itself * Gave certificate error within proxy * Lack of trust with the SSL Certificate. |
| 10 - **iPhone XR (iOS)** | Sezzle - Buy Now, Pay Later | 3.3.28 | * Secure * Did not display any private information * Gave certificate error within proxy * Lack of trust with the SSL Certificate. |

# 

# **RESULTS - SCREENSHOTS and DISCUSSIONS**

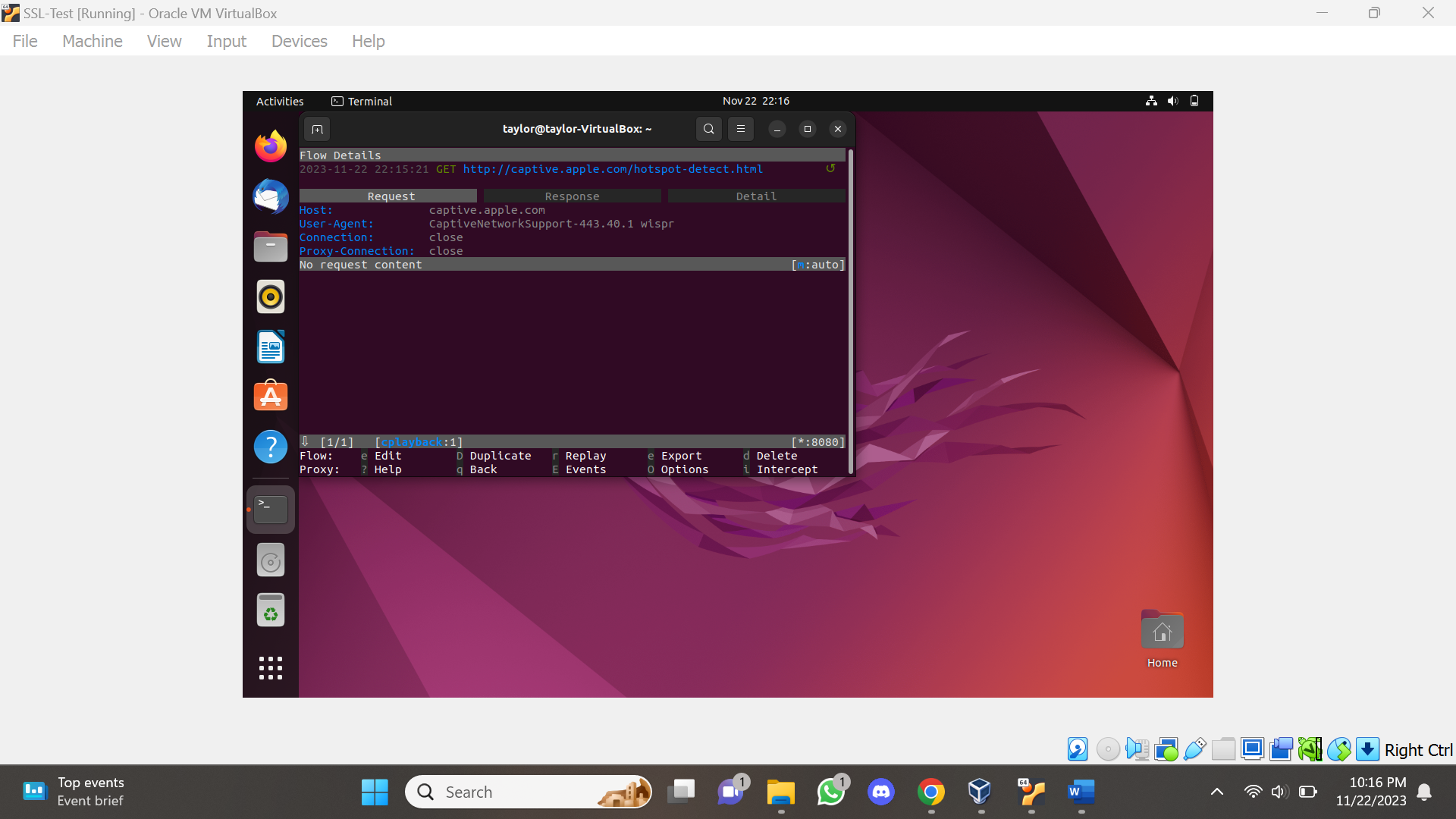
1. **SlickDeals**

* The proxy does detect the activity of opening the app but does not relay any confidential information.
* The app seems secure since no private information was given through the proxy server.
* No response data was given either, confirming this analysis as well.

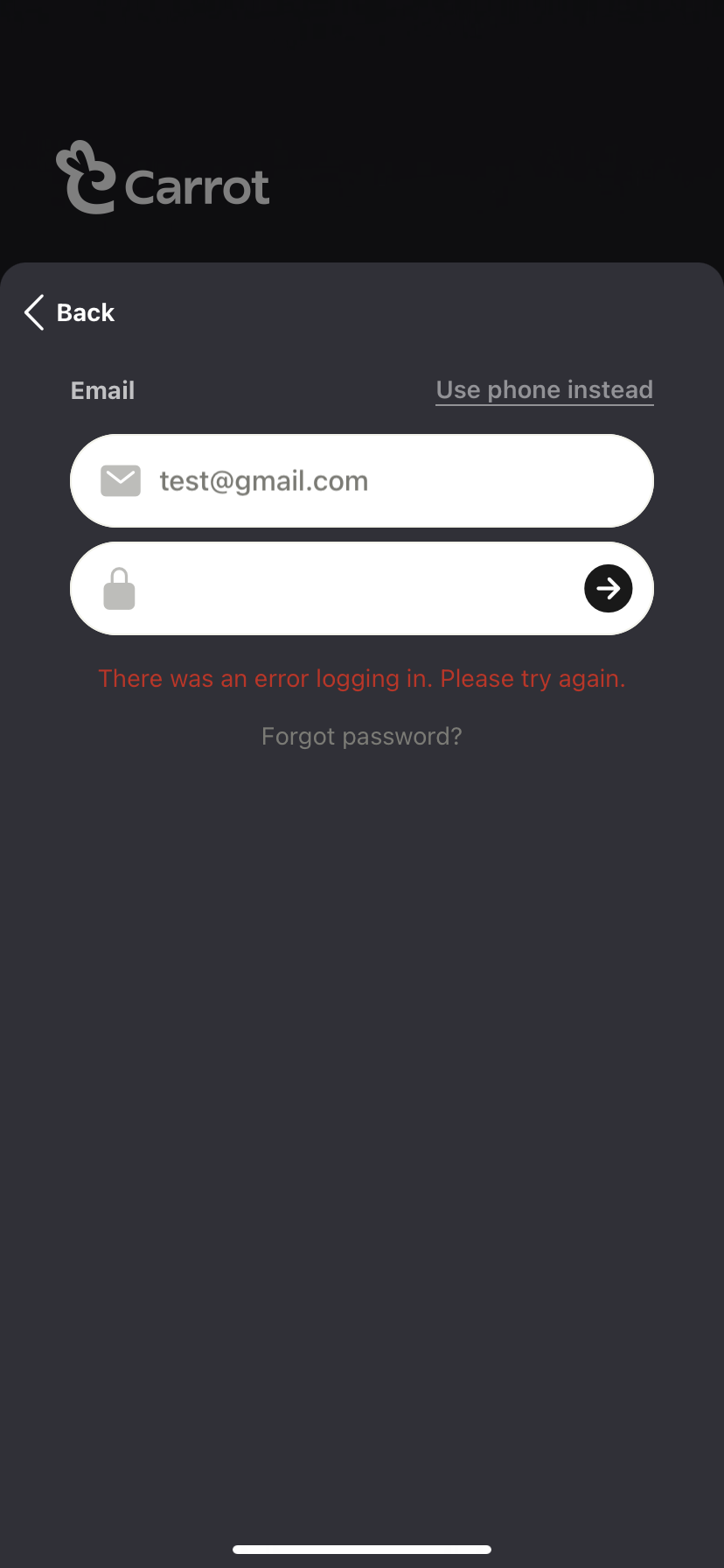


* When entering the app, an error message appeared, along with a constant loading pop-up that would not disappear.
* The app seems secure since no private information was given through the proxy server and the app only showed a connection error message.

1. **Carrot**

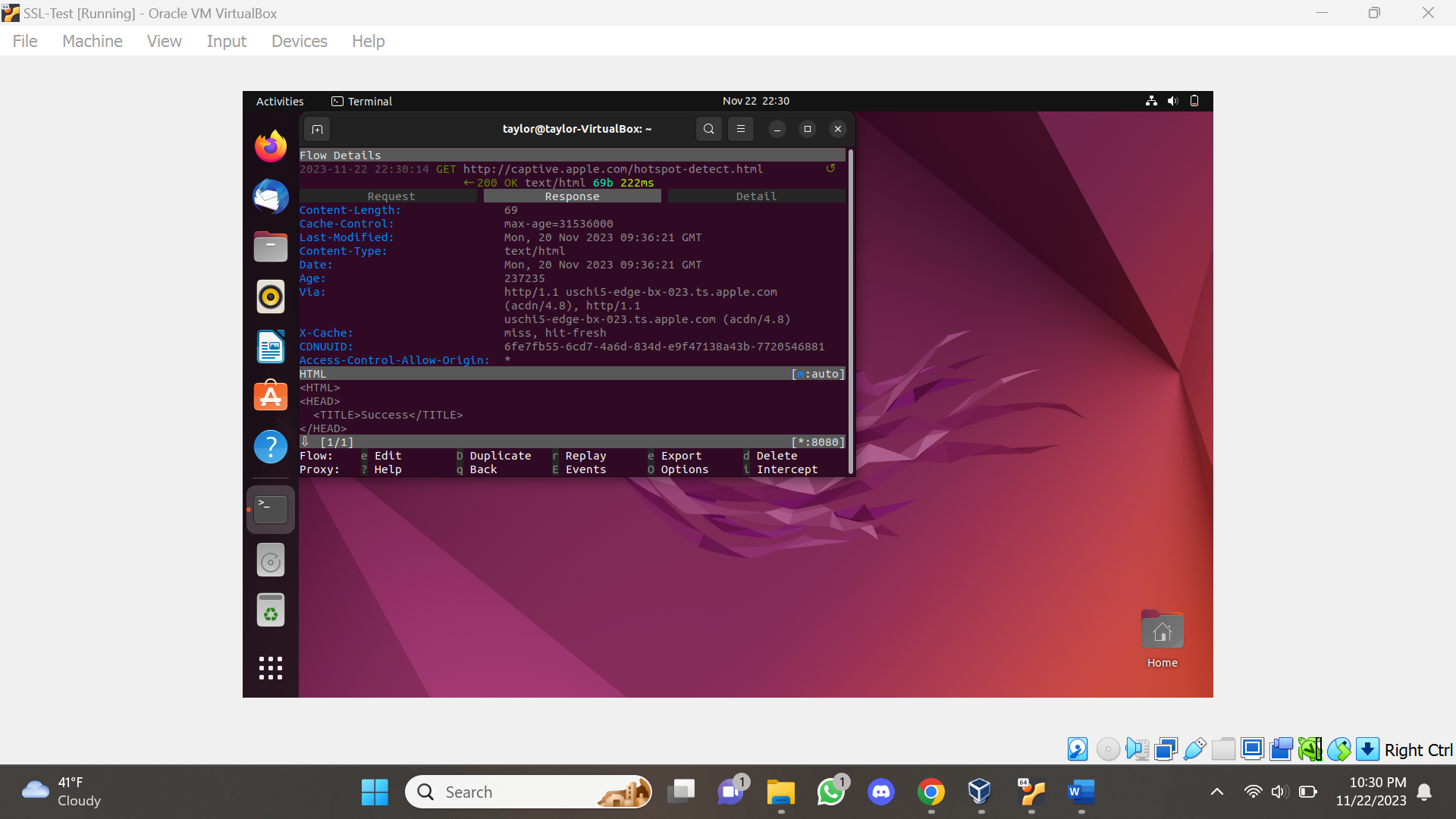


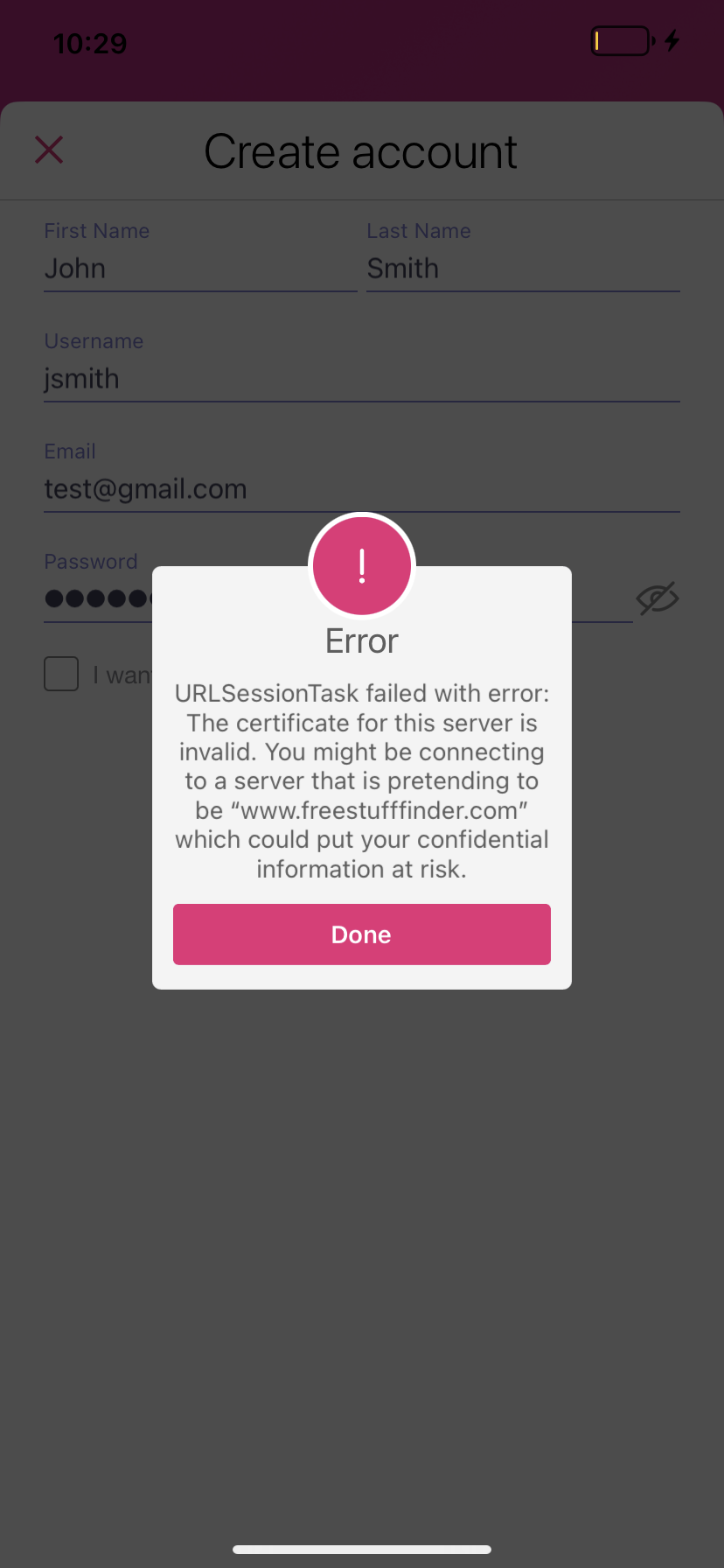
* Same as the previous app, no confidential information was given through the proxy server.
* The proxy does detect the activity of opening the app but does not relay any confidential information.
* No response data was given either, confirming this analysis as well.



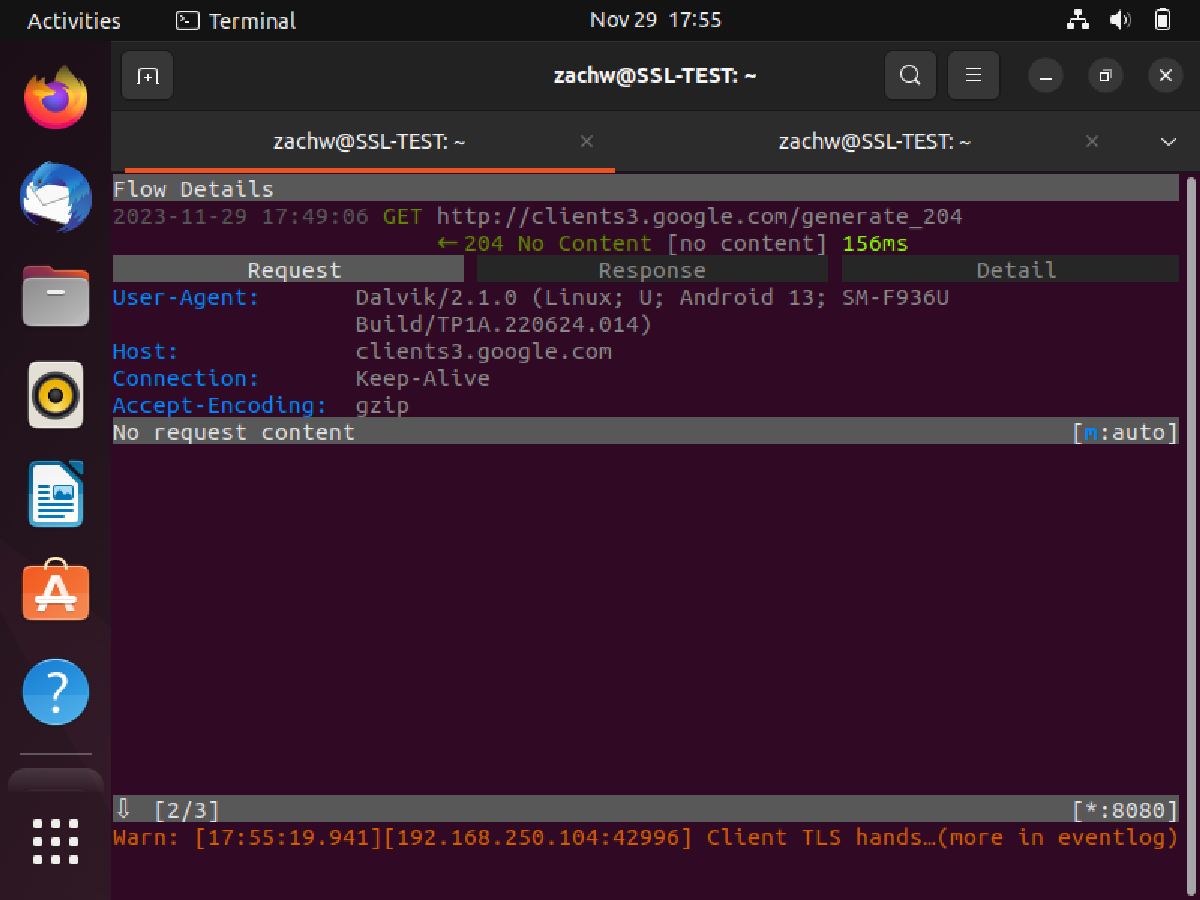
* When entering the app it was allowed to sign up via email and then enter a password, upon doing this none of the information was relayed to the proxy server.
* An error message does display upon pressing the continue arrow button, which can mean that this app is indeed secure.

1. **FreeStuffFinder**



* The proxy does detect the activity of opening the app but does not relay any confidential information.
* Some response data was given, unlike the previous apps, as shown in the screenshot.
* Upon entering the app, inputting confidential information was permitted, first name, last name, username, email, and password.
* When pressing the submit button, an error message was displayed, where it states that the URL session task failed indicating that the certificate for the server is invalid and might be connecting to a server pretending to be the app’s website “[www.freestufffinder.com](http://www.freestufffinder.com)” and would not let the process continue beyond that point.
* The app specifically warns the user that the connection is not secure, which is not something the previous apps have done. It is safe to assume that this app is secure.

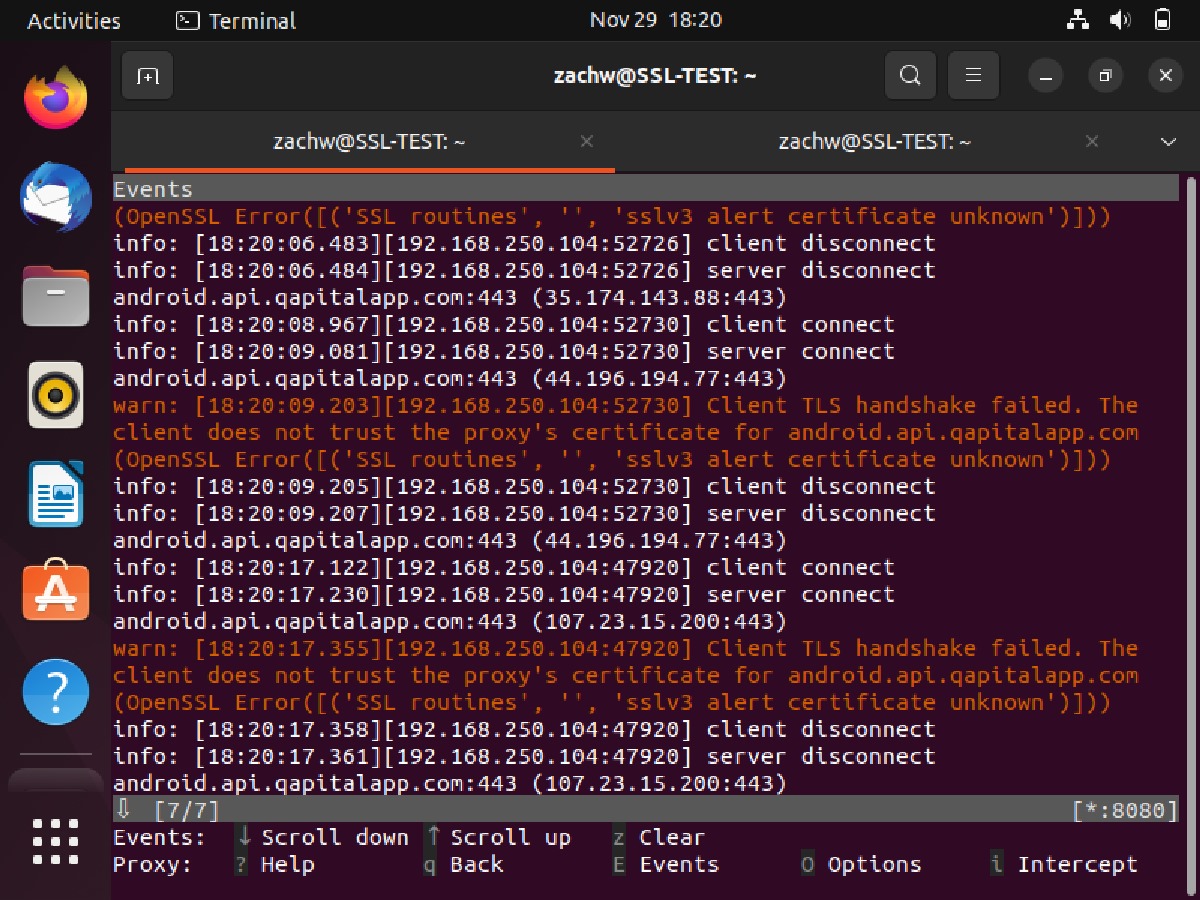
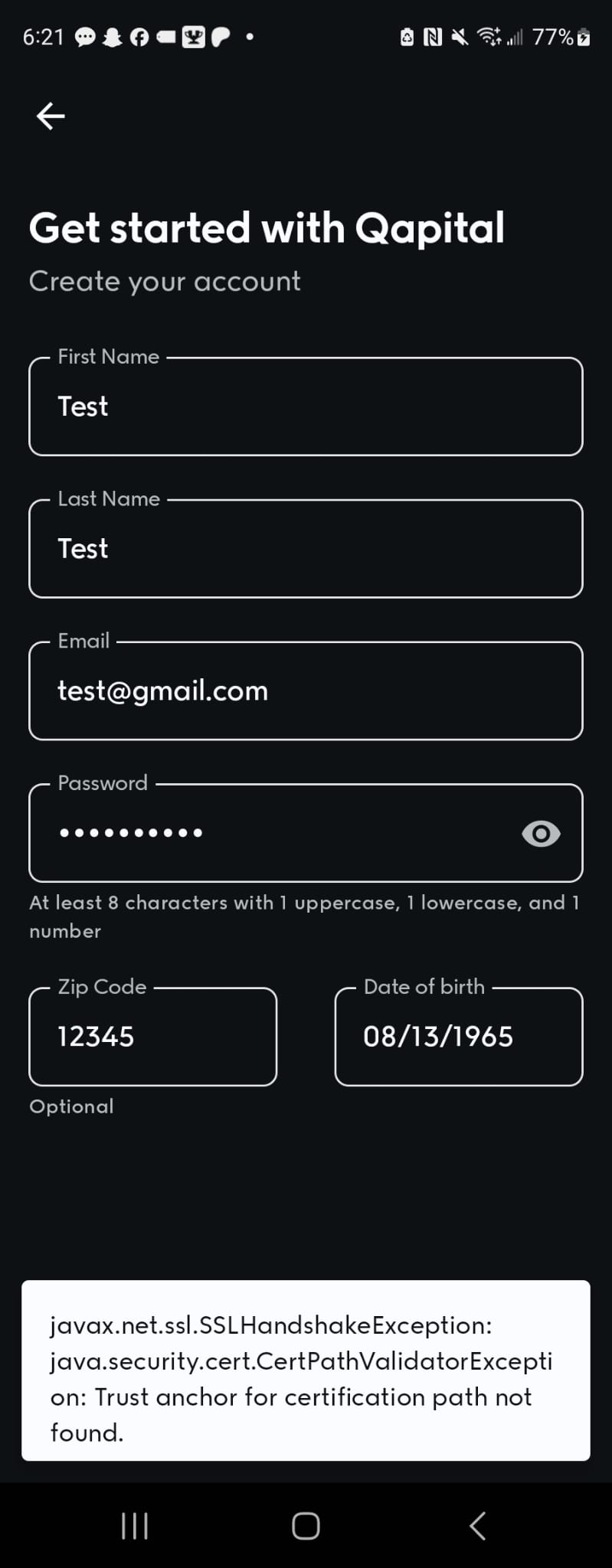
1. **HelloSkip**

****

* This proxy seemed to acknowledge the packets of data for when I pressed **submit** for creating a new account.
* While the app never was able to continue on from that account curation screen, the proxy wasn’t able to gather any sensitive information.
* App seems to be fairly secure with confirmation of TLS handshakes.



1. **Qapital**

****

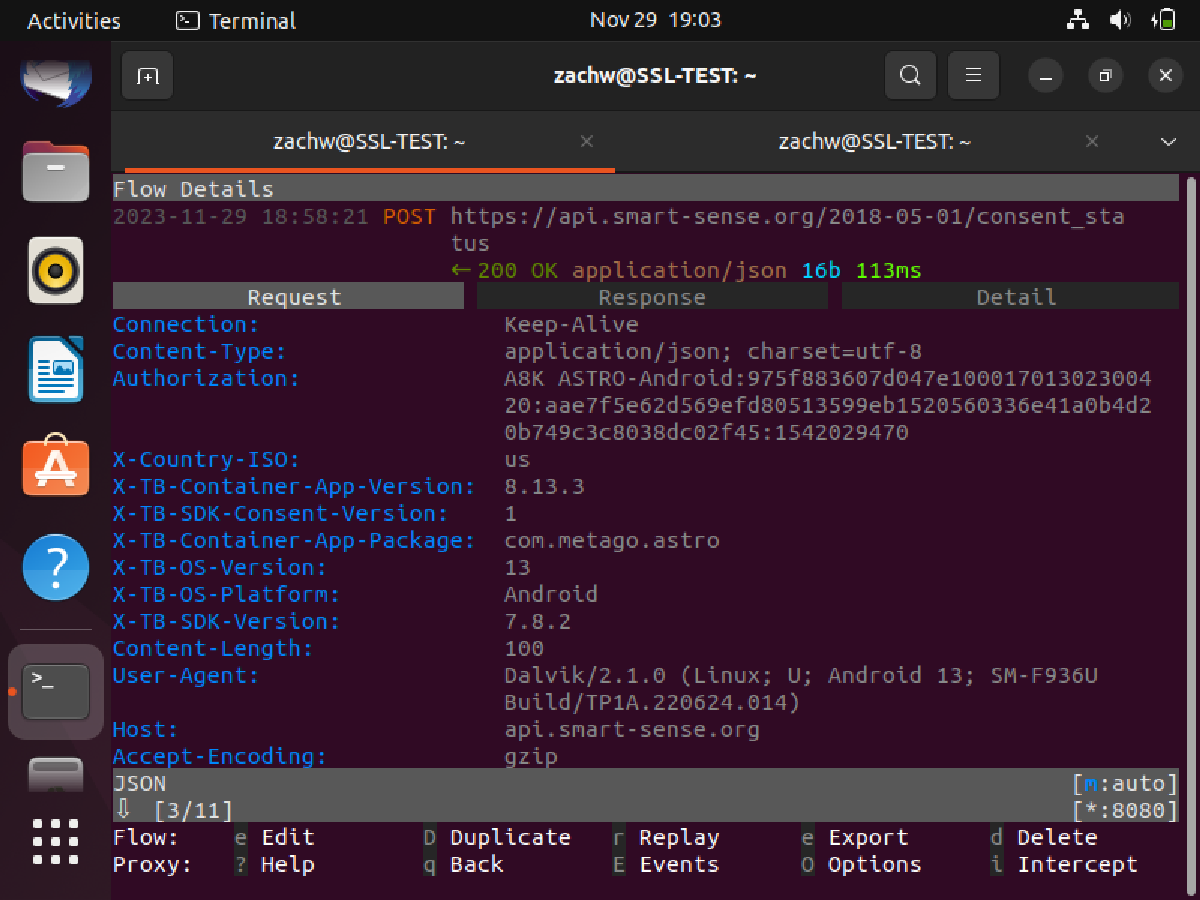
* This app never trusted the proxy’s certificate and so a packet was never received, but in the event logs you can clearly see that the TLS handshake failed when trying the request came through.
* The app also showed an error message detailing that there was a lack of trust in the SSL certificate.
* Definitely more secure than the last application.

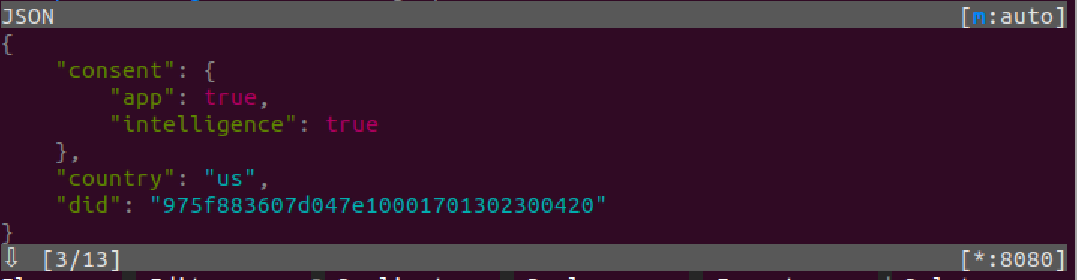
1. **Picsart**

****

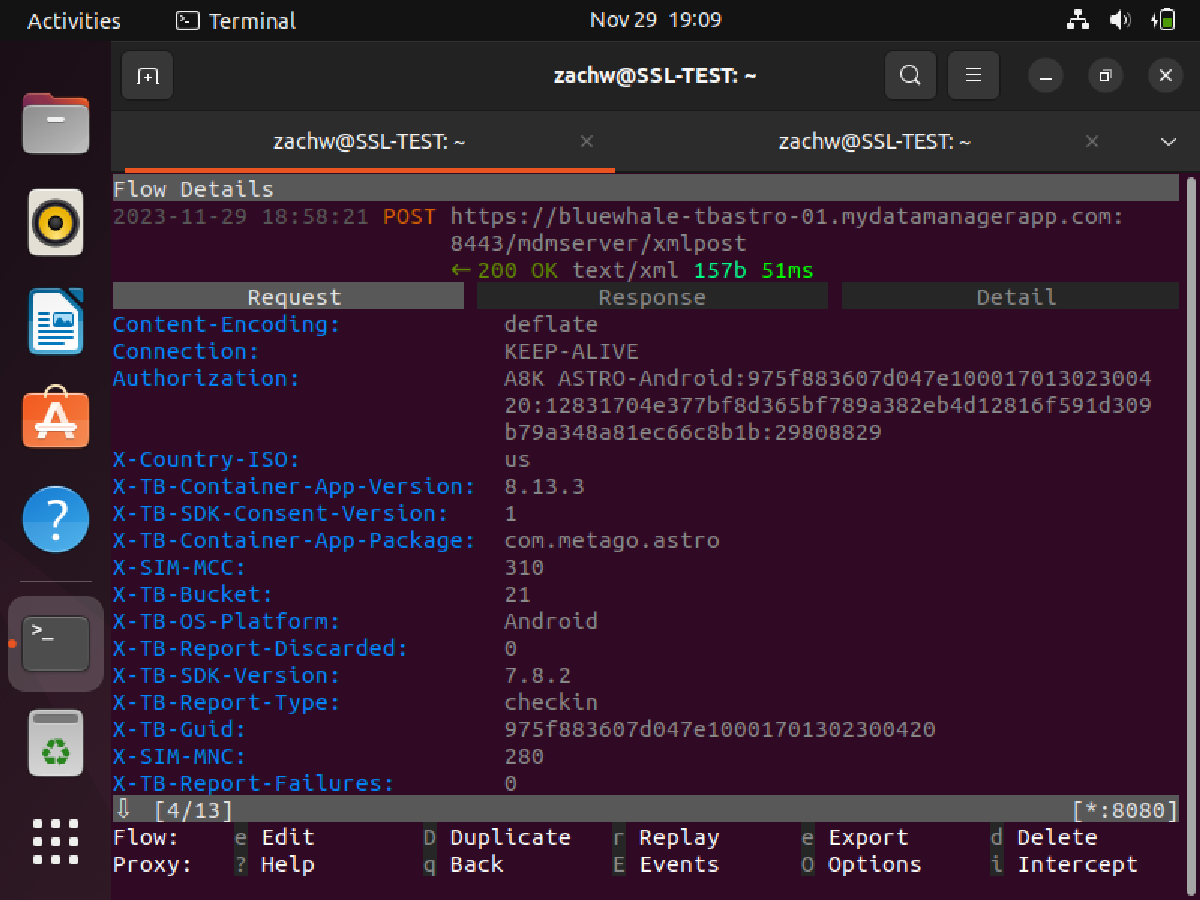
* The proxy wouldn’t get packets here either. When this application was known in the past to be a vulnerable app.
* Event logs show that the TLS handshake failed via the lack of trust with the certificate.
* App would just give a simple connection error.

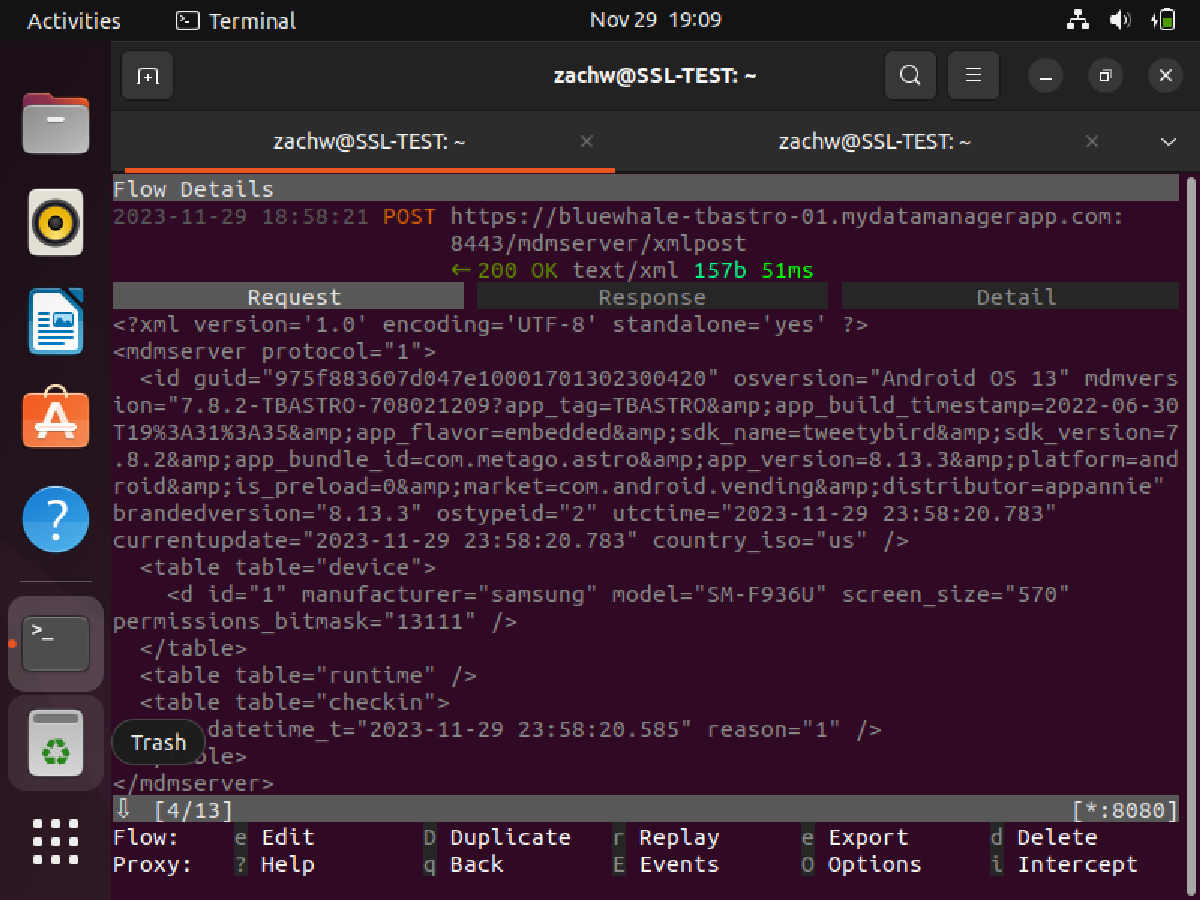
1. **ASTRO File Manager**

****

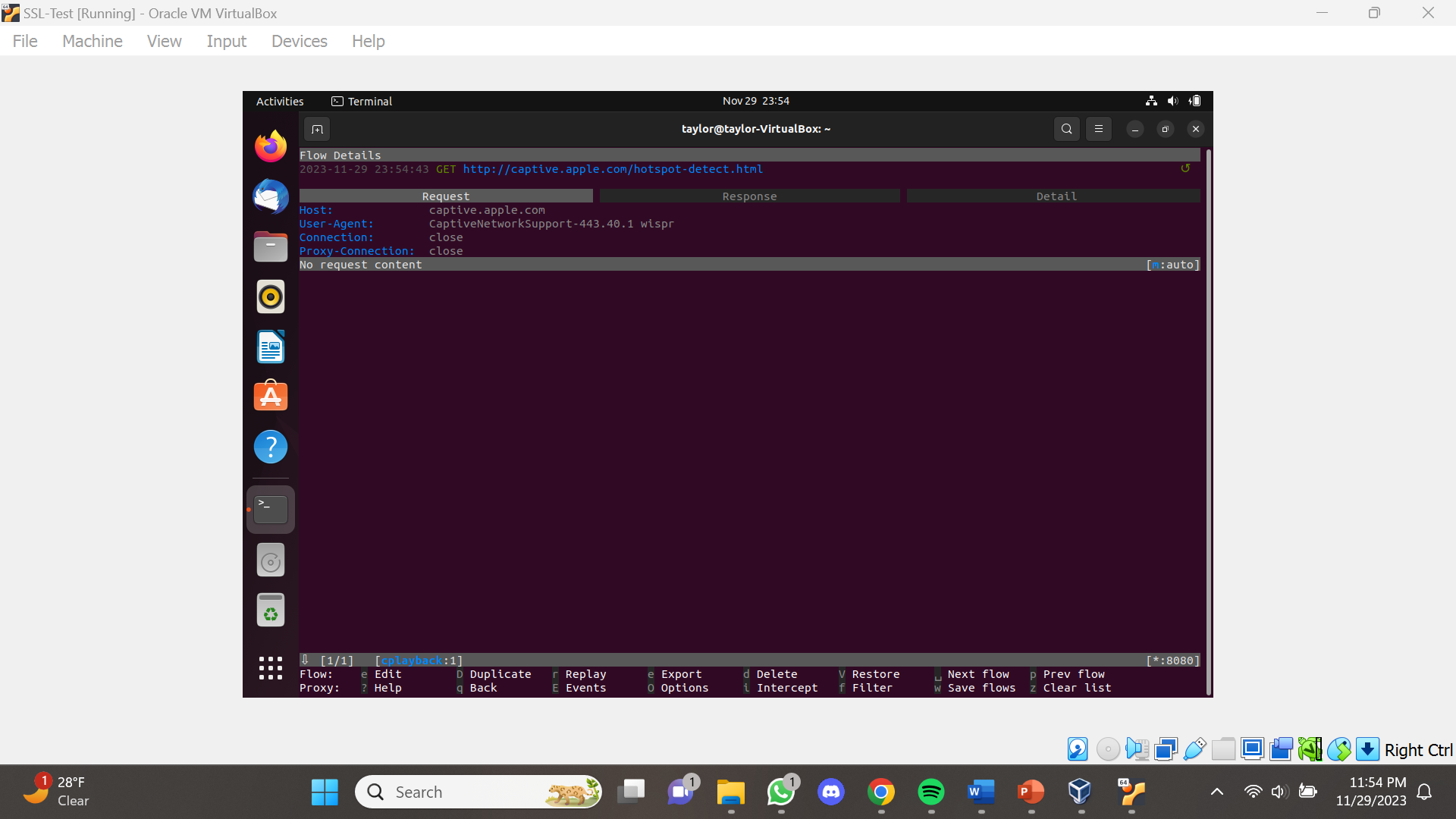
****

* As seen above, this app it unsafe. The proxy was easily able to identify a non robot-user just by running this consent\_status query. This was shortly after giving the app permissions for my phone as it is a file manager.
* Continuing in the pictures below, another query was ran in a xmlpost which was able to identify my phone’s brand/model and screen size of all things.
* Definitely an unsafe app, deleting right away.

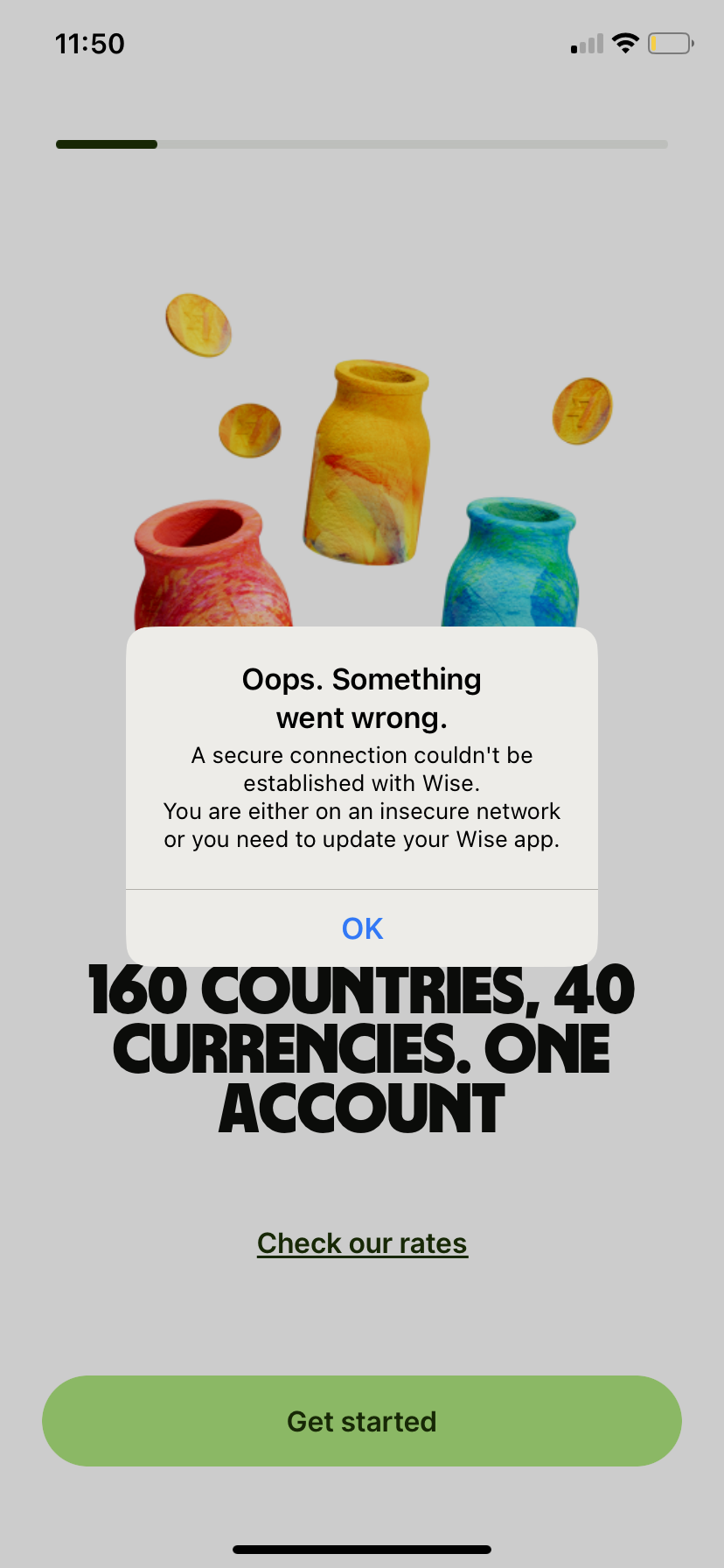
****

****

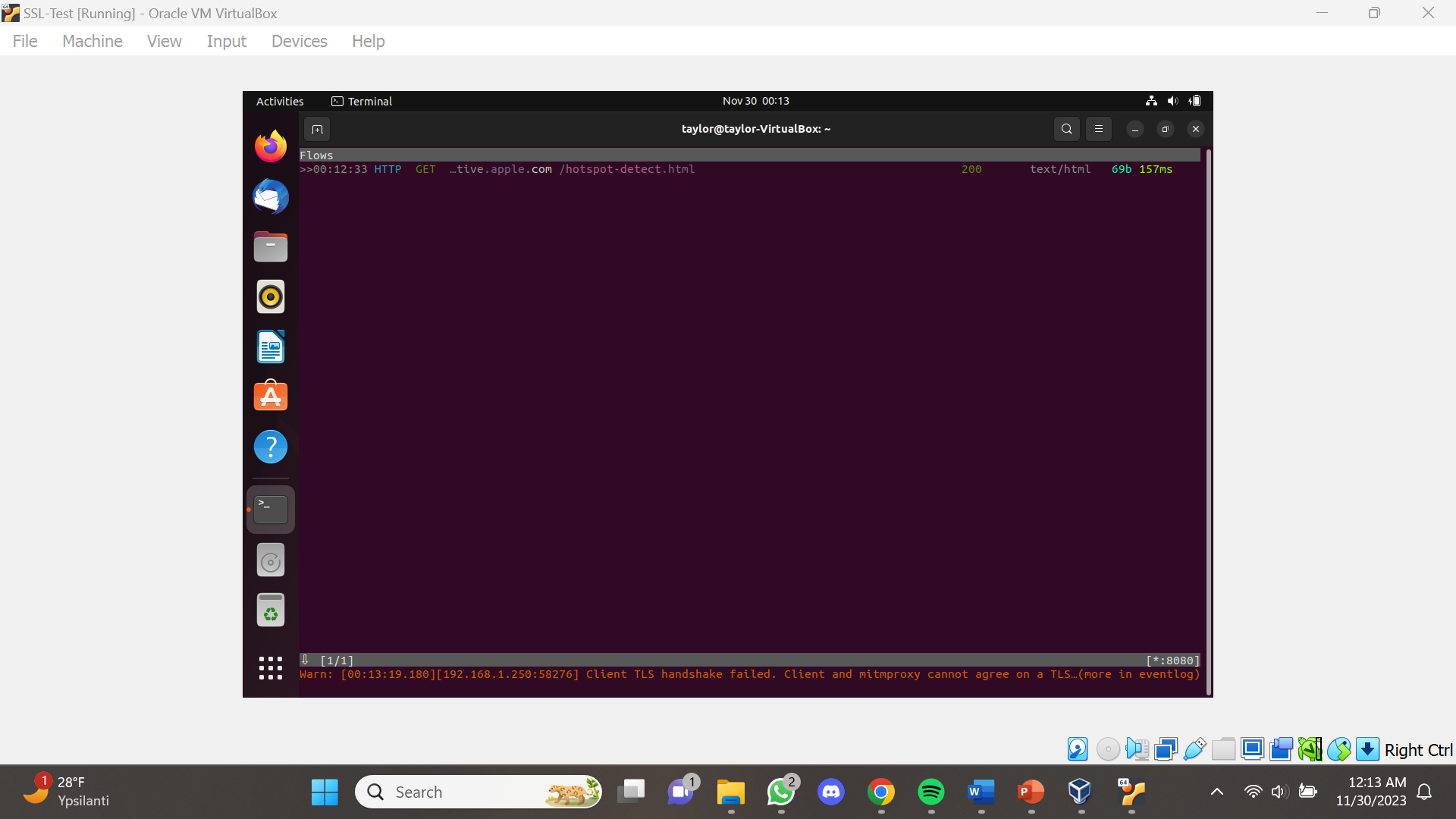
1. **Wise**

****

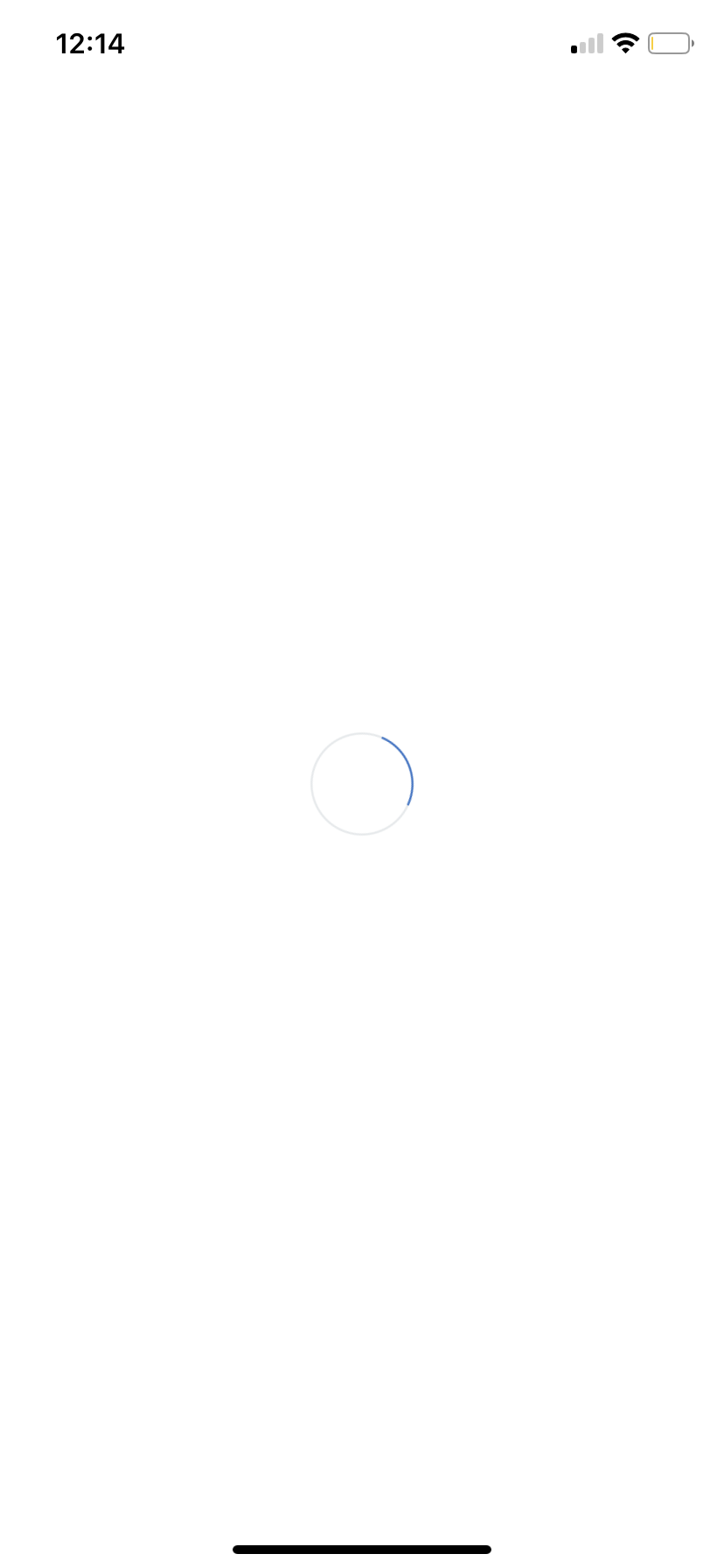
* Upon entering the app, the proxy server shows a single packet flow.
* When entering the app, a server connection error message appears and even when you press “ok”, nothing can be done within the app.

****

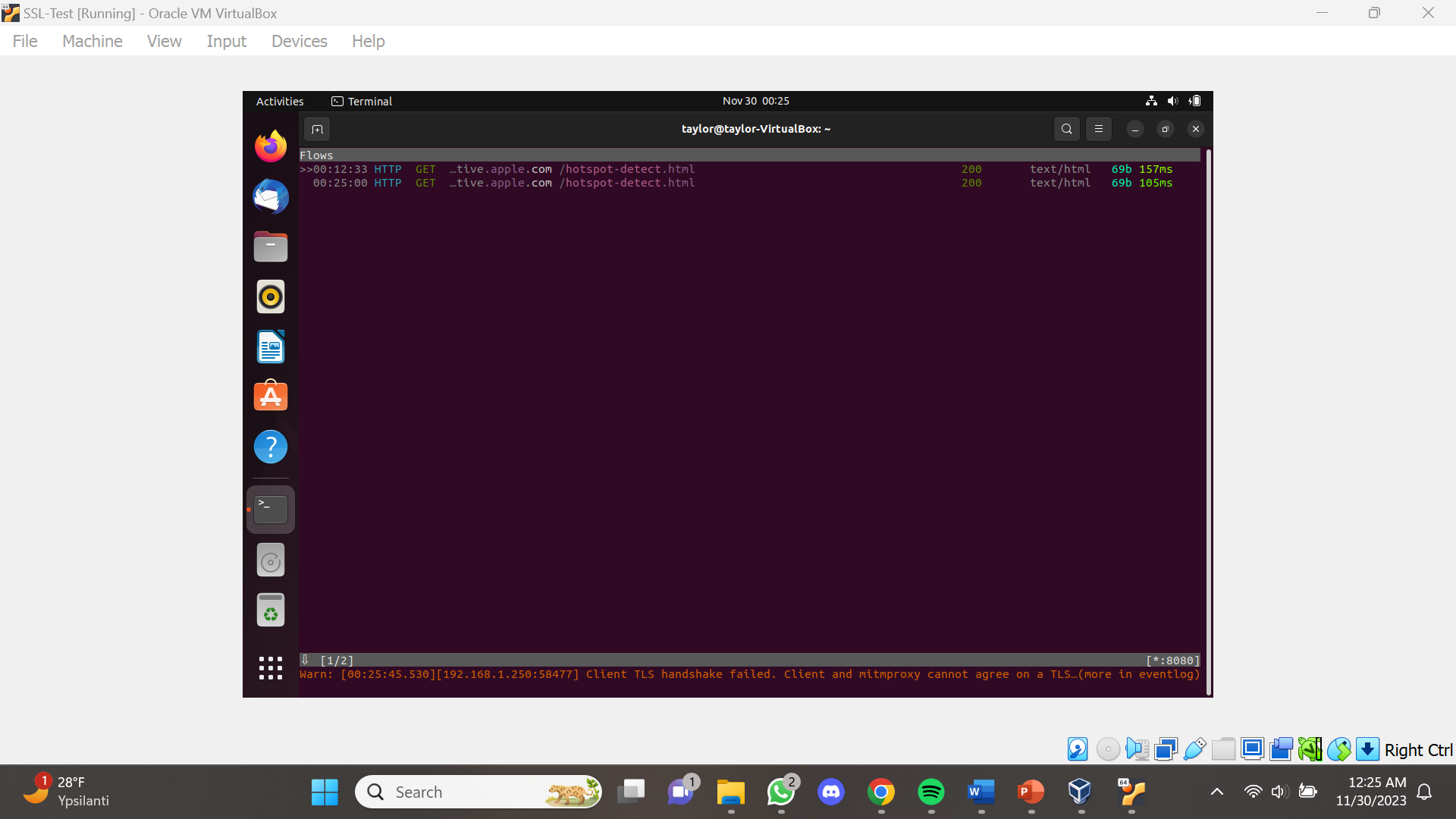
1. **Afterpay**

****

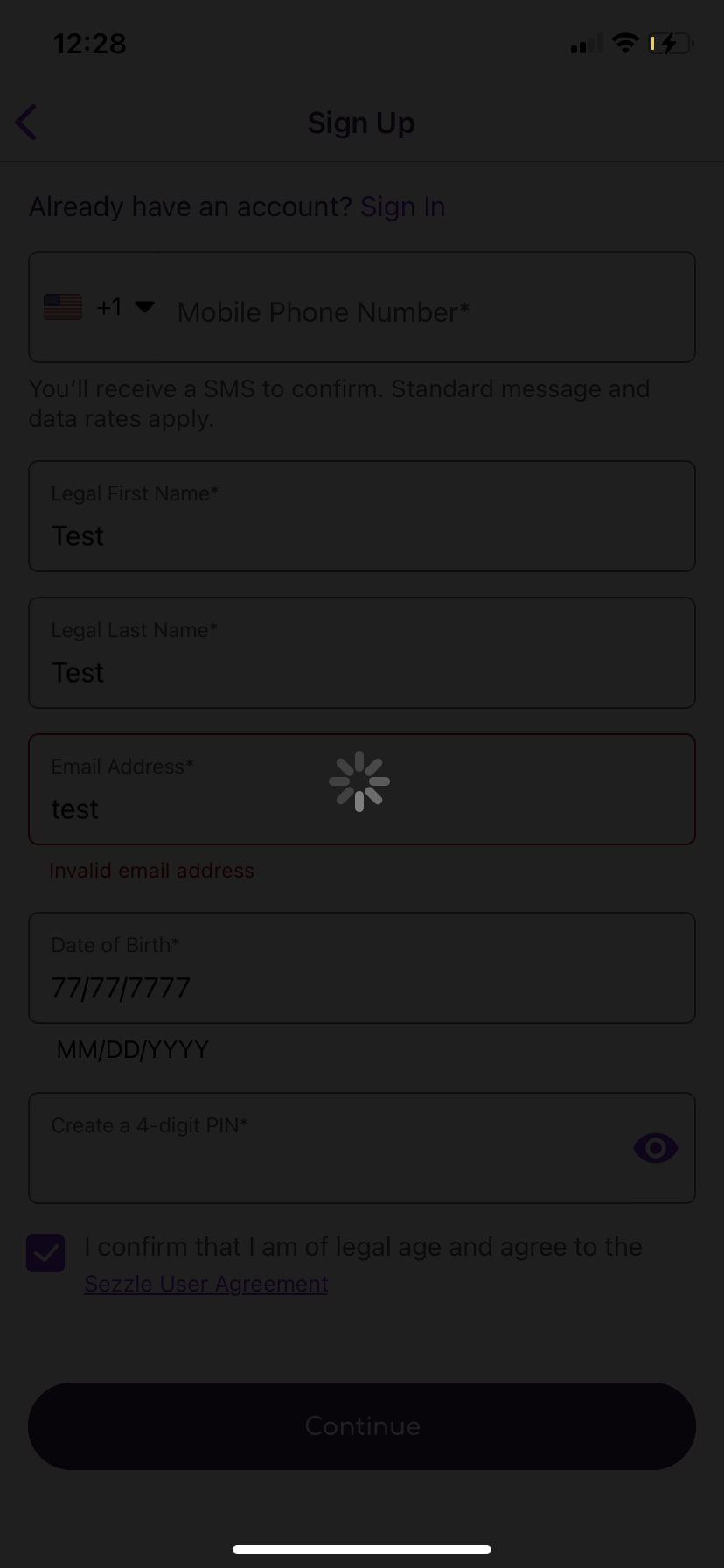
* Upon entering the app, the proxy server shows a single packet flow.
* Gave client TLS handshake error on the bottom of proxy server.
* Once entering the app, the app only loads continuously and no actions can be done.

****

1. **Sezzle**

****

* Proxy server was able to pick up two pocket flows when the app was opened.
* Gave client TLS handshake error on the bottom of proxy server.
* After entering the app, filling out login information, and pressing submit, the app refuses to do any more actions besides loading continuously.

****

**CONCLUSIONS**

In summary, our study into mobile application security has given our team valuable insights into the vulnerabilities present in this digital realm. Despite testing ten applications, only one exhibited flaws in its security measures concerning user data. This outcome underscores the unpredictable nature of identifying flawed apps merely by visual inspection. Intriguingly, our findings suggest a nuanced landscape, with applications within the Google Play Store appearing more susceptible to vulnerabilities compared to their counterparts in the Apple Store. This distinction highlights the importance of robust security protocols for developers and the necessity for users to exercise heightened caution, particularly when engaging with apps on certain platforms. As we reflect on our project, we recognize the dynamic challenges inherent in safeguarding user data and advocate for the continued lookout for possible security concerns and research in the pursuit of strengthening the security of all mobile applications across various platforms.

# **REFERENCES**

1. “iPhone Top Charts on the App Store - Apple.” *App Store*, https://apps.apple.com/us/charts/iphone. Accessed 22 November 2023.
2. Spiridonova, Kristina. “App Research — What Is It And How To Do It Right.” *Purrweb*, 26 October 2023, https://www.purrweb.com/blog/app-research-what-is-it-and-how-to-do-it/. Accessed 10 November 2023.
3. “App Vetting 101: How to Determine App Safety Before Downloading.” *TruSecAi*, 10 August 2023, https://trusecai.com/app-vetting-101-how-to-determine-app-safety-before-downloading/. Accessed 1 December 2023.