Tell us what your idea is.

Everyday we encounter numerous web-links across various apps ranging from social media to email, most of which are shortened URLs, making it impossible to know where they might lead to. Some malicious web-links are even disguised with legitimate website names to trick the average user. The internet is indeed a boon. But it is fair to say that for the average user, the internet is a black box and bad actors take advantage of this, whether it be stealing personal information, mining crypto-currencies or promoting illicit content.

So how do we provide a safe experience on the web? I believe that it should start even before the user visits a website. Say the user receives a suspicious web-link on some app. Upon clicking this link, we need to verify the content and credibility of the website it leads to and inform the user before he opens it in a browser.

To achieve this, I propose 2 stages of verification.

1. URL verification - Detect redirecting URLs, detect phishing URLs using simple decision tree based approaches.
2. Content verification - Extract and determine the content of a website using Text and Image classification. (adult content, illegal betting websites, advertising spams etc.)

Tell us how you plan on bringing it to life.

Android Implementation Details

One of the most crucial aspects of my proposal is providing a seamless experience to the user. The user shouldn't have to copy a web-link from one app, paste it onto another app to verify it and then open it in a browser! Thanks to the open nature of Android, this is not the case. The verification can be done as soon as the user clicks on a link and can be automatically redirected to a browser through the following steps:

1. Intercept web-links by registering an intent-filter for web-URL schemes. Show the verification dialog on top of the host app as soon as the link is clicked.
2. Extract the web content using web scraping tools like jsoup HTML parser. Perform verification in the 2 stages as mentioned above.
3. Inform the user the results of the verification on the same dialog and pass the web-link intent to the default browser if it is safe/white-listed or if the user explicitly chooses to visit it.

Listed below are two ways to implement this proposal. For the intents and purposes of this challenge, I have chosen to go with the first method just for the ease of prototyping.

a) Stand alone app

Act as an intermediary, intercept web-links, perform verification and pass the web-link to the default browser.

Pros

- Compatible with all browsers.

Cons

- Probably an Android anti-pattern to register as a browser when it actually isn't.
- Cannot intercept malicious web-links that are opened within a browser.

b) Bake feature inside Chrome app

Pros

- Makes sense from the user's perspective to have a browser perform the verification before opening it.
- Verify links that are clicked even within the browser.

Cons

- Incompatibility with other browsers.

On-Device Machine Learning

Leveraging on-device machine learning allows us to verify URLs and websites on the device without having to wait for a verification response from a central server. I plan to achieve this by creating custom TensorFlowLite models that can be deployed on-device using MLKit.

1. Soft Decision Tree to detect phishing URLs.
2. Text and Image Classification to determine if the website contains inappropriate content or suspicious intents.

How Google can help?

Basically connect me with my favorite Google developer advocates 😊

- Video session with Nick Butcher(@crafty) to discuss and finalize the UI/UX of the app.
- Video session with a technical representative from the Google SafeBrowsing team to discuss potential solutions for detecting malicious URLs and websites.
- Vaibhavi Desai as a mentor for my project, who is currently running Machine Learning Bootcamps. She has previously mentored me for a Google student program called Applied CS with Android back in India. Or a TensorFlowLite developer advocate. Or someone preferably from the Google Munich office.

Proposed Timeline

November

✔️ Setup base project architecture
✔️ Implement a PoC for the user-flow, intercept web-links and pass it to browser.
✔️ Setup jsoup HTML parser to extract website content.

December - mid January

🔵 Finalize the UI/UX of the app. (Video session with Nick Butcher)
🔵 Research machine learning solutions. (Video session with SafeBrowsing team)
🔵 Implement stage 1 URL verification.

mid January - mid March

🔵 Implement stage 2 content verification.
🔵 Benchmark results.
🔵 UI design and improvements.

mid March - April

🔵 Write tests.
🔵 Prepare release build.
🔵 Prepare store listing material for Google Play.
🔵 Begin brief alpha-testing.

Tell us about you.

I'm Chandramohan Sudar, a master's student at the Technical University of Munich where I specialize in Machine Learning and Analytics. I have been working with Android for the past 5 years on projects ranging from Network Security to Object detection for Autonomous vehicles. Along with my master's study program, I have been working as a part-time Android developer at P3 digital services where I am currently working on Android Automotive solutions for one of the major auto manufacturers in Germany.

I have also previously worked with on-device machine learning during an entrepreneurial course at my university where we partnered with Huawei to harness the on-device machine learning capabilities of the new Huawei smartphones running the Kirin 970/980 processors making use of their HiAi engine. We developed Regalo, an AI-based Android application that recommends gifts based on the preferences of the recipient gathered from social media profiles. View the marketing video of the app here.

During my bachelor's in India, I was part of Google's student program Applied CS with Android through which I delivered Android workshops to students in my university. I was one of 2 students from this program to receive a travel grant for Google I/O 2016. Looking back, the experience I gained through the program and by attending Google I/O, kick-started my journey into Android development and my involvement with the community through Google Developer Groups and other meetups. Blog post about my experience.

I also independently develop and publish android apps in my spare time.

Orble - https://play.google.com/store/apps/details?id=com.chandruscm.orble.android

A minimalist reflex-based game developed using LibGDx, a cross platform game development framework. It has grossed more than 150,000 downloads with over 1000 reviews. It has also been featured in a lot of youtube channels.

miniRTO - https://play.google.com/store/apps/details?id=com.chandruscm.minirto

An app that fetches the registration details of Indian vehicles using jsoup Java HTML parser. It grossed more than 50k downloads with a rating above 4.0. Unfortunately I had to discontinue the app due to govt. regulations, but this ended up making the users unhappy which resulted in the current 2.8 rating 😭

Thirukkural Puthiya Vadivam (Tamil | English)

https://play.google.com/store/apps/details?id=com.chandruscm.thirukkural

An intuitive and reliable digital version of a 2000-year old Tamil literature classic authored by Thiruvalluvar.

Having noticed the lack of good quality Android apps in my mother tongue Tamil, I began developing this app with the core intention of developing a Tamil app that meets the high standards of modern Android apps. Interestingly, I faced very specific challenges pertaining to Indian regional languages.

I have made use of the newest technologies and the recommended best practices conforming to what is now termed as "Modern Android Development". (Kotlin, Dependency Injection with Dagger, Clean MVVM Architecture, Data binding, Jetpack Architecture Components, new Material Library). Due to the commercial aspect of the app and the business relations with a 3rd party publisher, I cannot open-source the app. However, if the challenge committee wants to view this project source, I can definitely provide access to the GitHub repo.

💡 Blog - https://chandruscm.wordpress.com
💡 Medium - https://medium.com/@chandruscm
💡 GitHub - https://github.com/chandruscm
💡 LinkedIn - https://www.linkedin.com/in/chandruscm/