Using WebAPK Technology for Phishing Attacks

linkedin.com/pulse/using-webapk-technology-phishing-attacks-csirt-knf

CSIRT KNF

USING WEBAPK TECHNOLOGY
FOR PHISHING ATTACKS

CSIRT KNF

Zespół Reagowania na Incydenty Bezpieczeństwa Komputerowego polskiego sektora finansowego.

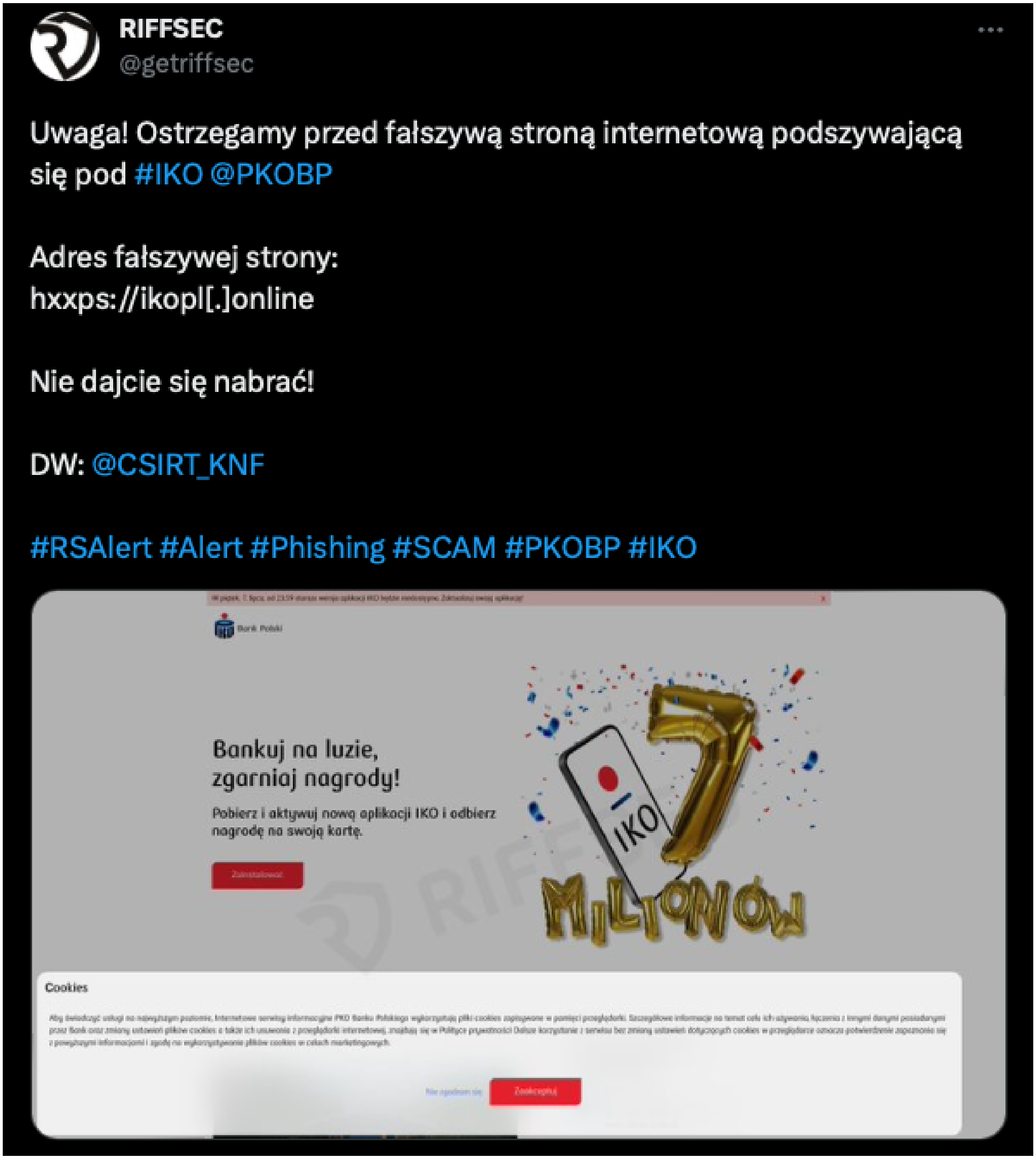
Published Jul 10, 2023

+ Follow

Report: Using WebAPK Technology for Phishing Attacks

Introduction

The CSIRT KNF team carried out a detailed analysis of a website reported by RIFFSEC (https://twitter.com/getriffsec/status/1676663509617131520). The result of this analysis was the discovery of a sophisticated phishing attack that used WebAPK technology to install a malicious application on Android devices.



RIFFSEC
@getriffsec

Uwaga! Ostrzegamy przed fałszywą stroną internetową podszywającą się pod #IKO @PKOBP

Adres fałszywej strony:
hxxps://ikopl[.]online

Nie dajcie się nabrać!

DW: @CSIRT_KNF

#RSAlert #Alert #Phishing #SCAM #PKOBP #IKO

W piątek, 7 lipca, od 23:59 stara wersja aplikacji IKO będzie niedostępna. Zaktualizuj swoją aplikację!

Bank Polski

Bankuj na luzie,
zgarniaj nagrody!

Pobierz i aktywuj nową aplikacji IKO i odbierz nagrodę na swoją kartę.

Zainstalować

Cookies

Aby świadczyć usługi na najwyższym poziomie, internetowe serwisy informacyjne PKO Banku Polskiego wykorzystują pliki cookies zapisywane w pamięci przeglądarki. Szczegółowe informacje na temat celu ich używania, łączenia z innymi danymi posiadanymi przez Bank oraz zmiany ustawień plików cookies a także ich usuwania z przeglądarki internetowej, znajdują się w Polityce prywatności Dalsze korzystanie z serwisu bez zmiany ustawień dotyczących cookies w przeglądarce oznacza potwierdzenie zapoznania się z powyższymi informacjami i zgodę na wykorzystywanie plików cookies w celach marketingowych.

WebAPK Technology

WebAPK is a technology that enables the creation of web applications that can be installed on Android devices as native applications. This is part of a broader trend called Progressive Web Apps (PWA), which aims to enhance the functionality and performance of web applications.

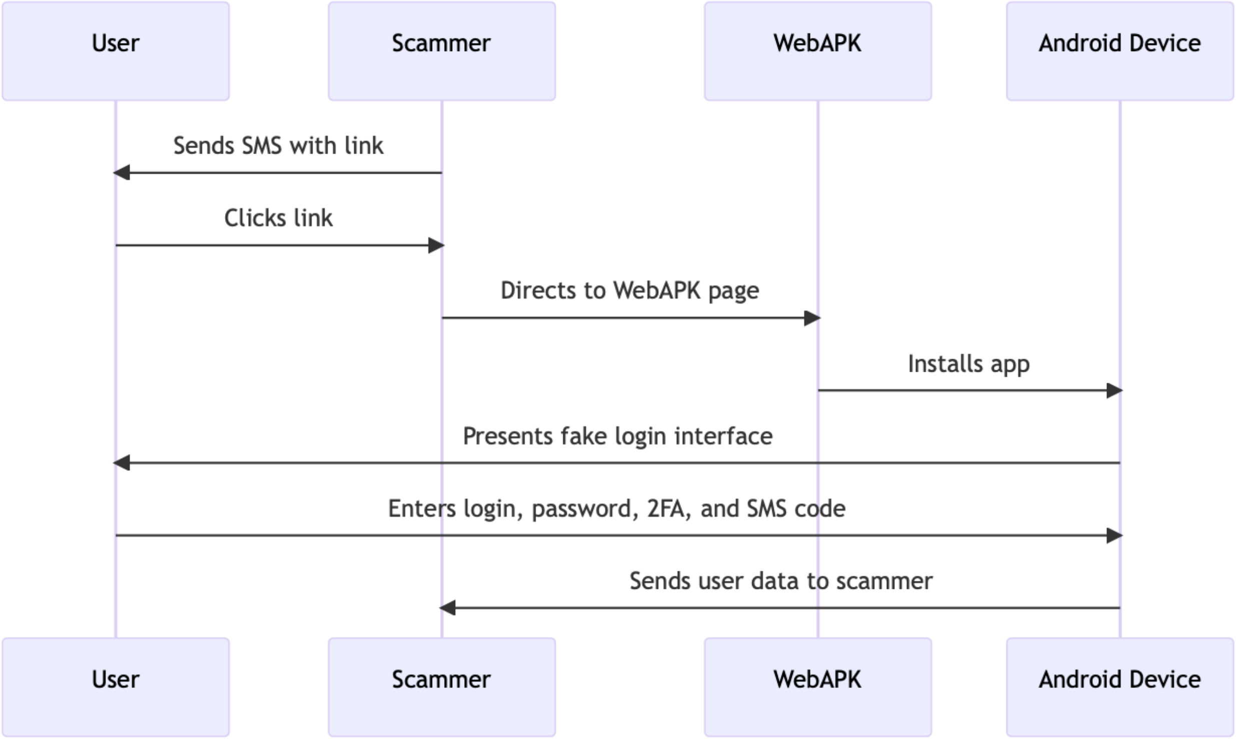
WebAPK works by generating an APK (Android Package Kit) file for a web application. APK is a file format used by the Android operating system to distribute and install mobile applications. When a user chooses to "add to home screen" a web application, the Android system automatically generates an APK file for that application.

A key element of WebAPK technology is the manifest.json file, which is a configuration file for the web application. The manifest.json file contains information such as the application name, icons, theme color, screen orientation, and start URL. This allows the Android system to know what the application should look like and how it should behave.

One of the main advantages of WebAPK technology is that it allows the installation of web applications directly from the browser, without the need to use the Google Play Store.

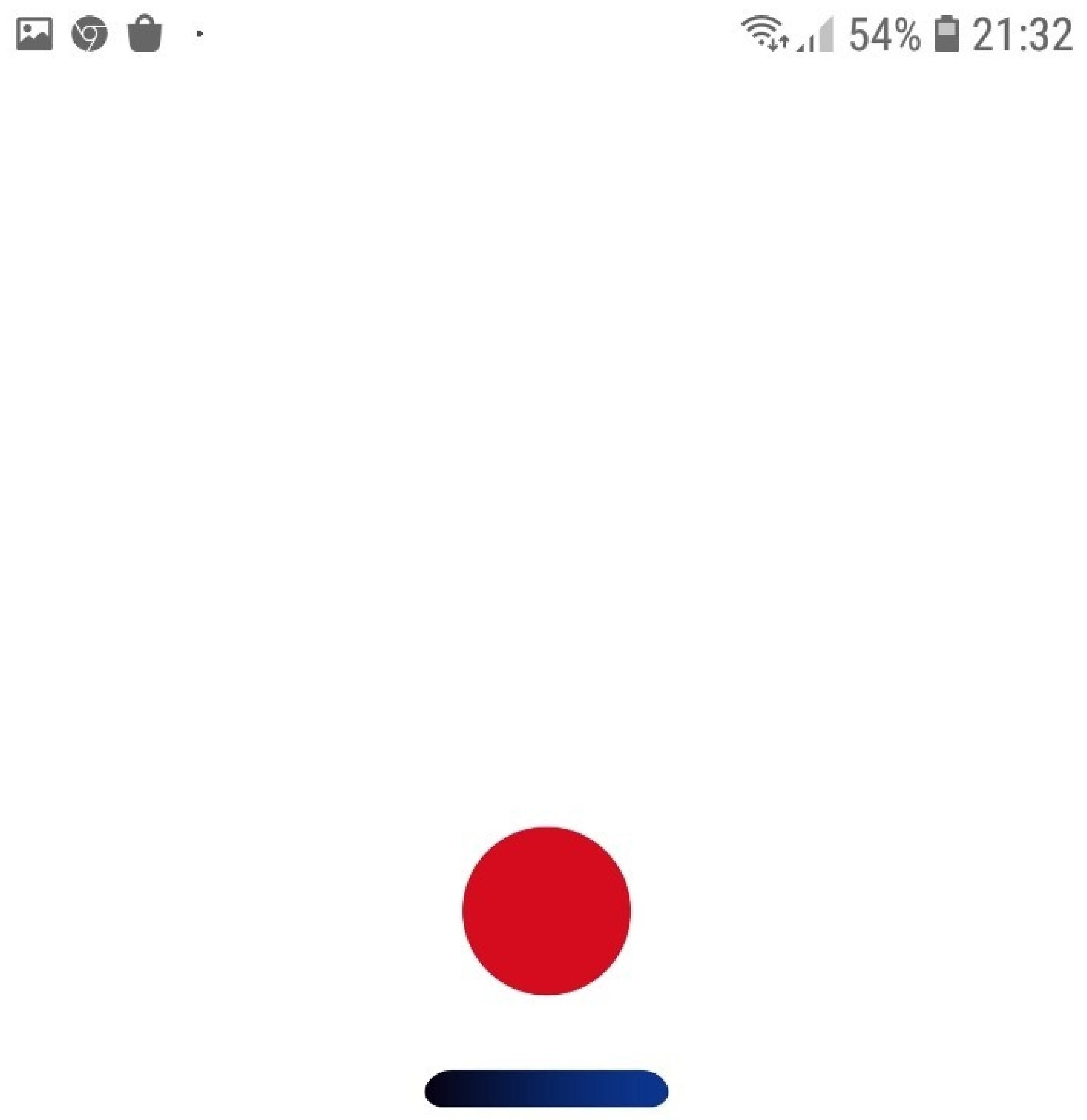
Attack Process

The attack began with victims receiving SMS messages suggesting the need to update a mobile banking application. The link contained in the message led to a site that used WebAPK technology to install a malicious application on the victim's device. Crucially, the installation process did not trigger typical warnings about installations from untrusted sources.

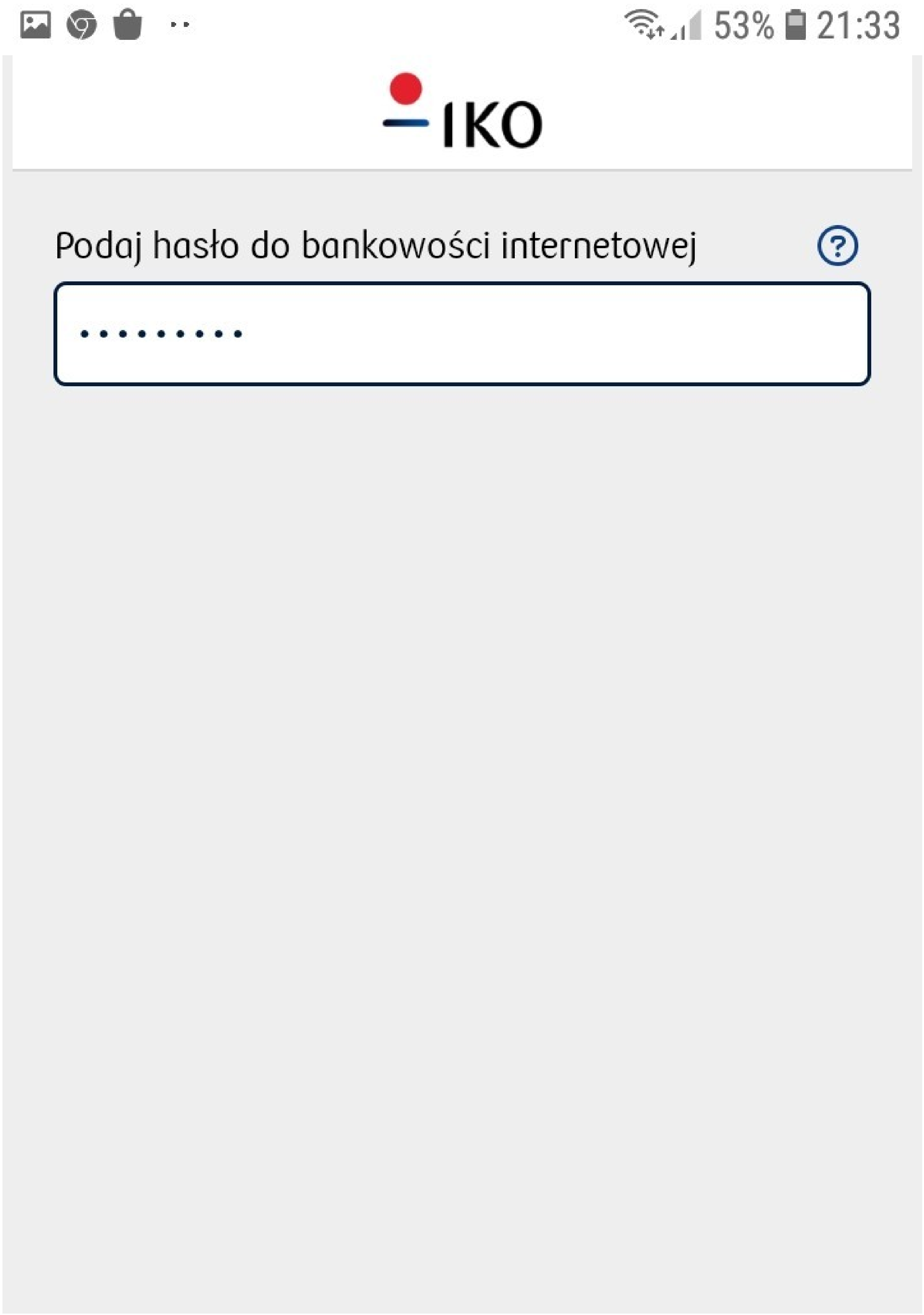


Malicious Application

After installation, the application presented the user with an interface imitating a mobile banking login panel. Subsequent screens asked the user for a login, password, 2FA code, and an SMS code, which was already being used for transaction authorization.


54%
21:32

IKO



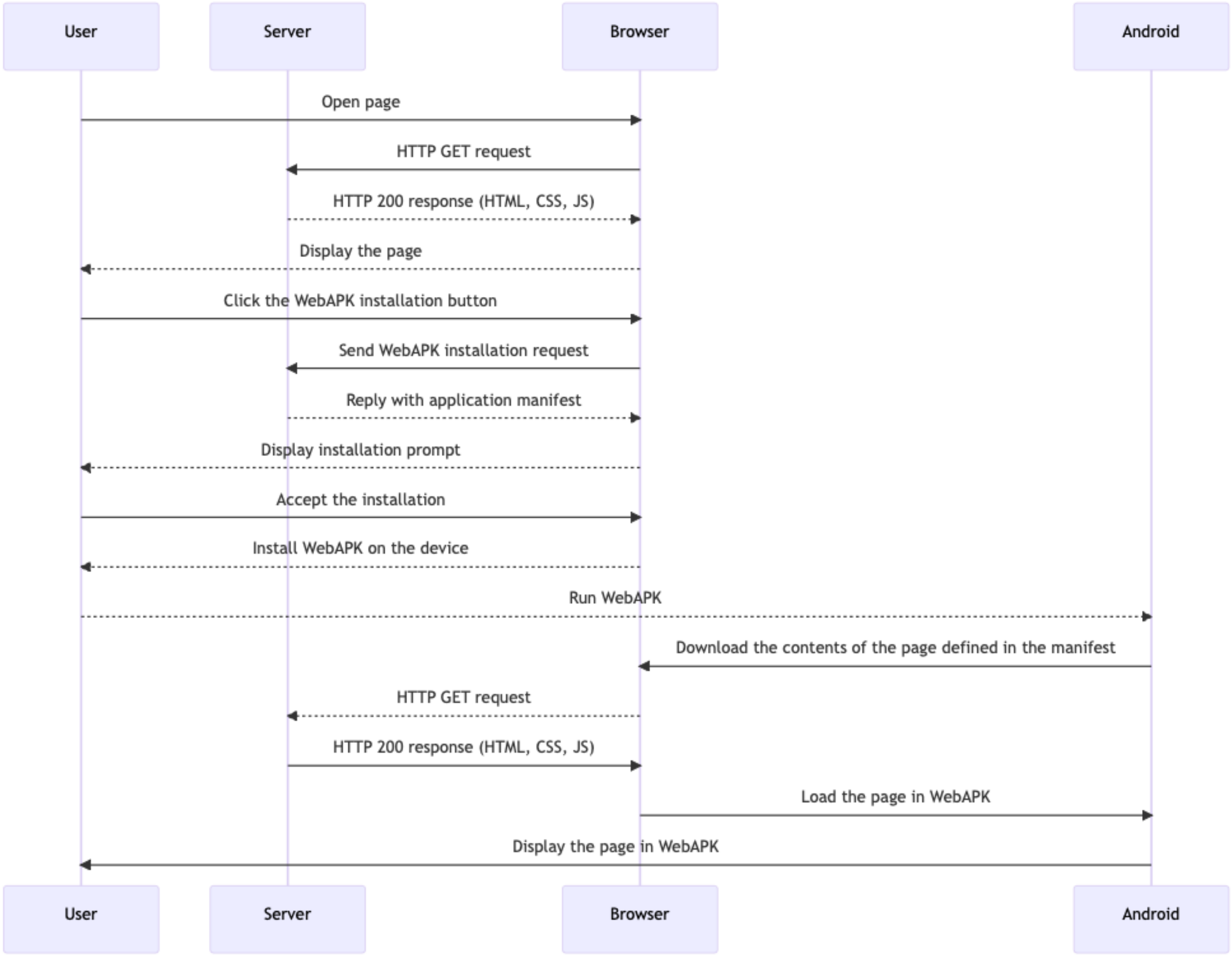
Podaj hasło do bankowości internetowej

•••••••••

Summary

Phishing attack based on WebAPK technology poses a serious threat due to the ability to install a malicious application without displaying typical warnings associated with installations from untrusted sources. The application then mimicked the mobile banking interface to defraud users of their login details and authorization codes. One of the main advantages of WebAPK technology is that it allows the installation of web applications directly from the browser, without the need to use the Google Play Store. This means that WebAPK applications can be distributed directly by developers, bypassing the Google Play Store approval process. However, the same feature that makes WebAPK attractive to developers also makes it attractive to those with malicious intent, as shown by the phishing attack described. Without appropriate safeguards, WebAPK technology can be used to install malicious applications that can steal user data or otherwise violate their privacy.

Technical Aspects


User
Server
Browser
Android
Open page
HTTP GET request
HTTP 200 response (HTML, CSS, JS)
Display the page
Click the WebAPK installation button
Send WebAPK installation request
Reply with application manifest
Display installation prompt
Accept the installation
Install WebAPK on the device
Run WebAPK
Download the contents of the page defined in the manifest
HTTP GET request
HTTP 200 response (HTML, CSS, JS)
Load the page in WebAPK
Display the page in WebAPK
User
Server
Browser
Android

How WebAPK works in practice:

- The user opens a page in the browser.
- The browser sends an HTTP GET request to the server.
- The server responds to the request, sending an HTTP 200 response containing HTML, CSS, JS.
- The browser displays the page to the user.
- The user clicks on the WebAPK installation button.
- The browser sends a WebAPK installation request to the server.
- The server responds to the request, sending the application manifest.
- The browser displays an installation prompt to the user.
- The user accepts the installation.
- The browser installs the WebAPK on the user's device.
- The user launches the WebAPK on their Android device.
- Android delegates to the browser to download the page content defined in the manifest.
- The browser sends another HTTP GET request to the server.

- The server responds to the request, sending an HTTP 200 response containing HTML, CSS, JS.
- The browser loads the page in the WebAPK on Android.
- Android displays the page in the WebAPK to the user.

Sample data:

MD5: ae12fd46fe868dc4384db26e6f745cce

SHA1: 1c24f4398caae9179028b5415ed980f0ad18f4a7

SHA256: 113be611bcb64b04dfaca2481d8108e94ff41a56fb81f8aef190d4161acd983d

App Name: IKO

Package Name: org.chromium.webapk.a798467883c056fed_v2

Main Activity: org.chromium.webapk.shell_apk.h2o.H2OOpaqueMainActivity

CERTIFICATE

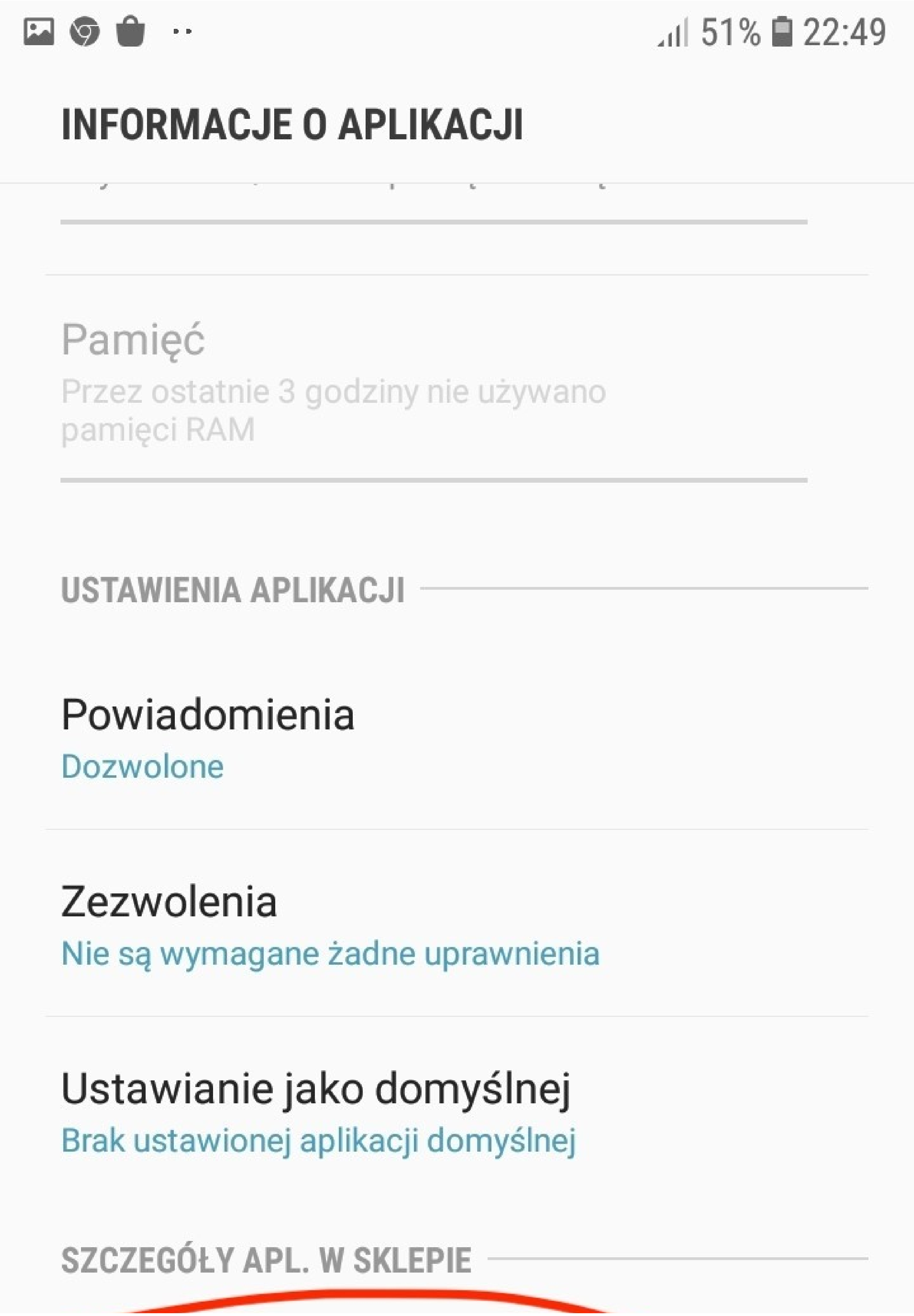
md5: 65044365734519 11f10b2be6ad7d560c

sha1: 0fa9a1b93dac4cc5a0a88d08f6949dc11e5275b0

sha256: fe71ef0d9897374f009d3c930c3eac31f523a29d42ad4898f366b1f220769bd1

Subject: C=US, O=Google, OU=Chrome WebAPK, CN=CA

The application is signed with the Google Chrome certificate, which is why it appears in the system settings as installed by Google Play Protect.



51% 22:49

INFORMACJE O APLIKACJI

Pamięć

Przez ostatnie 3 godziny nie używano pamięci RAM

USTAWIENIA APLIKACJI

Powiadomienia

Dozwolone

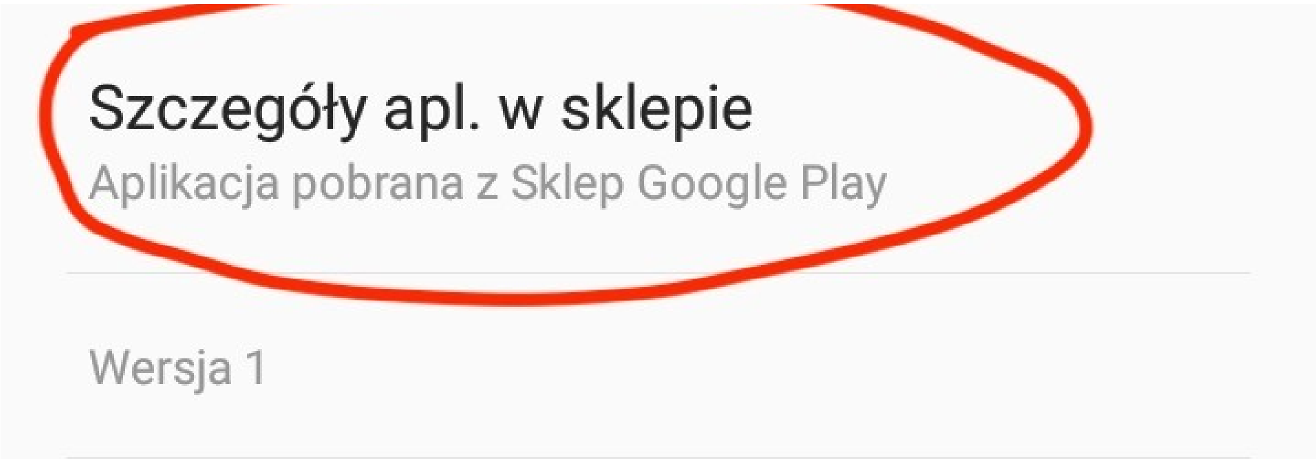
Zezwolenia

Nie są wymagane żadne uprawnienia

Ustawianie jako domyślnej

Brak ustawionej aplikacji domyślnej

SZCZEGÓŁY APL. W SKLEPIE



Szczegóły apl. w sklepie

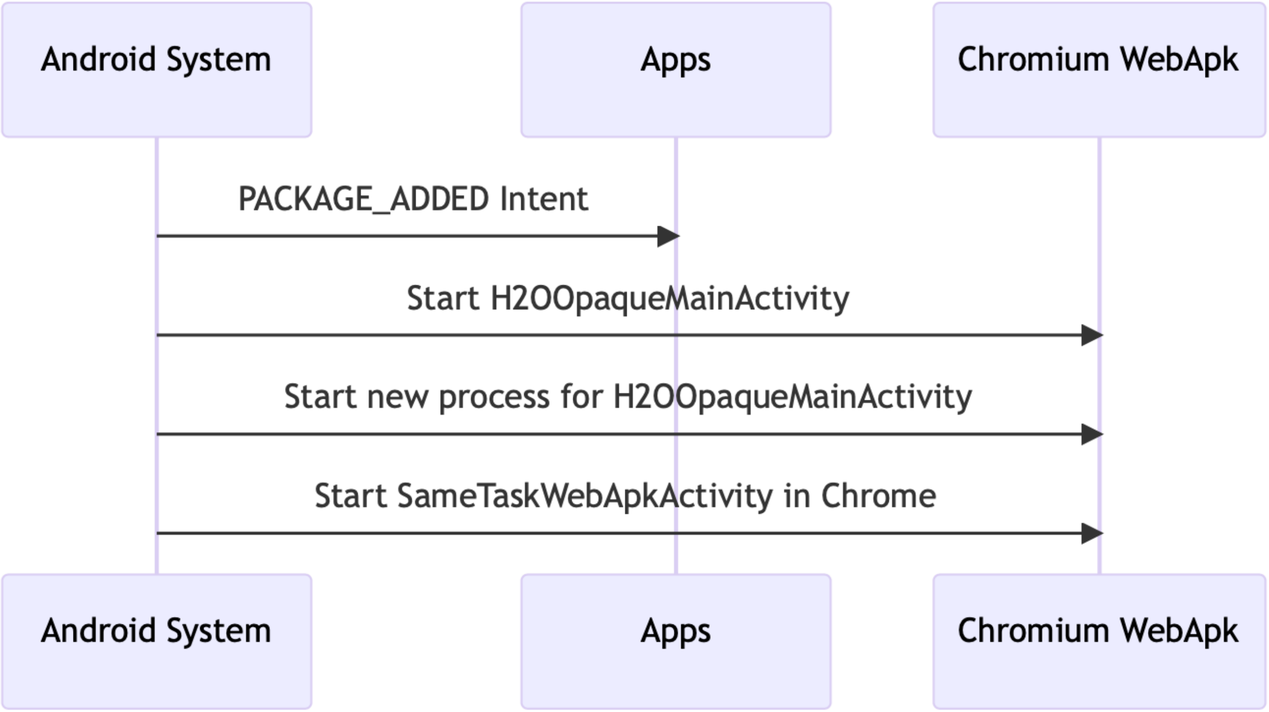
Aplikacja pobrana z Sklep Google Play

Wersja 1

Based on the analysis of AndroidManifest.xml (which is a file that defines the application and its behavior in the system), we can say that:

- SDK Version: The application was compiled using SDK version 34, codenamed "VanillaIceCream". The application's code version is "1", and the name version is also "1". This means that the application was built using the latest tools available at the time of creation.
- Package: The application's package name is "org.chromium.webapk.a798467883c056fed_v2". This is a unique identifier for the application in the Android system.
- Permissions: The application requires permission to send notifications. This means that the application can display notifications on the user's device.
- Intents: The application defines intents for handling VIEW actions and managing data from a Trusted Web Activity. This means that the application can handle website links and manage data related to trusted web activities.
- Activities: The application defines several activities, including those for displaying the page, managing data, displaying the start screen, and handling notifications. Each of these activities has specific behavior and is responsible for different aspects of the user's interaction with the application.
- Data: The application defines various metadata such as the shell APK version, the host runtime, the host runtime application name, the startup URL, thedisplay mode, the theme color, the background color, the icon identifier, the icon URL, and the network manifest URL.

The application is installed and run, and then the Chrome activity is launched.


Android System
Apps
Chromium WebApk
PACKAGE_ADDED Intent
Start H2OOpaqueMainActivity
Start new process for H2OOpaqueMainActivity
Start SameTaskWebApkActivity in Chrome
Android System
Apps
Chromium WebApk

- The Android system receives a PACKAGE_ADDED intent for the package org.chromium.webapk.a798467883c056fed_v2. This intent is sent when a package is added to the system. In this case, various applications receive this intent.
- The Android system launches the main activity (H2OOpaqueMainActivity) of the application org.chromium.webapk.a798467883c056fed_v2. This is typical for Android applications that have a defined main activity, which is launched when the user opens the application.
- The Android system then starts a new process for the H2OOpaqueMainActivity activity of the application org.chromium.webapk.a798467883c056fed_v2. In Android, each activity operates in its own process, which provides isolation and protects against crashes of other activities.
- Finally, the Android system launches the SameTaskWebApkActivity activity from the Chrome application to display the web page. This is typical for web applications that are displayed in the Chrome browser on Android.

Technical section summary

As a result of our analysis, we found that the malicious application, using WebAPK technology, operated by launching a Chrome activity. It then loaded a page defined in the AndroidManifest.xml file (<meta-data android:name="org.chromium.webapk.shell_apk.startUrl" android:value=„https://george.ikopl.online/app/„/>).

One of the challenges in countering such attacks is the fact that WebAPK applications generate different package names and checksums on each device. They are dynamically built by the Chrome engine, which makes the use of this data as Indicators of Compromise (IoC) difficult.

Additionally, detecting such applications by antivirus systems is complex and often impossible. For this reason, one of the most effective ways to counter such attacks is by detecting and blocking websites that use the WebAPK mechanism to carry out phishing attacks. We strongly recommend an approach focused on identifying and blocking these sites to minimize the risk to users.